Dealing with Domain Shift in Deep Learning: From Training-Time Generalization to Test-Time Adaptation

by

Mehrdad NOORI

MANUSCRIPT-BASED THESIS PRESENTED TO ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
IN PARTIAL FULFILLMENT FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY
Ph.D.

MONTREAL, MARCH 04, 2026

ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
UNIVERSITÉ DU QUÉBEC

BOARD OF EXAMINERS

THIS THESIS HAS BEEN EVALUATED

BY THE FOLLOWING BOARD OF EXAMINERS

Mr. Christian Desrosiers, Thesis supervisor
Department of Software Engineering and IT, École de Technologie Supérieure

Mr. Ismail Ben Ayed, Thesis Co-Supervisor
Department of Systems Engineering, École de Technologie Supérieure

Mr. Eric Granger, Chair, Board of Examiners
Department of Systems Engineering, École de Technologie Supérieure

Mr. José Dolz, Member of the Jury
Department of Software Engineering and IT, École de Technologie Supérieure

Ms. Negar Rostamzadeh, External Independent Examiner
Staff Research Scientist, Google; Associate Industrial Member, Mila; Department of Computer Science, McGill University

THIS THESIS WAS PRESENTED AND DEFENDED

IN THE PRESENCE OF A BOARD OF EXAMINERS AND THE PUBLIC

ON FEBRUARY 20, 2026

AT ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

ACKNOWLEDGEMENTS

Reaching the completion of this PhD marks a significant milestone in my life — the culmination of a journey filled with joy, challenges, growth, and deep human connection. It has been a transformative experience that has shaped both my professional and personal perspectives.

First and foremost, I would like to express my sincere gratitude to my supervisors, Professors Christian Desrosiers and Ismail Ben Ayed, for their continuous guidance, encouragement, and patience throughout this journey. Their insights and mentorship have profoundly influenced the way I approach research, critical thinking, and scientific collaboration.

I would also like to extend my appreciation to Arnaud Lina and Steve Massicotte, my main collaborators at Zebra Technologies. From the very beginning of our projects, they offered immense support and enthusiasm, turning every challenge into a rewarding experience. Their professionalism and openness made my time at the research chair both productive and truly enjoyable.

My heartfelt thanks go to my thesis committee members for their invaluable time, thoughtful feedback, and commitment to evaluating my work with such care and attention.

A very special note of gratitude is reserved for my beloved family — my mother and father, whose love and support have been my foundation. Although I have not seen them in person during the last four years of my PhD, their unwavering encouragement and faith in me have been a constant source of strength. Their values, sacrifices, and belief in my potential have shaped the person I am today. Even from afar, their presence accompanies me in every step I take, offering comfort in difficult times and joy in moments of success. This accomplishment belongs as much to them as it does to me.

Finally, I wish to express my wholehearted appreciation to the extraordinary people I have met during my time in Montréal. To David Osowiechi, it has been a privilege to collaborate, exchange ideas, and share both research victories and unforgettable adventures. To my dear friends and colleagues Ali, Milad, Gustavo, Moslem, Farzad, Sonia, and Samuel — thank you for

the countless memories, laughter, and inspiration you brought beyond the lab. Your friendship and perseverance have made this journey truly unforgettable.

Gérer le décalage de domaine en apprentissage profond : de la généralisation à l'entraînement à l'adaptation au temps d'inférence

Mehrdad NOORI

RÉSUMÉ

Les modèles d'apprentissage profond ont connu des avancées remarquables dans un large éventail de tâches de vision par ordinateur, allant de la classification d'images à la détection d'objets et à la segmentation. Malgré ces progrès, la plupart de ces modèles sont conçus sous l'hypothèse simplificatrice que les données rencontrées lors du déploiement suivront une distribution similaire à celle observée pendant l'entraînement. Dans la pratique, cette hypothèse est rarement vérifiée. Les conditions réelles — telles que les variations d'éclairage, de capteurs, de points de vue ou de textures — peuvent modifier considérablement la distribution des données et entraîner une chute notable des performances. Cette vulnérabilité soulève des préoccupations majeures quant à la robustesse et à la fiabilité des systèmes de vision déployés dans des environnements non contrôlés.

Pour pallier cette limitation, cette thèse explore des approches allant au-delà du paradigme classique centré sur l'entraînement, afin de concevoir des modèles capables de maintenir leurs performances face à des conditions nouvelles et imprévues. Plus précisément, elle étudie deux directions : la *Généralisation de Domaine* (DG), qui vise à apprendre des représentations invariantes aux domaines durant l'entraînement, et l'*Adaptation au Temps d'Inférence* (TTA), qui permet aux modèles de s'ajuster dynamiquement lors de l'inférence à partir de données tests non étiquetées. Ensemble, ces approches répondent au besoin croissant de modèles fiables et adaptatifs, tant pour les architectures visuelles classiques que pour les modèles fondamentaux modernes tels que les systèmes vision–langage.

Dans la première partie, nous étudions la DG et proposons deux méthodes novatrices. (1) TFS-ViT (Token-Level Feature Stylization for Vision Transformers) introduit le premier cadre de stylisation de caractéristiques au niveau des tokens pour les Vision Transformers, en mélangeant les statistiques de normalisation entre échantillons afin de favoriser des représentations dépendantes de la structure plutôt que de la texture. Une variante sensible à l'attention exploite les cartes d'attention du jeton de classe pour orienter la stylisation vers les régions sémantiquement pertinentes, atteignant des performances de pointe sur les principaux jeux de données de DG. (2) FDS (Feedback-Guided Domain Synthesis) présente un cadre fondé sur la diffusion qui entraîne un modèle conditionnel multi-source unique capable de générer des pseudo-domaines couvrant les écarts inter-domaines. Un mécanisme de filtrage guidé par la rétroaction sélectionne les échantillons synthétiques les plus difficiles afin de promouvoir explicitement l'apprentissage de caractéristiques invariantes aux domaines, tout en évitant tout coût de génération lors de l'inférence.

Avec l'émergence de modèles fondamentaux à grande échelle, préentraînés une seule fois puis réutilisés pour une multitude de tâches, il devient crucial de concevoir des mécanismes d'adaptation capables d'opérer au moment de l'inférence sans accès aux données sources. Cela

motive la deuxième partie de cette thèse, consacrée aux stratégies d'adaptation entièrement au temps d'inférence pour les *Modèles Vision–Langage* (VLMs). (3) MLMP constitue le premier cadre TTA pour la *segmentation sémantique à vocabulaire ouvert* (OVSS), combinant une fusion adaptative multi-couches à une optimisation multi-prompt afin d'exploiter la sensibilité intrinsèque des VLMs aux prompts comme signal stable d'adaptation. Nous établissons également le premier benchmark complet pour l'OVSS-TTA, couvrant neuf ensembles de données et plus de quatre-vingts scénarios de test, fournissant ainsi un protocole standardisé pour les recherches futures.

(4) Histopath-C introduit le premier benchmark d'évaluation de l'adaptation TTA des VLMs en histopathologie numérique, simulant des décalages de domaine cliniquement réalistes tels que les variations de coloration, le flou et la contamination. Ce cadre constitue une base solide pour l'étude de la robustesse des modèles en contexte médical. S'appuyant sur ce benchmark, nous proposons également LATTE (Low-rank Adaptation with Transductive Template Ensembling), une stratégie d'adaptation simple mais efficace combinant plusieurs gabarits textuels à des mises à jour de faible rang pour améliorer la stabilité et la robustesse du modèle. Spécifiquement conçue pour l'histopathologie, cette méthode offre des gains de performance significatifs sur divers ensembles de données et représente l'une des applications les plus réalistes et pertinentes de l'adaptation au temps d'inférence.

Mots-clés: généralisation de domaine, adaptation au temps d'inférence, modèles vision–langage

Dealing with Domain Shift in Deep Learning: From Training-Time Generalization to Test-Time Adaptation

Mehrdad NOORI

ABSTRACT

Deep learning models have achieved remarkable progress across a wide range of computer vision tasks, from image classification and object detection to segmentation. Despite these advances, most models are developed under the simplifying assumption that the data observed during deployment will resemble that seen during training. In real-world scenarios, this assumption is rarely valid. Real-world conditions, such as variations in illumination, imaging sensors, viewpoints, or textures, can substantially alter the data distribution and lead to a significant degradation in model performance. This vulnerability raises critical concerns about the robustness and reliability of vision systems deployed in the wild.

To address this limitation, this thesis explores methods that move beyond the traditional training-centered paradigm toward models capable of maintaining performance under novel and unseen conditions. Specifically, it investigates two directions: Domain Generalization (DG), which aims to learn domain-invariant representations during training, and Test-Time Adaptation (TTA), which adjusts models dynamically during inference using only unlabeled test data. Together, these approaches address the growing need for reliable and adaptive models in both traditional vision architectures and modern foundation models such as vision–language systems.

In the first part, we study DG and propose two novel methods. (1) TFS-ViT (Token-Level Feature Stylization for Vision Transformers) introduces the first token-level feature stylization framework for Vision Transformers, mixing normalization statistics across samples to enforce structure- rather than texture-dependent representations. An attention-aware variant further exploits class-token saliency to guide stylization toward semantically relevant regions, achieving state-of-the-art generalization across standard DG benchmarks. (2) FDS (Feedback-Guided Domain Synthesis) presents a diffusion-based framework that trains a single multi-source conditional model to generate pseudo-domains spanning inter-domain gaps. A feedback-driven filtering mechanism selects challenging synthetic samples that explicitly encourage domain-invariant feature learning, yielding substantial robustness gains while incurring no inference-time cost.

With the advent of large-scale foundation models, which are pretrained once and reused across diverse tasks, it becomes crucial to develop mechanisms that enable adaptation at test time without access to source data. This motivates the second part of this thesis, which explores fully test-time adaptation strategies for Vision–Language Models (VLMs). (3) MLMP is the first TTA framework for Open-Vocabulary Semantic Segmentation (OVSS), combining adaptive multi-layer fusion with multi-prompt optimization to exploit VLMs' inherent prompt sensitivity as a stable adaptation signal. Furthermore, we establish the first comprehensive OVSS-TTA benchmark covering nine datasets and over eighty test scenarios, providing a standardized protocol for future research in this field.

(4) Histopath-C introduces the first benchmark for evaluating VLM TTA in digital histopathology, simulating realistic clinical domain shifts such as stain variation, blur, and contamination, thereby providing a valuable foundation for studying model robustness under clinically relevant domain shifts. Building upon this benchmark, we also propose LATTE (Low-rank Adaptation with Transductive Template Ensembling), a simple yet powerful adaptation strategy that combines multiple textual templates with low-rank updates to enhance model stability and robustness. This histopathology-specific method achieves significant performance improvements across diverse datasets and represents one of the most realistic and practically important applications of TTA.

Keywords: Domain Gernalization, Test-Time Adaptation, Vision-Language Models

TABLE OF CONTENTS

Page

LIST OF TABLES

LIST OF FIGURES

Page

LIST OF ABBREVIATIONS AND ACRONYMS

TTA	Test-Time Adaptation
TTT	Test-Time Training
DA	Domain Adaptation
DG	Domain Generalization
PTBN	Prediction Time Batch Normalization
TENT	Test Entropy Minimization
ETA	Efficient Test-time Adaptation
COTTA	Continual Test-Time Adaptation
LAME	Laplacian Adjusted Maximum-likelihood Estimation
SAR	Sharpness-Aware and Reliable entropy minimization
CLIP	Contrastive Language-Image Pre-training
SigLIP	Sigmoid Loss for Language Image Pre-Training
VLM	Vision Language Model
TPT	Test-time Prompt Tuning
TDA	Training-free Dynamic Adapter
WATT	Weight Averaging at Test-Time
OOD	Out-of-distribution
CNN	Convolutional Neural Network
ViT	Vision Transformer

ERM	Empirical Risk Minimization
SWA	Stochastic Weight Averaging
SWAD	Stochastic Weight Averaging for Domain generalization
EMA	Exponential Moving Average
OVSS	Open Vocabulary Semantic Segmentation
TFS-ViT	Token-level Feature Stylization
FDS	Feedback-guided Domain Synthesis
MLMP	Multi-Level and Multi-Prompt
LATTE	Low-rank Adaptation with Transductive Template Ensembling

LIST OF SYMBOLS AND UNITS OF MEASUREMENTS

$\mathcal{X}_s$	Input space of the source domain
$\mathcal{Y}_s$	Label space of the source domain
$\mathcal{X}_t$	Input space of the target domain
$\mathcal{Y}_t$	Label space of the target domain
$P(\mathcal{X}_s, \mathcal{Y}_s)$	Joint distribution of inputs and labels in the source domain
$P(\mathcal{X}_t, \mathcal{Y}_t)$	Joint distribution of inputs and labels in the target domain
$\pi_{ss}(\cdot)$	Self-supervised task predictor
Θ	Complete set of model parameters
Θ_g	Parameters of the shared encoder
Θ_m	Parameters specialized for the main task
Θ_{ss}	Parameters specialized for the self-supervised task
$\mathcal{L}_{\text{m}}$	Main task loss
$\mathcal{L}_{\text{ss}}$	Self-Supervised loss
$\mathcal{L}_{\text{test}}$	Test-time adaptation loss (e.g., self-supervised loss or entropy loss)
$\mathcal{L}_{\text{ent}}$	Entropy loss used for unsupervised test-time adaptation
$\mathcal{L}_{\text{CL}}$	Contrastive learning loss
μ_s, μ_t	Mean feature vectors of source and target distributions
σ_s, σ_t	Standard deviation of feature vectors of source and target distributions
$p(\hat{y})$	Softmax probability output of the model prediction

γ, β	Scale and shift parameters of Batch Normalization layers
μ_t, σ_t^2	Batch mean and variance at test time for BN adaptation
τ	Temperature parameter in contrastive learning
z	Latent feature representation
$\mathbf{S}_v$	Image-to-image similarity matrix
$\mathbf{S}_t$	Text-to-text similarity matrix
$\mathbf{Q}$	Soft pseudo-label matrix
$\hat{\mathbf{P}}$	Image-to-text prediction matrix
$\mathbf{Z}_v$	Image embeddings (features)
$\mathbf{Z}_t$	Text embeddings (features)

INTRODUCTION

Deep learning has achieved remarkable success across a broad spectrum of computer-vision and machine-learning tasks, often surpassing human-level performance when the training and testing data share the same distribution. This success, however, hinges on a fundamental assumption that training and deployment samples are drawn independently and identically from a common data distribution. In practice, this assumption rarely holds. Once a model is deployed, it frequently encounters data that differ from its training distribution due to variations in sensors, environments, styles, or acquisition settings. This mismatch between training and testing distributions, commonly referred to as domain shift or distribution shift, can substantially degrade performance and reliability.

Domain shift manifests in diverse forms. In computer vision, it may arise from differences in illumination, weather, camera characteristics, rendering engines, or imaging modalities. In medical imaging, for instance, the same pathology may appear visually different across scanners or institutions. In autonomous driving, daytime and nighttime images of the same street can lead to drastically different model predictions. Such discrepancies can have critical implications for safety-sensitive applications, where performance failures are unacceptable.

Despite the enormous progress of modern architectures such as Convolutional Neural Networks (CNNs) and Vision Transformers (ViTs), and the availability of vast curated datasets, models remain vulnerable to even minor shifts in input statistics. Empirical studies consistently show that accuracy drops sharply when models are evaluated outside their training domain. For example, a classifier trained on clean, natural images may underperform when exposed to corrupted, blurred, or stylized inputs that alter low-level textures while preserving semantics. These limitations highlight a central weakness of deep networks—their tendency to memorize superficial cues in the data rather than learning domain-invariant, causal representations.

Table 0.1 Comparison of major learning paradigms for handling domain shift. Source data includes inputs x^s and labels y^s; target data includes inputs x^t and, when available, labels y^t. Each paradigm differs in the availability of target data, the loss functions used during training or testing, and the overall adaptation strategy

Paradigm	**Source Data**	**Target Data**	**Train Loss**	**Test Loss**	**Key Idea**
Fine-tuning	–	x^t, y^t	$\mathcal{L}(x^t, y^t)$	–	Adjust pretrained weights using labeled target samples to specialize the model for the new domain.
Domain Adaptation	x^s, y^s	x^t	$\mathcal{L}(x^s, y^s) + \mathcal{L}(x^s, x^t)$	–	Leverages unlabeled target data alongside source data during training to align source and target distributions.
Domain Generalization	x^s, y^s	–	$\sum \mathcal{L}(x^s, y^s)$	–	Learns domain-invariant representations from one or multiple source domains during training.
Test-Time Training	x^s, y^s	x^t	$\mathcal{L}(x^s, y^s) + \mathcal{L}_{aux}(x^s)$	$\mathcal{L}_{aux}(x^t)$	Trains with an auxiliary self-supervised loss and adapts using it at test time.
Test-Time Adaptation	–	x^t	–	$\mathcal{L}(x^t)$	Adapts model weights at inference using unlabeled target data, without requiring source samples.

More recently, foundation models trained on large and heterogeneous data—such as contrastive Vision Language Models (VLMs) (Radford *et al.*, 2021a)—have demonstrated unprecedented breadth of generalization. Yet even these models, despite their scale and diversity, remain susceptible to domain shifts. When transferred to new tasks or specialized environments, they can exhibit non-trivial performance degradation. This underscores that sheer data scale and model capacity alone do not guarantee robustness to distributional change. Addressing domain shift, therefore, remains a fundamental challenge for deploying deep learning systems in open-world and safety-critical settings.

Over the past decade, numerous strategies have been proposed to mitigate the negative effects of domain shift. These approaches differ primarily in when the model is adapted (during training or deployment) and what information from the target domain is accessible (labeled, unlabeled, or none). Broadly, these techniques can be grouped into five main paradigms: **fine-tuning**, **domain adaptation**, **domain generalization**, **test-time training**, and **test-time adaptation**. To contextualize existing research efforts, Table 0.1 summarizes the main learning paradigms

for handling domain shift, distinguished by the stage of adaptation and the availability of target-domain data.

These paradigms form a continuum from training-time to deployment-time adaptation. Fine-tuning and domain adaptation (Chidlovskii, Clinchant & Csurka, 2016; Wilson & Cook, 2020) methods assume access to target data before deployment, which limits their practicality when the target domain is unknown, evolving, or cannot be shared due to privacy constraints. In contrast, Domain Generalization (DG) (Gulrajani & Lopez-Paz, 2020; Wang *et al.*, 2022a) seeks to build models that perform robustly on unseen domains a priori, without ever observing target data. This setting is particularly challenging, as the model must infer domain-invariant cues solely from the variability present in its source domains, without knowing what type of shift it will eventually face.

Beyond DG, Test-Time Training (TTT) and Test-Time Adaptation (TTA) (Wang, Shelhamer, Liu, Olshausen & Darrell, 2020a) shift the adaptation process to the inference phase, where the model updates itself using only unlabeled target samples. While related in spirit, the two differ in their requirements. TTT relies on an auxiliary self-supervised task (e.g., rotation prediction or feature reconstruction) that is co-trained with the main task during source training. At deployment, this auxiliary loss serves as the sole signal to fine-tune the model on test data. In contrast, TTA requires no auxiliary head or joint training—it directly adapts model parameters at test time using unsupervised objectives derived from the prediction itself, such as entropy minimization, confidence calibration, or pseudo-label consistency.

With the emergence of large-scale foundation models, which are pre-trained on massive and diverse datasets, retraining or modifying their full weights during training has become increasingly impractical. In real-world deployments, data often arrive continuously, and distribution shifts are unpredictable. These trends make test-time adaptation an appealing and realistic direction—allowing models to retain the strengths of foundation pretraining while

dynamically adapting to new conditions. However, this setting remains highly challenging, as adaptation must occur without supervision, under unpredictable test conditions.

0.1 Research Statement

This thesis investigates novel methods to enable deep learning models to maintain reliable performance under distribution shifts when neither labeled target data nor retraining is feasible. It focuses on two complementary paradigms that address this challenge at different stages of deployment. The first, training-time generalization, or Domain Generalization (DG), aims to encourage models to learn domain-invariant and style-agnostic representations that remain robust to unseen environments. The second, fully test-time adaptation (TTA), seeks to adjust models dynamically to distributional changes during deployment using only unlabeled test samples—without requiring retraining, architectural modifications, or access to the original training data. This setting has become particularly relevant with the emergence of large-scale foundation models, such as VLMs, where source data are often inaccessible and there is an increasing need for novel adaptation methods that can operate effectively at deployment, even with few target samples, without relying on retraining or labeled supervision.

This thesis was carried out within the framework of an industrial research chair in collaboration with Zebra Technologies. Alongside the academic work presented here, several applied vision projects were conducted in partnership with Zebra, providing a valuable perspective on the relevance of the research directions pursued in this thesis.

0.2 Contributions

As outlined above, this research focuses on developing novel *Domain Generalization* methods for traditional vision architectures and fully *Test-Time Adaptation* strategies for large-scale foundation models. The main contributions of this thesis are organized into four chapters:

- **Chapter 2**: we introduce TFS-ViT (Token-Level Feature Stylization for Vision Transformers) Noori *et al.* (2024b), the first token-level stylization framework for Vision Transformers (ViTs) aimed at improving robustness to unseen domains. The proposed DG method stylizes token-level features by mixing normalization statistics across samples from different domains, encouraging ViTs to learn structure- rather than texture-dependent representations during training. An attention-aware variant (ATFS-ViT) further leverages class-token attention maps to selectively stylize salient regions, enhancing style invariance and interpretability. This approach achieves state-of-the-art domain generalization performance across multiple DG benchmarks.

  Related publication:

  - TFS-ViT: Token-Level Feature Stylization for Domain Generalization, published in *Pattern Recognition*, vol. 149, p. 110213, 2024.

- **Chapter 3**: we propose FDS (*Feedback-Guided Domain Synthesis*) Noori *et al.* (2025a), a novel generative framework that, for the first time, employs a single conditional diffusion model trained jointly on multiple source domains to synthesize *pseudo-domains* bridging the gaps between source distributions. The diffusion model is conditioned on both class and domain information to generate diverse yet semantically consistent samples that enrich the source domain space. Furthermore, an entropy-based feedback mechanism identifies and selects challenging synthetic samples, ensuring that the augmented data effectively expands distributional coverage. When these pseudo-domain samples are presented to the classifier, they encourage the learning of domain-invariant and robust feature representations, substantially improving generalization to unseen domains. This framework achieves state-of-the-art performance across standard DG benchmarks.

  Related publication:

- FDS: Feedback-Guided Domain Synthesis with Multi-Source Conditional Diffusion Models for Domain Generalization, published in the *IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2025.

- **Chapter 4**: we present MLMP (*Multi-Layer Multi-Prompt Test-Time Adaptation*) Noori *et al.* (2025b), the first plug-and-play framework designed to adapt VLMs for Open-Vocabulary Semantic Segmentation (OVSS) entirely at test time. Unlike prior TTA methods limited to image classification, MLMP introduces a dual strategy that combines *adaptive multi-level feature fusion*, aggregating intermediate vision-encoder layers weighted by entropy-based confidence, with *multi-prompt optimization* across diverse text templates. This joint formulation leverages VLMs' inherent prompt sensitivity as a stable adaptation signal, enabling robust performance under severe distribution shifts and even with a single test sample. Furthermore, we establish the first comprehensive OVSS-TTA benchmark suite encompassing nine segmentation datasets and over eighty test scenarios, providing a standardized protocol for future research in this field. Extensive experiments demonstrate that MLMP achieves consistent improvements over strong TTA baselines while maintaining a lightweight computational footprint.

  Related publication:

  - MLMP: Multi-Layer Multi-Prompt Test-Time Adaptation of Vision-Language Models, published in the *Conference on Neural Information Processing Systems (NeurIPS)*, 2025.

- **Chapter 5**: we introduce **Histopath-C** Noori *et al.* (2026), the first benchmark for evaluating TTA of VLMs in digital histopathology, and **LATTE**, the first TTA method specifically designed for this setting. Histopath-C simulates clinically grounded corruptions across ten categories, enabling dynamic, on-the-fly evaluation of model robustness. Building upon this benchmark, LATTE (*Low-rank Adaptation with Transductive Template Ensembling*) proposes a lightweight yet highly effective adaptation strategy that integrates *transductive*

vision–text pseudolabeling, *loss-level multi-template aggregation*, and *low-rank parameter updates* to mitigate prompt sensitivity and instability. Importantly, this work extends the scope of TTA to a critical real-world application domain—*medical imaging*—where models must often adapt on the fly to unseen data distributions, making it one of the most practically relevant use cases of TTA. Extensive experiments across multiple datasets and medical VLMs demonstrate that LATTE achieves substantial robustness gains over existing TTA baselines.

Related publication:

- Histopath-C: Towards Realistic Domain Shifts for Histopathology Vision–Language Adaptation, published in the *IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2026.

0.2.1 Additional Contributions

In addition to the main works presented in this thesis, several complementary studies were conducted that contributed to the broader understanding of model robustness under distribution shifts.

- We propose a multi-task domain generalization framework that enhances feature stylization by enforcing structural consistency across style-augmented samples Cheraghalikhani *et al.* (2024). The method combines instance-wise feature mixing with an auxiliary edge-reconstruction objective, enabling the network to preserve semantic and structural cues while improving robustness to unseen domains.

  Related publication:

  - *Structure-Aware Feature Stylization for Domain Generalization*, published in *Computer Vision and Image Understanding (CVIU)*, July 2024.

- We introduce a novel TTA framework for VLMs based on weight averaging theory to achieve stable and efficient adaptation under distribution shift without requiring retraining or access to source data Osowiechi *et al.* (2024b).

  Related publication:

  - *WATT: Weight Average Test-Time Adaptation of CLIP*, published in the *Conference on Neural Information Processing Systems (NeurIPS)*, 2024.

CHAPTER 1

LITERATURE REVIEW

1.1 Scope and positioning

Deep learning models have achieved remarkable generalization within their training distributions, yet their reliability deteriorates sharply under distribution shifts—a pervasive challenge across computer vision, medical imaging, and open-world deployment scenarios. Numerous research directions have emerged to address this problem, varying primarily in when adaptation occurs (during training or deployment) and what form of target information is available (labeled, unlabeled, or none). Among these, *Domain Generalization* (DG) and *Test-Time Adaptation* (TTA) have gained increasing attention for their practicality and data-efficiency. Both paradigms aim to enhance robustness without requiring access to labeled target data, but they approach the challenge from complementary angles: DG seeks to train models that generalize to unseen domains a priori, while TTA focuses on dynamically adapting pre-trained models during inference.

This chapter provides a comprehensive review of prior research related to these two paradigms, emphasizing advances most relevant to the contributions of this thesis. Specifically, we focus on: (i) DG methods for conventional vision architectures which aim to learn domain-invariant and style-agnostic representations through data-, feature-, and objective-level strategies; and (ii) TTA methods for both standard discriminative models and large-scale *foundation models*, including Vision–Language Models (VLMs) such as CLIP Radford *et al.* (2021a), that enable adaptation in fully test-time, source-free settings.

The discussion is organized to bridge traditional robustness research with emerging foundation-model adaptation trends. We first formalize the DG problem and review the major methodological categories, including data augmentation, feature stylization, weight-averaging ensembles, and recent diffusion-based generative approaches. We then shift focus to the TTA literature, tracing

the evolution from classic entropy-minimization and normalization-based techniques to modern foundation-model adaptation strategies.

In the subsequent chapters, this thesis builds upon the insights from the above literature to advance robustness under domain shift from both training and deployment perspectives. Chapters 2 and 3 advance the DG frontier by introducing token-level stylization and diffusion-based pseudo-domain synthesis, while Chapters 4 and 5 extend TTA to foundation models, open-vocabulary segmentation, and clinically realistic histopathology scenarios, which can be considered as one of the most important applications of TTA. Together, these studies contribute to a unified understanding of how models can generalize and adapt effectively under domain shift, both before and after deployment.

1.2 Domain Shift

The phenomenon of *domain shift*—also referred to as distribution shift or dataset bias—arises when the data encountered during deployment differ from those used for training. In such cases, the statistical relationship between inputs and labels changes, leading to a degradation in model performance. Domain shift is ubiquitous across learning modalities, manifesting in various forms such as changes in illumination, texture, sensor characteristics, or even linguistic style and acoustic environment. Figure 1.1 illustrates examples of domain shift across multiple data modalities, highlighting that this problem extends beyond vision to text, audio, and time-series domains.

Despite its generality, this thesis focuses on the vision domain, where domain shift remains a central obstacle to reliable deployment of deep models in open-world conditions. Vision models trained on curated datasets often fail when exposed to new styles, unseen sensors, or realistic corruptions, underscoring the need for methods that can either generalize across domains or adapt efficiently at test time.

Formally, let $P(\mathcal{X}_s, \mathcal{Y}_s)$ denote the joint distribution over input–label pairs $(\mathcal{X}_s, \mathcal{Y}_s)$ in the *source* domain, and $P(\mathcal{X}_t, \mathcal{Y}_t)$ the corresponding distribution in the *target* domain. Domain shift occurs

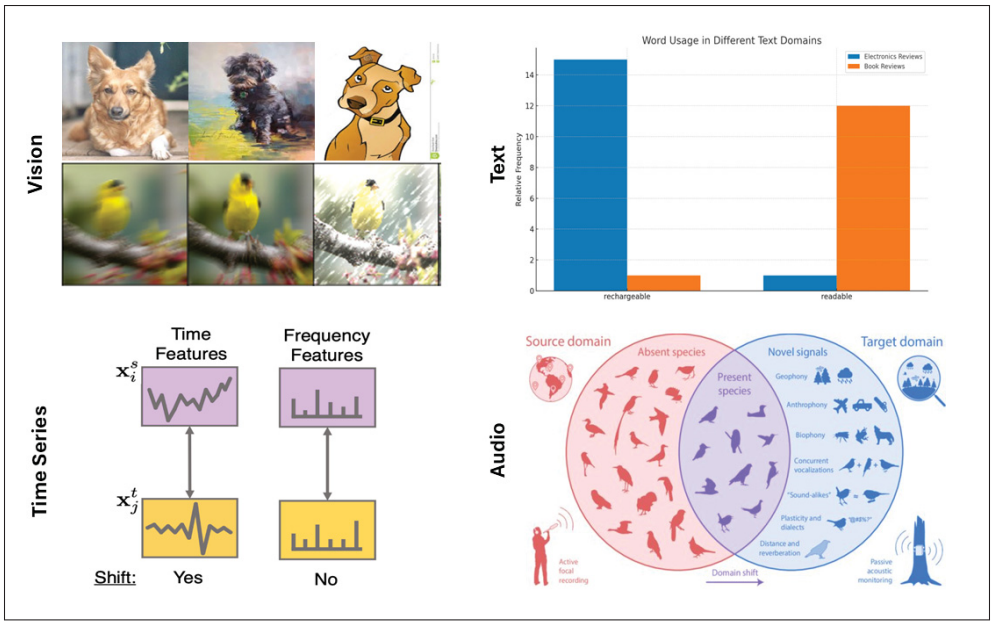

Figure 1.1 Illustration of domain shift across different modalities. Examples include changes in texture or rendering style in vision, vocabulary or sentiment polarity in text, time–frequency signal variations in time series, and environmental or species-related differences in audio

when these two distributions differ:

$$P(\mathcal{X}_s, \mathcal{Y}_s) \neq P(\mathcal{X}_t, \mathcal{Y}_t) \tag{1.1}$$

A common and practically relevant case is the *likelihood shift*, where the conditional distributions of inputs given labels differ between domains, while the label space remains shared:

$$P(\mathcal{X}_s|\mathcal{Y}_s) \neq P(\mathcal{X}_t|\mathcal{Y}_t), \quad \text{with} \quad \mathcal{Y}_s = \mathcal{Y}_t. \tag{1.2}$$

This formulation captures many real-world scenarios where the semantic categories are identical, but their visual appearance changes due to factors such as acquisition conditions, sensor types, or environmental variations. Addressing such shifts forms the foundation of the methods discussed in the remainder of this chapter.

1.3 Domain Generalization Research

1.3.1 Problem Definition

Let $\mathcal{X}$ and $\mathcal{Y}$ denote the input and label spaces, respectively. A *domain* for a classification task is defined as a joint probability distribution $P_{\mathcal{XY}}$ over $\mathcal{X}\times\mathcal{Y}$. For each domain, we denote by $P_{\mathcal{X}}$ the marginal distribution over inputs, $P_{\mathcal{Y}|\mathcal{X}}$ the posterior distribution of labels given inputs, and $P_{\mathcal{X}|\mathcal{Y}}$ the class-conditional distribution of inputs given labels.

In the standard *Domain Generalization* (DG) setting, we are given a collection of M related but distinct *source domains* $\mathcal{S} = \{S_i\}_{i=1}^{M}$, where each domain S_i is associated with its own joint distribution $P_{\mathcal{XY}}^{(i)}$. We assume these distributions differ, i.e., $P_{\mathcal{XY}}^{(i)} \neq P_{\mathcal{XY}}^{(i')}$ for $i \neq i'$. Each source domain provides N_i labeled samples $S_i = \{(x_j^{(i)}, y_j^{(i)})\}_{j=1}^{N_i}$. At deployment, the model encounters an unseen *target domain* $T = \{x_j^{(t)}\}_{j=1}^{N_t}$ with distribution $P_{\mathcal{XY}}^{(t)}$ that is distinct from those of the sources. The target labels are unknown during both training and adaptation. The objective of DG is to learn a predictive function $f : \mathcal{X} \to \mathcal{Y}$ that minimizes the expected risk on unseen domains, i.e., to find $f \in \mathcal{F}$ minimizing the target risk $\mathbb{E}_{(x,y)\sim P_{\mathcal{XY}}^{(t)}}[\mathcal{L}(f(x), y)]$ using only labeled samples from the sources. In practice, DG algorithms aim to capture invariances across the observed source domains that are likely to transfer to new, unseen domains.

Figure 1.2 summarizes the major methodological categories explored in the domain generalization literature. These include data-level augmentation techniques that diversify training samples via input-level transformations (Somavarapu, Ma & Kira, 2020; Yue *et al.*, 2019; Zhou, Yang, Hospedales & Xiang, 2020a); meta-learning frameworks that simulate domain shifts during training (Sharifi-Noghabi, Asghari, Mehrasa & Ester, 2020); self-supervised learning using auxiliary pretext tasks (Carlucci, D'Innocente, Bucci, Caputo & Tommasi, 2019a; Albuquerque, Naik, Li, Keskar & Socher, 2020); disentangled representation learning to separate domain-invariant and domain-specific features(Ilse, Tomczak, Louizos & Welling, 2020); domain alignment strategies enforcing statistical similarity across domains (Wang, Loog & Van Gemert,

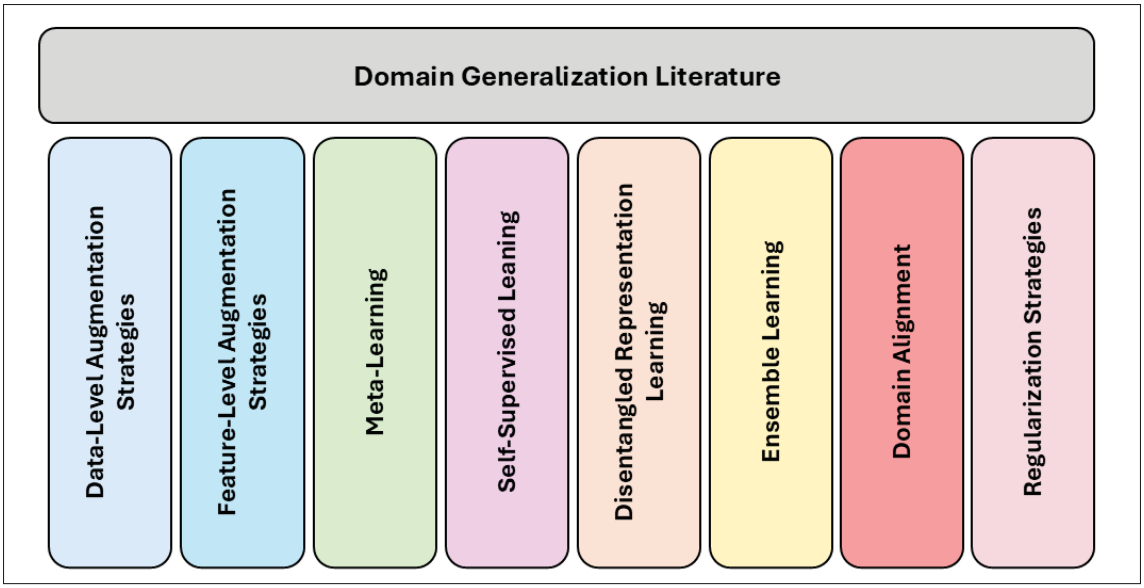

Figure 1.2 Taxonomy of major approaches in Domain Generalization

2021c; Jin, Lan, Zeng & Chen, 2020); and regularization-based approaches that penalize reliance on non-generalizable patterns (Huang, Wang, Xing & Huang, 2020).

While each category in Figure 1.2 contributes to advancing domain generalization, recent research has increasingly concentrated on a subset of particularly promising strategies. In this review, we highlight three directions that have demonstrated strong empirical performance and conceptual novelty: (1) *feature-level augmentation* methods that perturb internal representations to encourage structural or style-invariant learning; (2) *ensemble and weight-averaging* techniques that improve robustness by aggregating diverse hypotheses or navigating toward flatter minima; and (3) *diffusion-based methods*, a recent frontier that employs generative modeling to synthesize additional data to promote generalization.

1.3.2 Feature-Level Augmentation Methods

A key strategy for improving domain generalization involves augmenting the data or the intermediate feature representations to expose the model to a broader range of domain variations. Depending on where the augmentation occurs, such approaches can be broadly classified into

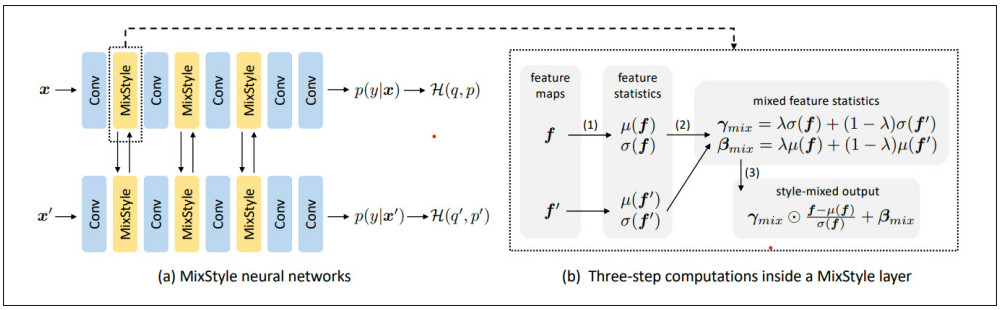

Figure 1.3 Overview of the MixStyle architecture
Taken from Zhou *et al.* (2024a)

data-level methods—which operate directly on image pixels—and *feature-level* methods—which perturb the latent features extracted by the network. In this section, we focus on *feature-level* augmentation techniques that manipulate internal representations to improve robustness across domains.

MixStyle (Zhou *et al.*, 2024a) proposed a plug-and-play, parameter-free module that can be seamlessly integrated between convolutional layers in CNNs. The method generates novel styles by mixing the feature statistics of two random instances using a convex combination with randomly sampled weights. The idea builds upon the principle of Adaptive Instance Normalization (AdaIN) (Huang & Belongie, 2017), which demonstrated that arbitrary style transfer can be achieved by manipulating channel-wise feature statistics.

Specifically, MixStyle randomly selects two random instances $(x, \tilde{x})$ from a batch and computes their corresponding feature statistics as:

$$\begin{aligned} \gamma_{mix} &= \lambda\sigma(x) + (1-\lambda)\sigma(\tilde{x}) \\ \beta_{mix} &= \lambda\mu(x) + (1-\lambda)\mu(\tilde{x}). \end{aligned} \tag{1.3}$$

In this equations, λ denotes instance-specific mixing coefficients drawn from a Beta distribution, while $\mu(x)$ and $\sigma(x)$ represent the channel-wise mean and standard deviation, computed across the spatial dimensions of each instance as follows:

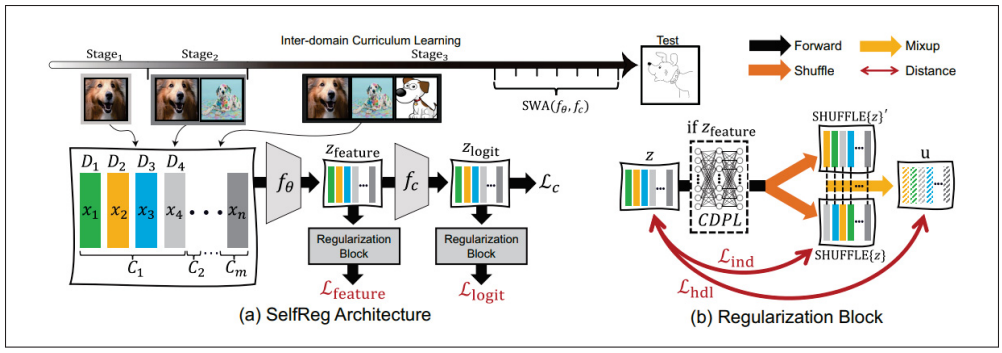

Figure 1.4 Overview of the SelfReg architecture
Taken from (Kim *et al.*, 2021b)

$$\begin{aligned}\mu(x)_{b,c} &= \frac{1}{HW}\sum_{h=1}^{H}\sum_{w=1}^{W} x_{b,c,h,w} \\ \sigma(x)_{b,c} &= \sqrt{\frac{1}{HW}\sum_{h=1}^{H}\sum_{w=1}^{W}(x_{b,c,h,w}-\mu(x)_{b,c})^2}\end{aligned} \tag{1.4}$$

The resulting mixed statistics are then formulated as

$$\text{MixStyle}(x) = \gamma_{mix} \odot \frac{x-\mu(x)}{\sigma(x)} + \beta_{mix}, \tag{1.5}$$

The overview of MixStyle architecture is shown in Fig. 1.3.

SelfReg (Kim *et al.*, 2021b) is another feature-level augmentation/regularization scheme that aligns *same-class* representations across source domains using only positive pairs—avoiding the instability of mining/maintaining negatives. Let $\mathbf{z}_i^c = f_\theta(x_i)$ be the latent feature of sample x_i with class c, and let f_{CDPL} denote a light projection head (Class-Specific Domain Perturbation Layer) that helps prevent representation collapse and supports inter-domain mixup. The *individualized in-batch dissimilarity* pulls a feature toward a shuffled same-class counterpart after CDPL:

$$\mathcal{L}_{\text{ind}}(\mathbf{z}) = \frac{1}{N}\sum_{i=1}^{N}\left\|\mathbf{z}_i^c - f_{\text{CDPL}}\left(\mathbf{z}_j^c\right)\right\|_2^2, \quad j \neq i,\ y_j{=}y_i. \tag{1.6}$$

To expose the model to cross-domain convex combinations while preserving class identity, SelfReg performs a same-class, two-domain mixup *in feature space* after CDPL. With $\gamma \sim \text{Beta}(\alpha, \beta)$ and a same-class partner j,

$$\bar{\mathbf{u}}_i^c = \gamma\, \mathbf{u}_i^c + (1-\gamma)\, \mathbf{u}_j^c, \qquad \mathbf{u}_\cdot^c := f_{\text{CDPL}}(\mathbf{z}_\cdot^c), \tag{1.7}$$

and the *heterogeneous in-batch dissimilarity* encourages consistency between original and mixed features:

$$\mathcal{L}_{\text{hdl}}(\mathbf{z}) = \frac{1}{N} \sum_{i=1}^{N} \left\| \mathbf{z}_i^c - \bar{\mathbf{u}}_i^c \right\|_2^2 . \tag{1.8}$$

Both penalties are applied at the feature and logit levels and combined as

$$\mathcal{L}_{\text{SelfReg}} = \lambda_{\text{feature}} \mathcal{L}_{\text{feature}} + \lambda_{\text{logit}} \mathcal{L}_{\text{logit}}, \tag{1.9}$$

yielding the overall training objective

$$\mathcal{L} = \mathcal{L}_{\text{c}} + \mathcal{L}_{\text{SelfReg}}. \tag{1.10}$$

Because $\mathcal{L}_{\text{SelfReg}}$ can dominate late in training, gradient stabilization is used: (i) *loss clipping* that scales SelfReg gradients by $\min(1.0, \mathcal{L}_{\text{c}})$; (ii) *stochastic weight averaging* (SWA), which averages snapshots to favor flatter minima,

$$\boldsymbol{\omega}_{\text{swa}} = \frac{1}{k+1} \sum_{i=0}^{k} \boldsymbol{\omega}_{m+ic}, \tag{1.11}$$

and (iii) *inter-domain curriculum learning* (IDCL), which orders source domains from "closer" to "farther" to reduce conflicting gradients. The overview of SelfReg architecture is shown in Fig. 1.4.

While these feature-level augmentation methods have proven effective for CNN architectures, the characteristic components of Vision Transformers (ViTs)—such as their token-based representations and self-attention mechanisms—have not yet been explored for domain

generalization. Consequently, the literature still lacks a feature-level augmentation framework specifically tailored to ViTs.

1.3.3 Ensemble and Weight Averaging

Ensembling and weight-averaging methods are effective strategies to enhance generalization by reducing variance and guiding optimization toward *flatter* regions of the loss landscape. In the context of domain generalization, these flatness-oriented approaches help improve robustness against distribution shifts. This subsection reviews several representative schemes proposed in this category.

Stochastic Weight Averaging Densely (SWAD) (Cha *et al.*, 2021) tackles domain shift by *seeking flat minima* rather than modifying architectures or objectives. While vanilla ERM may converge to sharp optima that fail under domain shift, SWAD is theoretically motivated by *robust risk minimization* (RRM), which favors parameters whose loss remains low within a neighborhood of the weights. SWAD operationalizes flatness via a dense, overfit-aware variant of stochastic weight averaging:

1. *Dense sampling:* Instead of averaging every K epochs (as in SWA), collect weights at *every iteration*, yielding a richer set of stochastic solutions.
2. *Overfit-aware window:* Use validation loss to select the averaging window $[t_s, t_e]$:

$$t_s := \min\{t \mid \hat{E}_{\text{val}}(t) \text{ does not decrease for } N_s \text{ consecutive iters}\}, \tag{1.12}$$

$$t_e := \min\{t > t_s \mid \hat{E}_{\text{val}}(t) > r \cdot \hat{E}_{\text{val}}(t_s) \text{ for } N_e \text{ iters}\}, \tag{1.13}$$

   with patience hyperparameters N_s, N_e and tolerance $r > 1$.

3. *Averaging:* The final SWAD weights are the uniform average over the selected window:

$$\theta_{\text{SWAD}} = \frac{1}{t_e - t_s + 1} \sum_{t=t_s}^{t_e} \theta_t. \tag{1.14}$$

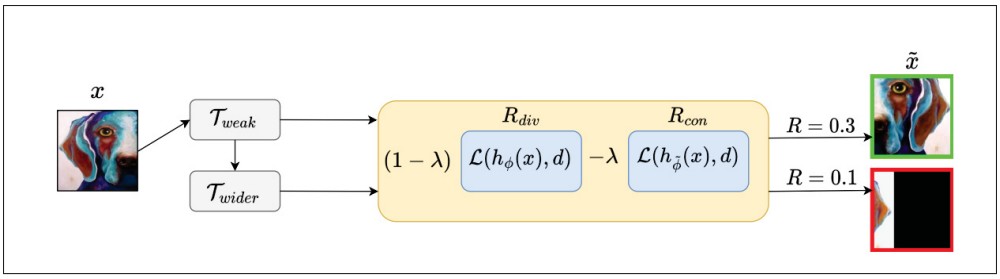

Figure 1.5 Overview of the DCAug Scheme
Taken from (Aminbeidokhti *et al.*, 2024a)

DCAug (Aminbeidokhti *et al.*, 2024a) is another ensemble training strategy that implicitly benefits from averaging across multiple augmentation-induced views of each sample. The method extends standard random augmentation by expanding the transformation magnitude range and introducing a rejection mechanism to discard harmful or overly distorted samples. For each input, DCAug selects between a weak (safe) and a strong (diverse) transformation based on a reward function based selection rule:

$$\tilde{x} = \begin{cases} T_{\text{wider}}(x), & \text{if } R(T_{\text{wider}}(x), z) \geq R(T_{\text{weak}}(x), z), \\ T_{\text{weak}}(x), & \text{otherwise.} \end{cases} \tag{1.15}$$

The reward function evaluates the *quality* of each augmented sample by jointly balancing *diversity* and *semantic consistency*:

$$R(\tilde{x}, z) = (1-\lambda)\, R_{\text{div}}(\tilde{x}, z) - \lambda\, R_{\text{con}}(\tilde{x}, z), \tag{1.16}$$

where λ controls the trade-off between the two objectives, and z represents either the domain label (d) or the class label (y). The diversity reward encourages exploration through higher student loss, while the consistency reward—computed using an exponential moving average (EMA) teacher—ensures that the semantic meaning of the augmented sample is preserved. By maximizing this unified reward, DCAug adaptively selects transformations that are challenging

yet semantically valid, leading to improved robustness and generalization. To implement the reward terms, DCAug adopts a teacher-student scheme where the diversity loss is computed using the student model, and the consistency loss is measured using its exponential moving average (EMA) teacher. For the domain-aware setting $\text{DCAug}_{\text{domain}}$, the rewards are defined as:

$$R_{\text{div}}(x, d) = L(h_{\phi}(x), d), \tag{1.17}$$

$$R_{\text{con}}(x, d) = L(h_{\tilde{\phi}}(x), d), \tag{1.18}$$

where $\tilde{\phi} = (1-\beta)\phi + \beta\tilde{\phi}$, and β controls the smoothness of the exponential moving average. This teacher–student mechanism allows DCAug to preserve semantic consistency while continuously adapting to diverse augmentations. The overview of DCAug scheme is shown in Fig. 1.5.

1.3.4 Diffusion-Based Methods

Recent studies have explored the use of diffusion models for domain generalization due to their strong generative and distributional modeling capabilities. Diffusion-based approaches aim to explicitly model or manipulate the underlying data distribution to improve robustness under domain shift. By leveraging their ability to learn the data manifold through iterative denoising, these methods can either synthesize diverse samples for training or adapt test-time samples to match the source distribution.

(Yu *et al.*, 2023b) proposed *Distribution Shift Inversion* (DSI), a diffusion-based framework that mitigates domain shift by transforming out-of-distribution (OoD) samples toward the source distribution before prediction. Instead of enforcing invariance during training, DSI reuses a diffusion model trained only on the source data to perform a two-step process: a noisy alignment that mixes the OoD input with Gaussian noise to match the diffusion model's noise space, followed by a reverse diffusion process that maps back toward the source manifold to obtain a semantically consistent sample. Theoretically, the generated distribution ω approaches the

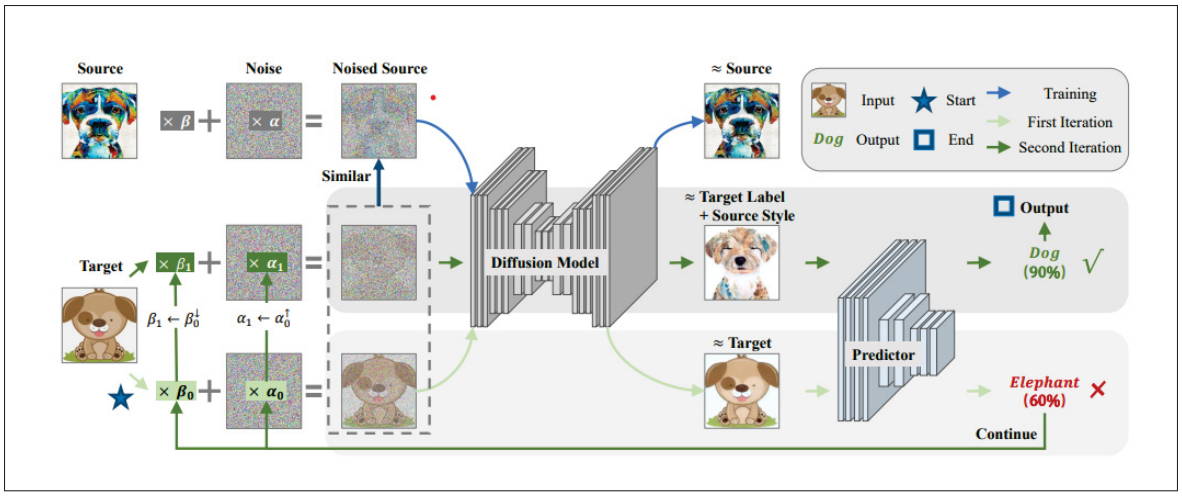

Figure 1.6 Overview of the Distribution Shift Inversion (DSI) framework
Taken from (Yu *et al.*, 2023b)

source distribution p under the bound

$$\mathrm{KL}(p\|\omega) \le \mathcal{J}_{\mathrm{SM}} + \mathrm{KL}(p_T\|\rho) + \mathcal{F}(\alpha), \tag{1.19}$$

where $\mathcal{J}_{\mathrm{SM}}$ denotes the score-matching error, $\mathrm{KL}(p_T\|\rho)$ measures the diffusion approximation, and $\mathcal{F}(\alpha)\to 0$ as the alignment strength α increases. By effectively reducing the distribution gap prior to prediction, DSI provides a lightweight and model-agnostic mechanism. In practice, it forms a noise-aligned input $\hat{\boldsymbol{x}} = \beta\boldsymbol{x} + \alpha\boldsymbol{\epsilon}$ at a chosen diffusion start time s, where (α, β) are determined by the forward noise schedule up to s. A source-trained diffusion model then runs the reverse process from s back to 0 to obtain $\tilde{\boldsymbol{x}}$ that preserves the target label while adopting the source-domain style. During inference, DSI adaptively increases s and repeats the transform–predict cycle until a confidence threshold is reached. The overview of DSI framework is shown in Fig. 1.6.

1.4 Test-Time Adaptation Research

1.4.1 Problem Definition

Let $\mathcal{X}$ and $\mathcal{Y}$ denote the input and label spaces, respectively. A *domain* is characterized by a joint probability distribution $P_{\mathcal{X}\mathcal{Y}}$ defined over $\mathcal{X}\times\mathcal{Y}$, with $P_{\mathcal{X}}$, $P_{\mathcal{Y}|\mathcal{X}}$, and $P_{\mathcal{X}|\mathcal{Y}}$ denoting the marginal, posterior, and class-conditional distributions, respectively.

In the TTA setting, a model f_θ is trained on one or multiple *source domains* $\mathcal{S} = \{S_i\}_{i=1}^{M}$ with corresponding joint distributions $\{P^{(i)}_{\mathcal{X}\mathcal{Y}}\}$, using labeled samples $S_i = \{(x_j^{(i)}, y_j^{(i)})\}_{j=1}^{N_i}$. At deployment, the model encounters data from an unseen *target domain* $T = \{x_j^{(t)}\}_{j=1}^{N_t}$ drawn from a different distribution $P^{(t)}_{\mathcal{X}\mathcal{Y}}$, such that $P^{(t)}_{\mathcal{X}\mathcal{Y}} \neq P^{(i)}_{\mathcal{X}\mathcal{Y}}$ for all $i \in \{1, \ldots, M\}$.

Unlike DG, TTA assumes access to unlabeled target samples at inference time, allowing the model to update its parameters or internal statistics using an unsupervised objective. Formally, given a pre-trained model f_{θ_0}, the goal is to obtain adapted parameters θ_t for each batch or instance of test data by minimizing an unsupervised loss $\mathcal{L}_{\text{unsup}}$, typically derived from the model's own predictions, such as entropy minimization, confidence calibration, or consistency regularization. This can be expressed as $\theta_t = \arg\min_\theta \mathbb{E}_{x\sim P^{(t)}_{\mathcal{X}}}\big[\mathcal{L}_{\text{unsup}}(f_\theta(x))\big]$. The adapted model f_{θ_t} is then used to predict labels for samples from the same or subsequent target batches.

Depending on the adaptation protocol, TTA can be *online*—where parameters are updated sequentially across batches—or *episodic*—where the model is reset to θ_0 before adapting to each test batch. The overall objective is to maintain or improve predictive performance under distribution shift using only unlabeled target data, without access to source data or retraining.

1.4.2 Test-Time Adaptation in Conventional Vision Models

Early developments in TTA primarily focused on conventional convolutional and transformer-based vision models trained on standard supervised datasets. Two key ideas have emerged as foundational to most subsequent TTA methods: *adaptation through normalization layers*

and *entropy minimization*. These approaches are often employed jointly and form the basis of modern lightweight adaptation strategies.

Prediction-Time Batch Normalization (Nado *et al.*, 2021) was among the first works to demonstrate that updating the statistics of normalization layers at inference can significantly mitigate the effect of domain shifts. For a given feature activation $f_{i,u}$ in channel u of sample i, the normalized feature $\hat{f}_{i,u}$ is computed as

$$\hat{f}_{i,u} = \frac{f_{i,u} - \mu_u}{\sqrt{\sigma_u^2 + \epsilon}}, \tag{1.20}$$

where $\mu_u = \frac{1}{N}\sum_{i=1}^{N} f_{i,u}$ and $\sigma_u^2 = \frac{1}{N}\sum_{i=1}^{N}(f_{i,u} - \mu_u)^2$ denote the batch mean and variance estimated from the current test batch. Unlike conventional inference, which uses fixed running statistics from training, PTBN dynamically recomputes these statistics on the incoming test data. Although simple, this mechanism allows the normalization layers to align feature distributions with the target domain, effectively reducing covariate shift without modifying model weights. PTBN thus established the importance of normalization alignment as a lightweight and universal adaptation mechanism.

Building upon PTBN, TENT (Wang, Shelhamer, Liu, Olshausen & Darrell, 2021a) introduced a gradient-based formulation that directly optimizes the model parameters at test time to produce more confident predictions. This approach employs the Shannon conditional entropy of the model's predictive distribution as an unsupervised loss function, often referred to as *entropy minimization*. Given the predicted class probabilities $\hat{y}_{i,c} = p(y_c|x_i)$ for sample x_i, the loss is defined as

$$\mathcal{L}_{\text{TENT}} = -\frac{1}{N}\sum_{i}\sum_{c}\hat{y}_{i,c}\log\hat{y}_{i,c}. \tag{1.21}$$

By minimizing this objective with respect to the model parameters, TENT encourages the network to make confident and consistent predictions on unlabeled target data. In practice, only the affine parameters of normalization layers (i.e., scale and bias) are updated, while the rest of the network remains frozen—striking a balance between flexibility and stability.

Although TENT does not fully resolve all adaptation challenges, its simplicity, generality, and effectiveness have made it one of the most widely adopted TTA baselines. Subsequent methods, such as entropy-regularized or sharpness-aware variants, build upon its core principle to improve stability, prevent overfitting, or extend adaptation to continual and dynamic test streams.

The Sharpness-Aware and Reliable Entropy Minimization (**SAR**) method (Niu *et al.*, 2023) extends entropy-based TTA by addressing two major limitations of earlier approaches such as TENT: sensitivity to noisy target samples and instability under severe domain shifts. SAR introduces the concept of *sharpness-aware regularization*, inspired by Sharpness-Aware Minimization (SAM) (Kirillov *et al.*, 2023), to encourage the model to converge toward *flat minima* in the loss landscape, thereby improving stability and generalization to unseen domains.

In conventional optimization, sharp minima correspond to regions where small parameter perturbations can cause large increases in loss, making the model highly sensitive to input or distributional noise. To counteract this, SAR formulates test-time adaptation as a bi-level optimization problem that minimizes an adversarially perturbed version of the entropy loss:

$$\min_{\theta} \mathcal{H}^{\mathrm{SA}}(\hat{\mathbf{y}};\theta), \quad \text{where} \quad \mathcal{H}^{\mathrm{SA}}(\hat{\mathbf{y}};\theta) = \max_{\|\epsilon\|_2 \le \rho} \mathcal{H}(\hat{\mathbf{y}};\theta+\epsilon), \tag{1.22}$$

where $\mathcal{H}(\hat{\mathbf{y}};\theta)$ denotes the entropy of the model predictions $\hat{\mathbf{y}}$, ϵ represents an adversarial perturbation of the model parameters, and ρ controls the radius of the perturbation ball. The inner maximization identifies the most sensitive perturbation directions (i.e., the sharpest regions), while the outer minimization adjusts the model parameters to reduce the loss sensitivity to such

perturbations. This process encourages the model to settle into flatter regions of the loss surface, yielding more reliable and stable adaptation across batches of test data.

Beyond sharpness-aware optimization, SAR also improves reliability by filtering out noisy or unstable samples during adaptation. Specifically, samples with excessively large gradients, indicative of unreliable predictions or outliers—are excluded from the optimization process. This selective update mechanism prevents the model from overfitting to noisy target instances and ensures that adaptation is guided by more consistent and trustworthy data.

By jointly enforcing flat-minima regularization and noise-aware sample selection, SAR achieves robust and stable test-time adaptation even under dynamic or non-stationary domain shifts. It represents an important step toward making entropy-based TTA methods both more reliable and resilient in real-world, continuously evolving deployment scenarios.

While the aforementioned methods (e.g., TENT and SAR) rely on gradient-based updates of model parameters at test time, an alternative direction explores parameter-free adaptation mechanisms that adjust predictions without modifying the network weights. Among these, **Laplacian Adjusted Maximum-Likelihood Estimation (LAME)** (Boudiaf, Mueller, Ben Ayed & Bertinetto, 2022) stands out as a simple yet powerful approach that achieves effective adaptation entirely in the prediction space.

LAME formulates test-time adaptation as a graph-regularized optimization problem that refines the model's output probabilities based on the local structure of the target data in feature space. For a batch of test samples $\{x_i\}_{i=1}^{N}$ and their corresponding predicted class probabilities $\hat{\mathbf{y}}_i = f_\theta(x_i)$, the method introduces an assignment variable $\tilde{\mathbf{z}}_i = [\tilde{z}_{i1}, \ldots, \tilde{z}_{iC}]$ lying on the probability simplex Δ^{C-1}, which represents the refined label distribution for each sample. The adaptation objective minimizes the divergence between the refined assignments and the model predictions while enforcing smoothness across similar samples in the feature space:

$$\mathcal{L}_{\text{LAME}} = \sum_i \text{KL}(\tilde{\mathbf{z}}_i \,\|\, \hat{\mathbf{y}}_i) - \sum_{i,j} w_{ij}\, \tilde{\mathbf{z}}_i^\top \tilde{\mathbf{z}}_j, \tag{1.23}$$

where $\mathrm{KL}(\cdot\|\cdot)$ denotes the Kullback–Leibler divergence and $w_{ij} = w(f_\theta(x_i), f_\theta(x_j))$ encodes the feature-space similarity between samples x_i and x_j. The first term aligns the refined assignments $\tilde{\mathbf{z}}_i$ with the model's original predictions, while the second Laplacian term promotes label consistency among neighboring samples by encouraging similar features to share similar label distributions.

Unlike gradient-based methods, LAME admits a closed-form analytical solution, making the adaptation process computationally efficient and stable. By avoiding parameter updates, it eliminates the risk of catastrophic forgetting and is well-suited for deployment scenarios with strict memory or time constraints. Moreover, its graph-based regularization enables local consistency among target samples, effectively capturing the manifold structure of the target domain. Despite its simplicity, LAME achieves strong robustness across a variety of domain-shift benchmarks, offering an elegant and efficient alternative to gradient-based TTA.

1.4.3 Test-Time Adaptation in Vision–Language Models

Recent advances in *contrastive Vision–Language Models* (VLMs) have revolutionized multimodal learning by aligning visual and textual representations within a shared embedding space. Models such as CLIP (Radford *et al.*, 2021a), and more recently SigLIP2 (Tschannen *et al.*, 2025) are pre-trained on massive collections of image–text pairs using a contrastive objective. Given an image x and a text description t, the model encodes them as feature vectors $v = f_{\text{img}}(x)$ and $u = f_{\text{text}}(t)$, and optimizes the cosine similarity $\mathrm{sim}(v, u)$ to maximize alignment between matching pairs while minimizing it for mismatched pairs. This contrastive training enables the emergence of rich, semantically aligned representations that generalize across a wide range of visual categories and downstream tasks with minimal supervision. Figure 1.7 illustrates the general architecture of CLIP, where dual encoders for image and text are jointly optimized using a contrastive loss.

The large-scale pretraining and zero-shot transfer capabilities of these models make them highly attractive candidates for TTA. Since VLMs predict similarity scores rather than explicit class

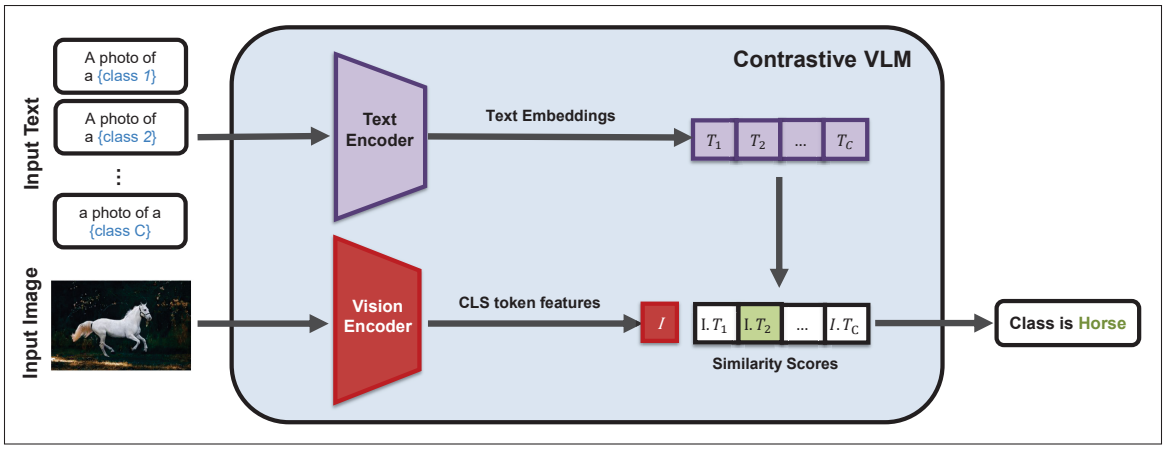

Figure 1.7 Overview of a contrastive Vision–Language Model such as CLIP. The image encoder and text encoder map visual and textual inputs into a shared embedding space, where semantic similarity is optimized using a contrastive loss

probabilities, adapting them to new domains can often be achieved by refining the representations or textual prompts rather than retraining the entire model. Moreover, because these models are trained on billions of diverse image–text pairs, they can be directly deployed to a wide range of downstream applications without task-specific supervision—making online adaptation particularly appealing.

However, despite their impressive generalization, foundation-scale VLMs are not immune to performance degradation under distribution shifts. Recent studies (Osowiechi *et al.*, 2024b) have shown that even small corruptions, such as noise, blur, or style variation, can significantly reduce the zero-shot accuracy of models like CLIP, particularly in specialized domains. This vulnerability highlights the need for effective *test-time adaptation strategies for Vision–Language Models*, which can adjust their representations or prompts on the fly to maintain robust performance under unseen conditions.

One of the first methods to explore Test-Time Adaptation for Vision–Language Models is **Test-Time Prompt Tuning (TPT)** (Shu *et al.*, 2022). TPT adapts CLIP-like models by optimizing the textual prompt embeddings while keeping the vision and text encoders frozen. The central idea is to modify the context words used in text prompts so that they better align with the visual

distribution of unseen target data, thereby improving zero-shot generalization without retraining the model. The overall architecture is illustrated in Figure 1.8.

Formally, in the text embedding space (before the application of the text encoder f_{text}), the prompt is represented as a learnable matrix $\mathbf{p} \in \mathbb{R}^{L \times D}$, where L denotes the number of tokens and D their embedding dimension. During adaptation, the model refines $\mathbf{p}$ by minimizing the entropy of the predicted class probabilities on the unlabeled target samples. The entropy loss is defined similarly to TENT, encouraging the model to make confident and consistent predictions across augmentations of the same image. For a test image x_i, a series of K transformations $\{x_i^{(k)}\}_{k=1}^{K}$ are applied, and the prompt is updated using only the most confident predictions (those with the lowest entropy). This objective can be expressed as

$$\mathcal{L}_{\text{TPT}} = -\frac{1}{K} \sum_{k=1}^{K} \sum_{c} \hat{y}_{i,c}^{(k)} \log \hat{y}_{i,c}^{(k)}, \tag{1.24}$$

where $\hat{y}_{i,c}^{(k)} = p(y_c | x_i^{(k)}, \mathbf{p})$ denotes the class probability obtained using the current prompt $\mathbf{p}$. The loss encourages the textual prompt to shift toward a context that produces more confident image–text alignments for the target distribution. The adaptation process thus refines only the prompt tokens, leaving the encoders f_{img} and f_{text} unchanged, which ensures that the adaptation remains lightweight and efficient.

To further enhance robustness, TPT averages predictions across all augmentations $\{x_i^{(k)}\}$, reducing overfitting to any specific view. However, since TPT requires iterative gradient-based optimization of the prompt embeddings for each test sample, it introduces a considerable computational overhead at inference time. Moreover, because no ground-truth supervision is available, the optimization landscape can be unstable and sensitive to the initialization of $\mathbf{p}$ and the learning rate. Despite these limitations, TPT demonstrated that prompt tuning is a promising direction for adapting large-scale VLMs to distribution shifts without modifying their architecture or retraining the encoders.

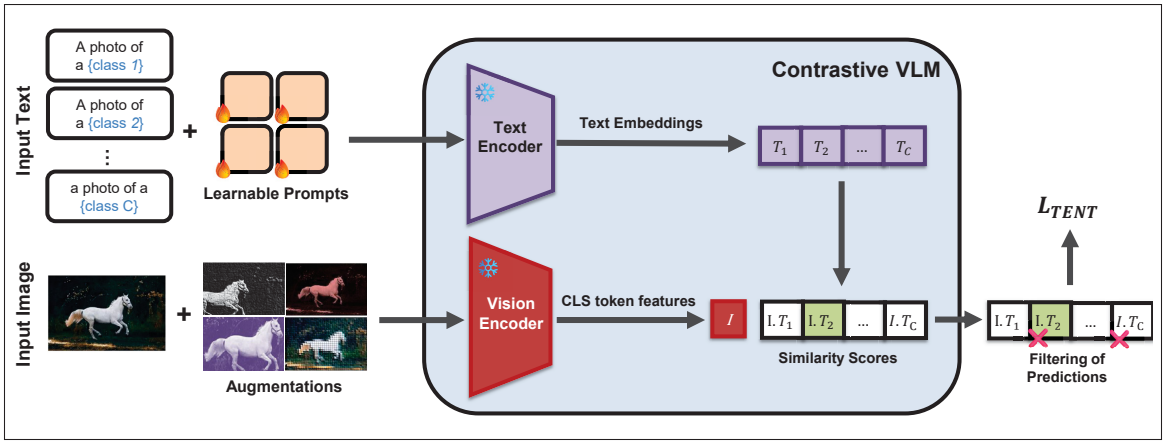

Figure 1.8 Overview of Test-Time Prompt Tuning (TPT). Only the textual prompt embeddings are optimized via entropy minimization, while the image and text encoders remain frozen. Multiple augmentations are used to enhance robustness and stability

Following prompt-based approaches such as TPT, subsequent research explored more structured and stable adaptation strategies for CLIP. **Weight-Averaged Test-Time Adaptation (WATT)** (Osowiechi *et al.*, 2024b) introduces an ensemble-based approach inspired by stochastic weight averaging (Izmailov, Podoprikhin, Garipov, Vetrov & Wilson, 2018b), leveraging model diversity to obtain flatter minima and improved generalization under distribution shifts. As illustrated in Figure 1.9, the key idea is to perform adaptation independently across multiple text templates and then average the resulting model weights, yielding a more robust final representation.

Given a batch of N test images, let $\mathbf{Z}^v \in \mathbb{R}^{N\times D}$ denote the image features and $\mathbf{Z}^t \in \mathbb{R}^{N\times D}$ the instance-specific text features. Unlike standard CLIP, which computes C class text embeddings, WATT uses N text features—one for each image—corresponding to the predicted class from an initial forward pass. The intra-modal similarity matrices are then computed as $\mathbf{S}^v = \mathbf{Z}^v(\mathbf{Z}^v)^\top$ and $\mathbf{S}^t = \mathbf{Z}^t(\mathbf{Z}^t)^\top$, capturing pairwise relationships within the visual and textual modalities. The combined pseudo-labels are defined as $\mathbf{Q} = \text{softmax}((\mathbf{S}^v + \mathbf{S}^t)/2\tau)$, while the standard image–text similarity predictions are given by $\mathbf{P} = \mathbf{Z}^v(\mathbf{Z}^t)^\top$. The adaptation objective minimizes the cross-entropy between $\mathbf{Q}$ and $\mathbf{P}$:

$$\mathcal{L}_{\text{WATT}} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{N} q_{ij}\log p_{ij}. \tag{1.25}$$

This process is repeated for each of the K available text templates in parallel, yielding a set of adapted weights $\{\theta_1, \theta_2, \ldots, \theta_K\}$. The final model parameters are then obtained via weight averaging, $\theta_{\text{avg}} = \frac{1}{K}\sum_{k=1}^{K}\theta_k$. In practice, the adaptation targets the parameters of the normalization layers within the CLIP encoders, as these layers play a critical role in distribution alignment for transformer-based architectures.

By averaging over multiple template-specific adaptations, WATT consolidates diverse solutions into a smoother region of the loss surface, enhancing robustness to domain shift and input corruption. This ensemble-based design leads to more stable performance across datasets and offers a compelling trade-off between adaptation flexibility and computational efficiency.

More recently, **CLIPArTT** (Hakim *et al.*, 2024) extends the idea of leveraging intra-modal similarities to refine CLIP's predictions in a fully test-time, label-free manner. CLIPArTT introduces two key innovations: (1) the use of instance-specific text prompts derived from the model's top-M predicted classes, and (2) the exploitation of image–image and text–text relationships through transductive pseudo-labeling.

For a given image x, CLIPArTT first identifies its M most probable classes ($M \ll C$) and constructs an instance-specific textual prompt of the form *"a photo of <class$_1$> or <class$_2$> or ... or <class$_M$>"*. This contextual disjunction enriches the text representation by embedding semantic uncertainty directly into the prompt. Using these adaptive prompts, the method computes the same intra-modal similarity matrices $\mathbf{S}^v$ and $\mathbf{S}^t$ as in WATT, which are then combined to produce transductive pseudo-labels $\mathbf{Q}$. The final image-to-text predictions $\mathbf{P}$ are contrasted with $\mathbf{Q}$ using the same cross-entropy loss as in Eq. 1.25.

In this framework, transduction plays a deeper role: the pseudo-labels $\mathbf{Q}$ capture the relational structure among all target samples and their predicted classes, enabling the model to exploit global information across the test batch. By integrating these cross-instance dependencies,

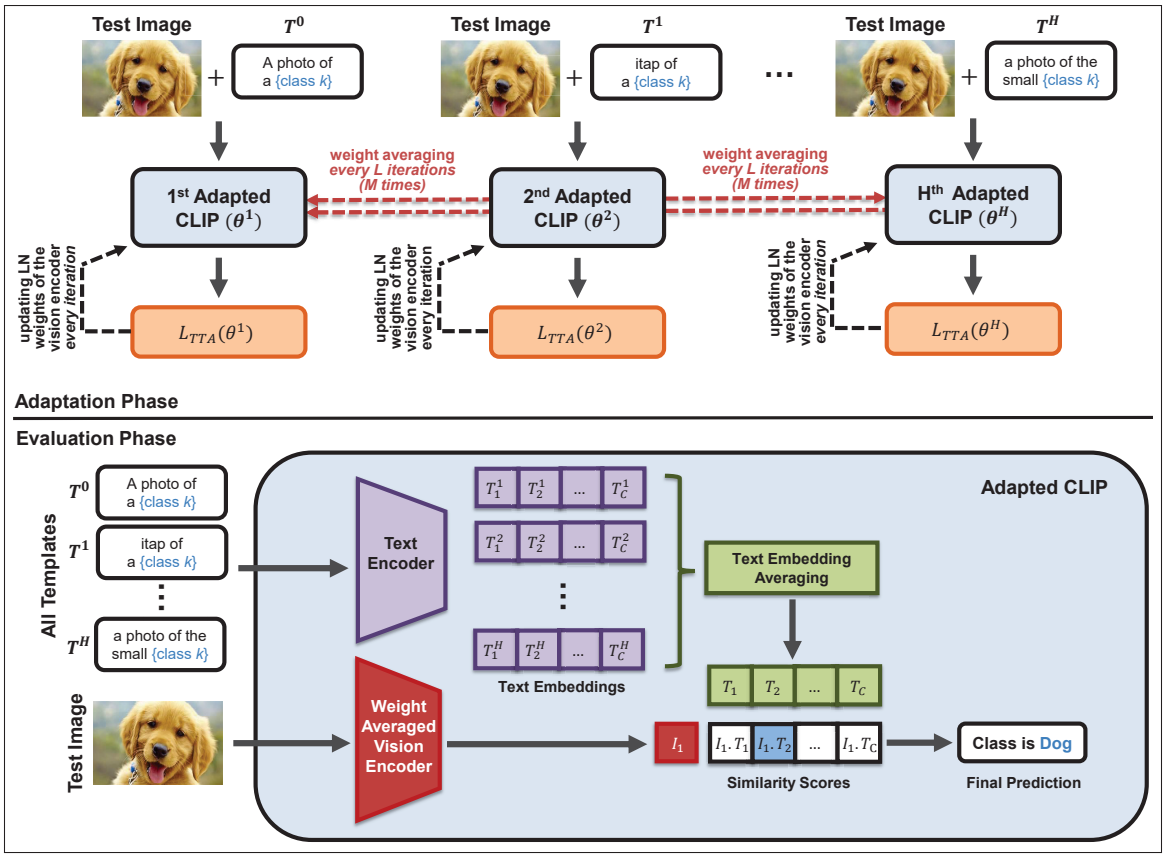

Figure 1.9 Overview of Weight-Averaged Test-Time Adaptation (WATT). The model is independently adapted for each available text template, and the resulting weights are averaged to obtain the final model parameters

CLIPArTT achieves improved consistency and robustness without any architectural modification or gradient-based fine-tuning. This lightweight and fully self-contained formulation makes it a practical solution for adapting large-scale Vision–Language Models under domain shift.

1.5 Final Remarks

This chapter surveyed robustness under domain shift through two complementary lenses: *Domain Generalization* (train-time robustness) and *Test-Time Adaptation* (deployment-time robustness). DG encourages models to learn domain-invariant structure *before* deployment, whereas TTA adjusts models *during* deployment using unlabeled target data. Recent progress in DG highlights feature-level perturbations, ensembling/weight averaging toward flatter minima, and diffusion-

driven data synthesis; TTA has evolved from normalization and entropy minimization to foundation-model strategies that adapt prompts, statistics, or similarity structures without labels.

Despite this progress, two DG gaps remain central to practical robustness: (i) while feature-level augmentation has proven effective for CNNs, the characteristic components of ViTs—tokenized representations and self-attention—remain under-explored; a dedicated feature-augmentation framework for ViTs that can benefit from its architecture specific characteristics is still missing; (ii) diffusion-based approaches either (a) emphasize data synthesis without explicitly encouraging the classifier to extract domain-invariant features, or (b) require running a generative model at inference time, which is often prohibitively costly in deployment.

On the TTA side, foundation-scale VLMs have seen steady advances, but *open-vocabulary segmentation* has largely been overlooked: existing VLM-TTA methods primarily target classification or retrieval and do not address the patch-/pixel-level adaptation required by segmentation.

This thesis responds to these gaps along both axes:

- **DG (Chs. 2, 3).** We introduce **TFS-ViT**, a token-level stylization framework (with attention-aware variants) specifically designed for ViTs and encourages structure, rather than texture-dependence. We further propose **FDS**, a diffusion-guided pseudo-domain synthesis framework that introduces a novel *controlled* way to diversify source samples by training a single conditional diffusion model across all source domains and generating new pseudo-domains through a novel conditioning mechanism coupled with an effective filtering strategy. This approach *explicitly* promotes domain-invariant feature learning while avoiding any inference-time overhead.
- **TTA for foundation models (Chs. 4, 5).** For the first time, we formulate and tackle a more challenging *open-vocabulary segmentation* under TTA for VLMs via **MLMP**, which aggregates multi-layer, multi-template vision–text cues to effectively adapt OVSS models at test time. In addition, we introduce a **comprehensive evaluation suite** encompassing over **80 test scenarios** across various domain-shift types, providing a unified foundation for

future research on VLM adaptation. As a particularly *practical and high-impact application*, we present **Histopath-C**, the first benchmark for VLM TTA in digital histopathology, and **LATTE**, a lightweight method that mitigates the substantial performance degradation experienced by VLMs under diverse histopathology-specific shifts, achieving consistent and significant performance gains.

CHAPTER 2

TFS-VIT: TOKEN-LEVEL FEATURE STYLIZATION FOR DOMAIN GENERALIZATION

Mehrdad Noori[*a] , Milad Cheraghalikhani[*a] , Ali Bahri[a] , Gustavo A. Vargas Hakim[b] , David Osowiechi[a] , Ismail Ben Ayed[b] , Christian Desrosiers[a]

[a] Department of Information Technologies Engineering, École de Technologie Supérieure
[b] Department of Systems Engineering, École de Technologie Supérieure
1100 Notre-Dame West, Montreal, Quebec, Canada H3C 1K3

[*]Equal Contribution
Paper published in *Pattern Reconition* Journal, May 2024

Abstract

Standard deep learning models such as convolutional neural networks (CNNs) lack the ability of generalizing to domains which have not been seen during training. This problem is mainly due to the common but often wrong assumption of such models that the source and target data come from the same i.i.d. distribution. Recently, Vision Transformers (ViTs) have shown outstanding performance for a broad range of computer vision tasks. However, very few studies have investigated their ability to generalize to new domains. This paper presents a first Token-level Feature Stylization (TFS-ViT) approach for domain generalization, which improves the performance of ViTs to unseen data by synthesizing new domains. Our approach transforms token features by mixing the normalization statistics of images from different domains. We further improve this approach with a novel strategy for attention-aware stylization, which uses the attention maps of class (CLS) tokens to compute and mix normalization statistics of tokens corresponding to different image regions. The proposed method is flexible to the choice of backbone model and can be easily applied to any ViT-based architecture with a negligible increase in computational complexity. Comprehensive experiments show that our approach is able to achieve state-of-the-art performance on five challenging benchmarks for domain generalization, and demonstrate its ability to deal with different types of domain shifts. The implementation is available at https://github.com/Mehrdad-Noori/TFS-ViT_Token-level_Feature_Stylization.

2.1 Introduction

Deep learning models like convolutional neural networks (CNNs), and more recently Vision Transformers (ViT), have enabled unprecedented progress in computer vision, achieving state-of-art performance on various tasks such as classification, semantic segmentation and object detection. However, most of these models rely on the naive assumption that the data used for training (the source domain) and the one encountered after deployment (the target domain) come from the same distribution. As they are not designed to tackle distribution shifts, the performance of such models typically degrades when out-of-distribution (OOD) data is encountered (Recht, Roelofs, Schmidt & Shankar, 2019; Hendrycks & Dietterich, 2019b). Domain adaptation (DA) approaches (Lu *et al.*, 2020; Saito, Watanabe, Ushiku & Harada, 2018) attempt to solve this problem by adapting a model trained on source domain data to a known target domain. A major limitation of such approaches is that they need target data for adaptation, which is not always available in practice. Moreover, adapting the source-trained model to each new target domain also incurs additional costs in terms of computations.

Domain generalization (DG) (Blanchard, Lee & Scott, 2011) seeks to overcome the domain shift problem in a different way: training a model with data from one or multiple source domains so that it can generalize to OOD data from any target domain. In recent years, a plethora of methods have been proposed for this challenging problem (Zhou, Liu, Qiao, Xiang & Loy, 2022b; Wang *et al.*, 2022a), exploiting various strategies including domain alignment (Hu, Zhang, Chen & Chan, 2020; Mahajan, Tople & Sharma, 2021; Li *et al.*, 2020), meta-learning (Li, Yang, Song & Hospedales, 2018a; Balaji, Sankaranarayanan & Chellappa, 2018), data augmentation (Shi, Yu, Sohn, Chandraker & Jain, 2020b; Shankar *et al.*, 2018), ensemble learning (Zhou, Yang, Qiao & Xiang, 2021), self-supervised learning (Carlucci *et al.*, 2019a; Albuquerque *et al.*, 2020) and regularization (Huang *et al.*, 2020; Cha *et al.*, 2021). While many of these methods have shown promising results using CNN architectures, very few have investigated the potential of ViTs for DG (Sultana, Naseer, Khan, Khan & Khan, 2022). One key reason for this is the more limited understanding of how ViTs learn compared to CNNs. For instance, it is well known that the first layers of CNN architectures like ResNet encode

domain-specific features, while those closer to the output capture features that are more related to class (Zhou *et al.*, 2021). This knowledge enables the development of efficient DG strategies, for example, augmenting the features in early network layers while enforcing the classification output to be consistent.

Compared to CNNs, which mainly learn by recognizing and composing local patterns, ViTs can model global relationships using so-called multiheaded self-attention (MSA) layers (Dosovitskiy *et al.*, 2020). Although ViT features are harder to interpret than those learned by CNNs, attention maps in MSA layers offer a powerful way to analyze the relationships between different parts of an image and their link to semantic classes. In particular, attention maps to the class (CLS) token measure of the contribution of each region to predicting the class of an image. In a recent work (Choi, Choi & Kim, 2022), authors exploit the attention maps of ViTs in a token-level data augmentation method for classification. While it improved performance by maximizing the saliency of augmented tokens, this method was designed for standard supervised learning, and not for a DG setting where the model can encounter OOD data.

In this paper, we propose a novel domain generalization approach, called Token-level Feature Stylization (TFS-ViT), which improves the generalization performance of ViTs to OOD data by synthesizing new domains. The core idea of our approach is to augment token-level features by mixing the normalization statistics of images from different domains. This encourages the model to learn meaningful relationships between different parts of an image, which do not depend on the image's style. We improve this approach with an attention-aware stylization strategy that leverages the attention maps of class (CLS) tokens to compute and mix normalization statistics of tokens corresponding to different image regions. The proposed method is flexible to the choice of backbone model and can be easily applied to any ViT-based architecture with a negligible increase in computational complexity.

Our contributions can be summarized as follows:

- We present a first token-level feature stylization approach for domain generalization in ViTs;

- We extend this approach with a novel attention-aware stylization strategy that uses attention maps in MSA layers to guide the augmentation toward more important regions of the image;
- We conduct extensive experiments on five challenging datasets, using different ViT architectures, and show our method to achieve state-of-art performance in most cases.

The rest of the paper is organized in the following way. In the next section, we provide an overview of related works on DG and recent methods using ViTs for this task. Section 2.3 then defines the DG problem addressed in this work, and presents our TFS-ViT approach for this problem. Section 2.4 describes the datasets and implementation details related to experiments. In Section 2.5, we present results evaluating the different components of our method and showing its advantage over existing approaches. Finally, we discuss the main results and possible future directions in Section 2.6.

2.2 Related Works

Domain Generalization The problem of generalizing to OOD data was initially introduced by Blanchard et al. in 2011 (Blanchard *et al.*, 2011) and has since then generated a growing interest in computer vision. The broad range of methods developed for this problem can mostly be grouped in seven categories based on domain alignment, meta-learning, data augmentation, ensemble learning, self-supervised learning, disentangled representation learning, and regularization. Domain alignment methods seek to learn domain-invariant representations by minimizing the difference among available source domains. This can be achieved in several ways, for instance by matching moments (Peng *et al.*, 2019), using discriminant analysis (Hu *et al.*, 2020), minimizing the maximum mean discrepancy (MMD) (Li, Pan, Wang & Kot, 2018b) or a contrastive loss (Motiian, Piccirilli, Adjeroh & Doretto, 2017), as well as with domain-adversarial learning (Li *et al.*, 2018c). Meta-learning approaches for DG (Li *et al.*, 2018a; Balaji *et al.*, 2018) typically consider a bi-level optimization problem where a model is updated using meta-source domains so that the test error on a given meta-target domain is minimized. This meta-learning is often done using episodic training, and can update all parameters of a network (Li *et al.*, 2018a) or a reduced set of regularization parameters (Balaji *et al.*, 2018).

Data augmentation is another popular approach for DG, which simulates domain shifts during training in hope of making the model more robust to such shifts. This can be achieved using various strategies, including learnable augmentation, off-the-shelf style transfer and feature-based augmentation. The first strategy employs an augmentation network to generate images from training samples so that their joint distribution is different from those of existing source domains (Carlucci, Russo, Tommasi & Caputo, 2019b; Zhou, Yang, Hospedales & Xiang, 2020b). On the other hand, data augmentation methods based on off-the-shelf style transfer try to map input images from one domain to another using techniques like Fourier-based augmentation (Xu *et al.*, 2023) or to change the style of these images (Yue *et al.*, 2019). This can be done, for example, using Adaptive Instance Normalization (AdaIN) (Huang & Belongie, 2017; Somavarapu *et al.*, 2020). While most data augmentation methods operate on pixels, feature-level augmentation techniques have also been proposed for DG (Mancini, Akata, Ricci & Caputo, 2020; Zhou *et al.*, 2024a). Such techniques are motivated by the observation that style-related information is often captured in statistics of CNN features (Zhou *et al.*, 2024a).

Ensemble learning approaches for DG try to increase the robustness to OOD data by training multiple domain-specific models (Zhou *et al.*, 2021). The ensemble prediction for target domain examples can then be obtained as a weighted average of the individual models' predictions, with the weights measuring the similarity of the target sample to each source domain or the models' confidence (Mancini, Bulò, Caputo & Ricci, 2018). In contrast, self-supervised learning methods aim to learn representations that better generalize across domains by pre-training a model on some unsupervised auxiliary (pretext) tasks (Carlucci *et al.*, 2019a; Kim, Yoo, Park, Kim & Lee, 2021a). Pretext tasks can be solving a jigsaw puzzle (Carlucci *et al.*, 2019a; Wang, Yu, Li, Fu & Heng, 2020b), reconstructing an image with an autoencoder (Ghifary, Bastiaan Kleijn, Zhang & Balduzzi, 2015), or using a contrastive objective (Kim *et al.*, 2021a). In a more recent study (Zhang, Zhang, Wang & Liu, 2023a), the connection between the information bottleneck principle and DG has been studied, leading to enhanced domain generalization through domain-invariant representation learning.

Instead of directly learning a domain-invariant representation, disentangled representation learning approaches try to separate features in two groups encoding domain-specific and domain-invariant information (Li *et al.*, 2017; Chattopadhyay, Balaji & Hoffman, 2020). This can be achieved by learning domain-specific masks that can dynamically select relevant features for a given image of the target domain (Chattopadhyay *et al.*, 2020). The last category of methods for DG regularize the training of a model to learn features which can better generalize across domains (Cha *et al.*, 2021; Sagawa, Koh, Hashimoto & Liang, 2019; Sultana *et al.*, 2022). Such methods typically extend the empirical risk minimization (ERM) approach (Gulrajani & Lopez-Paz, 2021) by adding a regularization objective, for example, based on distillation (Sultana *et al.*, 2022), dense stochastic weight averaging (Cha *et al.*, 2021) or distributionally robust optimization (DRO) (Sagawa *et al.*, 2019).

Vision transformers (ViTs) The methods mentioned above are mostly based on CNN architectures. Despite the outstanding performance of ViTs for classification (Dosovitskiy *et al.*, 2020), object detection (Dai *et al.*, 2021) and semantic segmentation (Strudel, Garcia, Laptev & Schmid, 2021), very few works have explored their potential for domain generalization. Zhang et al. analyzed the robustness of ViTs to distribution shifts, and proposed a novel architecture based on self-supervised learning and information theory that better generalizes to data from unseen domains (Zhang *et al.*, 2021a). Recently, Sultana et al. proposed a Self-Distilled Vision Transformer (SDViT) for DG which employs auxiliary losses in intermediate transformer blocks to alleviate the problem of overfitting source domains (Sultana *et al.*, 2022). Our proposed method follows a different approach: designing an token-level features stylization strategy that exploits the information of attentions maps to effectively and efficiently synthesize new domains during training.

2.3 Method

In this section, we first define the problem of domain generalization. We then detail our Token-Level Feature Stylization (TFS-ViT) method for DG and explain how attention maps in MSA layers can be used to further improve its performance.

2.3.1 Problem Definition

Referring to the input space as $\mathbf{X}$ and the target space as $\mathbf{Y}$, we define a domain for a classification task as the joint distribution of $P_{\mathbf{XY}}$ on $\mathbf{X}\times\mathbf{Y}$. For a particular domain, the marginal distribution on $\mathbf{X}$ is denoted as $P_{\mathbf{X}}$, the posterior distribution of $\mathbf{Y}$ given $\mathbf{X}$ as $P_{\mathbf{Y}|\mathbf{X}}$, and the class-conditional distribution of $\mathbf{X}$ given $\mathbf{Y}$ as $P_{\mathbf{X}|\mathbf{Y}}$. In the standard DG setup, we have access to M source domains that are related to one another but are not the same, $\mathcal{S} = \{S_i\}_{i=1}^{M}$. In other words, we proceed on the assumption that the joint distribution of each domain, $P_{\mathbf{XY}}^{(i)}$, is unique in comparison to that of other domains, $P_{\mathbf{XY}}^{(i)} \neq P_{\mathbf{XY}}^{(i')}$ when $i \neq i'$. Each source domain consists of N_i samples, $S_i = \{(x_j^{(i)}, y_j^{(i)})\}_{j=1}^{N_i}$. $T = \{x_j^{\mathbf{T}}\}_{j=1}^{N_{\mathbf{T}}}$ represents the target domain, which has a joint distribution distinct from the one of the source domain. The goal is to predict the labels for target domain examples without having access to such examples. We thus try to minimize a loss function, $\mathcal{L} : \mathbf{Y}\times\mathbf{Y} \rightarrow [0, \infty]$, to find the learning function $f : \mathbf{X} \rightarrow \mathbf{Y}$ that best estimates $P_{\mathbf{Y}|\mathbf{X}}$.

2.3.2 Token-Level Feature Stylization (TFS)

Normalization-based feature stylization techniques, such as Adaptive Instance Normalization (AdaIN) (Huang & Belongie, 2017) and MixStyle (Zhou *et al.*, 2024a) have been shown to improve the generalization performance of CNNs. However, their potential and effectiveness in the context of ViT models has not been explored. Motivated by the success of these techniques in CNNs and also by leveraging the sequential nature of ViTs, we propose a token-level feature stylization method, TFS-ViT, that is able to enhance the generalization capacity of ViTs on unseen domains. Our proposed method is designed to selectively stylize a subset of tokens at each layer, resulting in generating more divers samples during training. Figure 2.1 illustrates the overall architecture of TFS-ViT.

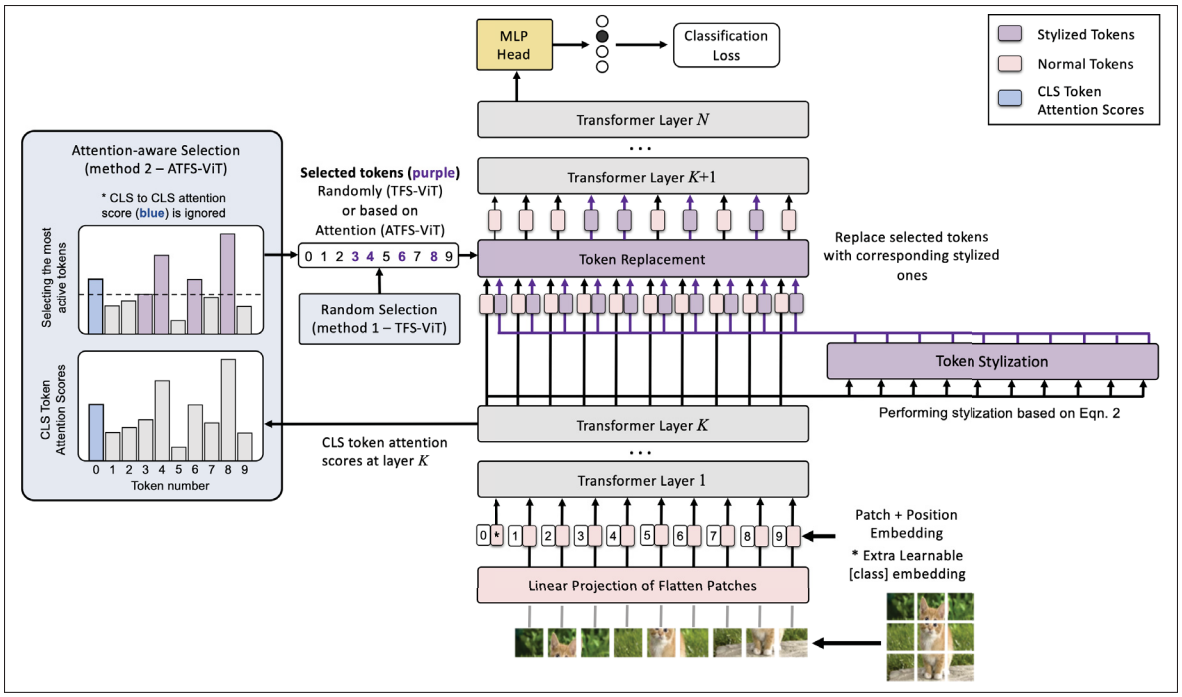

Figure 2.1 Overview of the proposed architecture for Token-level Feature Stylization (TFS-ViT)

In the proposed method, we first estimate token-level statistics of the feature embedding in layer k, denoted by x^k, by computing the mean and standard deviation across token sequences:

$$\begin{aligned} \mu_c(x^k) &= \frac{1}{S}\sum_{s=1}^{S} x_{c,s}^k \\ \sigma_c(x^k) &= \sqrt{\frac{1}{S}\sum_{s=1}^{S}\left(x_{c,s}^k - \mu_c(x^k)\right)^2} \end{aligned} \tag{2.1}$$

where S is the length of the token embedding sequence. We then randomly choose another sample $\tilde{x}$ from the batch and synthesize a stylized version of x^k, denoted as $\phi(x^k)$, in the

following manner:

$$\begin{aligned} \gamma_{\text{mix}} &= \alpha\sigma(x^k) + (1-\alpha)\sigma(\tilde{x}^k) \\ \beta_{\text{mix}} &= \alpha\mu(x^k) + (1-\alpha)\mu(\tilde{x}^k) \\ \phi(x^k) &= \gamma_{\text{mix}}\frac{x^k - \mu(x^k)}{\sigma(x^k)} + \beta_{\text{mix}} \end{aligned} \tag{2.2}$$

where mixing coeffcient α is sampled from the Beta distribution, $\alpha \sim \text{Beta}(0.1, 0.1)$. Afterwards, to generate the input for the subsequent layer, we randomly choose a given number of tokens and replace their original feature, x^k, with their corresponding stylized version from $\phi(x^k)$. The percentage of replaced tokens is controlled by a hyper-parameter d. While training the network, at each iteration, we randomly choose n layers from the total of N layers that form the backbone and perform token-level stylization on those layers as described above.

The detailed process of our TFS-ViT method is illustrated in Figure 2.2. The reason for stylizing some of the tokens while leaving others unchanged is to increase the diversity of the generated samples during training. By randomly selecting a subset of tokens to stylize at each layer, we are effectively creating new combinations of stylized tokens while also maintaining the underlying structure of the input tokens[1]. This cannot be achieved so easily in a CNN architecture. Our approach, specifically designed for ViTs, allows exploring a wider range of feature distributions, thereby increasing the model's capacity to generalize to unseen domains. Our method not only proves to be effective in DG settings (Section 2.5.1), but also enhances in-domain performance, as evidenced by the results presented in Section 2.5.2.

2.3.3 Attention-Aware TFS

One of the key aspects of ViTs that distinguishes them from traditional CNNs is their use of self-attention. In ViTs, self-attention maps are employed to encode the relationships between features corresponding to different regions of an image. In particular, the attention maps from

[1] We simulate our proposed stylization method by using an Encoder-Decoder ViT and visualize the effect of this selection and replacement of tokens in Figure 1 and Figure 2 of the supplementary materials.

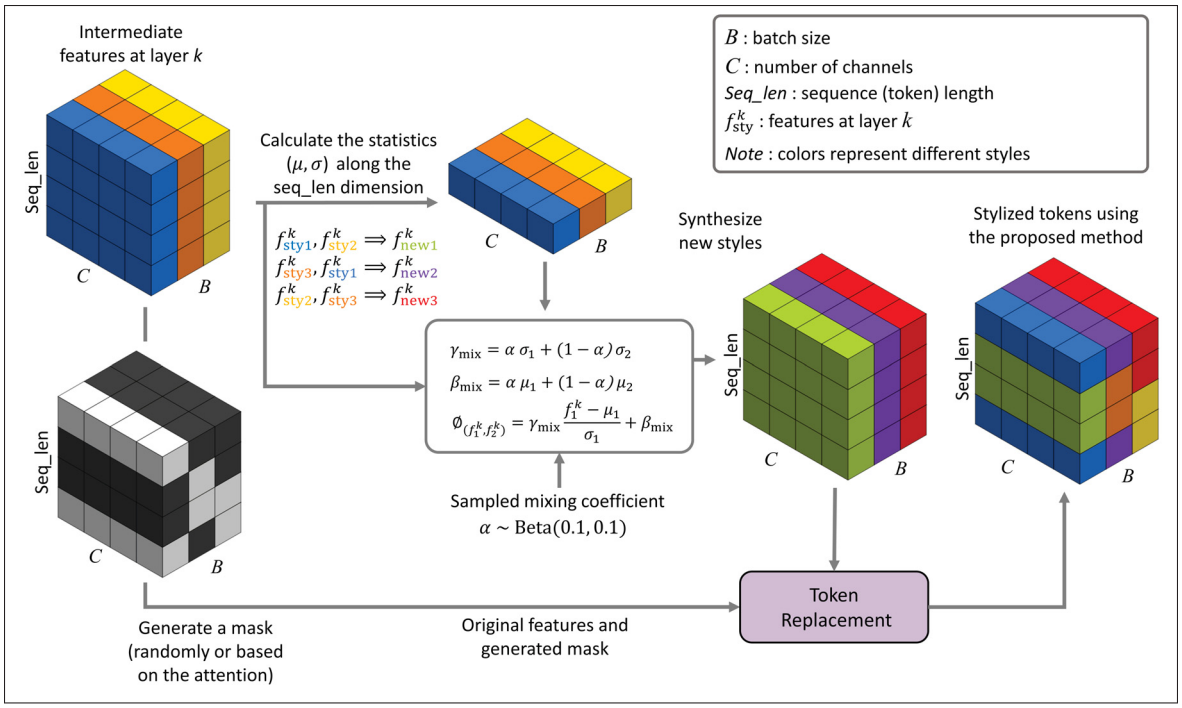

Figure 2.2 Synthesized features using our proposed method. Different colors denote different styles. By randomly selecting a subset of tokens to stylize at each layer, our method generates diverse samples while preserving the underlying structure of the tokens. This leads to forcing the network to only focus on the structure-related information which eventually results in improving the generalization performance. It is worth mentioning that we perform our stylization method on multiple layers of the ViT network

tokens to the class (CLS) token offer a measure of saliency which can be exploited in feature stylization. Based on this idea, we extend our proposed TFS-ViT to have an Attention-aware Token-Level Feature Stylization (ATFS-ViT).

To this end, we compute the mean of attention matrices of the CLS token over the different attention heads:

$$\begin{aligned} A(Q,K)_{h,s,s} &= \text{Softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right) \\ M_{s,s} &= \frac{1}{H}\sum_{h=1}^{H} A_{h,s,s} \\ M_{\text{cls}} &= M[0, 1:S] \end{aligned} \tag{2.3}$$

Here, Q and K are the backbone's Query and Key, H is the number of attention heads, and S is the length of the token sequence. In TFS-ViT, we swapped D tokens with their stylized counterparts, the number of which is determined by hyper-parameter d. For ATFS-ViT, instead of choosing the tokens randomly, we select those with highest activation in M_{cls}. The rationale behind this strategy is that, by picking the most active tokens with respect to the CLS token, our method will focus on stylizing the image's foreground, which is more important than the background for the final prediction. Experimental evidence supporting this approach can be found in Sec 2.5.2.

2.4 Experimental Setup

2.4.1 Datasets

Following the work of Gulrajani and Lopez-Paz (Gulrajani & Lopez-Paz, 2021), we compare our approach's performance to the current state-of-art using five challenging datasets, `PACS` (Li *et al.*, 2017), `VLCS` (Fang, Xu & Rockmore, 2013), `OfficeHome` (Venkateswara, Eusebio, Chakraborty & Panchanathan, 2017), `TerraIncognita` (Beery, Van Horn & Perona, 2018) and `DomainNet` (Peng *et al.*, 2019), which we describe below.

`PACS` (Li *et al.*, 2017) has a total of 9,991 photos organized into four distinct domains, $d \in \{\text{Art, Cartoons, Photos, Sketches}\}$, and seven distinct classes. `VLCS` (Fang *et al.*, 2013) is comprised of four different domains, $d \in \{\text{Caltech101, LabelMe, SUN09, VOC2007}\}$, five different classes, and 10,729 different photos. `OfficeHome` (Venkateswara *et al.*, 2017) includes four domains, $d \in \{\text{Art, Clipart, Product, Real}\}$, 65 classes, and a total of 15,588 photos. `TerraIncognita` (Beery *et al.*, 2018) contains four camera-trap domains, with 10 categories and a total of 24,778 images. Finally, DomainNet has 6 domains, $d \in \{\text{Clipart, Infograph, Painting, Quickdraw, Real, Sketch}\}$, 345 classes and 586,575 photos.

2.4.2 Implementation

To have a fair comparison, we implement our method using DomainBed (Gulrajani & Lopez-Paz, 2021) – a recently introduced framework that contains the main existing DG methods and is developed to offer comparisons under a fair evaluation protocol. Accordingly, we follow the same leave-out-one-domain strategy to evaluate performance for different DG datasets, where one domain is used for testing, and the remaining ones are employed for training. Additionally, to choose the best model, 20% of the training data is used as the validation set[2]. The final result corresponding to each dataset is the average of the accuracy values calculated when using different domains of that dataset as the test domain. To obtain statistically meaningful results, we repeat each experiment three times with different seeds.

Similar to (Sultana *et al.*, 2022), for all of our experiments, we employ the AdamW optimizer and use the default hyperparameters of DomainBed, including a batch size of 32, a learning rate of $5e$-05, and a weight decay of 0.0. Additionally, to select the best values of our method-specific hyperparameters, d, n, we perform a grid search with $d \in \{0.1, 0.3, 0.5, 0.8\}$ and $n \in \{1, 2, 3, 4\}$ using the validation set. Most existing methods for DG incorporate ResNet50 as backbone in their architecture. To have a fair comparison, we explore two different ViT-based backbones in our experiments: DeiT (Touvron *et al.*, 2021) and T2T-ViT (Yuan *et al.*, 2021). Specifically, we use DeiT-Small, containing 22M parameters, and T2T-ViT-14, containing 21.5M parameters, since they have a number of parameters comparable to ResNet50 which has 23.5M parameters.

2.5 Results

We first compare the performance of our proposed methods against state-of-art DG methods, across five DG benchmarks. We then present a detailed analysis investigating several aspects of our methods, including the impact of method-specific hyperparameters, the efficacy of various token selection strategies, our proposed method performance in single-source domain

[2] During training, the model that maximizes the accuracy on this overall validation set is chosen as the best model. The best model is then evaluated on the test domain to report the out-of-domain (unseen domain) accuracy.

Table 2.1 Comparison to the state-of-art on five benchmarks, reporting the mean and standard deviation across three runs.The best and second best results are in **bold** and underlined fonts, respectively

Method	Backbone	# Params	VLCS	PACS	OfficeHome	TerraInc	DomainNet	Average
ERM (Gulrajani & Lopez-Paz, 2021)	ResNet-50	23.5M	77.5± 0.4	85.5± 0.2	66.5± 0.3	46.1± 1.8	40.9± 0.1	63.3
IRM (Arjovsky, Bottou, Gulrajani & Lopez-Paz, 2019)	ResNet-50	23.5M	78.5± 0.5	83.5± 0.8	64.3± 2.2	47.6± 0.8	33.9± 2.8	61.5
GroupDRO (Sagawa *et al.*, 2019)	ResNet-50	23.5M	76.7± 0.6	84.4± 0.8	66.0± 0.7	43.2± 1.1	33.3± 0.2	60.7
Mixup (Yan, Song, Li, Zou & Ren, 2020)	ResNet-50	23.5M	77.4± 0.6	84.6± 0.6	68.1± 0.3	47.9± 0.8	39.2± 0.1	63.4
MLDG (Li *et al.*, 2018a)	ResNet-50	23.5M	77.2± 0.4	84.9± 1.0	66.8± 0.6	47.7± 0.9	41.2± 0.1	63.5
CORAL (Sun & Saenko, 2016)	ResNet-50	23.5M	78.8± 0.6	86.2± 0.3	68.7± 0.3	47.6± 1.0	41.5± 0.1	64.5
MMD (Li *et al.*, 2018b)	ResNet-50	23.5M	77.5± 0.9	84.6± 0.5	66.3± 0.1	42.2± 1.6	23.4± 9.5	58.8
DANN (Ganin *et al.*, 2016)	ResNet-50	23.5M	78.6± 0.4	83.6± 0.4	65.9± 0.6	46.7± 0.5	38.3± 0.1	62.6
CDANN (Li *et al.*, 2018c)	ResNet-50	23.5M	77.5± 0.1	82.6± 0.9	65.8± 1.3	45.8± 1.6	38.3± 0.3	62.0
MTL (Blanchard, Deshmukh, Dogan, Lee & Scott, 2017)	ResNet-50	23.5M	77.2± 0.4	84.6± 0.5	66.4± 0.5	45.6± 1.2	40.6± 0.1	62.8
SagNet (Nam, Lee, Park, Yoon & Yoo, 2021a)	ResNet-50	23.5M	77.8± 0.5	86.3± 0.2	68.1± 0.1	48.6± 1.0	40.3± 0.1	64.2
ARM (Zhang *et al.*, 2021b)	ResNet-50	23.5M	77.6± 0.3	85.1± 0.4	64.8± 0.3	45.5± 0.3	35.5± 0.2	61.7
VREx (Krueger *et al.*, 2021)	ResNet-50	23.5M	78.3± 0.2	84.9± 0.6	66.4± 0.6	46.4± 0.6	33.6± 2.9	61.9
RSC (Huang *et al.*, 2020)	ResNet-50	23.5M	77.1± 0.5	85.2± 0.9	65.5± 0.9	46.6± 1.0	38.9± 0.5	62.6
SelfReg (Kim *et al.*, 2021a)	ResNet-50	23.5M	77.5± 0.0	86.5± 0.3	69.4± 0.2	51.0± 0.4	44.6± 0.1	65.8
mDSDI (Bui, Tran, Tran & Phung, 2021)	ResNet-50	23.5M	79.0± 0.3	86.2± 0.2	69.2± 0.4	48.1± 1.4	42.8± 0.1	65.0
SWAD (Cha *et al.*, 2021)	ResNet-50	23.5M	79.1± 0.1	88.1± 0.1	70.6± 0.2	50.0± 0.3	46.5± 0.1	66.8
ERM-ViT (Touvron *et al.*, 2021)	DeiT-Small	22M	78.8± 0.5	84.9± 0.9	71.4± 0.1	43.4± 0.5	45.5± 0.0	64.8
SDViT (Sultana *et al.*, 2022)	DeiT-Small	22M	78.9± 0.4	86.3± 0.2	71.5± 0.2	44.3± 1.0	45.8± 0.0	65.3
TFS-ViT (ours)	DeiT-Small	22M	80.19± 0.45	87.27± 0.38	72.08± 0.13	48.60± 0.61	46.60± 0.06	66.95
ATFS-ViT (ours)	DeiT-Small	22M	80.65± 0.36	87.54± 0.39	71.44± 0.16	46.06± 0.70	46.18± 0.07	66.37
ERM-ViT (Yuan *et al.*, 2021)	T2T-ViT-14	21.5M	78.9± 0.3	86.8± 0.4	73.7± 0.2	48.1± 0.2	48.1± 0.1	67.1
SDViT (Sultana *et al.*, 2022)	T2T-ViT-14	21.5M	79.5± 0.8	88.0± 0.7	74.2± 0.3	50.6± 0.8	48.2± 0.2	68.1
TFS-ViT (ours)	T2T-ViT-14	21.5M	80.03± 0.25	88.99± 0.45	74.59± 0.21	**51.76± 0.54**	**48.34± 0.13**	68.74
ATFS-ViT (ours)	T2T-ViT-14	21.5M	**80.98± 0.40**	**89.56± 0.41**	**74.65± 0.24**	51.20± 0.43	47.94± 0.21	**68.87**

generalization, its capability for in-domain regularization, and the computational overhead involved.

2.5.1 Comparison with the state-of-the-art

In Table 2.1, we provide a comparison between our TFS-ViT method and 18 recent algorithms for DG implemented in the DomainBed framework (Gulrajani & Lopez-Paz, 2021). We also compare our results with vanilla ERM on the same ViT (ERM-ViT) as well as against SDViT (Sultana *et al.*, 2022) which is currently the only ViT-based method for DG. Using DeiT-Small as backbone, our method improves over ERM-ViT by 2.64% in PACS, 1.85% in VLCS, 0.68% in OfficeHome, 5.2% in TerraIncognita, and 1.1% in DomainNet. Moreover, it outperforms the recent SDViT on the same backbone by 1.24% in PACS, 1.75% in VLCS, 0.58% in OfficeHome, 4.3% in TerraIncognita, and 0.80% in DomainNet.

As can be seen, an even greater improvement is achieved when switching the backbone to T2T-ViT. Specifically, TFS-ViT then increases the baseline accuracy by 2.76% in PACS, 2.08% in VLCS, 0.95% in OfficeHome, 3.66% in TerraIncognita, and 0.24% in DomainNet. Compared to SDViT on this backbone, our method yields accuracy improvements of 1.56% in PACS, 1.48% in VLCS, 0.45% in OfficeHome, 1.16% in TerraIncognita, and 0.14% in DomainNet.

On the PACS dataset, our TFS-ViT method achieves a 4.6% higher accuracy than the vanilla ERM baseline with ResNet-50 backbone, and improved the previous state-of-art by 1.46%, which was previously held by SWAD (Cha *et al.*, 2021) with a 88.1% accuracy. Likewise, we observe a 3.48% improvement compared to vanilla ERM on the VLCS dataset. Once again, TFS-ViT outperformed the previous state-of-art by a 1.48% margin, previously held by SDViT (Sultana *et al.*, 2022) with a 79.5% accuracy. By achieving an accuracy of 75.65%, our method also improves by 0.45% the previous record of 74.2% on the OfficeHome dataset, established by SDViT (Sultana *et al.*, 2022). For this dataset, a large improvement of 8.15% over the ERM baseline is achieved by our method. A similar result is observed for the TerraIncognita dataset, for which we witness an increase of 5.66% over the vanilla ERM, and where TFS-ViT beats the previous record of SDViT (Sultana *et al.*, 2022) by a 0.76% margin. In DomainNet, we see an improvement of 7.44% compared to the ERM baseline. For this last dataset, our method's accuracy of 48.34% outperforms the previous record of SDViT (Sultana *et al.*, 2022) by 0.14%.

2.5.2 Further Analyses

2.5.2.1 Hyperparameter impact

Domain Generalization (DG) is an inherently challenging task given the unpredictable nature of unseen domains. For a comprehensive understanding, and to ensure robust outcomes, it is essential to delve into the effects of the method-specific hyperparameters introduced in our paper. As discussed in the Sec 2.4.2, our TFS-ViT and ATFS-ViT methods introduce two specific hyperparameters: n, which represents the number of random layers where stylization is applied, and d, the fraction of tokens to be replaced with their stylized counterparts. TFS-ViT chooses d

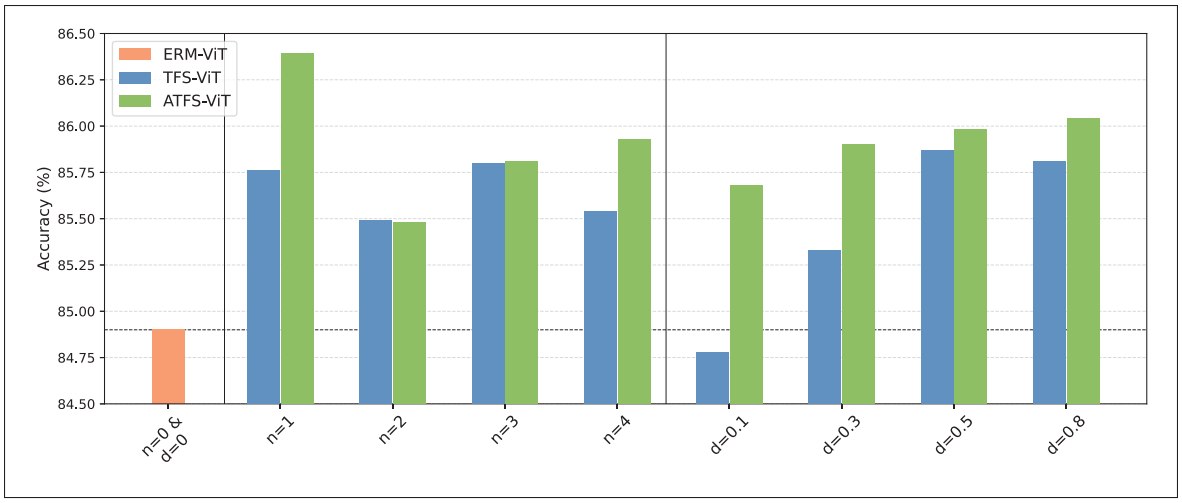

Figure 2.3 Effects of varying hyperparameters on the PACS dataset. The figure shows the influence of n, the number of layers where stylization is performed, with results averaged over different d values, alongside the impact of d, the fraction of tokens to be replaced with their stylized counterparts, averaged over various n values

tokens randomly for stylization, whereas ATFS-ViT selects d tokens with highest activations in M_{cls}. For all experiments, we conducted a grid search to determine the optimal values for each domain or dataset using the validation set. The grid search parameters were selected in $d \in \{0.1, 0.3, 0.5, 0.8\}$ and $n \in \{1, 2, 3, 4\}$ unless otherwise specified.

It is important to note that there is no "one-size-fits-all" solution for hyperparameters across different datasets and domains. The optimal choice often depends on the specific characteristics of the dataset and the degree of domain shift. To provide a clearer understanding, we performed an extensive study on the effects of these hyperparameters on the proposed TFS-ViT and ATFS-ViT. The resulting insights of varying these hyperparameters, n and d, on the PACS dataset are presented in Fig. 2.3.

As can be seen, the majority of hyperparameter combinations yield enhanced generalization when compared to the baseline, ERM-ViT. The best hyperparameters are around n=3 and d=0.5 for the PACS dataset. It is worth mentioning that this figure reports average results across various domains. However, when the domain shift is more severe, we find that higher values of d and n are needed, and vice versa. For example, in the PACS dataset, n=2 and d=0.1 are the ideal

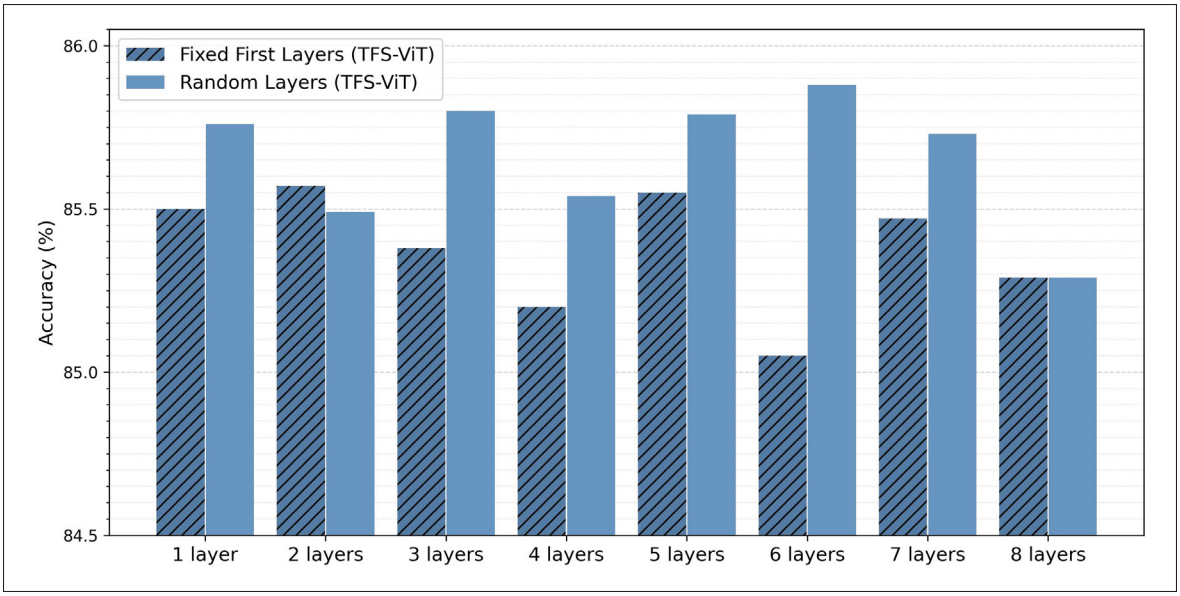

Figure 2.4 Performance comparison of stylization applied to a fixed initial set of layers versus random layer selection on the PACS dataset. Results are averaged over the different d values

hyperparameters when target domain is Cartoon, which contains less domain shift compared to source domains. On the other hand, when the Sketch domain presenting a more severe domain shift stands as the target, n=4 and d=0.8 emerges as the most effective combination, aligning with our expectations. Additionally, we can see that the performance of ATFS-ViT is more robust than TFS-ViT in all cases, as it chooses the candidate tokens for stylization in a more intelligent way.

2.5.2.2 Fixed Layers vs Random Layers

The use of CNNs for DG relies on several principles, one of which being that information pertaining to style is encoded in the first few layers. As we move closer to the classification head, more information related to classes is included in the feature maps. As a result, CNN-based feature stylization techniques focus on the first few layers of the network for augmenting features. However, the same strategy may not be optimal for ViTs, where features encoding structure can be found in all layers.

Table 2.2 Performance comparison of token selection strategies for stylization across the PACS dataset domains. The best results are highlighted in **bold**. Descriptions: All - select and stylize all tokens; Random - random selection of tokens for stylization; High and Low - selection based on the highest and lowest activations in M_{cls} for stylization, respectively

Token Selection	Art	Cartoon	Photo	Sketch	Average
Baseline (ERM-ViT)	87.4	81.5	98.1	72.6	84.9
All	88.88	82.20	98.40	76.73	86.43
Random (TFS-ViT)	89.63	**83.03**	**98.58**	77.83	87.27
Low	**90.46**	82.93	98.39	77.39	87.29
High (ATFS-ViT)	**90.46**	83.00	98.43	**78.25**	**87.54**

To explore this question, we compare in Figure 2.4 the accuracy on the PACS dataset while conducting feature stylization on the first n layers *vs* randomly selecting n layers from the first 75% of the layers (i.e., the first 8 layers of DeiT). As can be seen, applying stylization to randomly selected layers is usually better than in the first ones. The improved performance achieved by randomly selected layers is also due to the added stochasticity, which exposes the model to a broader range of domain shifts.

2.5.2.3 Token Selection Choices

While our proposed TFS-ViT method has demonstrated impressive accuracy in various domain generalization scenarios, the strategy for token selection, particularly the one based on CLS attention scores, can further improve the performance. In this section, we present a comprehensive analysis to understand the impact of various selection strategies on model performance.

Table 2.2 showcases the comparative performance of diverse token selection strategies applied to the PACS dataset. Evidently, the ATFS-ViT method, which stylizes tokens with the highest activations in the CLS attention maps, outperforms not only the TFS-ViT approach that employs random selection, but also strategies that stylizes the lowest activations in CLS attention maps or involve stylizing all tokens (denoted as “All” in the table). “All” is analogous to TFS-ViT when $d = 1$, and can be seen as an adaptation of the MixStyle method (Zhou *et al.*, 2024a) into ViTs, a technique originally designed for CNNs. While this approach does exhibit improvement over

the baseline, our proposed methods, which introduce greater diversity to the model (please see the Fig 2.2), offer a substantial performance boost.

The results of the table suggests that tokens with higher activations in CLS attention maps contain more discriminative features. When stylized, they contribute more effectively to the domain generalization task, compared to those with lower activations which might be less informative or potentially noisy. Furthermore, our observations indicate that using a more powerful ViT backbone (for example T2T-ViT-14) significantly amplifies the performance benefits of the high activation token selection strategy. This observation is evident from the main results in Table 2.1.

2.5.2.4 Single Source Domain Generalization

While it is generally assumed that all samples from a given domain originate from the same distribution, this assumption may not always hold in practice. Based on this idea, we evaluate the advantage of using our TFS-ViT method when training with images from the same domain. Figure 2.5 compares the performance of ERM-ViT and our TFS-ViT method, when training with a single source domain of the PACS dataset and testing on all others. As shown, TFS-ViT considerably increases the generalization capability of ERM-ViT for every source domain.

2.5.2.5 Regularization Effect

In the next analysis, we evaluate whether supplementing the network with synthetic features generated by TFS-ViT can also improve accuracy when evaluating the network on the same domain it was trained on. For this analysis, we train and test the model separately on each domain of the PACS dataset. Results presented in Figure 2.6 reveal that TFS-ViT also achieves a significantly higher accuracy than ERM-ViT in this setting. This demonstrates the usefulness of TFS-ViT in a standard in-domain setting, in addition to the OOD scenario of DG. Therefore, our method can be also regarded as a regularization technique, and can be employed across a variety of applications.

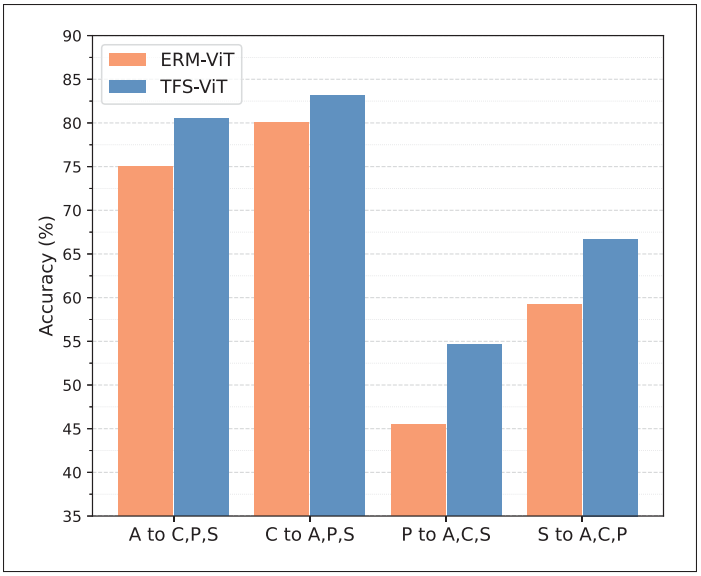

Figure 2.5 Comparison of ERM-ViT and TFS-ViT performance in Single-Source Domain Generalization setting on the PACS dataset

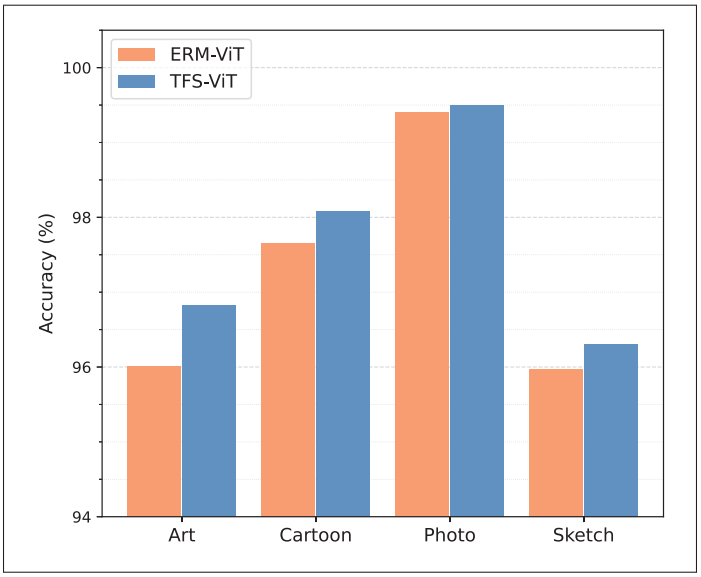

Figure 2.6 Regularization effect. Comparison of ERM-ViT and TFS-ViT performance when training and evaluation is done on the same domain for different domains of the PACS dataset

Table 2.3 Our proposed method performance on different domains of the PACS (Li *et al.*, 2017) dataset. Mean and Standard Deviation are reported across three runs. The best and second best average is in **bold** and underlined fonts, respectively

Method	Backbone	# of Params	Art	Cartoon	Photos	Sketch	Average
ERM	ResNet-50	23.5M	81.3± 0.6	80.9± 0.3	96.3± 0.6	78.0± 1.6	84.1± 0.4
ERM-ViT	DeiT-Small	22M	87.4± 1.2	81.5± 0.8	98.1± 0.1	72.6± 3.3	84.9± 0.9
SDViT (Sultana *et al.*, 2022)	DeiT-Small	22M	87.6± 0.3	82.4± 0.4	98.0± 0.3	77.2± 1.0	86.3± 0.2
TFS-ViT (ours)	DeiT-Small	22M	89.63± 0.86	83.03± 0.31	98.58 ± 0.19	77.83 ± 1.21	87.27± 0.38
ATFS-ViT (ours)	DeiT-Small	22M	90.46± 0.67	83.00± 0.31	98.43± 0.15	78.25 ± 1.37	87.54± 0.39
ERM-ViT	T2T-ViT-14	21.5M	89.6± 0.9	81.0± 0.9	**98.9± 0.2**	77.6± 2.6	86.8± 0.4
SDViT(Sultana *et al.*, 2022)	T2T-ViT-14	21.5M	90.2± 1.2	82.7± 0.7	98.6± 0.2	80.5± 2.2	88.0 ± 0.7
TFS-ViT (ours)	T2T-ViT-14	22M	90.48± 0.72	83.62± 0.52	98.80± 0.21	83.04± 1.56	88.99± 0.45
ATFS-ViT (ours)	T2T-ViT-14	22M	**90.48± 0.15**	**84.86± 1.14**	98.53± 0.02	**84.38± 1.18**	**89.56± 0.41**

2.5.2.6 Detailed Results on the PACS Dataset

The PACS dataset contains highly different domains, ranging from photos to basic sketches. For the following analysis, we use this dataset to evaluate the robustness of TFS-ViT to such domain variability. Toward this goal, we give in Table 2.3 a breakdown of our method's performance across all PACS domains. As can be seen, TFS-ViT outperforms vanilla ERM-ViT on all domains but one (Photos), improving the average accuracy by 2.76%. In particular, it achieves a significant improvement of 6.78% for the Sketch domain, which is the most challenging one due to its large domain shift.

2.5.2.7 Computational Overhead

The computational overhead of domain generalization approaches compared to vanilla models is a major roadblock to their use in real-world applications. To demonstrate our method's computational efficiency, we compare in Table 2.4 the training times and GPU memory requirements of TFS-ViT against ERM-ViT, for our biggest model which has four layers. As reported, TFS-ViT only incurs a 1.08 percent increase in training time and a 0.06 percent increase in GPU memory. This suggests that TFS-ViT can be employed without having to worry about added computational costs.

Table 2.4 Computational Statistics for training on three source domains of the PACS dataset for 5000 steps with a batch size of 32

Model	Training Time (hrs)	GPU Mem (GB)
ERM-ViT	0.36769	7.02028
TFS-ViT (ours)	0.37166	7.02428
ATFS-ViT (ours)	0.37317	7.02450

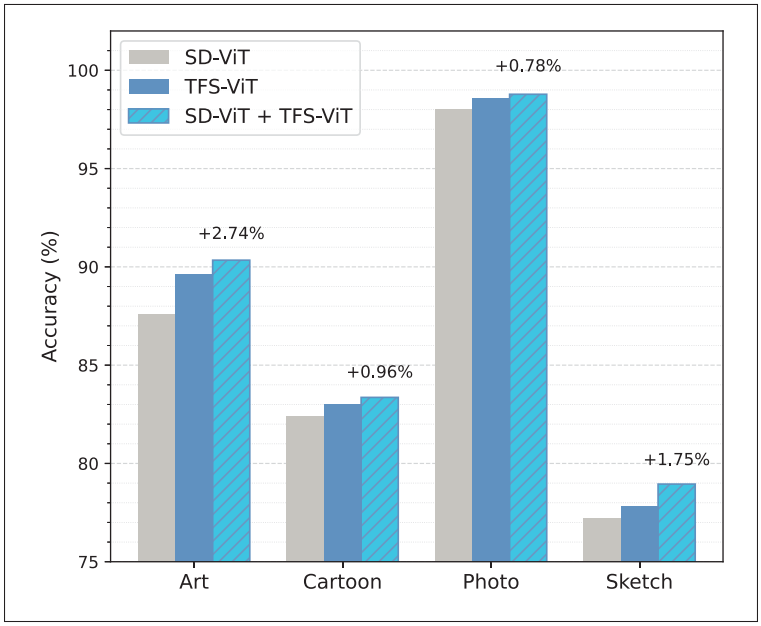

Figure 2.7 Comparison between the performance of TFS-ViT when it is applied to SDViT and the original SDViT and TFS-ViT on different domains of the PACS dataset. Annotations on the bars indicate the percentage increase in accuracy achieved by SDViT + TFS-ViT, compared to the original SDViT. The results show the extendability of our method which can be applied on top of any ViT-based method with negligible increased computational complexity

2.5.2.8 Extendability Analysis

Our TFS-ViT method is flexible and, since it has a low computational cost and simply requires to mix inner tokens of the backbone, it can be used as a module on top of any backbone or in tandem with other domain generalization techniques. To show the complementary benefit brought by our method, we added it on top of the Self-Distilled Vision Transformer (SDViT)

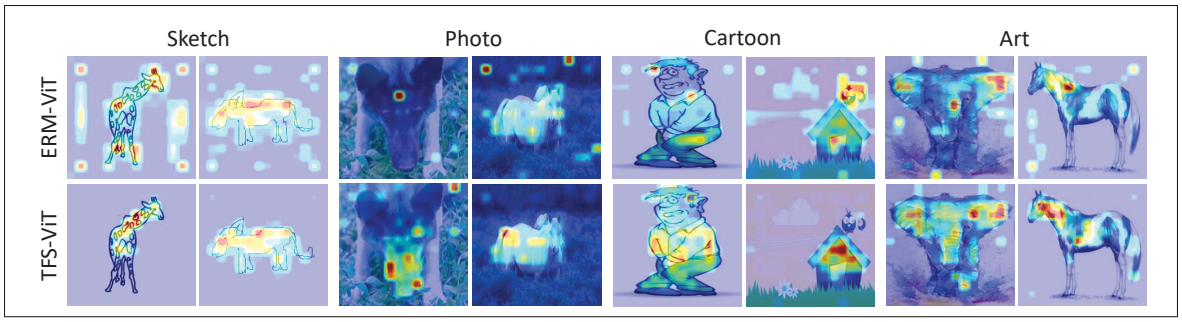

Figure 2.8 Comparison of attention maps for the CLS token of the last layer generated by two models, ERM-ViT (baseline) and TFS-ViT (with DeiT-Small backbone), on various domains of the PACS dataset as the unseen/target domain

approach for DG (Sultana *et al.*, 2022), which regularizes training with auxiliary losses in intermediate layers. As shown in Figure 2.7, TFS-ViT further improves the performance of SDViT on different domains of the PACS dataset.

2.5.2.9 Visualization of Attention Maps

In contrast to CNNs, which learn from local patterns, ViTs attempts to represent global relationships using multi-head self-attention (MSA) layers. As a result, by visualizing the attention maps of CLS token, we may get access to the most decisive parts of the input. Figure 2.8 depicts visual comparisons of final layer attention maps for ERM-ViT and TFS-ViT. It demonstrates that our method assists networks in attending to features that are more indicative of the semantics of the picture in all target domains, which lead to improving the overall generalization performance.

2.6 Conclusion

In this paper, we presented the first token-level feature stylization approach to improve the generalization capabilities of ViTs in out-of-distribution scenarios. We also proposed an innovative attention-aware stylization technique that makes use of attention maps in MSA layers to guide the augmentation toward relevant regions of the image. In a comprehensive set of

experiments using five challenging benchmark datasets, we showed our TFS-ViT method to outperform existing alternatives for DG, and to achieve state-of-art accuracy on these datasets. Detailed analyses revealed the benefit of randomly selecting layers on which to perform stylization, as well as its usefulness in single-source domain generalization and in-domain settings. Our method provides consistent improvements for very different domains, ranging from photos to sketches, has a negligible overhead in terms of training time and GPU memory, and can further boost performance when used in conjunction with other DG strategies such as self-distillation.

In this work, we demonstrated the advantage our method on two well-known ViT backbones, DeiT-Small and T2T-ViT-14, which have a number of parameters comparable to the standard ResNet-50 architecture. However, additional improvements could be achieved for more recent ViT architectures, for instance, the Swin transformer (Liu *et al.*, 2021c) which uses a shifted window strategy to learn image representations in a hierarchical manner. Moreover, while we exploited the attention maps of class tokens to steer the stylization toward salient regions in the image, more sophisticated techniques could be considered. For instance, future work could investigate the idea of stylizing foreground and background regions separately.

CHAPTER 3

FDS: FEEDBACK-GUIDED DOMAIN SYNTHESIS WITH MULTI-SOURCE CONDITIONAL DIFFUSION MODELS FOR DOMAIN GENERALIZATION

Mehrdad Noori[a] , Milad Cheraghalikhani[a] , Ali Bahri[a] , Gustavo A.Vargas Hakim[a] , David Osowiechi[a] , Moslem Yazdanpanah[a] , Ismail Ben Ayed[b] , Christian Desrosiers[a]

[a] Department of Information Technologies Engineering, École de Technologie Supérieure
[b] Department of Systems Engineering, École de Technologie Supérieure
1100 Notre-Dame West, Montreal, Quebec, Canada H3C 1K3

Paper published in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, February 2025

3.1 Abstract

Domain Generalization techniques aim to enhance model robustness by simulating novel data distributions during training, typically through various augmentation or stylization strategies. However, these methods frequently suffer from limited control over the diversity of generated images and lack assurance that these images span distinct distributions. To address these challenges, we propose **FDS**, **F**eedback-guided **D**omain **S**ynthesis, a novel strategy that employs diffusion models to synthesize novel, pseudo-domains by training a single model on all source domains and performing domain mixing based on learned features. By incorporating images that pose classification challenges to models trained on original samples, alongside the original dataset, we ensure the generation of a training set that spans a broad distribution spectrum. Our comprehensive evaluations demonstrate that this methodology sets new benchmarks in domain generalization performance across a range of challenging datasets, effectively managing diverse types of domain shifts. The code can be found at: https://github.com/Mehrdad-Noori/FDS

3.2 Introduction

Deep learning architectures, including Convolutional Neural Networks (CNNs) and Vision Transformers (ViTs), have significantly advanced the field of computer vision, achieving state-of-art results in tasks like classification, semantic segmentation, and object detection. Despite these

advancements, such models commonly operate under the simplistic assumption that training data (source domain) and post-deployment data (target domain) share identical distributions. This overlook of distributional shifts results in performance degradation when models are exposed to out-of-distribution (OOD) data (Recht *et al.*, 2019; Hendrycks & Dietterich, 2019b). Domain adaptation (DA) (Lu *et al.*, 2020; Saito *et al.*, 2018) and Test-Time Adaptation (or Training) (Wang *et al.*, 2020a; Hakim *et al.*, 2023; Osowiechi *et al.*, 2023a) strategies have been developed to mitigate this issue by adjusting models trained on the source domain to accommodate a predefined target domain. Nonetheless, these strategies are constrained by their dependence on accessible target domain data for adaptation, a prerequisite that is not always feasible in real-world applications. Furthermore, adapting models to each novel target domain entails considerable computational overhead, presenting a practical challenge to their widespread implementation.

Domain generalization (DG) (Blanchard *et al.*, 2011) aims to solve the issue of domain shift by training models using data from one or more source domains so that they perform well *out-of-the-box* on new, unseen domains. Recently, various techniques have been developed to tackle this problem (Zhou *et al.*, 2022b; Wang *et al.*, 2022a), including *domain aligning* (Hu *et al.*, 2020; Mahajan *et al.*, 2021; Li *et al.*, 2020), *meta-learning* (Li *et al.*, 2018a; Balaji *et al.*, 2018), *data augmentation* (Shi *et al.*, 2020b; Volpi *et al.*, 2018; Shankar *et al.*, 2018), *ensemble learning*(Zhou *et al.*, 2021), *self-supervised learning* (Carlucci *et al.*, 2019a; Albuquerque *et al.*, 2020), and *regularization methods* (Huang *et al.*, 2020; Cha *et al.*, 2021). These strategies are designed to make models more adaptable and capable of handling data that they were not explicitly trained on, making them more useful in real-world situations where the exact nature of future data cannot be predicted. Among these techniques, a notable category focuses on synthesizing samples from different distributions to mimic target distributions. This is achieved through strategies like *image transformation* (Shi *et al.*, 2020b; Zhang *et al.*, 2020), *style transfer* (Somavarapu *et al.*, 2020; Borlino, D'Innocente & Tommasi, 2021), *learnable augmentation networks* (Zhou *et al.*, 2020a,b; Carlucci *et al.*, 2019b), and *feature-level stylization* (Zhou *et al.*, 2021; Noori *et al.*, 2024a; Cheraghalikhani *et al.*, 2024). However, many of these methods face challenges

in controlling the synthesis process, often resulting in limited diversity where primarily only textures are altered.

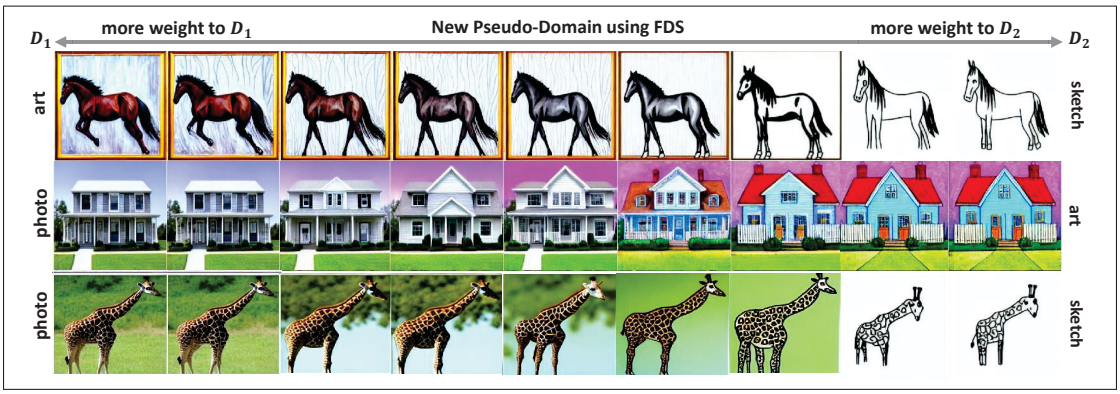

Figure 3.1 Generating new, pseudo-domains with FDS: Comprehensive distribution coverage from domain D_1 to D_2

In this study, we introduce an innovative approach using diffusion model, named Feedback-guided Domain Synthesis (FDS), to address the challenge of domain generalization. Known for their exceptional ability to grasp intricate distributions and semantics, diffusion models excel at producing high-quality, realistic samples (Dhariwal & Nichol, 2021; Ho, Jain & Abbeel, 2020; Rombach, Blattmann, Lorenz, Esser & Ommer, 2022; Song, Meng & Ermon, 2020). We exploit this strength by training a single diffusion model that is conditioned on various *domains* and *classes* present in the training dataset, aiming to master the distribution of source domains.

As illustrated in Figure 3.1, through the process of domain interpolation and mixing during generation, we create images that appear to originate from novel, pseudo-domains. To ensure the development of a robust classifier, we initially select generated samples that are difficult for a model trained solely on the original source domains to classify. Subsequently, we train the model using a combination of these challenging images and the original dataset. In this manner, we make sure that our model is fed with images of the widest possible diversity, thereby significantly enhancing its ability to generalize across unseen domains. Our contributions can be summarized as follows:

- We introduce a novel DG approach that leverages diffusion models conditioned on multiple domains and classes to generate samples from novel, pseudo-domains through domain interpolation. This approach increases the diversity and realism of the generated images;
- We propose an innovative strategy for selecting images that pose a challenge to a classifier trained on original images, ensuring the diversity of the final sample set. By incorporating these challenging images with the original dataset, we ensure a comprehensive and diverse training set, significantly improving the model's generalization capabilities;
- We conduct extensive experiments across various benchmarks and perform different analyses to validate the effectiveness of our method. These experiments demonstrate FDS's ability to significantly improve the robustness and generalization of models across a wide range of unseen domains, achieving SOTA performance.

3.3 Related Works

3.3.1 Domain Generalization (DG)

Domain Generalization, a concept first introduced by Blanchard et al. in 2011 (Blanchard *et al.*, 2011), has seen growing interest in the field of computer vision. This interest has spurred the development of a broad spectrum of methods aimed at enabling models to generalize across unseen domains. These include approaches based on *domain alignment* like moment matching (Peng *et al.*, 2019), discriminant analysis (Hu *et al.*, 2020) and domain-adversarial learning (Li *et al.*, 2018c), *meta-learning* approaches (Li *et al.*, 2018a; Balaji *et al.*, 2018) which solve a bi-level optimization problem where the model is fine-tuned on meta-source domains to minimize the error on a meta-target domain, *ensemble learning* techniques (Zhou *et al.*, 2021; Mancini *et al.*, 2018) improving robustness to OOD data by training multiple models tailored to specific domains, *self-supervised learning* methods fostering domain-agnostic representations through the pre-training of models on unsupervised tasks (Carlucci *et al.*, 2019a; Wang *et al.*, 2020b; Ghifary *et al.*, 2015) or via contrastive learning (Kim *et al.*, 2021b), approaches leveraging *disentangled representation learning* (Li *et al.*, 2017; Chattopadhyay *et al.*, 2020) segregating

domain-specific features from those common across all domains, and *regularization* methods which build on the empirical risk minimization (ERM) framework (Gulrajani & Lopez-Paz, 2021), incorporating additional objectives such as distillation (Wang, Li, Chau & Kot, 2021b; Sultana *et al.*, 2022), stochastic weight averaging (Cha *et al.*, 2021), or distributionally robust optimization (DRO) (Sagawa *et al.*, 2019) to promote generalization.

A key strategy in DG, *data augmentation* aims to enhance model robustness against domain shifts encountered during deployment. This objective is pursued through a variety of techniques, such as learnable augmentation, off-the-shelf style transfer, and augmentation at the feature level. Learnable augmentation models utilizes networks to create images from training data, ensuring their distribution diverges from that of the source domains (Zhou *et al.*, 2020a; Carlucci *et al.*, 2019b; Zhou *et al.*, 2020b). Meanwhile, off-the-shelf style transfer based methods seek to transform the appearance of images from one domain to another or modify their stylistic elements, often through Adaptive Instance Normalization (AdaIN)(Huang & Belongie, 2017; Somavarapu *et al.*, 2020). Unlike most augmentation approaches that modify pixel values, some propose altering features directly, a technique inspired by the finding that CNN features encapsulate style information(Mancini *et al.*, 2020; Zhou *et al.*, 2021) in their statistics.

3.3.2 Diffusion Models

Diffusion models have recently surpassed Generative Adversarial Networks (GANs) as the leading technique for image synthesis. Innovations like denoising diffusion probabilistic models (DDPMs) (Ho *et al.*, 2020) and denoising diffusion implicit models (DDIMs) (Song *et al.*, 2020) have significantly sped up the image generation process. Rombach et al. (Rombach *et al.*, 2022) introduced latent diffusion models (LDMs), also known as stable diffusion models, which enhance both training and inference efficiency while facilitating text-to-image and image-to-image conversions. Extensions of these models, such as stable diffusion XL (SDXL) (Podell *et al.*, 2023) and ControlNet (Zhang, Rao & Agrawala, 2023b), have been developed to further guide the generation process with additional inputs, such as depth or semantic information, allowing for more controlled and versatile image creation. Recent studies have shown that augmenting datasets

using diffusion models can improve performance in general vision tasks (Azizi, Kornblith, Saharia, Norouzi & Fleet, 2023; Dunlap *et al.*, 2023).

Despite the capabilities of diffusion models in generating images, their application in domain generalization has been minimally explored. Miao et al. (Miao, Yuan, Zhang, Wu & Kuang, 2024) introduced DomainDiff, a method that boosts OOD generalization by training a Word-to-Image Mapping (WIM) with diffusion models to generate additional synthetic data. Yue et al. (Yu, Liu, Yang & Wang, 2023a) proposed a method called DSI to address OOD prediction by transforming testing samples back to the training distribution using multiple diffusion models, each trained on a single source distribution. While effective, DSI requires multiple models during prediction, making it impractical for real-time deployment. In contrast, our method only uses diffusion during training to synthesize pseudo-domains and select more challenging images, keeping the computational complexity the same during testing while achieving significantly better performance.

In another study, CDGA (Hemati *et al.*, 2023) uses diffusion models to generate synthetic images that fill the gap between different domain pairs through a simple interpolation. While effective, CDGA relies on generating a large number of synthetic images (e.g., 5M for PACS) and employs naive interpolation. Additionally, CDGA generates samples with additional text descriptions that are not provided in standard DG benchmarks. In contrast, our method employs novel and more efficient interpolation techniques and a unique filtering mechanism that selects challenging images. We demonstrate that both of these innovations are crucial for improving generalization.

3.4 Theoretical Motivation

Our classifier, represented by f with parameters θ, aims to craft a unified model from n source domains $\{\mathcal{D}^1, \ldots, \mathcal{D}^n\}$ that adapts to a novel target domain $\mathcal{D}^{\mathbf{T}}$. Within any domain $\mathcal{D}$, we measure classification loss by

$$\mathcal{L}_{\mathcal{D}}(\theta) \;=\; \mathbb{E}_{(x,y)\sim\mathcal{D}}\left[\ell(f(x;\theta),y)\right], \tag{3.1}$$

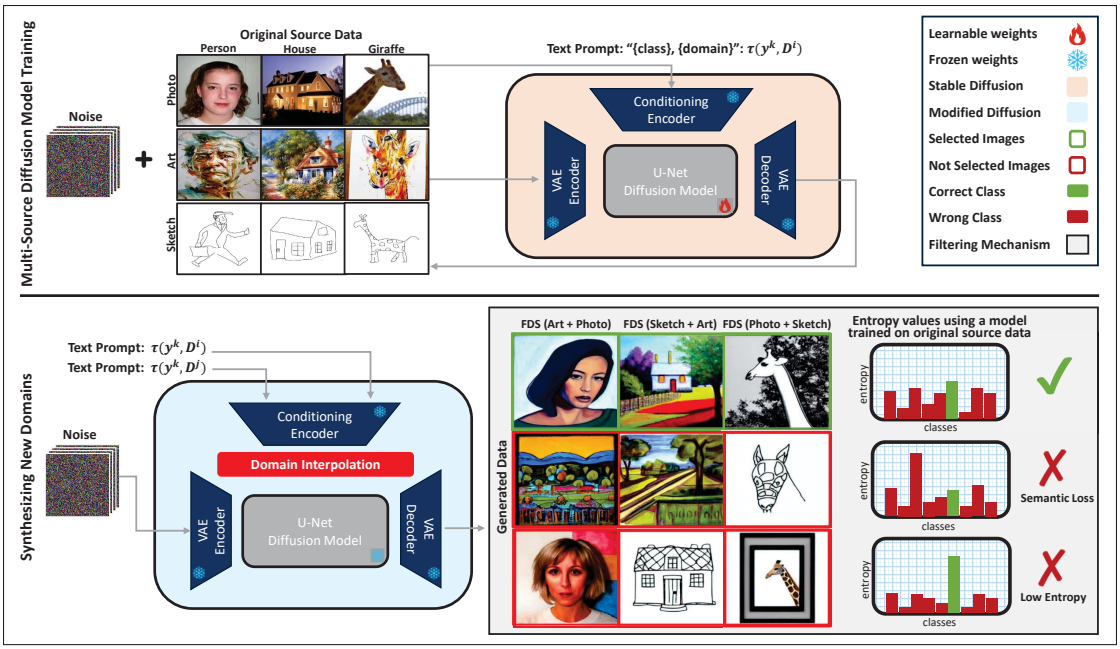

Figure 3.2 Overview of the proposed architecture for FDS. (top) Multi-source training of diffusion model conditioned on class and domain of the training images. (bottom) Generating novel pseudo-domain using the proposed interpolation and filtering mechanism of FDS

where x and y denote the input and its corresponding label, respectively, and $\ell(f(x;\theta), y)$ is the cross entropy loss in this work.

Empirical Risk Minimization (ERM) (Sain, 1996) forms the foundation for training our models, aiming to reduce the mean loss across training domains:

$$\min_{\theta} \sum_{i} \frac{1}{|\mathcal{D}^i|} \sum_{k=1}^{|\mathcal{D}^i|} \ell(f(x_i^k;\theta), y_i^k) \tag{3.2}$$

Here, $|\mathcal{D}^i|$ counts the number of samples in domain i, with x_i^k as the k-th sample and y_i^k its label. However, the accuracy of ERM-trained models drops when data shifts occur across domains due to inadequate OOD generalization. To enhance ERM, Chapelle et al. introduced Vicinal Risk Minimization (VRM) (Chapelle, Weston, Bottou & Vapnik, 2000), which substitutes point-wise estimates with density estimation in the vicinity distribution around every observation within each domain. Practical implementation often involves data augmentation to introduce synthetic

samples from these density estimates. Traditional augmentation processes a single data point x_i^k from domain i to yield $\tilde{x}_i^k = g(x_i^k)$, where $g(\cdot)$ denotes a basic transformation and $\tilde{x}_i^k$ is the modified data point.

Despite the potential of VRM to boost performance on OOD samples, it does not completely bridge domain gaps. The root cause lies in the inability of ERM methods to anticipate shifts in data distribution, which simple augmentation within domains does not address. Hence, domain generalization requires more robust transformations to extend the model's applicability beyond training domains. Muller et al. (Müller & Hutter, 2021) have demonstrated that a wider range of transformations outperforms standard methods. However, Aminbeidokhti et al. (Aminbeidokhti *et al.*, 2024b) advise that aggressive augmentations could distort the essential characteristics of images, pointing to the necessity of a mechanism to filter out extreme alterations. The emergence of diffusion models, proficient in reproducing diverse data distributions, facilitates such advanced sampling approaches. Our goal, therefore, is to generate images that span across domain gaps, lessen the variability between data distributions, and introduce a system to exclude trivial or excessive modifications.

3.5 Method

To advance the generalization ability of our classifier, we aim to utilize generated image samples that traverse domain gaps yet retain class semantics. This objective is realized through a three-step methodology at the core of our FDS approach. **Step 1:** we begin by training an image generator that masters the class-specific distributions of all source domains, enabling the synthesis of class-consistent samples within those domains. **Step 2:** we then introduce a strategy for generating inter-domain images to span the domain gaps effectively. **Step 3:** last, we filter overly simplistic samples from synthetic inter-domain images and train the final model on this refined set alongside original domain images for enhanced OOD generalization. The process is illustrated in Figure 3.2.

3.5.1 Image Generator

Diffusion Models are designed to approximate the data distribution $p(x)$ by reversing a predefined Markov Chain of T steps, effectively denoising a sample in stages. These stages are modeled as a sequence of denoising autoencoder applications $e_\theta(x_t, t)$, for $t = 1 \dots T$, which gradually restore their input x_t. This iterative restoration is formalized by the following objective:

$$\mathcal{L}_{DM} = \mathbb{E}_{x,\epsilon\sim\mathcal{N}(0,1),t}\left[\|\epsilon - \epsilon_\theta(x_t, t)\|^2\right], \tag{3.3}$$

where t is drawn uniformly from $\{1, \dots, T\}$. Utilizing perceptual compression models, encoded by $\mathcal{E}$ and decoded by $\tilde{\mathcal{E}}$, we access a latent space that filters out non-essential high-frequency details. By placing the diffusion process in this compressed space, our objective is then reformulated as

$$\mathcal{L}_{LDM} = \mathbb{E}_{\mathcal{E}(x),\epsilon\sim\mathcal{N}(0,1),t}\left[\|\epsilon - \epsilon_\theta(z_t, t)\|^2\right]. \tag{3.4}$$

Diffusion models can also handle conditional distributions $p(z|c)$. To enable this conditioning, building upon the work of Rombach et al. (Rombach *et al.*, 2022), we utilize a condition encoder τ_θ that maps c onto an intermediate representational space $\tau(c) \in \mathbb{R}^{M\times d_\tau}$. Following Stable Diffusion, we employ the CLIP-tokenizer and implement τ as a transformer to infer a latent code. This representation is subsequently integrated into the UNet using a cross-attention mechanism, culminating in our final enhanced objective:

$$\mathcal{L}^{\text{cond}}_{LDM} = \mathbb{E}_{\mathcal{E}(x),\, c,\, \epsilon\sim\mathcal{N}(0,1),t}\left[\|\epsilon - \epsilon_\theta(z_t, t, \tau(c))\|^2\right], \tag{3.5}$$

This approach enables a single diffusion model to understand and generate images across different domains for each class, using text-based conditions (prompts). By dynamically adjusting conditions, the model efficiently learns varied representations without needing multiple models for each domain. Thus, we employ a textual template $c(x)$, denoting "`[y], [D]`" where `y` is the class label and `D` is the domain name of the input x, and proceed to train our diffusion

model using this template across all images from the source domains. Upon completing training, we can create a new sample for class y^k, belonging to the set $\{y^1, \ldots, y^m\}$, within domain $\mathcal{D}^i$. This is achieved by decoding a denoised representation after t timesteps, $\tilde{x}_t^{i,k} = \tilde{\mathcal{E}}(\Phi_t(\mathcal{D}^i, y^k))$, where $\Phi_t(\mathcal{D}^i, y^k)$ is the denoised representation that originates from random Gaussian noise conditioned on $\mathcal{D}^i$ and y^k.

3.5.2 Domain Mixing

We propose two mixing strategies to synthesize images from new, pseudo distributions, based on *noise-level interpolation* and *condition-level interpolation*.

3.5.2.1 Noise Level Interpolation

Consider our dataset comprising n source domains $\{\mathcal{D}^1, \ldots, \mathcal{D}^n\}$ and target classes y^k from the set $\{y^1, \ldots, y^m\}$. Utilizing a trained image generator that initiates with random Gaussian noise, we denoise this input over t steps to produce a synthetic, denoised representation $\tilde{z}_t^{i,k} = \Phi_t(\mathcal{D}^i, y^k)$ indicative of domain $\mathcal{D}^i$ and class y^k. To synthesize a sample that merges the characteristics of domains $\mathcal{D}^i$ and $\mathcal{D}^j$ for class y^k, we employ a single diffusion model, conditioned dynamically to capture the essence of both domains. This process aims to generate a sample that embodies the transitional features between these domains, effectively bridging the domain gap.

The model begins its process from the same initial random Gaussian noise, adapting its denoising trajectory under two distinct conditions, $\tau(\mathcal{D}^i, y^k)$ and $\tau(\mathcal{D}^j, y^k)$, up to a specific timestep T. This dual-conditioned approach ensures that the evolving representation up to T incorporates influences from both domains, guided by the respective conditional inputs.

From timestep T onwards, until the final representation is formed, the model blends the outputs from these dual paths at each step. Specifically, for each step $t > T$, we form a mixed representation $\tilde{z}_t$ as:

$$\tilde{z}_t = \alpha\, \Phi_t(\mathcal{D}^i, y^k) + (1-\alpha)\, \Phi_t(\mathcal{D}^j, y^k), \tag{3.6}$$

where α is a predefined mixing coefficient that dictates the blend of domain characteristics in the output. This combined representation $\tilde{z}_t$ is then used as the basis for the model's next denoising step, integrating features from both $\mathcal{D}^i$ and $\mathcal{D}^j$ for class y^k. Through this iterative mixing, the model ensures a gradual and cohesive fusion of domain-specific attributes, leading to a synthesized sample that seamlessly spans the gap between the domains for class y^k.

3.5.2.2 Condition Level Interpolation

We also propose Condition Level Interpolation to generate images that effectively bridge domain gaps. This technique relies on manipulating the conditions fed into our diffusion model to guide the synthesis of new samples. Specifically, for a target class y^k and two distinct domains, $\mathcal{D}^i$ and $\mathcal{D}^j$, we employ our encoder $\tau(c)$ to create separate condition representations for each domain-class pair: $(y^k, \mathcal{D}^i)$ and $(y^k, \mathcal{D}^j)$.

The core of this strategy involves blending these condition representations using a mixing coefficient α, leading to a unified condition:

$$c_{\text{mixed}} = \alpha\, \tau(c_{y^k}, \mathcal{D}^i) + (1-\alpha)\, \tau(c_{y^k}, \mathcal{D}^j). \tag{3.7}$$

This mixed condition c_{mixed} then orchestrates the generation process from the initial step, ensuring that the diffusion model is consistently influenced by attributes from both domains. By initiating this conditioned blending from the beginning of the diffusion process, we ensure a harmonious integration of domain characteristics throughout the generation of the synthetic image.

3.5.3 Filtering Mechanism

Through our mixing strategies, we create synthetic samples $\tilde{x}^k_{i,j}$ that not only synthesize class y^k traits but also blend features from domains $\mathcal{D}^i$ and $\mathcal{D}^j$, thereby aiming to bridge the domain gaps. This approach generates a synthetic dataset comprising $\tilde{N}$ samples for each combination of class index k and domain index pair (i, j), structured as $S^k_{i,j} = \{\tilde{x}^{k,(r)}_{i,j} \mid 1 \leq i < j \leq n,\ 1 \leq$

$k \leq m,\ 1 \leq r \leq \tilde{N}\}$, ensuring diversity via distinct random Gaussian noise initiation for each sample.

The utility of these synthetic samples in improving model generalization varies, prompting an entropy-based evaluation to identify those with the greatest potential. High entropy scores, indicating prediction uncertainty by a classifier $h(x)$ trained on the original dataset, suggest that such samples may come from previously unseen distributions. This characteristic posits these high-entropy samples as prime candidates for training, hypothesized to challenge the classifier significantly and aid in covering the domain gaps. Further refining this selection, we only include samples correctly predicted as their target class by $h(x)$, ensuring the exclusion of samples that have lost semantic integrity during the diffusion process.

Let $C_{i,j}^k$ be the subset of samples in $S_{i,j}^k$ which are correctly classified by $h(x)$:

$$C_{i,j}^k \ = \ \left\{\tilde{x}_{i,j}^{k,(r)} \mid h(\tilde{x}_{i,j}^{k,(r)}) = y^k\right\}. \tag{3.8}$$

We choose from $C_{i,j}^k$ the N_L samples with highest entropy (the entropy of a k-class discrete probability distribution p is given by $\mathcal{H}(p) = -\sum_k p_k \log p_k$) to form a set of selected samples $\tilde{\mathcal{D}}_{i,j}^k$. Last, we combine the synthetic samples, created for each class and domain pair, to the original dataset O to obtain the final augmented training set

$$\mathcal{A} \ = \ O \ \cup \ \left\{\tilde{\mathcal{D}}_{i,j}^k \mid 1 \leq i < j \leq n,\ 1 \leq k \leq m\right\} \tag{3.9}$$

Training the final classifier on $\mathcal{A}$ not only enriches the dataset but also ensures robust model generalization across diverse domain landscapes.

3.6 Experimental Setup

Datasets. Following (Gulrajani & Lopez-Paz, 2021), we compare our proposed approach to the current state-of-art using three challenging datasets–**12 individual target domains**–with different characteristics: `PACS` (Li *et al.*, 2017), `VLCS` (Fang *et al.*, 2013), and `OfficeHome` (Venkateswara

et al., 2017). The PACS dataset has a total of 9,991 photos divided into four distinct domains, $d \in$ {Art, Cartoon, Photo, Sketch}, and seven distinct classes. The second dataset, `VLCS`, comprises 10,729 photos from four separate domains, $d \in$ {Caltech101, LabelMe , SUN09, VOC2007}, and five different classes. The third dataset, `OfficeHome`, includes a total of 15,588 photos taken from four domains, $d \in$ {Art, Clipart, Product, Real}, and 65 classes.

Implementation Details. To ensure a fair comparison, we adopt the DomainBed framework (Gulrajani & Lopez-Paz, 2021), a comprehensive benchmark that encompasses prominent domain generalization (DG) methodologies under a uniform evaluation protocol. Following this framework, we employ a leave-one-out strategy for DG dataset assessment where one domain serves as the test set while the others form the training set. A subset of the training data, constituting 20%, is designated as the validation set[1]. The aggregate result for each dataset represents the mean accuracy derived from varying the test domain. To ensure reliability, experiments are replicated three times, each with a unique seed. Moreover, to rigorously test our method, we try both ERM and SWAD classifiers with FDS. The former is considered a baseline in standard training methods, while the latter serves as a baseline for weight averaging (WA) methods. SWAD is essentially ERM but with weight averaging applied during training using multiple steps based on the validation set. To have a fair comparison with other methods, for the ERM baseline, we strictly follow the hyperparameter tuning proposed in (Cha *et al.*, 2021). For SWAD, as in the original work, we did not tune any parameters and used the default values. For image synthesis, the original Stable Diffusion framework (Rombach *et al.*, 2022) is utilized with DDIM = 50 steps. The PACS and VLCS datasets prompt the generation of $N = 32,000$ samples per class, whereas OfficeHome, with its 65 classes, necessitates $N = 16,000$ samples. Image generation spans an interpolation range of $\alpha \in [0.3, 0.7]$ and a Noise Level Interpolation range of $\mathcal{T} \in [20, 45]$. This diversified parameter selection, rather than optimizing hyperparameters per dataset, acknowledges each domain's unique shift. Our filtering mechanism then identifies the most informative images from the generated pool. The impact of dataset size, N_L, is studied further in Section 3.7.2.

[1] The model with peak accuracy on this validation set is selected for evaluation on the test domain, providing unseen domain accuracy.

Table 3.1 Leave-one-out accuracy (%) results on the PACS, VLCS, and OfficeHome benchmarks. Aug. indicates whether advanced augmentation or domain mixing techniques are used. The **best results** and second-best results are highlighted

	Method	Aug.	PACS	VLCS	Office	Avg.
Standard Methods	ERM (*baseline*) (Gulrajani & Lopez-Paz, 2021)	✗	85.5± 0.2	77.5± 0.4	66.5± 0.3	76.5
	ERM (*reproduced*)	✗	84.3± 1.1	76.2± 1.1	64.6± 1.1	75.0
	IRM (Arjovsky *et al.*, 2019)	✗	83.5± 0.8	78.5± 0.5	64.3± 2.2	75.4
	GroupDRO (Sagawa *et al.*, 2019)	✗	84.4± 0.8	76.7± 0.6	66.0± 0.7	75.7
	Mixup (Yan *et al.*, 2020)	✓	84.6± 0.6	77.4± 0.6	68.1± 0.3	76.7
	CORAL (Sun & Saenko, 2016)	✗	86.2± 0.3	78.8± 0.6	68.7± 0.3	77.9
	MMD (Li *et al.*, 2018b)	✗	84.6± 0.5	77.5± 0.9	66.3± 0.1	76.1
	DANN (Ganin *et al.*, 2016)	✗	83.6± 0.4	78.6± 0.4	65.9± 0.6	76.0
	SagNet (Nam *et al.*, 2021a)	✓	86.3± 0.2	77.8± 0.5	68.1± 0.1	77.4
	RSC (Huang *et al.*, 2020)	✓	85.2± 0.9	77.1± 0.5	65.5± 0.9	75.9
	Mixstyle (Zhou *et al.*, 2021)	✓	85.2± 0.3	77.9± 0.5	60.4± 0.3	74.5
	mDSDI (Bui *et al.*, 2021)	✗	86.2± 0.2	79.0± 0.3	69.2± 0.4	78.1
	SelfReg (Kim *et al.*, 2021b)	✓	85.6± 0.4	77.8± 0.9	67.9± 0.7	77.1
	DCAug (Aminbeidokhti *et al.*, 2024b)	✓	86.1± 0.7	78.6± 0.4	68.3± 0.4	77.7
	DomainDiff (Miao *et al.*, 2024)	✓	85.6± 0.6	–	63.7± 0.6	–
	DSI (Yu *et al.*, 2023a)	✓	86.9± 1.4	–	–	–
	CDGA (Hemati *et al.*, 2023)	✓	88.5± 0.5	79.6± 0.3	68.2± 0.6	78.8
	ERM + FDS (ours)	✓	**88.8± 0.1**	**79.8± 0.5**	**71.1± 0.1**	**79.9**
WA Methods	SWAD (*baseline*) (Cha *et al.*, 2021)	✗	88.1± 0.1	79.1± 0.1	70.6± 0.2	79.3
	SWAD (*reproduced*)	✗	88.1± 0.4	78.9± 0.5	70.3± 0.4	79.1
	SelfReg SWA (Kim *et al.*, 2021b)	✓	86.5± 0.3	77.5± 0.0	69.4± 0.2	77.8
	DNA (Chu *et al.*, 2022)	✗	88.4± 0.1	79.0± 0.1	71.2± 0.1	79.5
	DIWA (Rame *et al.*, 2022b)	✓	88.8± 0.4	79.1± 0.2	71.0± 0.1	79.6
	TeachDCAug (Aminbeidokhti *et al.*, 2024b)	✓	88.4± 0.2	78.8± 0.4	70.4± 0.2	79.2
	SWAD + FDS (ours)	✓	**90.5± 0.3**	**79.7± 0.5**	**73.5± 0.4**	**81.3**

3.7 Results

We first compare the performance of our FDS approach against SOTA DG methods across three benchmarks. We then present a detailed analysis investigating several key aspects of our approach, including the effectiveness of each proposed component, a comparison of different mixing strategies, its regularization capabilities, domain diversity visualization and quantification, the impact of data size, the efficacy of our filtering mechanism, and stability analysis during training.

Table 3.2 Comparative analysis of FDS component effects on accuracy (%) across PACS dataset domains. "Basic Gen." refers to generation without interpolation or filtering

Module	Target Domains				
	Art	**Cartoon**	**Photo**	**Sketch**	**Avg.**
Baseline (SWAD (Cha *et al.*, 2021))	89.49± 0.2	83.65± 0.4	97.25± 0.2	82.06± 1.0	88.11± 0.4
+ Basic Gen.	89.87± 0.1	85.59± 0.6	97.50± 0.3	83.07± 0.4	89.01± 0.4
+ Interpolation	91.38± 0.2	85.20± 0.6	97.73± 0.1	84.27± 0.9	89.65± 0.4
+ Filtering	**91.80± 0.3**	**86.03± 0.8**	**98.05± 0.2**	**86.11± 0.1**	**90.50± 0.3**

Additional analysis, visualizations, and detailed tables can be found in the Supplementary Material.

3.7.1 Comparison with the State-of-the-art

In Table 3.1, we compare our approach with recent methods for domain generalization as outlined in the DomainBed framework (Gulrajani & Lopez-Paz, 2021). Our method, when added to the baseline ERM classifier, shows an impressive improved accuracy of 4.5% on the PACS dataset using the ResNet-50 model. On the VLCS dataset, our method sees a 3.6% increase in accuracy over the ERM. For the OfficeHome dataset, our method outperforms the ERM baseline by 6.5%. To test the strength of our method, we also applied it to SWAD, which is a the baseline for weight averaging (WA) domain generalization methods. Here, our method improves the performance by 2.4%, 0.8%, and 3.2% on the PACS, VLCS, and OfficeHome datasets, respectively.

Furthermore, when comparing with previous SOTA methods, our method outperforms CDGA (Hemati *et al.*, 2023) by 0.3% on PACS, 0.2% on VLCS, and 2.9% on OfficeHome. In the context of weight averaging methods, our approach surpasses DIWA (Rame *et al.*, 2022b), which trains multiple independent models, by 1.7% on PACS, 0.6% on VLCS, and 2.5% on OfficeHome, setting a new benchmark for domain generalization. Please refer to the Supplementary Material for the full, detailed results for each dataset and its domains.

Table 3.3 Impact of different interpolation strategies of FDS on PACS accuracy (%)

Strategy	Target Domains				
	Art	**Cartoon**	**Photo**	**Sketch**	**Avg.**
Baseline (SWAD (Cha *et al.*, 2021))	89.49± 0.2	83.65± 0.4	97.25± 0.2	82.06± 1.0	88.11± 0.4
Noise Level Interpol.	**91.95± 0.3**	83.23± 0.1	97.56± 0.1	85.85± 0.3	89.65± 0.2
Condition Level Interpol.	91.80± 0.3	**86.03± 0.8**	**98.05± 0.2**	**86.11± 0.1**	**90.50± 0.3**
Both	91.13± 0.4	82.98± 0.2	97.90± 0.2	85.62± 0.8	89.41± 0.4

3.7.2 Further Analysis

For this section, we use the SWAD baseline to analyze the performance of our proposed method FDS, due to its stable performance as a WA method. Final settings are applied to the ERM baseline. All analyses in this section use the PACS dataset and the SWAD baseline unless otherwise stated.

Ablation Study on Different Components. To evaluate the impact of each component on generalization, we systematically introduce each module and observe the enhancements. As detailed in Table 3.2, first incorporating domain-specific synthetic samples, generated without interpolation or filtering (Basic Gen.), yields a 0.9% gain over the baseline. This increment validates our diffusion model's proficiency in capturing and replicating the class-specific distributions within domains, thus refining OOD performance through enhanced density estimation near original samples. Subsequently, integrating images from pseudo-novel domains via our interpolation strategy leads to a further 0.5% enhancement, underscoring the mechanism's effectiveness in connecting distinct domains. Lastly, applying our entropy-based filtering to eliminate overly simplistic images and images with diminished semantic relevance results in a significant 2.4% improvement against the SWAD model, a robust benchmark. These outcomes collectively underscore the efficacy of our approach in improving OOD generalization.

Mixing Strategies. Our ablation study, presented in Table 3.3, reveals that both Noise Level Interpolation and Condition Level Interpolation significantly outperform the baseline, yet their

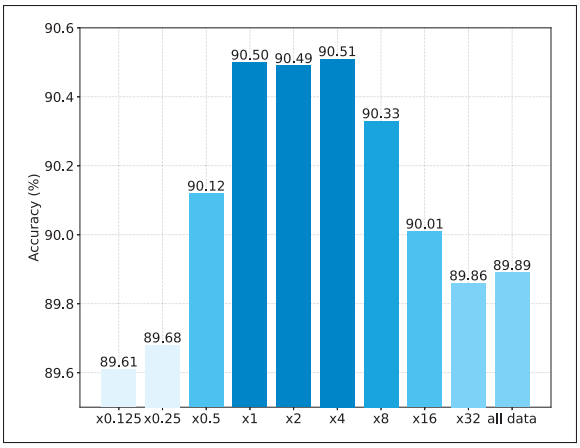

Figure 3.3 Impact of varying scales of sample size N_L relative to the average number of images per class on PACS dataset

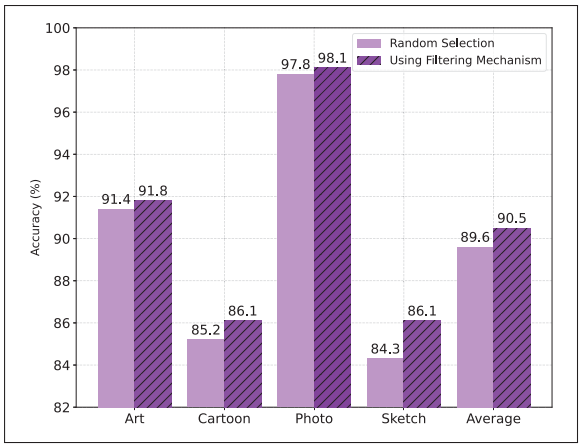

Figure 3.4 Impact of using Random Selection vs. Proposed Filtering Strategy of FDS on PACS accuracy (%)

effectiveness varies with the dataset's attributes and the nature of the domain shift. Noise Level Interpolation is optimal for minimal domain shifts, focusing on adjusting the noise aspect to bridge domain gaps. However, it falls short in scenarios with substantial domain differences, such as the transition to Cartoon or Sketch, where the source and target domains diverge significantly. In these cases, Condition Level Interpolation proves more advantageous, offering a robust mechanism for navigating complex domain shifts by manipulating higher-level semantic representations. When applying both methods simultaneously, the performance did not improve compared to using Condition Level Interpolation alone, possibly due to the increased complexity. Nonetheless, it still performed better than the baseline. Based on these results, we use Condition Level Interpolation as our final interpolation method.

Impact of Sample Size. Next, we assess the impact of the selected data size, denoted as N_L, on final performance. We consider the PACS dataset for this analysis, where the average number of images per class is 570. We systematically explore varying scales relative to this average class size, aiming to discern the optimal dataset size for enhancing OOD generalization. Our findings, detailed in Figure 3.3, reveal an initial improvement in OOD generalization with increased data size. However, excessively enlarging the dataset size begins to diminish the benefits of our

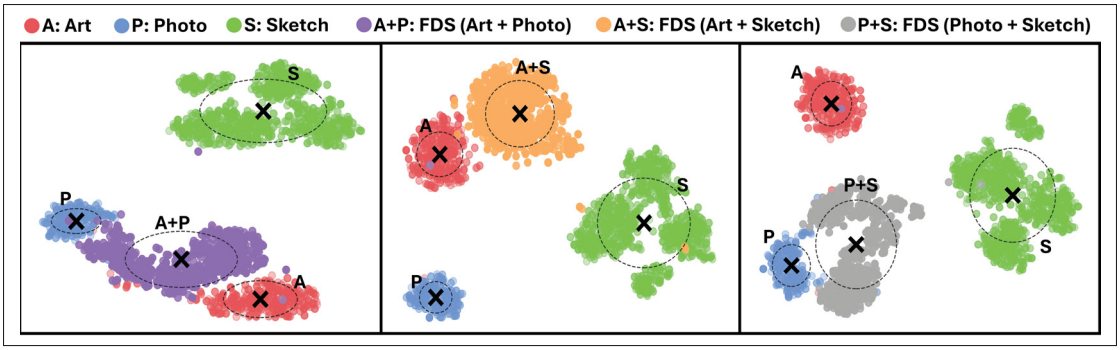

Figure 3.5 t-SNE plots showcasing the original "giraffe" class samples for the "Art", "Photo", and "Sketch" source domains of the PACS dataset, as well as the data generated with FDS

Table 3.4 Impact of filtering strategy components of FDS on accuracy (%), using three benchmarks

Method	PACS	VLCS	Office	Avg.
Baseline (SWAD (Cha *et al.*, 2021))	88.11± 0.4	78.87± 0.5	70.34± 0.4	79.11
Filtered Based on Entropy	90.50± 0.4	79.33± 0.8	71.97± 0.3	80.60
+ Reject Semantic Loss	**90.50± 0.3**	**79.73± 0.5**	**73.51± 0.5**	**81.25**

filtering mechanism, as it incorporates a broader array of samples, including those that are overly simplistic and not conducive to model improvement.

Filtering Mechanism. One might consider that the enhanced accuracy observed with our filtering mechanism might be the result of constraining the sample size for a balanced final training set. To investigate this hypothesis, Figure 3.4 contrasts the outcome of selecting N_L samples at random from all generated images per class against employing our entropy-based filtering strategy. The consistent improvement in out-of-domain (OOD) generalization across all domains, facilitated by our filtering approach, underscores its effectiveness.

Additionally, to examine the impact of excluding samples that deviate semantically which will be identified through misclassification by a classifier trained solely on the original dataset, we contrasted the accuracy between selections purely based on entropy and those refined this way

Table 3.5 Domain diversity metric (Ye *et al.*, 2022) between source domains and the target domain of the PACS dataset. "Basic Gen." refers to generation without interpolation or filtering

Data	Target Domains				
	Art	**Cartoon**	**Photo**	**Sketch**	**Avg.**
Original PACS	0.39	0.87	0.54	0.99	0.70
Basic Gen.	0.54	0.72	0.45	1.01	0.68
FDS	0.44	0.58	0.49	0.92	**0.61**

in Table 3.4. This comparison highlights the significance and efficiency of our filtering strategy in preserving semantic integrity.

Domain Diversity Visualization. To illustrate the effectiveness of our method, we present t-SNE plots of original PACS dataset samples and those generated by FDS in Figure 3.5. The t-SNE plots show distinct clusters for original source domains (here Art, Photo, Sketch) and highlight how the generated samples clearly bridge the gaps between these clusters, thus enhancing domain diversity. This expanded diversity is crucial for improving the generalization capabilities of models, as it ensures a broader spectrum of data distributions in the training set. By providing a continuous representation of the domain space, our method facilitates smoother transitions and better prepares models to handle unseen domains, ultimately contributing to more robust performance in real-world applications. Please check the supplementary materials (Figure 8) for further details.

Domain Diversity Quantification. We further validate the effectiveness of our method using a domain diversity metric based on the methodology proposed in (Ye *et al.*, 2022). This metric quantifies the diversity shift between the source domains and the target domain, providing valuable insight into how the newly generated domains using our method can improve generalization on unseen domains. Table 3.5 presents the domain diversity metric for the original PACS dataset, comparing it with samples generated using the diffusion model in its basic form (no interpolation and no filter) and our FDS method (including interpolation and filtering). We observe that our FDS method results in a lower diversity metric compared to using the original

PACS dataset or the samples generated with basic generation, suggesting that our method more effectively promotes generalization to new, unseen domains.

In-Domain Regularization Effect. In this section, we study the impact of incorporating images generated by our method on the network's accuracy, when testing on the same domains as those used for training. To provide a more comprehensive analysis, we adopted an 80/10/10 split for the source domains[2]. We selected the best model on the validation set and reported its performance on the held-out in-domain test set. As indicated in Table 3.6, our FDS approach surpasses the baseline accuracy in this setup for both the ERM and SWAD backbones. Additionally, we compared our method to two alternative strategies: *i*) duplicating and augmenting original samples (Dup. Aug.), and *ii*) generating synthetic in-domain images (same number of images as FDS) without domain interpolation (Basic Gen.). While these strategies also improve the performance, FDS achieves the highest accuracy due to its ability to generate more diverse and challenging pseudo-domains, rather than simply increasing dataset size. This confirms the value of our method not only in OOD conditions but also in standard in-domain validation, aligning with the principles of Vicinal Risk Minimization (VRM) (Chapelle *et al.*, 2000). Hence, our method may also be viewed as a regularization strategy, suitable for a broad spectrum of applications. For completeness, we also report the out-domain performance of FDS and these strategies in Table 3.7. As can be seen, the beneficial impact of FDS on OOD generalization is not simply the result of increasing the dataset size.

3.8 Conclusion

This work presented FDS, a domain generalization (DG) technique that leverages diffusion models for domain mixing, generating a diverse set of images to bridge the domain gap between source domains distribution. We also proposed an entropy-based filtering strategy that enriches the pseudo-novel generated set with images that test the limits of classifiers trained on original data, thereby boosting generalization. Our extensive experiments across multiple benchmarks

[2] This setting was only used for in-domain experiments, while the rest of the paper follows the standard 80/20 setting as used in DomainBed.

Table 3.6 Impact of FDS on in-domain PACS accuracy (%). 'A', 'C', 'P', and 'S' refer to 'Art', 'Cartoon', 'Photo', and 'Sketch'

Method	Source Domains				
	C, S, P	**A, P, S**	**A, C, S**	**A, C, P**	**Avg.**
ERM	98.05± 0.5	96.06± 0.9	95.66± 0.3	96.83± 0.6	96.65
ERM + Dup. Aug.	98.01± 0.2	97.07± 0.3	96.46± 0.2	97.28± 0.3	97.21
ERM + Basic Gen.	96.96± 0.4	96.74± 0.2	96.23± 0.2	97.37± 0.5	96.83
ERM + FDS (ours)	97.77± 0.4	97.53± 0.5	96.71± 0.1	98.02± 0.4	**97.51**
SWAD	98.48± 0.3	97.87± 0.5	97.61± 0.2	98.44± 0.0	98.10
SWAD + Dup. Aug.	98.35± 0.2	97.86± 0.3	97.38± 0.1	98.16± 0.1	97.94
SWAD + Basic Gen.	98.42± 0.4	98.23± 0.3	97.87± 0.5	98.64± 0.2	98.29
SWAD + FDS (ours)	98.70± 0.3	98.25± 0.5	97.80± 0.4	98.61± 0.0	**98.34**

Table 3.7 Leave-one-out accuracy (%) of FDS compared to other augmentation strategies

Method	Target Domains				
	Art	**Cartoon**	**Photo**	**Sketch**	**Avg.**
ERM	86.94± 0.6	80.21± 0.7	96.61± 0.4	74.45± 2.9	84.30
ERM + Dup. Aug.	85.34± 0.9	80.99± 0.9	94.98± 0.8	76.88± 2.0	84.55
ERM + Basic Gen.	87.21± 0.3	80.90± 1.9	95.71± 0.4	80.31± 2.3	86.03
ERM + FDS (ours)	90.69± 0.9	84.19± 0.6	97.21± 0.1	82.99± 0.4	**88.77**
SWAD	89.49± 0.2	83.65± 0.4	97.25± 0.2	82.06± 1.0	88.11
SWAD + Dup. Aug.	89.55± 0.3	83.00± 1.7	97.65± 0.2	81.86± 1.1	88.02
SWAD + Basic Gen.	89.87± 0.1	85.59± 0.6	97.50± 0.3	83.07± 0.4	89.01
SWAD + FDS (ours)	91.80± 0.3	86.03± 0.8	98.05± 0.2	86.11± 0.1	**90.50**

demonstrate that our method not only surpasses existing DG techniques but also sets new records for accuracy. Our analysis indicate that our approach contributes to more stable training processes when confronted with domain shifts and serves effectively as a regularization method in in-domain contexts. Notably, our technique consistently enhances performance across diverse scenarios, from realistic photos to sketches. While we exploited our trained diffusion model for covering the domain gap, more sophisticated techniques could be considered. For instance, future work could investigate the idea of generating pseudo-novel distributions via extrapolation in the domain space, in addition to interpolation. Additionally, exploring generative models for domain

synthesis, or performing domain mixing directly in a learned feature or latent space, represent promising directions that could offer greater efficiency while retaining semantic consistency.

CHAPTER 4

TEST-TIME ADAPTATION OF VISION-LANGUAGE MODELS FOR OPEN-VOCABULARY SEMANTIC SEGMENTATION

Mehrdad Noori[*a] , David Osowiechi[*a] , Gustavo A.Vargas Hakim[b] , Ali Bahri[a] , Moslem Yazdanpanah[a] , Sahar Dastani[a] , Farzad Beizaee[a] , Ismail Ben Ayed[b] , Christian Desrosiers[a]

[a] Department of Information Technologies Engineering, École de Technologie Supérieure
[b] Department of Systems Engineering, École de Technologie Supérieure
1100 Notre-Dame West, Montreal, Quebec, Canada H3C 1K3

[*]Equal Contribution
Paper published in *The Thirty-Ninth Annual Conference on Neural Information Processing Systems (NeurIPS)*, December 2025

4.1 Abstract

Recently, test-time adaptation has attracted wide interest in the context of vision-language models for image classification. However, to the best of our knowledge, the problem is completely overlooked in dense prediction tasks such as Open-Vocabulary Semantic Segmentation (OVSS). In response, we propose a novel TTA method tailored to adapting VLMs for segmentation during test time. Unlike TTA methods for image classification, our Multi-Level and Multi-Prompt (MLMP) entropy minimization integrates features from intermediate vision-encoder layers and is performed with different text-prompt templates at both the global CLS token and local pixel-wise levels. Our approach could be used as plug-and-play for any segmentation network, does not require additional training data or labels, and remains effective even with a single test sample. Furthermore, we introduce a comprehensive OVSS TTA benchmark suite, which integrates a rigorous evaluation protocol, nine segmentation datasets, 15 common synthetic corruptions, and additional real and rendered domain shifts, **with a total of 87 distinct test scenarios**, establishing a standardized and comprehensive testbed for future TTA research in open-vocabulary segmentation. Our experiments on this suite demonstrate that our segmentation-tailored method consistently delivers significant gains over direct adoption of TTA classification baselines. Code and data are available at https://github.com/dosowiechi/MLMP.

4.2 Introduction

Contrastive Vision-Language Models (VLMs) such as CLIP (Radford *et al.*, 2021a) have demonstrated remarkable generalization capabilities by aligning vision and language modalities through large-scale pre-training. This versatility has positioned VLMs as powerful foundation models for numerous downstream tasks (Lin *et al.*, 2022b; Guzhov, Raue, Hees & Dengel, 2022b; Liu *et al.*, 2023b). A promising direction for leveraging VLMs beyond classification is Open-Vocabulary Semantic Segmentation (OVSS), where models aim to segment objects beyond a pre-defined set of categories, via VLMs' zero-shot recognition capabilities. Unlike traditional segmentation methods that require pixel-wise supervision, OVSS enables generalization to unseen object categories through language-driven representations.

Although existing OVSS methods have made significant progress, they remain vulnerable to domain shifts at test time, such as environmental changes or image corruptions, which may dramatically degrade segmentation quality. In the absence of a mechanism enabling them to adapt to unseen test-time distributions, these models might lose their generalization capabilities, which limits their reliability in real-world applications. Consequently, there is an unresolved gap for the Test-Time Adaptation (TTA) of OVSS models, which would enable models to dynamically adjust both to the task shift of VLM-based segmentation and to the domain shifts encountered during inference.

To close this gap, we present a novel **M**ulti-**L**evel **M**ulti-**P**rompt (MLMP) test time adaptation strategy, the first fully test-time adaptation framework that could be plugged into *any OVSS model*, to the best of our knowledge. MLMP is lightweight and plug-and-play, boosting performance on the fly without access to labels. Its power comes from two key ideas: (*i*) adaptively integrating intermediate vision-encoder layers to harvest complementary, shift-resilient features, and (*ii*) a multi-prompt optimization that exploits VLMs' template sensitivity to provide a robust adaptation signal across diverse text-template conditions.

The core requirement for test-time adaptation is a reliable signal that faithfully reflects the current input distribution—even under severe domain shifts or corruptions. To meet this need,

MLMP begins by adaptively integrating intermediate layers of the vision encoder: earlier layers preserve fine-grained edges and textures, while deeper blocks encode semantic context, and each layer reacts differently when the data distribution changes. By aggregating these multi-level features into the adaptation process and weighting them by their confidence, MLMP harvests the most trustworthy signals for each input sample.

Beyond multi-level fusion, MLMP leverages VLMs' prompt sensitivity to model uncertainty. Prior work (Osowiechi *et al.*, 2024b; Shu *et al.*, 2022) shows that changing a prompt template, e.g., from ''a photo of a {class}" to ''an origami of a {class}", could drastically change the classification performance. *In segmentation, we show that this effect is even more extreme: per-pixel predictions under different prompts diverge far more than the single CLS token used for classification (Figure 4.1a: CLS token vs. last-layer spatial tokens). This sensitivity effect is even more pronounced in intermediate feature maps. The intermediate layers that we fuse for reliability exhibit even stronger template-specific shifts (Figure 4.1a: intermediate-layer tokens).* Instead of viewing this inconsistency as a weakness, MLMP models it directly by incorporating multi-prompt, multi-level predictions into its adaptation objective function. This multi-prompt approach not only smooths out template-specific noise, thereby reducing gradient variance and preventing degenerate collapse, but also ensures that the model yields segmentations that are consistent across diverse linguistic formulations. In this way, MLMP transforms prompt sensitivity into a powerful adaptation signal that complements its multi-level feature integration. As illustrated in Figure 4.1b, our MLMP method consistently outperforms the non-adapted baseline, achieving about 8–9 absolute mIoU improvements on domain-shifted inputs, while also boosting performance on original (non-corrupted) images. This demonstrates the benefit of our joint multi-level, multi-prompt adaptation.

We outline our key contributions as follows:

- **Plug-and-Play TTA Framework for OVSS:** We introduce MLMP, which is, to the best of our knowledge, the first fully test-time adaptation method that could be easily applied to any OVSS backbone.

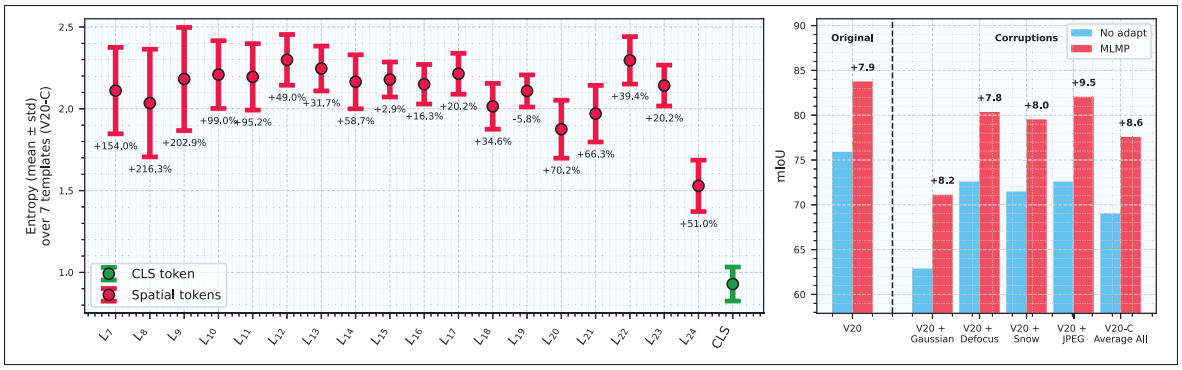

Figure 4.1 Motivation. (a) Left: Mean ± std entropy across seven text templates for the CLS token and the spatial tokens of the final and intermediate vision layers. Even the final-layer spatial tokens exhibit higher entropy and variability than CLS, and this sensitivity grows further in intermediate layers (numbers show % std increase relative to CLS). These patterns highlight pronounced prompt-induced uncertainty at multiple depths and motivate both multi-level and multi-prompt adaptation. (b) Right: mIoU of the baseline vs. MLMP on clean and corrupted data, showing consistent absolute improvements and underscoring the effectiveness of our joint adaptation strategies. Here, V20 denotes the Pascal VOC 20 dataset, and V20-C represents the average performance over its 15 synthetic corruption types. The variance in (a) is computed across all samples and all corruptions

- **Adaptive Multi-Level Fusion:** MLMP integrates features from intermediate vision-encoder layers to capture complementary, shift-resilient cues. To further enhance robustness, we propose an uncertainty-aware strategy that re-weights features from individual layers based on their prediction entropy.
- **Multi-Prompt Local-Global Test-Time Optimization:** MLMP turns prompt sensitivity into signal by directly minimizing entropy across different text prompt templates at both the global CLS token and local pixel-wise levels. This optimization naturally complements our multi-level feature fusion by enforcing consistency across linguistic perspectives and feature depths.
- **Comprehensive OVSS TTA Benchmark Suite:** We curate a rigorous evaluation protocol spanning nine mainstream segmentation datasets and 15 common synthetic corruptions, and additional real and rendered domain shifts, **with a total of 87 distinct test scenarios**, establishing a standardized and comprehensive testbed for future TTA research

in open-vocabulary segmentation. Our experiments on this suite demonstrate that MLMP consistently delivers significant gains over baselines across all scenarios.

4.3 Related Work

Test-time adaptation (TTA) for open-vocabulary semantic segmentation (OVSS) remains unexplored—existing TTA methods focus on classification or single-modality segmentation, while OVSS approaches use VLMs without any online adaptation. We bridge this gap with our proposed method MLMP, a plug-and-play TTA framework that can be applied to any OVSS method.

Test-Time Adaptation. TTA addresses domain shifts by adapting pre-trained models to unlabeled target data without source samples. Methods like PTBN (Nado *et al.*, 2020) and TENT (Wang *et al.*, 2020a) update batch statistics and affine parameters via entropy minimization but rely on large batches or augmentations. MEMO (Zhang, Levine & Finn, 2022) simplifies this with single-sample augmentations, LAME (Boudiaf *et al.*, 2022) clusters features via Laplacian smoothing, and SAR (Niu *et al.*, 2023) stabilizes adaptation using batch-agnostic normalization and sharpness-aware entropy minimization.

Test-Time Adaptation on Segmentation. TTA enhances segmentation robustness against domain shifts without source data. Methods include self-supervised adaptation via entropy minimization or contrastive learning (Wang *et al.*, 2020a; Liu *et al.*, 2021a), single-image adaptation optimizing per-image predictions (He, Zhang, Wang & Huang, 2021), and continual TTA that leverages clustering to prevent forgetting (Mummadi, Arens & Brox, 2021). Multi-modal adaptation uses cross-modal self-supervision (Valvano, Leo, Saito & Tommasi, 2023), while active TTA integrates minimal human feedback for guided refinement (Wang, Zhang, Yu & Zhang, 2023). These approaches assume a fixed, vision-only label space and rely on spatial or surrogate tasks, making them ill-suited for zero-shot, text-driven OVSS. Consequently, none have been applied to VLMs.

Open-Vocabulary Semantic Segmentation. OVSS enables segmentation of unseen categories using vision-language models like CLIP. Approaches fall into fully-supervised, weakly-supervised, and training-free categories. Fully-supervised methods use pixel-wise annotations (Barsellotti, Amoroso & Caputo, 2023; Ghiasi *et al.*, 2022; Liang *et al.*, 2023), while weakly-supervised ones leverage image-text pairs (Cai *et al.*, 2023; Cha, Mun & Roh, 2023; Chen, Li, Zhang & Liu, 2023; Xu, De Mello, Liu & Wang, 2022). Training-free OVSS avoids adaptation data but may rely on auxiliary pre-trained models (Barsellotti, Amoroso & Caputo, 2024; Corradini, Bianchi, Nozza & Hovy, 2024; Karazija, Bousselham, Bursuc, Alldieck & Pérez, 2023). Training-free OVSS approaches aim to enhance segmentation without additional training data. Some methods, such as SCLIP (Wang, Mei & Yuille, 2025), adjust self-attention mechanisms to improve feature localization, while others, like MaskCLIP (Zhou, Loy & Dai, 2022a), refine feature extraction from CLIP's visual backbone. GEM (Bousselham, Bursuc, Alldieck & Pérez, 2024) introduces additional optimization techniques to extract better dense features without fine-tuning. Among these, NACLIP (Hajimiri, Ben Ayed & Dolz, 2025) enhances CLIP's dense prediction capabilities by introducing neighborhood attention, which ensures that image patches focus on nearby regions, and by refining similarity measures to improve spatial consistency.

To the best of our knowledge, this is the first work to address TTA for OVSS models, filling a previously unexplored intersection between these fields.

To better situate this contribution within the broader landscape of general adaptation methods, Table 4.1 summarizes the key distinctions among zero-shot inference, domain generalization (DG) (Noori *et al.*, 2024b, 2025a), few-shot segmentation (FSS) (Wang, Liew, Zou, Zhou & Feng, 2019), test-time training (TTT) (Osowiechi *et al.*, 2023b; Vargas Hakim *et al.*, 2023; Osowiechi *et al.*, 2024a), and our fully unsupervised test-time adaptation (TTA). This comparison highlights that MLMP addresses the most challenging and realistic scenario, adapting models entirely from unlabeled test data without any source access or annotated support samples.

Table 4.1 Comparison of general learning and adaptation paradigms. Here, x^s, y^s denote labeled source samples and x^t, y^t denote target (test) samples and labels. Domain Generalization trains on labeled multi-domain source data to improve robustness on unseen targets, while few-shot and test-time training methods rely on labeled or source data during or after training. Our approach (Fully TTA) adapts solely using unlabeled test samples, requiring neither supervision nor source access

Setting	Source Data	Target Data	Train Loss	Test Loss
Zero-Shot Inference	✗	x^t	✗	✗
Domain Generalization (DG)	x^s, y^s (often multi-domain)	x^t	$\mathcal{L}(x^s, y^s)$ (DG objectives)	✗
Few-Shot Learning (FSS)	✗	x^t, y^t (few)	$\mathcal{L}(x^t, y^t)$ (FSS objectives)	✗
Test-Time Training (TTT)	x^s, y^s	x^t	$\mathcal{L}(x^s, y^s) + \mathcal{L}_{aux}(x^s)$	$\mathcal{L}_{aux}(x^t)$
Fully Test-Time Adaptation (TTA, Ours)	✗	x^t	✗	$\mathcal{L}_{unsup}(x^t)$

4.4 Methodology

We first revisit the contrastive vision–language model (VLM) for open-vocabulary semantic segmentation (OVSS), and then present our Multi-Level Multi-Prompt (MLMP) adaptation strategy.

4.4.1 OVSS with VLMs

Given an input image $\mathbf{X} \in \mathbb{R}^{H\times W\times 3}$ and a set of concepts $C_k \in C$ expressed in natural language, OVSS seeks a semantic mask $\mathbf{y} \in \{1, \ldots, K\}^{H\times W}$ that assigns one concept to every pixel.

Following recent approaches for OVSS (Hajimiri *et al.*, 2025; Wang *et al.*, 2025), we employ a transformer-based VLM to extract visual and text features from the image and concepts in natural language. Specifically, we feed the image $\mathbf{X}$ into the ViT-based vision encoder to extract a visual token matrix $\mathbf{F} = \left[\mathbf{f}_{[\texttt{cls}]}, \mathbf{f}_1, \ldots, \mathbf{f}_N\right]$ with each $\mathbf{f}_i \in \mathbb{R}^D$, where $N = \lfloor H/s \rfloor \times \lfloor W/s \rfloor$ is the number of patches of size $s\times s$ in the image and `[cls]` is the CLS token for classification. We define $\mathbf{Q} = \left[\mathbf{q}_{[\texttt{cls}]}, \mathbf{q}_1, \ldots, \mathbf{q}_N\right]$, with each $\mathbf{q}_i \in \mathbb{R}^{D'}$, the output features before the projection layer: $\mathbf{F} = \text{proj}(\mathbf{Q})$. At the same time, the text encoder is employed to extract text features $\mathbf{t}_k \in \mathbb{R}^D$ for each concept $C_k \in C$. This is achieved by combining C_k with a text prompt template, for instance "`A photo of a [`C_k`]`" or "`An image of a [`C_k`]`" where C_k is an arbitrary text description like "`white horse`".

The standard approach for classifying images with a contrastive VLM such as CLIP (Radford *et al.*, 2021a) computes the cosine between the CLS token features and text embeddings of classes, and assigns the image to the class with highest similarity:

$$\arg\max_k \; sim\big(\mathbf{f}_{\texttt{[cls]}}, \mathbf{t}_k\big), \quad \text{where } sim(\mathbf{x}, \mathbf{y}) = \frac{\mathbf{x} \cdot \mathbf{y}}{\|\mathbf{x}\|\|\mathbf{y}\|}. \tag{4.1}$$

For extending this approach to segmentation, we instead compute the similarity between *patch* embeddings $\mathbf{f}_i$ and text embeddings $\mathbf{t}_k$ and assign a class/concept to each patch.

4.4.2 MLMP: Proposed Method

Figure 4.2 illustrates our full test-time adaptation pipeline, **MLMP**. MLMP integrates three complementary ideas: **uncertainty-aware multi-level fusion**, **image-level entropy minimization** and **multi-prompt adaptation**.

We begin by modifying the entropy minimization objective of TENT (Wang *et al.*, 2020a) from image classification to work with *spatial tokens*. More specifically, for a batch of B images, each containing N tokens, the probability that token i belongs to concept k is

$$p_{ik} \;=\; \frac{\exp\big(sim(\mathbf{f}_i, \mathbf{t}_k)/\tau\big)}{\sum_{k'=1}^{|C|} \exp\big(sim(\mathbf{f}_i, \mathbf{t}_{k'})\big/\tau\big)}. \tag{4.2}$$

where τ is a softmax temperature scaling parameter, $\mathbf{T} = \big[\mathbf{t}_1, \ldots, \mathbf{t}_{|C|}\big]$, and $\text{norm}(\cdot)$ denotes a function normalizing the columns of its input matrix to unit length. Also, let $\mathbf{P}$ denote a matrix containing the probabilities in (4.2), which could be expressed more compactly as follows:

$$\mathbf{P} \;=\; \text{softmax}\big(\text{norm}(\mathbf{F}) \cdot \text{norm}(\mathbf{T})^\top / \tau\big). \tag{4.3}$$

The batch-wise entropy, which is minimized for adaptation, is then defined as follows:

$$\mathcal{H}(\mathbf{P}) \;=\; -\frac{1}{B \cdot N} \sum_{i=1}^{B \cdot N} \sum_{k=1}^{|C|} p_{ik} \log p_{ik}. \tag{4.4}$$

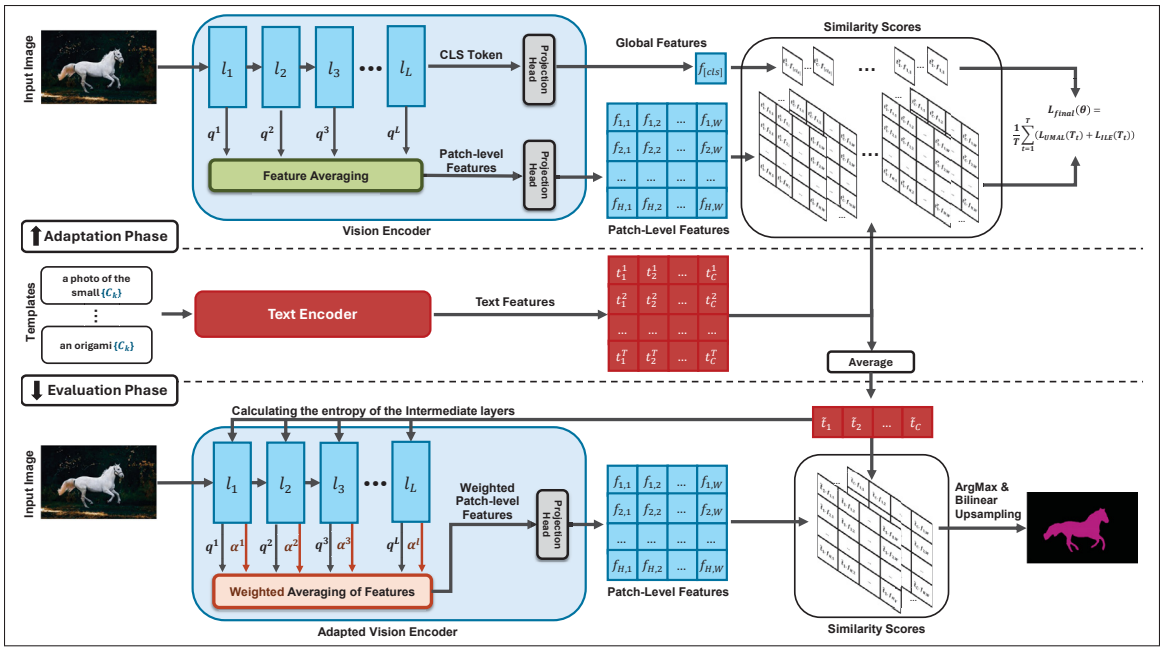

Figure 4.2 Overview of our MLMP method. In the Adaptation Phase, the model is adapted by leveraging multiple prompt templates alongside various intermediate feature layers, as well as the global feature. During the Evaluation Phase, the model computes weights based on the entropy of the intermediate features to perform a weighted averaging. These averaged features, combined with the different templates, are then used to generate the final segmentation map

Following (Osowiechi *et al.*, 2024b; Hakim *et al.*, 2024), we keep the entire text encoder frozen and update only the *LayerNorm* parameters of the vision encoder during adaptation. Freezing the text encoder greatly reduces computational overhead, since text embeddings can be precomputed and reused across all test samples.

Uncertainty-Aware Multi-Level Fusion. In VLM-based classification, the CLS token in the last layer of the visual encoder is typically used to compute the class label probabilities. This approach relies on the idea that the relevant information for classification lies at the end of the ViT and that intermediate layers serve to transform features. In segmentation, however, features from intermediate layers are often used to capture complementary information at different scales (Ronneberger, Fischer & Brox, 2015). This hypothesis is validated in Table 4.2 as well as

Figure 4.3, showing that a higher segmentation mIoU is obtained when combining the features from different intermediate layers.

Inspired by this result, and leveraging the useful property of ViTs that the output of each layer has the same shape, we extend the entropy-based loss described above to use features from *multiple layers*. Denoting as $\mathbf{q}_i^\ell$ the visual features of patch i obtained at layer ℓ, we seek to aggregate the multi-level features into a single vector $\overline{\mathbf{q}}_i$ for segmentation prediction. A simple approach for doing this is to compute $\overline{\mathbf{q}}_i$ by averaging $\mathbf{q}_i^\ell$ across all layers ℓ. However, this approach ignores the relative contribution and confidence of each layer in the final segmentation. To address this limitation, we estimate a confidence weight α^ℓ for each layer ℓ based on its prediction entropy. First, we get visual features $\mathbf{F}_i^\ell = \text{proj}(\mathbf{Q}_i^\ell)$ using the *same* projection head as for the final segmentation. Following the same approach as before, we then compute the batch-wise entropy of layer ℓ as

$$h^\ell \;=\; \mathcal{H}(\mathbf{P}^\ell), \;\; \text{with } \mathbf{P}^\ell = \text{softmax}\big(\text{norm}(\mathbf{F}^\ell) \cdot \text{norm}(\mathbf{T})^\top / \tau\big). \tag{4.5}$$

Finally, the confidence weight of the layer is obtained using a softmax as follows:

$$\alpha^\ell \;=\; \frac{\exp(-\beta \cdot h^\ell)}{\sum_{\ell'=1}^{L} \exp(-\beta \cdot h^{\ell'})}. \tag{4.6}$$

Here, β is a parameter controlling the "sharpness" of the weight distribution. During adaptation, we set $\beta = 0$ to promote a uniform contribution from all layers in the prediction. During inference, we sharpen the distribution with a value of $\beta = 1$, emphasizing the more confident layers in the final prediction.

With these confidence weights, we can now obtain our uncertainty-aware multi-level (UAML) features as

$$\overline{\mathbf{F}} \;=\; \text{proj}(\overline{\mathbf{Q}}), \;\; \text{with } \overline{\mathbf{Q}} = \sum_{\ell=1}^{L} \alpha^\ell \mathbf{Q}^\ell, \tag{4.7}$$

giving the following entropy-based loss to minimize:

$$\mathcal{L}_{\text{UAML}}(\mathbf{T}) \;=\; \mathcal{H}(\overline{\mathbf{P}}), \quad \text{with } \overline{\mathbf{P}} = \text{softmax}\big(\text{norm}(\overline{\mathbf{F}}) \cdot \text{norm}(\mathbf{T})^{\top}/\tau\big). \tag{4.8}$$

Image-Level Entropy Minimization. Since the CLS token is not directly linked to individual patch predictions but rather captures a more global representation of the input, we also include an image-level entropy (ILE) minimization term specifically for this token. As illustrated in Figure 4.1, the CLS token demonstrates increased robustness and reliability. This term, which encourages the model to produce more confident global predictions is expressed as:

$$\mathcal{L}_{\text{ILE}}(\mathbf{T}) \;=\; -\frac{1}{B}\sum_{b=1}^{B}\sum_{k=1}^{|C|} p_{b,k}^{[\texttt{cls}]} \log p_{b,k}^{[\texttt{cls}]}. \tag{4.9}$$

Here, $p_{b,k}^{[\texttt{cls}]}$ denotes the predicted probability for concept C_k obtained using the CLS token in the last layer for the b-th sample.

Multi-Prompt Adaptation. Prior work on TTA for classification (Osowiechi *et al.*, 2024b) has shown the usefulness of leveraging multiple prompt templates in VLM to encode class labels, based on the idea that the templates capture complementary information about these classes. As shown in Figure 4.1a, the sensitivity to the choice of prompt templates is even more pronounced in segmentation tasks, where fine-grained spatial predictions are required. Using multiple templates acts as cross-modal regularization, encouraging more stable and generalized learning signals. While different from image augmentation, it can be seen as a strong, safe, and lightweight text-space augmentation. Rather than averaging the weights adapted from different prompt templates as in (Osowiechi *et al.*, 2024b)—a computationally expensive approach for dense prediction tasks such as segmentation—our method minimizes our proposed UAML and ILE losses across these templates. Let $\mathbf{T}_t$ be the text features obtained using the t-th template.

Our final adaptation loss is defined as:

$$\mathcal{L}_{\text{final}}(\theta) \;=\; \frac{1}{T}\sum_{t=1}^{T}(\mathcal{L}_{\text{UAML}}(\mathbf{T}_t) + \mathcal{L}_{\text{ILE}}(\mathbf{T}_t)). \tag{4.10}$$

Theoretical Justification. Each template t contributes its own adaptation loss, as we optimize their average in Eq. (4.10). By optimizing the adaptation loss of each prompt directly, we force the model to correct for the unique wording and visual cue of each template, rather than 'averaging' these differences in the text embedding space. This loss-level integration treats each template as an independent critic, translating diverse linguistic perspectives into separate gradient signals. Averaging those signals produces an unbiased descent direction whose variance decays as $1/T$, enabling each adaptation step to represent the full prompt ensemble while being stable under noisy shifts.

Proposition 1 (Unbiasedness and Variance Bound). Assume that each per-template gradient $g_t(\theta) = \nabla_\theta\big[\mathcal{L}_{\text{UAML}}(\mathbf{T}_t) + \mathcal{L}_{\text{ILE}}(\mathbf{T}_t)\big]$ has variance bounded by σ^2, then the ensemble gradient, defined by $\nabla_\theta\mathcal{L}_{\text{final}} = \frac{1}{T}\sum_{t=1}^{T} g_t(\theta)$, is unbiased and satisfies the following variance bound:

$$\mathbb{E}\Big[\nabla_\theta\mathcal{L}_{\text{final}}\Big] = \mathbb{E}\big[g_t(\theta)\big]; \quad \text{Var}\Big(\nabla_\theta\mathcal{L}_{\text{final}}\Big) = \frac{1}{T^2}\sum_{t=1}^{T}\text{Var}\big(g_t(\theta)\big) \le \frac{\sigma^2}{T} \tag{4.11}$$

Proof. A proof of Prop. 1 is provided in the Appendix.

The $1/T$ reduction in gradient variance, as stated in Prop. 1, explains the improved stability we observe. Table 4.4 confirms that this loss-level ensemble outperforms alternative fusion strategies.

4.5 Experimental Settings

Experimental Setup. Following prior work on TTA in classification (Hakim *et al.*, 2024; Osowiechi *et al.*, 2024b), we restrict updates to the normalization layers within the vision

encoder. The adaptation process is carried out over 10 iterations using the Adam optimizer with a constant learning rate of 10^{-3} across all datasets. We use a batch size of 2 images during adaptation across all datasets. For each new batch, the model undergoes a reset, restoring it to its initial weights before adaptation is applied.

Datasets. In traditional TTA for segmentation, two datasets are commonly employed to simulate domain shifts—one for model training and another for adaptation during inference (e.g., GTAV and Cityscapes)—as both must share the same semantic label space. In our study, as this is the first exploration of TTA for VLMs in segmentation tasks and given that VLMs are pre-trained, we draw inspiration from ImageNet-C (Hendrycks & Dietterich, 2019a) to introduce 15 synthetic corruptions on segmentation datasets. Our experiments are conducted on Pascal VOC 20 (v20), Pascal VOC 21 (v21) (Everingham, Gool, Williams, Winn & Zisserman, 2010a), Pascal Context 59 (P59), Pascal Context 60 (P60) (Mottaghi *et al.*, 2014), and Cityscapes (Cordts *et al.*, 2016), incorporating both original version (clean) and the synthetic 15 corruptions (denoted with a "-C" suffix). For COCO-Stuff (Caesar, Uijlings & Ferrari, 2016) and COCO-Object (Lin *et al.*, 2014), we use only the original versions. To further evaluate robustness under real and rendered distributional shifts, we additionally include ACDC (Sakaridis, Dai & Van Gool, 2021)—capturing real-world adverse conditions such as fog, night, rain, and snow—and GTA-V (Richter, Vineet, Roth & Koltun, 2016), which provides photorealistic, game-rendered urban scenes. This extended setup results in **87 distinct test scenarios** encompassing synthetic, real, and rendered shifts, enabling a comprehensive evaluation of MLMP across diverse conditions.

Benchmarking. While MLMP is compatible with any OVSS framework, we incorporate NACLIP (Hajimiri *et al.*, 2025) with ViT-L/14 as our baseline OVSS model, which leverages neighborhood attention to enhance spatial consistency in a training-free manner. The compared methods include TENT (Wang *et al.*, 2020a), which serves as a baseline and minimizes entropy during adaptation; CLIPArTT (Hakim *et al.*, 2024), which employs pseudo-labels generated via conformal learning; WATT (Osowiechi *et al.*, 2024b), which averages learnable parameters across multiple parallel branches; and TPT (Shu *et al.*, 2022), which performs prompt tuning to adapt VLMs at test time. For a fair comparison, we modified all methods for the segmentation

setting by processing all spatial tokens extracted from the VLM, rather than relying solely on the CLS token.

4.6 Results

4.6.1 Ablation studies

Effect of Intermediate Layers. To analyze the impact of layer selection in our uncertainty-aware multi-level adaptation strategy, we use different ranges of intermediate layers. As shown in Table 4.2, performance varies notably with the fusion range. While using only the final layers yields moderate improvements, incorporating the last 75% of the layers consistently achieves the best performance across both clean and corrupted inputs. This highlights that multi-level fusion is a key driver of adaptation performance: earlier layers, although less semantically abstract, contribute valuable low-level features—such as texture and edge cues—that enhance robustness to distribution shifts. In this ablation, we isolate the effect of multi-level fusion by applying only the first term in Eq. 4.10, using a single prompt template and omitting the L_{ILE} term.

Effect of Uncertainty-Aware Layer Fusion. We investigate strategies for aggregating multi-level features by comparing uniform averaging ($\beta = 0$) and uncertainty-aware fusion ($\beta = 1$) during evaluation, using the same 75% layer range identified in the previous ablation. As shown in Table 4.3, incorporating entropy-based weighting improves performance by 4.29% on V20 and 3.31% on V20-C. This highlights the importance of leveraging layer-wise confidence when aggregating features.

Visualization of Layer-Wise Confidence Weights. We visualize the mean and standard deviation of the learned layer weights to better understand the behavior of our uncertainty-aware fusion strategy. As shown in Figure 4.3, deeper layers tend to receive higher confidence, though earlier layers also contribute, especially under corrupted conditions. This variation is most pronounced in the Cityscapes dataset, where the distribution fluctuates more across layers and corruption types. In contrast, the COCO-Stuff dataset shows a flatter distribution, where the

Table 4.2 mIoU performance when using different layer ranges in the proposed multi-level adaptation

ViT-L/14 Layer Range	L_{24} (last)	$L_{23–24}$ (last two)	$L_{22–24}$ (last three)	$L_{19–24}$ (last 25%)	$L_{13–24}$ (last 50%)	$L_{7–24}$ (last 75%)	$L_{1–24}$ (all layers)
V20 (Original)	77.00± 0.04	77.65± 0.02	77.66± 0.09	80.61± 0.05	80.50± 0.03	**81.67± 0.04**	78.79± 0.02
Gaussian Noise	63.02± 0.06	64.41± 0.05	65.39± 0.13	66.88± 0.18	66.88± 0.02	**67.82± 0.01**	63.06± 0.09
Defocus Blur	72.06± 0.12	72.93± 0.19	72.84± 0.02	76.10± 0.16	76.37± 0.05	**78.78± 0.02**	77.56± 0.09
Snow	71.04± 0.05	72.09± 0.04	72.56± 0.02	74.47± 0.12	74.41± 0.01	**76.39± 0.02**	73.72± 0.07
JPEG Compression	71.84± 0.15	73.88± 0.11	74.40± 0.07	76.96± 0.02	77.67± 0.03	**78.73± 0.08**	75.87± 0.19
V20-C Average	69.33	70.33	70.78	72.89	73.45	**74.90**	72.02

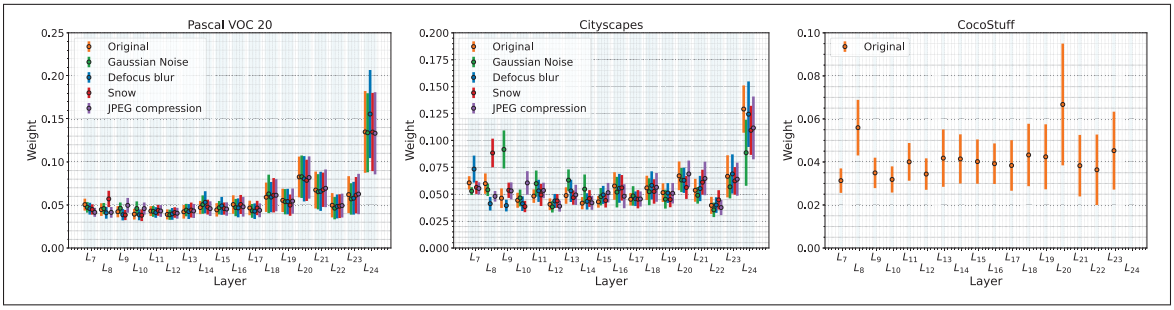

Figure 4.3 Mean and standard deviation of layer-wise confidence weights of MLMP across datasets. The fusion mechanism adaptively emphasizes more reliable layers based on input conditions

Table 4.3 mIoU comparison of MLMP components, showing individual and combined contributions

Multi-Level Fusion	✗	✓	✓	✗	✗	✓	✓	✓	✓	✗	✓	✓
Multi-Prompt Loss	✗	✗	✗	✓	✗	✓	✓	✗	✗	✓	✓	✓
Image-Level Entropy	✗	✗	✗	✗	✓	✗	✗	✓	✓	✓	✓	✓
Uncertainty-Aware Weighting	✗	✗	✓	✗	✗	✗	✓	✗	✓	✗	✗	✓
V20 (Original)	77.00	77.38	81.67	79.70	78.74	78.97	83.00	77.69	82.70	81.15	79.13	**83.76**
Gaussian Noise	63.02	65.42	67.82	66.75	65.66	65.96	69.13	66.17	69.00	69.62	67.35	**71.13**
Defocus Blur	72.06	76.65	78.78	74.31	75.00	76.46	78.78	77.29	79.78	77.14	77.79	**80.36**
Snow	71.04	72.64	76.39	74.66	74.16	73.25	77.31	74.05	78.50	77.20	74.94	**79.53**
JPEG Compression	71.84	74.38	78.73	75.56	74.77	76.77	80.81	74.61	79.79	77.98	77.94	**82.06**
V20-C Average	69.33	71.59	74.90	72.58	71.99	72.66	75.97	72.41	76.18	75.08	73.89	**77.58**

final layer is not consistently the most influential. These results underscore the core strength of our fusion mechanism: its ability to adaptively reweight layers based on input conditions,

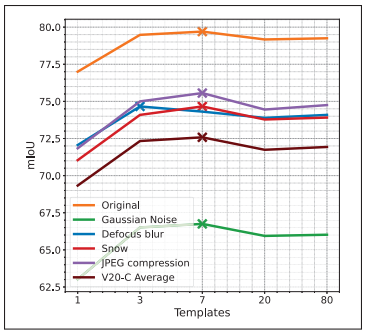

Figure 4.4 mIoU performance of our method for different numbers of templates

Table 4.4 mIoU performance for prompt-integration strategies (Text, Params, Loss) on clean and corrupted data

Dataset: V20	Text	Params	Loss
Original	78.91± 0.07	74.46± 0.21	**79.70± 0.06**
Gaussian Noise	66.27± 0.00	62.83± 0.04	**66.75± 0.01**
Defocus Blur	74.05± 0.10	70.28± 0.16	**74.31± 0.09**
Snow	73.78± 0.02	70.10± 0.30	**74.66± 0.01**
JPEG Compression	74.98± 0.05	70.55± 0.11	**75.56± 0.02**
V20-C Average	71.92	68.44	**72.58**

assigning greater importance to those that remain more reliable under distribution shifts and corruption. Please refer to the Appendix for additional results on other datasets.

Effect of Global Image-Level Adaptation Term. The image-level entropy term, L_{ILE} complements our patch-level adaptation by encouraging consistent global predictions through the CLS token. While the multi-level loss targets fine-grained spatial predictions, the ILE term introduces global context that helps stabilize adaptation. As shown in Table 4.3, when added in isolation, L_{ILE} improves performance by 1.74% and 2.66% on V20 and V20-C, respectively, demonstrating the benefit of incorporating global context under distribution shift.

Effect of Number of Prompt Templates. To isolate this effect, we evaluate performance using different numbers of prompt templates while disabling both the multi-level fusion and the image-level entropy (ILE) term in Eq. 4.10. As shown in Figure 4.4, increasing the number of templates improves performance up to 7, after which the gains begin to saturate or slightly decline. This trend holds across both clean and corrupted settings, indicating that a moderate number of diverse prompt templates is sufficient for MLMP. We therefore use 7 templates by default in our main experiments. Please refer to the Appendix for template details.

Where to Integrate Multi-Prompt Information. Here, we empirically compare strategies for integrating multi-prompt information into the adaptation process. Specifically, we evaluate:

(1) text-level averaging, where prompt embeddings are averaged before computing logits—a technique commonly used in zero-shot learning (Radford *et al.*, 2021a); (2) a learnable parameter averaging baseline (Params) inspired by WATT (Osowiechi *et al.*, 2024b); and (3) our proposed method (Loss), which incorporates all prompt templates directly into the adaptation loss (Eq. 4.10). To isolate the effect of prompt integration, this analysis excludes other components such as multi-level fusion and the image-level entropy (ILE) term. As shown in Table 4.4, our loss-level formulation consistently outperforms the alternatives across both clean and corrupted settings.

Full Component Analysis. Table 4.3 presents an extensive ablation evaluating the contribution of each component in our MLMP strategy. Each proposed element yields consistent gains when added independently, but it is their combination that delivers the highest overall performance across both clean and corrupted settings, highlighting their strong complementary effects within a unified framework. Additionally, we provide further ablations in the Appendix, including alternative OVSS backbones, different VLMs, ViT architecture variants, computational complexity analysis, and MLMP segmentation map visualizations, as well as discussions on the effect of longer prompts, effect of adaptation iterations, and episodic vs. online adaptation.

4.6.2 Final Comparison with Alternative Adaptation Methods

Performance on Clean Data (No Distributional Shift). We begin by evaluating MLMP on clean test data (original), where no distributional shift is present. This setting is crucial, as TTA methods must avoid degrading performance when adaptation is unnecessary. As shown in Table 4.5, MLMP achieves strong mIoU gains of +7.85, +5.66, +3.72, and +3.04 on V20, V21, P59, and P60, and +5.04/+2.91 on challenging datasets COCOObject and COCOStuff. In contrast, most alternative adaptation methods fail to improve performance in this setting. These gains highlight the robustness and generalization of our method, even when no explicit domain shift is present.

Table 4.5 mIoU comparison of MLMP and baselines across several datasets. CLIPArTT could not be run for a few cases owing to GPU memory shortages. Full per-dataset results are in the Appendix

OVSS Backbone: NACLIP		Adaptation Method					
Dataset		No Adapt.	TENT	TPT	WATT	CLIPArTT	MLMP
V20 (Original)		75.91	77.00± 0.04	75.93± 0.01	57.73± 0.06	72.77± 0.14	**83.76± 0.00**
V20-C	Gaussian Noise	62.89	63.02± 0.06	62.98± 0.01	36.44± 0.04	53.36± 0.25	**71.13± 0.09**
	Shot noise	66.26	65.88± 0.06	66.33± 0.02	40.95± 0.05	58.15± 0.28	**75.02± 0.03**
	Impulse Noise	63.16	64.17± 0.04	63.12± 0.01	34.90± 0.06	54.83± 0.03	**71.34± 0.11**
	Defocus blur	72.59	72.06± 0.12	72.55± 0.02	52.43± 0.03	65.39± 0.45	**80.36± 0.06**
	Glass blur	71.44	70.74± 0.07	71.40± 0.01	49.96± 0.05	64.62± 0.13	**78.84± 0.05**
	Motion blur	73.10	73.50± 0.10	73.16± 0.02	53.35± 0.06	67.48± 0.17	**81.41± 0.05**
	Zoom blur	59.03	61.36± 0.07	59.00± 0.01	41.39± 0.08	52.37± 0.12	**69.41± 0.12**
	Snow	71.49	71.04± 0.05	71.44± 0.01	51.18± 0.06	66.97± 0.02	**79.53± 0.05**
	Frost	65.38	67.01± 0.02	65.46± 0.01	45.75± 0.05	60.48± 0.08	**73.20± 0.07**
	Fog	70.69	70.54± 0.07	70.70± 0.01	52.96± 0.04	67.85± 0.10	**79.81± 0.06**
	Brightness	74.95	75.61± 0.02	74.95± 0.01	55.82± 0.05	71.52± 0.14	**83.51± 0.01**
	Contrast	71.51	70.51± 0.04	71.49± 0.02	50.74± 0.06	66.01± 0.06	**79.06± 0.16**
	Elastic transform	62.86	65.78± 0.05	62.95± 0.01	45.45± 0.04	60.41± 0.10	**74.03± 0.01**
	Pixelate	77.28	76.95± 0.12	77.31± 0.01	59.76± 0.05	73.14± 0.17	**84.97± 0.04**
	JPEG compression	72.59	71.84± 0.15	72.56± 0.01	53.44± 0.05	68.21± 0.07	**82.06± 0.01**
	Average	69.01	69.33	69.03	48.30	63.39	**77.58**
V21 (Original)		45.12	45.65± 0.02	45.17± 0.01	28.58± 0.05	39.50± 0.04	**50.78± 0.02**
V21-C Average		40.75	40.95	40.77	24.12	34.16	**46.25**
P59 (Original)		28.23	28.73± 0.02	28.26± 0.01	16.55± 0.04	24.60± 0.03	**31.95± 0.02**
P59-C Average		23.88	23.88	23.88	13.37	19.72	**27.03**
P60 (Original)		24.95	25.29± 0.01	24.98± 0.01	14.77± 0.03	21.88± 0.03	**27.99± 0.03**
P60-C Average		21.39	21.25	21.49	12.08	17.79	**24.07**
CityScapes (Original)		29.49	30.54± 0.04	29.57± 0.01	20.77± 0.06	–	**33.35± 0.03**
CityScapes-C Average		21.63	21.64	21.60	13.45	–	**23.02**
COCOObject (Original)		23.80	24.88± 0.01	23.84± 0.01	14.14± 0.06	21.34± 0.03	**28.84± 0.01**
COCOStuff (Original)		18.34	18.76± 0.01	18.35± 0.01	9.49± 0.02	15.48± 0.01	**21.25± 0.01**

Performance Under Distributional Shift. Under distributional shifts, the advantages of MLMP become even more apparent. As shown in Table 4.5, MLMP consistently outperforms both the zero-shot baseline and existing adaptation methods, achieving mIoU gains of +8.60, +5.50, +3.15, and +2.68 on V20-C, V21-C, P59-C, and P60-C, respectively. Beyond standard corruptions, we further evaluate on the Cityscapes dataset, which presents natural domain shifts such as environmental variation, weather conditions, and resolution differences. Despite its challenging nature and low zero-shot performance, MLMP improves mIoU by +3.86, demonstrating its real-world adaptability. To push this further, we apply 15 corruption types to create Cityscapes-C,

Table 4.6 mIoU comparison on realistic (ACDC) and rendered (GTA-V) domain shifts. Full ACDC results (including reference/clean views) are provided in the Appendix

OVSS: NACLIP		Adaptation Method		
Dataset		No Adapt.	TENT	MLMP
ACDC	Fog	23.88	26.89± 0.04	**33.33± 0.04**
	Night	22.12	24.17± 0.00	**24.76± 0.03**
	Rain	23.86	26.84± 0.04	**32.44± 0.04**
	Snow	23.54	27.25± 0.05	**30.59± 0.03**
	Average	23.35	26.29	**30.28**
GTA-V		25.09	26.62± 0.01	**28.84± 0.02**

where MLMP still yields a +1.39 gain. While TENT provides modest improvements, most other adaptation methods—including ClipArTT, TPT, and WATT—either fail to improve or degrade performance. These results highlight that naive strategies like pseudo-labeling, prompt tuning, or weight averaging are insufficient for open-vocabulary segmentation, and emphasize the need for segmentation-specific adaptation techniques such as MLMP. Detailed results can be found in the Appendix.

While Cityscapes already embodies natural distributional shifts caused by variations in lighting, camera viewpoint, and urban layouts, we further assess MLMP on datasets explicitly designed to capture targeted domain shifts. Specifically, we evaluate on ACDC (Sakaridis *et al.*, 2021), which contains real-world adverse conditions (Fog, Night, Rain, and Snow), and GTA-V (Richter *et al.*, 2016), a photorealistic, game-rendered dataset that introduces a distinct synthetic distribution shift relative to real imagery seen during CLIP pre-training. As summarized in Table 4.6, MLMP achieves consistent improvements over both the non-adapted baseline and TENT across all ACDC domains, yielding on average +6 mIoU gains under real-world conditions. Similarly, on the GTA-V dataset, MLMP improves by +3.8 mIoU over the baseline, further confirming its robustness across both realistic and rendered distribution shifts.

4.7 Conclusion

We presented MLMP, a plug-and-play test-time adaptation framework for open-vocabulary semantic segmentation that can be integrated with any OVSS method. By combining uncertainty-aware fusion of intermediate ViT features with a novel loss-level integration of multiple prompt templates, MLMP consistently enhances performance across both clean and shifted domains—including common corruptions and natural distributional shifts. Our comprehensive OVSS-TTA benchmark—covering nine datasets and 87 distinct test scenarios—demonstrates MLMP's broad applicability and establishes a rigorous evaluation protocol for future work in adaptive, language-aware segmentation. While MLMP demonstrates strong, consistent gains, there remain opportunities to further refine its components. In particular, our current layer-weighting mechanism relies on entropy estimates from a shared projection head, which may not fully reflect each layer's unique characteristics. Future work could investigate more flexible architectures—such as lightweight adapters or dedicated projection modules per layer—to more accurately assess and fuse intermediate features.

CHAPTER 5

HISTOPATH-C: TOWARDS REALISTIC DOMAIN SHIFTS FOR HISTOPATHOLOGY VISION-LANGUAGE ADAPTATION

Mehrdad Noori[a] , Gustavo A.Vargas Hakim[b] , David Osowiechi[a] , Fereshteh Shakeri[b] , Ali Bahri[a] , Moslem Yazdanpanah[a] , Sahar Dastani[a] , Ismail Ben Ayed[b] , Christian Desrosiers[a]

[a] Department of Information Technologies Engineering, École de Technologie Supérieure
[b] Department of Systems Engineering, École de Technologie Supérieure
1100 Notre-Dame West, Montreal, Quebec, Canada H3C 1K3
Paper submitted for publication in *IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2026

5.1 Abstract

Medical Vision-language models (VLMs) have shown remarkable performances in various medical imaging domains such as histopathology by leveraging pre-trained, contrastive models that exploit visual and textual information. However, histopathology images may exhibit severe domain shifts, such as staining, contamination, blurring, and noise, which may severely degrade the VLM's downstream performance. In this work, we introduce Histopath-C, a new benchmark with realistic synthetic corruptions designed to mimic real-world distribution shifts observed in digital histopathology. Our framework dynamically applies corruptions to any available dataset and evaluates Test-Time Adaptation (TTA) mechanisms on the fly. We then propose LATTE, a transductive, low-rank adaptation strategy that exploits multiple text templates, mitigating the sensitivity of histopathology VLMs to diverse text inputs. Our approach outperforms state-of-the-art TTA methods originally designed for natural images across a breadth of histopathology datasets, demonstrating the effectiveness of our proposed design for robust adaptation in histopathology images. Our code repository is available at https://github.com/Mehrdad-Noori/Histopath-C.

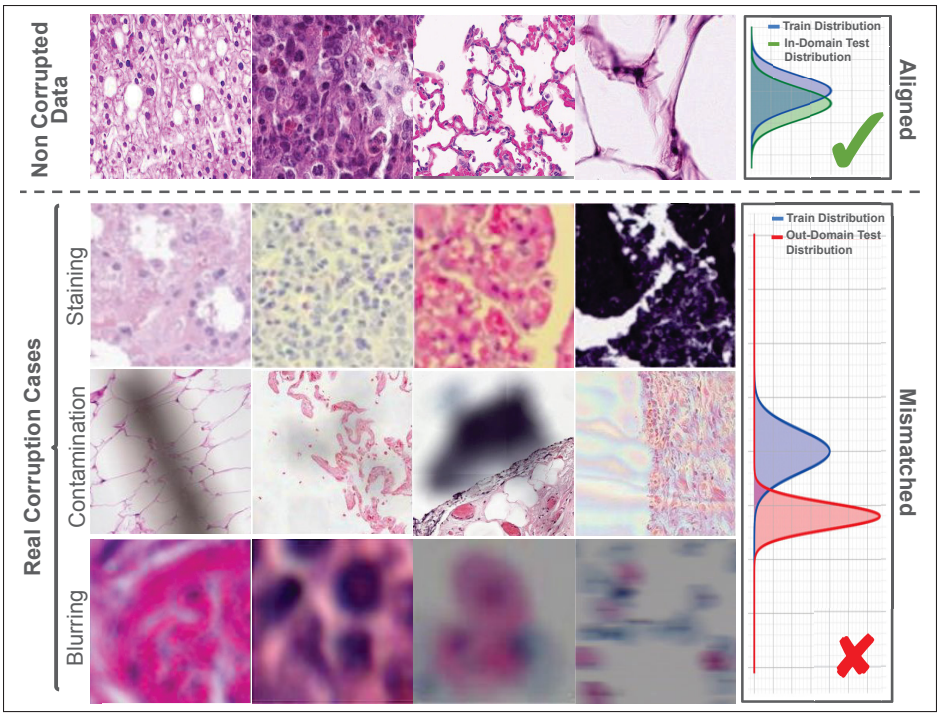

Figure 5.1 Illustrative examples of real-world corruption artifacts in histopathology slides, as documented in prior literature. The top row shows representative clean (non-corrupted) images with train–test alignment. The bottom rows depict real instances of staining (Ochi *et al.*, 2024b; Hoque *et al.*, 2024; Xu *et al.*, 2025), contamination (Jurgas *et al.*, 2024; Satapute *et al.*, 2020), and blurring artifacts (Jiang *et al.*, 2020; Senaras *et al.*, 2018; Wu *et al.*, 2015b), each introducing domain shifts that can significantly degrade model performance. These examples motivate the need for a test-time adaptation benchmark, such as Histopath-C, that simulates realistic perturbations and enables evaluation of adaptation methods in histopathology

5.2 Introduction

Vision-Language Models (VLMs) (Radford *et al.*, 2021a; Jia *et al.*, 2021) have recently emerged as a powerful paradigm for solving a wide range of downstream visual tasks using a single large-scale pretraining objective that aligns visual and textual representations in a shared embedding space. While originally developed for natural image domains, VLMs have shown remarkable versatility and have been rapidly extended to medical imaging applications(Ikezogwo *et al.*, 2024; Wang, Wu, Agarwal & Sun, 2022c; Silva-Rodríguez, Chakor, Kobbi, Dolz & Ben Ayed, 2025). By leveraging large-scale paired image–text datasets—often sourced from clinical

reports or curated descriptions—these models are able to learn transferable representations that outperform traditional convolutional or transformer-based architectures trained from scratch or with supervised objectives. Their strong zero-shot and few-shot generalization capabilities make them particularly appealing for medical domains, where labeled data is scarce and task diversity is high.

Despite their promising capabilities, medical VLMs inherit a critical limitation observed in standard computer vision models trained on datasets such as ImageNet (Deng *et al.*, 2009b): a pronounced sensitivity to domain shifts. In real-world clinical settings, variations in acquisition protocols, scanners, staining procedures, or patient populations are very common (Satapute *et al.*, 2020) and can lead to substantial distributional discrepancies between training and deployment data, which degrade the performance and reliability of deep learning models in clinical practice (Hoque *et al.*, 2024; Aubreville *et al.*, 2021; Kohlberger *et al.*, 2019a). To mitigate domain shift, Test-Time Adaptation (TTA) (Wang *et al.*, 2021a) has emerged as a compelling paradigm, enabling models to adapt at inference time by leveraging unlabeled target samples. While recent work has begun to explore TTA for VLMs (Shu *et al.*, 2022; Hakim *et al.*, 2024), these efforts have been largely confined to natural image domains and have yet to be formally extended or systematically evaluated in the context of medical imaging.

To address this gap, we introduce a novel benchmark that simulates realistic domain shifts commonly encountered in histopathology. These shifts include staining, contamination, blurring, noise, illumination variations,. All corruptions can be applied on the fly to existing datasets, enabling systematic evaluation of model robustness under controlled distributional perturbations. Representative examples are shown in Figure 5.1. While recent benchmarks such as Histo-VL (Majzoub *et al.*, 2025) aggregate heterogeneous datasets to study broad generalization and prompt/adversarial sensitivity, our focus is factor-isolated robustness. Histopath-C applies controlled, graded corruptions (stain, contamination, blur, illumination, noise) to the same images, allowing us to pinpoint which real-world artifacts most degrade each model and to stress-test TTA methods head-to-head under identical conditions.

We show that multiple pathology VLMs such as Quilt (Ikezogwo *et al.*, 2024), PathGen (Sun *et al.*, 2024), and CONCH (Lu *et al.*, 2024) degrade substantially under these corruptions, and that existing TTA methods exhibit inconsistent performance across datasets and corruption types. This brittleness stems in part from the reliance of medical VLMs, on text templates derived from clinical reports, which can introduce high variability depending on the choice of prompt and input image. To overcome these limitations, we propose a novel TTA framework that leverages multiple text templates to stabilize model predictions and enhance robustness under distribution shift. By aggregating information across diverse prompts, our method improves consistency and reduces sensitivity to template selection, a key weakness in existing VLM-based pipelines. Our main contributions are summarized as follows:

- A novel and challenging benchmark comprising 10 diverse domain shifts that closely resemble real-world corruptions encountered in histopathology imaging. We devise a framework for applying these corruptions *on the fly* to any existing dataset.
- An evaluation of recent TTA methods for VLMs, demonstrating that the proposed benchmark introduces significant challenges that severely degrade performance, which existing baselines fail to address.
- We introduce Low-rank Adaptation with Transductive Template Ensembling (LATTE), a simple yet effective adaptation method for VLMs. This technique addresses text template ambiguity in histopathology imaging by leveraging loss-level aggregation across templates, transductive pseudolabeling, and low-rank adaptation.
- In addition to improving robustness under domain shift, LATTE consistently enhances zero-shot performance even in the absence of corruptions—demonstrating its effectiveness as a general-purpose enhancement for VLMs in medical imaging.

5.3 Related Work

Medical Vision Language Models. Building on the success of CLIP (Radford *et al.*, 2021a), significant efforts have been made to develop vision-language models (VLMs) for medical applications. These models have proven valuable in diverse fields, including retinal analysis

(Silva-Rodríguez *et al.*, 2025), radiology (Wang *et al.*, 2022c), and histopathology (Ikezogwo *et al.*, 2024), by leveraging both visual and textual modalities. Their success largely stems from the availability of extensive visual data paired with detailed text descriptions (Zhang *et al.*), along with powerful contrastive learning techniques that enhance the robustness of visual encoders. Despite their strong performance in downstream tasks, research has focused on adapting VLMs to new challenges with minimal supervision, such as few-shot learning (Shakeri *et al.*, 2024; Hussein, Shamshad, Naseer & Nandakumar, 2024). Given the inherent scarcity of medical images, we argue that a zero-shot adaptation approach is more suitable, aligning with TTA, where the model continuously learns from each new sample in an unsupervised manner.

Test-Time Adaptation. In traditional TTA, a deep model is adapted *on the fly* to new target domains that differ from the source training domain. Key aspects of this setting include the absence of labels during adaptation, the inaccessibility of the source dataset, and adaptation to data streams (*i.e.,* , batches) without access to the entire target dataset. In the spirit of exploiting the connection between the model's pre-training loss (*e.g.*, crossentropy) and an unsupervised adaptation loss, TENT (Wang *et al.*, 2021a) introduces conditional entropy minimization, and targets normalization layers' affine parameters. An important body of research has built on this methodology, offering increasing performance scores (Niu *et al.*, 2022; Goyal, Sun, Raghunanthan & Kolter, 2022; Niu *et al.*, 2023; Gong, Kim, Lee, Chottananurak & Lee, 2024; Yu, Sheng, He & Liang, 2024). Alternative directions have been also investigated, including parameter-free Laplacian optimization (Boudiaf *et al.*, 2022), contrastive learning (Wang, Fink, Van Gool & Dai, 2022b; Chen, Wang, Darrell & Ebrahimi, 2022), and pseudolabeling (Liang, Hu & Feng, 2020; Jang, Chung & Chung, 2022). Test-Time Training (TTT) methods, such as (Sun *et al.*, 2020; Osowiechi *et al.*, 2023b, 2024a; Vargas Hakim *et al.*, 2023; Colussi, Mascetti, Dolz & Desrosiers, 2024), train an unsupervised sub-branch during pretraining for refinement during testing.

TTA for VLMs has broader implications, as it enables model adaptation to any new downstream task, regardless of its domain. TPT (Shu *et al.*, 2022) extended the framework of TENT by using entropy minimization on confident image augmentations' predictions, and using text prompt

finetuning. However, this approach has inherent limitations in VLMs, as entropy minimization can lead to model degradation by adapting based on wrong and overconfident predictions. Additionally, the interactions between visual and textual modalities is not directly exploited. In response, recent methods have explored image-text pseudolabeling (Hakim *et al.*, 2024) and weight ensembling (Osowiechi *et al.*, 2024b) in hopes of leveraging both modalities. However, as demonstrated in Section 5.6, the improvements from these methods remain limited and inconsistent across the various domain shifts in our benchmark. Specifically, their working mechanisms appear to be suboptimal in the histopathology adaptation scenario.

Histopathology Benchmarks. Several benchmarks have been proposed to assess the performance of foundation models in computational pathology. Histo-VL (Majzoub *et al.*, 2025) aggregates heterogeneous datasets to study broad generalization and sensitivity to prompts and adversarial inputs in histopathology VLMs. More recently, THUNDER (Marza *et al.*, 2025) introduced a comprehensive tile-level benchmark evaluating 23 foundation models across 16 diverse datasets, spanning multiple cancer types and magnifications. THUNDER examines downstream task performance, feature space geometry, and robustness—the latter assessed by measuring the displacement of model embeddings under input transforms rather than classification accuracy degradation under realistic corruptions. While THUNDER provides a broad comparison of feature extractors under clean conditions, it does not model histopathology-specific acquisition artifacts such as staining variability, contamination, or blur. Histopath-C is therefore complementary: rather than ranking models under ideal conditions, it isolates and quantifies the impact of corruptions that reflect real clinical failure modes, and provides a rigorous testbed for TTA methods designed to recover from such degradations.

5.4 Benchmark

We propose Histopath-C, a new TTA benchmark for histopathology imaging. Our setting is based on the application of ten different corruptions that realistically simulate real-world perturbations. Different from natural images (Hendrycks & Dietterich, 2019c), our perturbations are specifically designed to reflect failure modes relevant to histopathology imaging. Moreover,

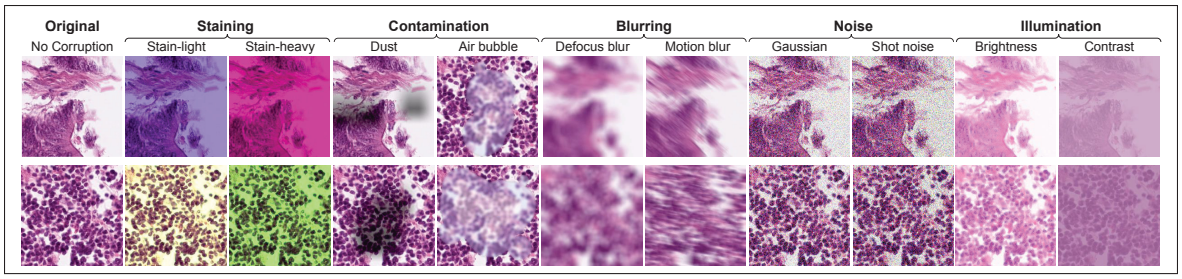

Figure 5.2 Representative examples of the ten corruption types introduced in the Histopath-C benchmark, spanning five categories: Staining, Contamination, Blurring, Noise, and Illumination. These synthetic corruptions are designed to mimic real-world perturbations in histopathology and can be dynamically applied to any dataset for robust evaluation

Histopath-C can be applied to any dataset to evaluate the robustness of foundation models and adaptation algorithms. We categorize the chosen corruptions as five groups —*Staining*, *Contamination* (dust & air bubbles), *Blurring*, *Noise*, and *Illumination*. Representative examples of several important real-world artifacts, drawn from prior pathology studies, are shown in Figure 5.1. We next outline how each corruption is simulated to reflect real-world variation.

5.4.1 Staining

Variations in staining procedures are one of the most pervasive and impactful sources of domain shift in histopathology imaging (Ochi *et al.*, 2024b; Hoque *et al.*, 2024; Xu *et al.*, 2025). These variations can arise from differences in reagent concentration, staining protocols, scanner calibration, or tissue fixation practices across laboratories and institutions. The resulting shifts in color composition and intensity patterns (as shown in Figure 5.1 can drastically alter the appearance of tissue structures and reduce model generalization, especially for deep learning models trained on a narrow staining distribution.

To simulate these realistic distributional shifts, we adopt a biologically grounded perturbation model based on color deconvolution in the Hematoxylin-Eosin-DAB (HED) color space. Inspired by the motivation behind stain-invariant training (Tellez *et al.*, 2018), we introduce this

perturbation in the context of test-time evaluation, using controlled shifts in the HED space to model realistic staining variability across clinical settings.

First, each image $\mathbf{x} \in \mathbb{R}^{H\times W\times 3}$ is projected to the Hematoxylin–Eosin–DAB (HED) optical-density space via a fixed color-deconvolution matrix $\mathbf{M}_{\text{HED}}$ (Ruifrok, Johnston *et al.*, 2001):

$$\mathbf{s} = \text{vec}\big(\text{rgb2hed}(\mathbf{x})\big) = \text{vec}\big(-\log(\mathbf{x}\,\mathbf{M}_{\text{HED}})\big) \in \mathbb{R}^{3HW} \tag{5.1}$$

where $\text{vec}(\cdot)$ flattens the tensor to a column vector.

Each stain channel $c \in \{\text{H}, \text{E}, \text{D}\}$ is then perturbed independently by a multiplicative and an additive jitter:

$$\mathbf{s}'_c = \alpha_c\,\mathbf{s}_c + \beta_c, \quad \alpha_c \sim \mathcal{U}(1-\theta,\, 1+\theta),\ \beta_c \sim \mathcal{U}(-\theta,\, \theta) \tag{5.2}$$

where θ controls corruption severity. We set $\theta = 0.05$ for **Stain-Light** and $\theta = 0.2$ for **Stain-Heavy**. Finally, the perturbed optical densities $\mathbf{s}'$ are reshaped to $\mathbb{R}^{H\times W\times 3}$, converted back to RGB with $\text{hed2rgb}(\cdot)$, and linearly rescaled to $[0, 255]$. A fixed seed is used once to sample $(\boldsymbol{\alpha}, \boldsymbol{\beta})$, ensuring deterministic corruptions per image while preventing the model from exploiting inter-sample consistency.

The intensity scaling controlled by α_c simulates real-world variations such as under- or over-staining, or differences in scanner amplification. Similarly, the additive shifts introduced by β_c reflect changes in dye absorption caused by factors like pH, reagent quality, or fixation protocols. Because each of the three stain channels (Hematoxylin, Eosin, and DAB) is perturbed independently, the resulting color changes capture complex and realistic variations in staining that generic color jitter techniques (e.g., HSV or PCA jitter) fail to model accurately. In Section 5.6.1, we show that even the light setting ($\theta = 0.05$) leads to a substantial drop in the zero-shot performance of a state-of-the-art histopathology VLM, pre-trained on a vast corpus of image–text pairs, underscoring the practical need for robust stain adaptation.

5.4.2 Contamination

Real-world histopathology slides are sometimes affected by physical contaminants introduced during slide preparation, scanning, or storage. As shown in Figure 5.1, artifacts such as dust particles, air bubbles, and tissue folds can obscure diagnostically relevant structures and introduce variability that challenges both pathologists and automated systems (Jurgas *et al.*, 2024; Satapute *et al.*, 2020; Taqi, Sami, Sami & Zaki, 2018). These issues are especially critical in digital pathology, where even minor occlusions may compromise tissue interpretation. To simulate such scenarios in a controlled and reproducible way, we design two of the most commonly encountered contamination artifacts: **Dust** and **Air Bubble**.

Dust. We model particulate contamination through semi-opaque smudges and fine linear artifacts, commonly seen due to static particles or slide friction. Each dust instance is simulated as a blurred, darkened region using one of two shape types: (1) large rectangular smudges with vertical gradient opacity to mimic streaking or residue, and (2) narrow opaque lines to mimic hairline debris or scratches. A random number of such artifacts (sampled from a uniform range) are placed per image, with Gaussian blur applied to the alpha mask to enhance realism. The final dust mask $M \in [0,1]^{H\times W}$ is applied multiplicatively as an occlusion:

$$\mathbf{x}' = \mathbf{x} \odot (1 - M) \tag{5.3}$$

where $\mathbf{x}$ is the original normalized RGB image.

Air Bubble. We simulate bubble artifacts by overlaying translucent circular regions, combined with local defocus blur and specular highlights. Each bubble is defined by a randomly sampled center and radius, with an alpha-composited RGBA layer simulating light refraction through the bubble's surface. To enhance realism, we apply circular defocus blur *only* within the bubble region, specified by a binary mask $B \in \{0,1\}^{H\times W}$. Pixels inside the mask are blurred, while

pixels outside remain sharp:

$$\mathbf{x}' \;=\; (1 - B) \odot \mathbf{x} \;+\; B \odot \mathrm{Blur}_{\sigma}(\mathbf{x}), \tag{5.4}$$

where $\mathrm{Blur}_{\sigma}(\cdot)$ denotes a circular defocus kernel of severity σ. Specular highlights are added by drawing brighter inner ellipses and Gaussian-blurring them to mimic reflections, yielding a visually coherent and spatially localized occlusion that resembles micro-bubbles or mounting-medium residue.

5.4.3 Blurring

In addition to staining and contamination artifacts, histopathology images are often affected by acquisition-related blurring (see Figure 5.1), typically caused by imperfect slide preparation, mechanical vibrations, or misaligned focal planes during whole-slide scanning. These degradations obscure fine-grained tissue structures critical for diagnosis and negatively impact both human interpretation and algorithmic performance (Ochi *et al.*, 2024a; Wu *et al.*, 2015a; Kohlberger *et al.*, 2019a). To simulate such degradations, we adapt established corruption techniques originally developed for robustness benchmarking in natural images (Hendrycks & Dietterich, 2019c), modifying them for use in histopathology. For **Motion blur**, each image is convolved with a 1D linear kernel of length r and Gaussian spread σ, sampled according to a high severity setting ($r = 20,\ \sigma = 15$), and applied at a random direction $\theta \sim \mathcal{U}(-45°, 45°)$, simulating scanner shake and stage instability. For **Defocus blur**, a disk-shaped convolutional kernel is applied independently to each color channel, with parameters corresponding to a strong focal distortion (e.g., radius 10, aliasing blur 0.5). These perturbations introduce strong yet plausible degradation to tissue boundaries and cellular features, challenging the spatial precision of visual encoders and providing a rigorous testbed for evaluating test-time adaptation under real-world imaging imperfections.

5.4.4 Noise and Illumination

Finally, we incorporate two additional categories that are less commonly encountered in histopathology but are introduced to induce greater distributional shifts and create more challenging out-of-distribution evaluation scenarios.

More generally, microscopy images—acquired under varying illumination conditions, sensor types, and photon counts—are often corrupted by signal-dependent shot (Poisson) noise and additive Gaussian noise (Neary-Zajiczek *et al.*, 2023; Kohlberger *et al.*, 2019b; Laine, Jacquemet & Krull, 2021), both of which can obscure subtle tissue structures critical for accurate analysis. These degradations have been shown to impair classification performance and image-diagnosis quality (Kohlberger *et al.*, 2019b). To simulate these effects, we apply **Gaussian noise** sampled from a standard normal distribution and **Shot noise** sampled from the Poisson distribution. We inject both at a high-severity setting: Gaussian noise ($\sigma = 0.38$) and shot noise (photon-count factor $c = 3$).

In addition to noise-related degradations, histopathology images may be subject to intensity variations arising from differences in illumination or scanner calibration. Uneven illumination across fields of view, often resembling shading artifacts, can cause intensity distortions that affect downstream analysis (Tak *et al.*, 2020). To simulate such effects, we apply two transformations: **Contrast**, where pixel value differences are scaled to increase or decrease the dynamic range of the image, and **Brightness**, where an additive constant is applied to all pixels, shifting the overall luminance. Both are applied at a high-severity setting to mimic strong intensity changes—contrast (scaling factor = 0.05) and brightness (additive shift $\Delta I = 0.5$).

Representative examples of the proposed Histopath-C corruptions are illustrated in Figure 5.2.

5.5 Method

To introduce our method, we begin by explaining how VLMs are applied to image classification and providing an overview of the TTA setting considered in this work. We identify the essential

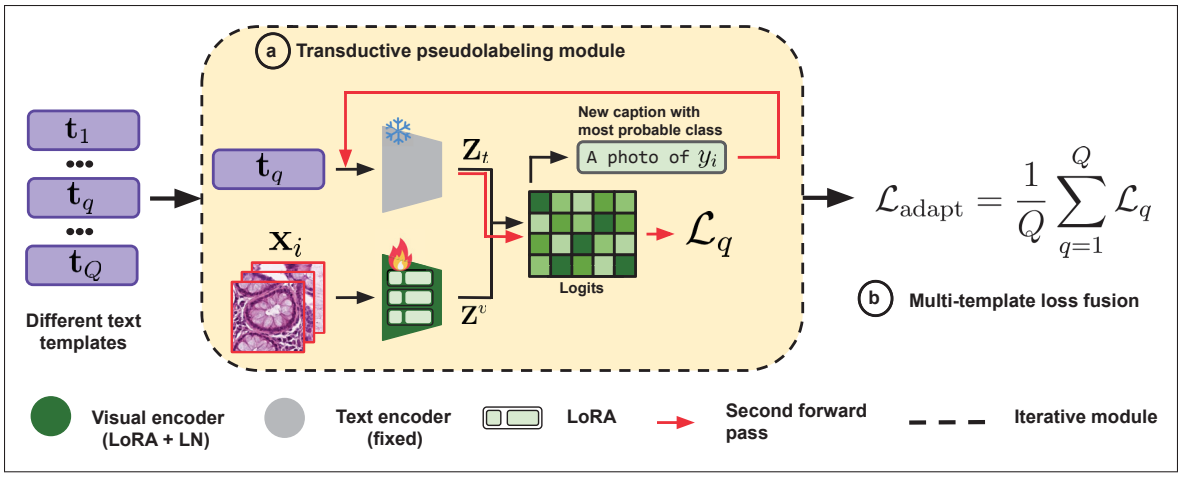

Figure 5.3 The overall framework of LATTE. All templates are used parallely to compute a loss function. a) We use a transductive pseudolabeling module to align the prediction of each image and its corresponding text caption, and the image-wise and text-wise similarities. b) The losses of the different templates are averaged to build the final adaptation loss used to finetune the model. LoRA and normalization layers are crucial components for adaptation

points leading towards an effective adaptation against histopathology-plausible domain shifts: *a*) a transductive vision-text loss, *b*) low-rank and normalization layers adaptation, and *c*) loss-level text ensembling.

Vision-language classification. Our method relies on a pre-trained VLM consisting of a visual encoder f_v and a text encoder f_t, parameterized by θ_v and θ_t, respectively. From a batch of B images $\boldsymbol{x}_i$ and a set of C classes described using text prompts $\boldsymbol{t}_k$, the normalized visual and text features are computed as $z_v = f_v(\boldsymbol{x}_i)$ and $z_t^k = f_t(\boldsymbol{t}_k)$. The probability of assigning images to class c is then calculated as

$$p(y = c \,|\, \boldsymbol{x}) \;=\; \frac{\exp\left(z_v^\top z_t^c / \tau\right)}{\sum_{k=1}^{C} \exp\left(z_v^\top z_t^k / \tau\right)} \tag{5.5}$$

where τ is a temperature parameter. The predicted class for an image $\boldsymbol{x}_i$ is $\hat{y}_i = \arg\max_c p(y = c \,|\, \boldsymbol{x}_i)$. In TTA, the objective is to adapt the model, via f_v or f_t, to an unlabeled target dataset $\mathcal{D}_{target} = \{\boldsymbol{x}_i\}_{i=1}^{N}$.

Transductive pseudolabeling. Entropy minimization, the most widely used TTA paradigm, presents severe limitations in VLMs, where using conditional entropy on probabilistic pseudolabels (*e.g.,* , softmax logits) tends to cause different degrees of collapse due to overconfidence. This issue is further compounded when multiple text templates are involved, a common scenario in the adaptation of medical VLMs. To address this, we propose integrating both vision and multi-text information to generate more robust pseudolabels, enhancing the adaptation process.

Following (Osowiechi *et al.*, 2024b), we first compute an image-wise similarity matrix $\mathbf{S}_v = \mathbf{Z}_v^\top \mathbf{Z}_v \in \mathbb{R}^{B\times B}$, which quantifies the degree of similarity between each pair of images. Similarly, we use the predicted class label $\hat{y}_i$ of the i-th image to generate its corresponding text feature $\hat{\mathbf{Z}}_{t,i} = f_t(template(\hat{y}_i))$. To capture the similarity between text-based predictions across images, we then compute the text-wise similarity matrix as $\mathbf{S}_t = \hat{\mathbf{Z}}_t^\top \hat{\mathbf{Z}}_t \in \mathbb{R}^{B\times B}$. Finally, the transductive pseudolabels are obtained by combining the image-wise and text-wise similarities as follows:

$$\hat{\mathbf{P}}_q \;=\; softmax\Big(\frac{\mathbf{S}_v + \mathbf{S}_t}{2}\Big). \tag{5.6}$$

With the new logits $\hat{\mathbf{Y}}^{\text{new}} = \mathbf{Z}_v^\top \hat{\mathbf{Z}}_t$, we use the cross-entropy between $\hat{\mathbf{Y}}^{\text{new}}$ and $\hat{\mathbf{P}}_q$ as our adaptation loss:

$$\mathcal{L}_q \;=\; \mathcal{H}(\hat{\mathbf{Y}}_q^{\text{new}}, \hat{\mathbf{P}}_q). \tag{5.7}$$

Minimizing this loss, the visual and text encoders are encouraged to output similar embeddings for (**potentially different**) samples which are related visually or semantically.

Loss-level text ensemble. To account for the different contribution of templates in solving the classification task, we utilize Q different text templates per task. The loss function in Eq. 5.7 is computed for all text features $\hat{\mathbf{Z}}_{t,i} = f_t(template_q(\hat{y}_i))$. Finally, the contribution of each template is aggregated on the loss level as a linear combination:

$$\mathcal{L}_{\text{adapt}} \;=\; \sum_{q=1}^{Q} \alpha_q \mathcal{L}_q \tag{5.8}$$

with $\sum_q \alpha_q = 1$. Besides being a natural solution, a uniformly distributed combination (*i.e.,* , $\alpha_q = 1/Q$) demonstrated competitive performance across benchmarks, providing a sufficiently diverse gradient update from the full spectrum of text information. During evaluation, we apply text averaging to derive the final prediction.

Adapting beyond normalization layers: Contrary to the widely established practice of only focusing on normalization layers (Wang *et al.*, 2021a) for adaptation, we also exploit the recently introduced Low-Rank (LoRA) adaptation (Hu *et al.*, 2022), where the *Queries*, *Keys*, *Values* and MLP layers are fine-tuned through parameter-efficient adapters modeled as the product of two low-rank matrices. A given set of weights $\mathbf{W} \in \mathbb{R}^{d_{in} \times d_{out}}$ is adapted as $\mathbf{W}' = \mathbf{W} + \mathbf{BA}$, with $\mathbf{A} \in \mathbb{R}^{d_{in} \times r}$ and $\mathbf{B} \in \mathbb{R}^{r \times d_{out}}$ as matrices of rank $r \ll d_{in}$.

5.6 Experiments

Datasets. To assess the effectiveness and generalization capabilities of LATTE, we perform extensive evaluations across a suite of diverse and challenging histopathology datasets, encompassing various organs, cancer types, imaging conditions, and annotation granularities, including colorectal cancer (NCT-7K/100K (Kather, Halama & Marx, 2024)), lung/colon adenocarcinoma (LC25000 (Borkowski *et al.*, 2019)), skin tumors (SkinCancer (Kriegsmann *et al.*, 2022)), renal cell carcinoma textures (RenalCell (Brummer, Pölönen, Mustjoki & Brück, 2022)), and colorectal polyp subtypes (MHIST (Wei *et al.*, 2021a)). Together, these datasets span binary to multi-class classification tasks, fine-grained to coarse-grained labels, and intra-/inter-organ variability. For each, we apply the ten corruptions in Histopath-C, denoted as *Dataset-C*. Full dataset details are provided in the supplementary.

Baselines. We follow the well established practices in TTA, and incorporate popular methods into our baseline evaluation. We use TENT (Wang *et al.*, 2021a), the most general entropy minimization method, TPT (Shu *et al.*, 2022) for prompt tuning, LAME (Boudiaf *et al.*, 2022) as a parameter-free method, and CLIPArTT (Hakim *et al.*, 2024) as a VLM-oriented method. Adaptation is performed for 10 iterations, with a learning rate of 10^{-3} on the

	VLM: Quilt Dataset	Source	TENT	LAME	TPT	CLIPArTT	LATTE (Ours)
	NCT7K	60.86	61.65	68.91	58.00	67.01	**69.24** (+8.4)
NCT7K-C	Stain-Light	46.77	51.71	53.39	46.45	55.61	**65.26** (+18.5)
	Stain-Heavy	35.72	13.18	35.85	34.29	53.99	**66.95** (+31.2)
	Dust	55.53	60.86	35.85	54.47	67.80	**67.92** (+12.4)
	Air Bubble	63.89	63.19	65.01	63.19	65.02	**67.44** (+3.6)
	Defocus Blur	53.36	40.75	51.11	52.04	56.33	**64.45** (+11.1)
	Motion Blur	34.25	14.86	32.46	34.54	**49.47**	46.70 (+12.5)
	Gaussian Noise	26.77	9.59	18.25	26.91	62.92	**69.72** (+43.0)
	Shot Noise	21.59	10.11	12.18	21.41	57.63	**67.42** (+45.8)
	Brightness	43.75	28.66	48.30	43.73	**57.94**	57.09 (+13.3)
	Contrast	22.63	9.35	23.92	22.52	42.85	**44.82** (+22.2)
	Mean	40.43	30.23	37.63	39.96	56.96	**61.78** (+21.4)
Other Datasets	NCT100K	55.98	41.42	64.06	52.83	59.86	**68.14** (+12.2)
	NCT100K-C	39.89	28.30	37.20	38.42	51.18	**56.13** (+16.2)
	LC25K-Lung	82.87	70.34	88.00	83.03	–	**89.41** (+6.5)
	LC25K-Lung-C	72.54	52.65	74.08	72.06	–	**78.80** (+6.3)
	LC25K-Colon	94.41	89.28	98.70	94.50	–	**99.21** (+4.8)
	LC25K-Colon-C	78.13	55.69	79.86	77.58	–	**92.17** (+14.0)
	LC25K-All	79.28	71.39	**87.13**	79.22	80.47	86.97 (+7.7)
	LC25K-All-C	57.13	40.26	56.61	56.64	65.76	**71.68** (+14.6)
	Skin	44.22	24.31	40.09	45.16	46.42	**50.62** (+6.4)
	Skin-C	22.21	8.78	17.30	22.47	29.50	**33.81** (+11.6)
	Renal	49.76	43.19	**50.77**	50.29	43.28	46.14 (-3.6)
	Renal-C	30.46	26.95	30.40	30.20	29.37	**38.29** (+7.8)
	MHIST	62.95	63.15	63.10	61.51	–	**64.02** (+1.1)
	MHIST-C	57.75	55.05	53.29	57.60	–	**62.01** (+4.3)

Table 5.1 Comparison of test-time adaptation methods with Quilt (Ikezogwo *et al.*, 2024) as the base VLM. Results are reported on multiple datasets under clean and corrupted settings. Gains of our method over the source model are highlighted in green. Note that CLIPArTT is not applicable to datasets with fewer than three classes due to its method constraints. For detailed corruption-specific results and corresponding statistics (mean ± standard deviation over three runs), please refer to the supplementary material

Adam optimizer, applied on batches of 128 images. We report results across three pathology VLMs—Quilt (Ikezogwo *et al.*, 2024), PathGen (Sun *et al.*, 2024), and CONCH (Lu *et al.*, 2024). Quilt uses ViT-B/32 at 224×224; PathGen uses ViT-B/16 at 224×224; CONCH uses ViT-B/16 at 448×448. Unless otherwise stated (e.g., in ablations), Quilt serves as the default VLM.

5.6.1 Baseline evaluation

Performance on Histopathology Images. Different pathology VLMs rely on distinct backbones, tokenization schemes, and training strategies. To comprehensively evaluate Histopath-C, we report results separately for Quilt, PathGen, and CONCH across datasets and adaptation methods in Tables 5.1, 5.2, and 5.3, respectively.

VLM: PathGen Dataset	Source	TENT	LAME	TPT	CLIPArTT	LATTE (Ours)
NCT7K	64.74	73.02	73.86	64.42	64.95	**79.95** (+15.2)
NCT7K-C	43.52	45.07	44.46	43.39	57.51	**74.05** (+30.5)
NCT100K	66.41	67.30	68.73	65.73	69.84	**81.73** (+15.3)
NCT100K-C	45.91	40.25	45.53	45.41	58.22	**71.68** (+25.8)
LC25K-Lung	82.79	81.85	77.74	81.41	–	**96.05** (+13.3)
LC25K-Lung-C	67.09	52.88	63.94	66.44	–	**89.37** (+22.3)
LC25K-Colon	95.92	97.69	**98.98**	95.82	–	98.27 (+2.4)
LC25K-Colon-C	72.96	70.18	74.07	72.82	–	**94.39** (+21.4)
LC25K-All	83.25	90.47	83.74	82.38	87.40	**93.27** (+10.0)
LC25K-All-C	55.66	50.00	56.95	55.39	70.52	**82.08** (+26.4)
Skin	55.26	56.02	63.52	54.71	63.22	**68.34** (+13.1)
Skin-C	26.22	18.92	23.71	26.18	38.07	**50.65** (+24.4)
Renal	48.61	49.66	50.28	48.67	50.42	**56.81** (+8.2)
Renal-C	23.04	15.81	19.59	22.82	34.68	**44.08** (+21.0)
MHIST	57.11	39.00	46.21	57.22	–	**60.49** (+3.4)
MHIST-C	55.29	46.08	49.54	54.94	–	**56.22** (+0.9)

Table 5.2 Comparison of test-time adaptation methods with PathGen (Sun *et al.*, 2024) as the base VLM. Results are reported on multiple datasets under clean and corrupted settings. Gains of our method over the source model are highlighted in green

VLM: CONCH Dataset	Source	TENT	LAME	TPT	CLIPArTT	LATTE (Ours)
NCT7K	67.55	68.66	73.59	67.41	58.28	**82.12** (+14.6)
NCT7K-C	37.35	38.19	38.66	37.29	39.09	**67.91** (+30.6)
NCT100K	63.79	65.00	71.82	63.73	59.66	**82.76** (+19.0)
NCT100K-C	32.98	32.32	34.24	32.94	37.01	**63.53** (+30.6)
LC25K-Lung	87.95	90.22	92.67	87.45	–	**97.02** (+9.1)
LC25K-Lung-C	56.13	52.00	55.95	56.17	–	**83.72** (+27.6)
LC25K-Colon	95.31	96.32	97.46	95.15	–	**99.18** (+3.9)
LC25K-Colon-C	71.19	69.93	71.57	71.10	–	**97.42** (+26.2)
LC25K-All	86.58	87.98	90.55	86.17	89.49	**95.81** (+9.2)
LC25K-All-C	51.17	49.26	51.98	51.17	48.06	**80.38** (+29.2)
Skin	34.56	33.34	35.98	34.11	31.39	**64.32** (+29.8)
Skin-C	21.52	19.64	21.51	21.44	21.85	**47.61** (+26.1)
Renal	46.16	48.74	51.26	45.65	52.87	**57.22** (+11.1)
Renal-C	20.62	18.16	20.21	20.58	22.64	**48.99** (+28.4)
MHIST	59.98	62.59	60.39	59.72	–	**63.00** (+3.0)
MHIST-C	57.89	57.45	**58.78**	58.06	–	56.59 (-1.3)

Table 5.3 Comparison of test-time adaptation methods with CONCH (Lu *et al.*, 2024) as the base VLM. Results are reported on multiple datasets under clean and corrupted settings. Gains of our method over the source model are highlighted in green

Recent state-of-the-art TTA methods based on entropy minimization, such as TENT and TPT, can lead to performance degradation on certain datasets, notably on NCT-100K, as shown in the Table 5.1. This suggests that confidence-based adaptation strategies are not consistently reliable in the context of histopathology. In particular, TENT, which often serves as a strong baseline for TTA in natural image benchmarks, frequently degrades performance in our setting. These findings indicate that entropy minimization applied to normalization layers may be ill-suited for histopathology images, potentially due to the high intra-class variability and subtle structural differences characteristic of this domain. In contrast, pseudo-labeling approaches, such as CLIPArTT and our proposed method, LATTE, demonstrate significantly greater robustness. Notably, LAME also achieves strong performance without requiring backpropagation. However, LATTE consistently outperforms all methods across many evaluated datasets, improving performance by approximately 9% over the baseline for NCT7K, for instance.

Beyond Quilt, we validate our findings on PathGen (Table 5.2) and CONCH (Table 5.3), two recent pathology-specific VLMs with distinct backbones and training pipelines. Both models

show similar trends: entropy-based methods (TENT, TPT) often degrade performance, while pseudo-labeling and parameter-free approaches are more robust. Notably, although CONCH achieves strong accuracy on clean datasets, it degrades more severely under corruptions—an interesting robustness gap. Across datasets and corruption types, LATTE consistently achieves the largest improvements, in many cases exceeding 20–30% over the source baseline.

Performance on Real-World Corruptions. We apply the proposed corruptions to all datasets, leading to a performance drop for most methods compared to the baseline. Interestingly, in the case of NCT7K, applying the *Air Bubble* corruption unexpectedly improves performance over the baseline in Table 5.1, suggesting that Quilt may have encountered similar artifacts during pretraining on colorectal cancer images. Our proposed method, LATTE, consistently improves performance across all corruptions and datasets compared to the baseline. On average, LATTE also outperforms other TTA methods. However, it can be less effective on specific corruptions, such as *Motion Blur* and *Brightness* on NCT7K, where CLIPArTT achieves slightly better results. Nevertheless, LATTE remains highly robust across different corruptions, achieving substantial improvements of approximately 10% to 20% over CLIPArTT and the baseline for *Stain-Light* for example. Moreover, it achieves an average performance improvement of at least 4% over the second-best method across the corrupted datasets.

5.6.2 Ablation Studies

On the updated parameters. We investigate the optimal strategy for updating our model in the context of test-time adaptation. Specifically, we compare three approaches: updating only the normalization layers, as commonly adopted in prior TTA studies; updating only the attention mechanism using LoRA; and updating both components simultaneously. As shown in Table 5.5, the effectiveness of each strategy varies depending on the type of corruption. For instance, updating only the normalization layers proves beneficial for *Contrast* distortions, whereas leveraging LoRA is preferable for *Dust* perturbations. However, in a broader sense, jointly updating both LoRA and normalization layers tends to yield more accurate results across a wider range of corruptions.

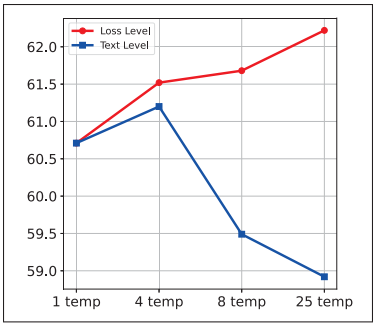

Figure 5.4 Comparison of Text and Loss averaging over several templates

VLM: Quilt Dataset		LNs	LoRA	LoRA + LNs
NCT7K		69.42± 0.02	69.63± 0.40	**69.93± 0.18**
NCT7K-C	Defocus Blur	60.00± 0.31	63.01± 0.36	**63.38± 0.03**
	Contrast	**43.74± 0.16**	41.52± 0.28	43.40± 0.48
	Stain-Light	58.23± 0.24	60.82± 0.09	**60.90± 0.13**
	Stain-Heavy	62.66± 0.09	65.23± 0.02	**65.96± 0.39**
	Dust	69.72± 0.28	**70.05± 0.21**	69.91± 0.15
	Mean	58.87	60.13	**60.71**

Figure 5.5 Comparison on the updated parameters

	Template
T^1:	"a histopathology slide showing {class k}"
T^2:	"histopathology image of {class k}"
T^3:	"pathology tissue showing {class k}"
T^4:	"presence of {class k} tissue on image"
T^5:	"a photomicrograph showing {class k}"
T^6:	"a photomicrograph of {class k}"
T^7:	"an image of {class k}"
T^8:	"an image showing {class k}"

Figure 5.6 Some examples of the used text templates.The complete list is provided in the supplementary material

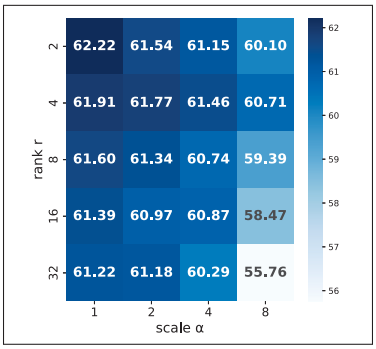

Figure 5.7 Comparison of rank r and scale α to leverage LoRA

Text averaging vs Loss averaging. As discussed in Section 5.2, leveraging multiple templates is crucial due to the nature of the textual data used to train medical VLMs. Figure 5.4 demonstrates that employing multiple templates yields better performance compared to using a single one. Specifically, using four templates improves accuracy, whether through text averaging or loss averaging. However, when increasing the number of templates further, performance degrades during adaptation with text averaging, whereas loss averaging continues to improve results. Consequently, we adopt loss averaging as our preferred strategy, using a set of 25 templates (examples of which are provided in Table 5.6).

LoRA parameters. We investigate the optimal hyperparameters for our LoRA-based adaptation strategy by evaluating different scaling factors α and ranks r, as presented in Figure 5.7. The results indicate that smaller values of α and r lead to improved performance. Specifically, setting $\alpha = 1$ and $r = 2$ optimizes the LoRA-based strategy, yielding the best adaptation performance.

5.6.3 Discussion and limitations

This paper explores TTA not only as a tool to safely deploy medical VLMs in histopathology, but also as a necessary paradigm for the application of deep models. In that spirit, our benchmark, Histopath-C, is one of the first of its type to emulate real-world shifts, and to severely challenge previous adaptation methods. We recognize, however, that the obtained performance represents the first steps towards a clinically-ready application.

We evaluate LATTE as a simple, yet effective ensembling approach for TTA in the medical field. We show that combining multiple text semantics using different loss signals, provides more positive feedback to the model (coupled with LoRA). For instance, our technique outperforms the major exponents of TTA per category; entropy-based (TENT), parameter-free (LAME), prompt tuning (TPT), and transductive adaptation (CLIPArTT). Despite these promising results, several limitations warrant discussion. As a transductive TTA method, LATTE relies on the quality of pseudolabels estimated from the target batch. Under extreme corruption severities — where visual features are severely distorted — pseudolabel reliability may degrade, reducing the effectiveness of the low-rank adaptation signal. Furthermore, like most TTA methods, LATTE's performance is sensitive to batch size, as very small batches may yield unreliable transductive estimates.

5.7 Conclusion

In this work, we introduce Histopath-C, a new benchmark featuring 10 corruption types applicable to any histopathology dataset to simulate real-world distribution shifts. We also present the first test-time adaptation framework for vision-language models in this domain, evaluating

recent TTA methods under these challenging conditions. To enhance adaptation, we propose LATTE a novel approach that integrates loss averaging across multiple templates, transductive pseudo-labeling, and low-rank adaptation. We further introduce a diverse set of 25 templates to support generalization to medical reports. Through an extensive ablation study, we provide deeper insights into our method's design choices and their impact. Comparative evaluations across multiple histopathology datasets demonstrate the superiority of our approach across all scenarios. To further advance realistic adaptation settings, future work could explore batch-based adaptation with mixed corruption types.

CONCLUSION AND RECOMMENDATIONS

Throughout this thesis, we addressed the fundamental problem of domain shift in visual recognition systems under two scenarios: Domain Generalization (DG) during training and Test-Time Adaptation (TTA) during deployment. While DG seeks to endow models with domain-invariant representations before deployment, TTA focuses on equipping them with the ability to self-adapt when confronted with unseen data distributions at inference. Together, these paradigms aim to extend the effective lifespan of models that would otherwise become brittle in real-world environments without adaptation mechanisms.

From a training-time perspective, we introduced two novel DG frameworks. First, we moved beyond conventional CNN-based paradigms to explore transformer-specific strategies for generalization. In Chapter 2, we proposed TFS-ViT, the first token-level feature stylization framework tailored for ViTs. By leveraging the unique representational properties of ViTs, TFS-ViT performs feature stylization directly at the token level, an operation not feasible in convolutional architectures, and thereby promotes generalization. This approach provides a more principled and architecture-aware route to DG, demonstrating that transformer models can benefit from internal feature mixing mechanisms that improve robustness to unseen domains. Furthermore, in Chapter 3, we introduced FDS, a diffusion-based framework that tackles domain shift from a complementary generative perspective. Instead of relying on handcrafted augmentations or uncontrolled style transfer, FDS trains a single multi-source conditional diffusion model capable of synthesizing novel pseudo-domains through domain interpolation. By selectively incorporating synthetic samples that are both semantically consistent and challenging for the classifier, identified via an entropy-based feedback mechanism, FDS generates a more diverse and informative training distribution. This controlled synthesis process broadens the support of source domains and explicitly bridges their distributional gaps, which forces the model to focus on extracting domain-invariant features and gain substantial improvements in out-of-distribution generalization without introducing any inference-time overhead.

With the advent of large-scale foundation models, which are pretrained once and reused across diverse tasks, it becomes increasingly crucial to develop mechanisms that enable their adaptation at test time without access to source data. This motivates the second part of this thesis, which explores fully test-time adaptation strategies for Vision–Language Models (VLMs). While prior work has mainly focused on adapting VLMs for image classification, we extend this paradigm to the more challenging setting of dense prediction, where the pixel-level nature of the task exacerbates distributional shifts and limits the effectiveness of existing techniques. In Chapter 4, we introduced MLMP, a fully test-time adaptation framework for VLM-based segmentation that leverages multi-layer and multi-prompt aggregation to dynamically recalibrate both vision and language representations. By fusing entropy-weighted patch-wise similarities across layers and templates, MLMP achieves robust and fine-grained adaptation under diverse corruptions, demonstrating that even large foundation models can severely degrade without proper adaptation mechanisms.

Finally, in Chapter 5, we presented Histopath-C and LATTE, the first benchmark and method for evaluating TTA of VLMs in digital histopathology. Histopath-C simulates clinically realistic domain shifts such as stain variation, blur, and contamination, while LATTE introduces a simple yet effective adaptation strategy that combines low-rank modulation with transductive vision–text ensembling. Together, they establish a principled foundation for studying and advancing the reliability of VLMs in real-world, high-stakes deployment scenarios.

Finally, the findings of this doctoral research lead to several key conclusions. Across a variety of domains and tasks, our experiments consistently show that domain shift remains a primary source of performance degradation—even for large-scale foundation models that are often perceived as universally robust. This challenge becomes particularly critical in high-stakes applications such as autonomous driving and medical imaging, where reliability under unseen conditions is non-negotiable. Our work demonstrates that addressing domain shift requires

equipping vision models with dedicated mechanisms for robustness, explored here through the complementary paradigms of DG and TTA. As the field continues to move toward ever larger and more general-purpose foundation models, traditional DG approaches may face inherent scalability limits, since no single training-time solution can anticipate all possible target domains. In this context, we argue that test-time adaptation—lightweight, and label-free—will play an increasingly central role in maintaining model reliability in the wild.

Additionally, this thesis also initiates two research directions with significant potential for future work: (i) TTA for open-vocabulary semantic segmentation, aimed at enabling foundation models to generalize across novel domains, and (ii) practical, domain-specific adaptation in clinical settings through our Histopath-C benchmark.

We hope this thesis inspires further research toward domain-universal and self-adaptive vision systems capable of enduring the diversity and unpredictability of real-world data.

APPENDIX I

SUPPLEMENTARY MATERIAL FOR THE PAPER TITLED TFS-VIT: TOKEN-LEVEL FEATURE STYLIZATION FOR DOMAIN GENERALIZATION

1. Result Reproduction

Our proposed TFS-ViT method is implemented in Python and PyTorch framework, and we use an NVIDIA Tesla V100 GPU for all of our experiments. The original implementation and the instructions for reproducing the results can be found in this repository.

2. Stylization Visualization

In order to get a better understanding of our proposed token-level feature stylization method, we train a simple ViT-based encoder-decoder network[1] without performing any stylization. When the training is finished, we perform token-level stylization in the encoder using a batch of input images - precisely like what we do in TFS-ViT - and try to reconstruct images to see the effect of stylization at the pixel level. For this section, we conduct two experiments: the effects of single layer selection as well as random selections of multiple layers on the stylization using two values, $\{0.5, 1.0\}$, for the parameter d.

Single Layer Figure I-1 demonstrates the reconstructed images when token-level stylization is performed on different layers of the Vision Transformer model (encoder). As can be observed from the figure, stylizing tokens at different layers of the model preserves the original structure of the input images. This observation is in contrast with CNNs in which feature-level stylization can only be adopted on the very first convolutional layers where style-related features are extracted (Jeon, Hong, Lee, Lee & Byun, 2021; Zhou *et al.*, 2021). Additionally, from the figure,

[1] We use ViT-base (Dosovitskiy *et al.*, 2020) as encoder, and an architecture similar to (He *et al.*, 2022) as our decoder. The "Photo", "Art", "Cartoon" domains of PACS dataset is used to train the model. During the training, no feature stylization is used. We train the network for 800 epochs using the default hyperprameters in (He *et al.*, 2022).

we observe that choosing a random number of tokens (in this case $d = 0.5$) and replacing them with stylized ones leads to creating more diverse samples.

Multiple Random Layers Figure I-2 illustrates the effect of choosing up to four random layers on the reconstructed images. As can be seen from the figure, stylizing more layers in addition to stylizing random tokens (in this case $d = 0.5$) results in even more diverse images compared to single-layer selection. These diverse synthetic features generated from different domains are able to simulate different kinds of domain shifts during training and accordingly force the network to learn domain-invariant features.

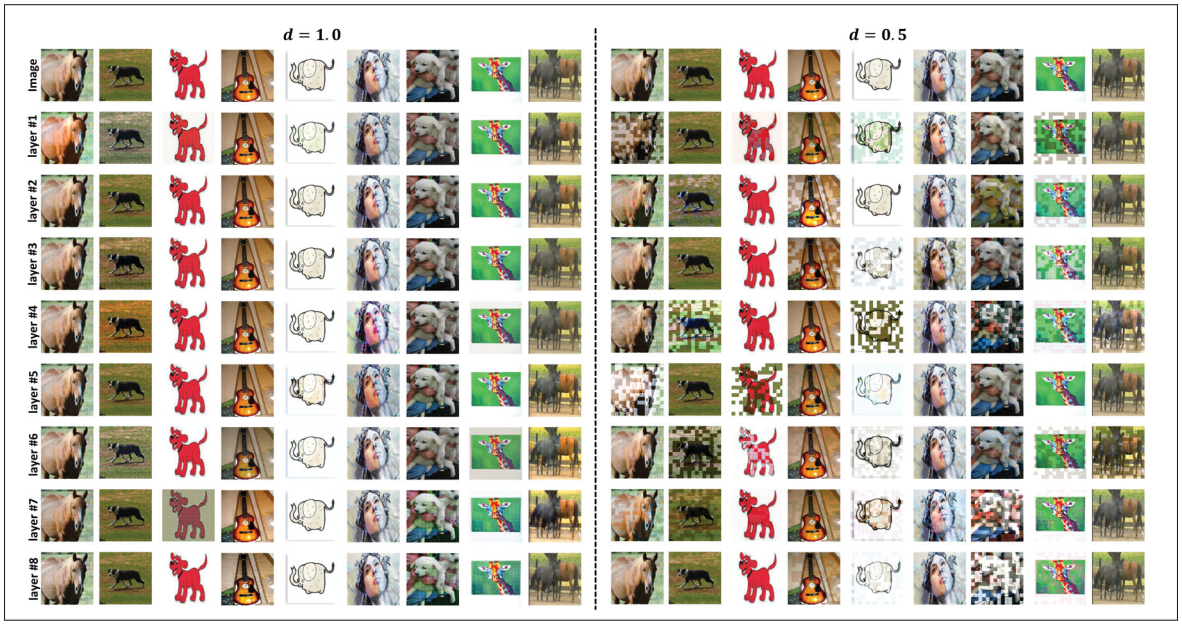

Figure-A I-1 The effect of the token-level stylization on different layers of the Vision Transformer model (encoder). The left and right figures show the results of the reconstructed images when all and half of the tokens are stylized, respectively

3. Detailed Results

The detailed breakdown of our method's performance across all domains for VLCS (Fang *et al.*, 2013), OfficeHome (Venkateswara *et al.*, 2017), TerraIncognita (Beery *et al.*, 2018), and DomainNet (Peng *et al.*, 2019) can be seen in Tables. I-1, I-2, I-3, and I-4, respectively. As can be seen, our method has improves the average accuracy of baseline (ERM-ViT) and SD-ViT,

Figure-A I-2 The effect of stylizing multiple randomly-selected layers on the reconstructed images. The left and right figures show the results of the reconstructed images when all and half of the tokens are stylized, respectively

Table-A I-1 Our proposed method performance on different domains of the VLCS (Fang *et al.*, 2013) dataset. Mean and Standard Deviation are reported across three runs. The best and second best average is in **bold** and underlined fonts, respectively

Method	Backbone	# of Params	Caltech101	LableMe	SUN09	VOC2007	Average
ERM-ViT	DeiT-Small	22M	96.7± 0.8	65.2± 1.0	73.9± 0.3	77.4± 0.3	78.3± 0.5
SDViT (Sultana *et al.*, 2022)	DeiT-Small	22M	96.8± 0.5	64.2± 0.8	76.2± 0.4	78.5± 0.4	78.9± 0.4
TFS-ViT (ours)	DeiT-Small	22M	98.70± 0.35	66.57± 0.51	76.5± 1.38	79.00± 1.00	80.19± 0.45
ATFS-ViT (ours)	DeiT-Small	22M	**99.12± 0.25**	**66.71± 0.77**	75.96± 0.93	80.82± 0.74	80.65± 0.36
ERM-ViT	T2T-ViT-14	21.5M	96.5± 0.5	64.5± 0.1	76.4± 0.4	78.2± 1.0	78.9± 0.3
SDViT(Sultana *et al.*, 2022)	T2T-ViT-14	21.5M	96.9± 0.4	64.0± 0.5	76.7± 1.4	80.4± 1.3	79.5± 0.8
TFS-ViT (ours)	T2T-ViT-14	21.5M	98.32± 0.29	63.86± 0.83	77.57± 0.43	80.37± 0.15	80.03± 0.25
ATFS-ViT (ours)	T2T-ViT-14	21.5M	98.82± 0.13	65.84± 1.30	**78.32± 0.64**	**80.92± 0.65**	**80.98± 0.40**

which is the only ViT-based method for Domain generalisation that is currently available for all these four datasets. This is also true for most of the target domains on these datasets, not just on average, which can demonstrate our method's robustness in dealing with different kinds of domain shift.

4. Additional Attention Map Visualizations

In Figs. I-3, I-4, and I-5, we compared attention maps of CLS Token of our method with baseline for several images from all possible target domains on the VLCS (Fang *et al.*, 2013), OfficeHome (Venkateswara *et al.*, 2017), and TerraIncognita (Beery *et al.*, 2018)

Table-A I-2 Our proposed method performance on different domains of the OfficeHome (Venkateswara *et al.*, 2017) dataset. Mean and Standard Deviation are reported across three runs. The best and second best average is in **bold** and underlined fonts, respectively

Method	Backbone	# of Params	Art	Clipart	Product	Real World	Average
ERM-ViT	DeiT-Small	22M	67.6± 0.3	57.0± 0.6	79.4± 0.1	81.6± 0.4	71.4± 0.1
SDViT (Sultana *et al.*, 2022)	DeiT-Small	22M	68.3± 0.8	56.3± 0.2	79.5± 0.3	81.8± 0.1	71.5± 0.2
TFS-ViT (ours)	DeiT-Small	22M	68.52± 0.22	57.73± 0.08	80.15± 0.30	81.90± 0.37	72.08± 0.13
ATFS-ViT (ours)	DeiT-Small	22M	67.39± 0.20	56.98± 0.31	79.40± 0.25	81.97± 0.45	71.44± 0.16
ERM-ViT	T2T-ViT-14	21.5M	70.2± 0.5	59.0± 0.6	81.9± 0.3	83.6± 0.6	73.7± 0.2
SDViT(Sultana *et al.*, 2022)	T2T-ViT-14	21.5M	71.1± 0.5	59.2± 0.3	**82.8± 0.4**	83.5± 0.3	74.2± 0.3
TFS-ViT (ours)	T2T-ViT-14	21.5M	71.37± 0.60	60.60± 0.53	82.40± 0.17	**83.97± 0.18**	74.59± 0.21
ATFS-ViT (ours)	T2T-ViT-14	21.5M	**71.99± 0.10**	**60.67± 0.10**	82.35± 0.70	83.59± 0.65	**74.65± 0.24**

Table-A I-3 Our proposed method performance on different domains of the TerraIncognita (Beery *et al.*, 2018) dataset. Mean and Standard Deviation are reported across three runs. The best and second best average is in **bold** and underlined fonts, respectively

Method	Backbone	# of Params	location_100	location_38	location_43	location_46	Average
ERM-ViT	DeiT-Small	22M	50.2± 1.4	30.6± 0.9	53.2± 0.2	39.6± 1.0	43.4± 0.5
SDViT (Sultana *et al.*, 2022)	DeiT-Small	22M	55.9± 1.7	31.7± 2.6	52.2± 0.3	37.4± 0.6	44.3± 1.0
TFS-ViT (ours)	DeiT-Small	22M	58.70± 1.43	40.85± 1.92	53.72± 0.47	41.14± 0.22	48.60± 0.61
ATFS-ViT (ours)	DeiT-Small	22M	57.47± 2.61	36.45± 0.47	52.19± 0.56	38.11± 0.62	46.06± 0.70
ERM-ViT	T2T-ViT-14	21.5M	52.5± 1.7	43.0± 1.3	53.7± 1.1	43.0± 1.6	48.1± 0.2
SDViT(Sultana *et al.*, 2022)	T2T-ViT-14	21.5M	57.2± 2.9	45.4± 2.4	57.7± 0.8	41.9± 0.4	50.6± 0.8
TFS-ViT (ours)	T2T-ViT-14	21.5M	59.62± 1.44	**47.40± 1.28**	**58.19± 0.34**	41.84± 0.89	**51.76± 0.54**
ATFS-ViT (ours)	T2T-ViT-14	21.5M	**60.32± 0.51**	43.72± 1.34	56.39± 0.23	**44.38± 0.93**	51.20± 0.43

datasets. In almost all of these scenarios, the cls token of our method mostly uses tokens that represent the foreground object instead of background's token which mostly contains style-related information.But the baseline approach (ERM-ViT) focuses more on the features of the background and less on the features of the object in the foreground.

Table-A I-4 Our proposed method performance on different domains of the DomainNet (Peng *et al.*, 2019) dataset. Mean and Standard Deviation are reported across three runs. The best and second best average is in **bold** and underlined fonts, respectively

Method	Backbone	# of Params	Clipart	Infograph	Painting	Quickdraw	Real	Sketch	Average
ERM-ViT	DeiT-Small	22M	62.9± 0.2	23.3± 0.1	53.1± 0.2	15.7± 0.1	65.7± 0.1	52.4± 0.2	45.5± 0.0
SDViT (Sultana *et al.*, 2022)	DeiT-Small	22M	63.4± 0.1	22.9± 0.0	53.7± 0.1	15.0± 0.4	67.4± 0.1	52.6± 0.2	45.8± 0.0
TFS-ViT (ours)	DeiT-Small	22M	64.85± 0.14	23.51± 0.11	53.60± 0.15	16.60± 0.00	67.61± 0.20	53.44± 0.23	46.60± 0.06
ATFS-ViT (ours)	DeiT-Small	22M	64.36± 0.30	23.37± 0.19	53.35± 0.21	15.80± 0.00	67.39± 0.04	52.79± 0.16	46.18± 0.07
ERM-ViT	T2T-ViT-14	21.5M	67.0± 0.3	25.2± 0.2	55.3± 0.3	15.3± 0.2	**70.3± 0.1**	55.9± 0.2	48.1± 0.1
SDViT(Sultana *et al.*, 2022)	T2T-ViT-14	21.5M	**67.6± 0.2**	25.0± 0.2	55.8± 0.4	15.2± 0.3	70.0± 0.1	**55.9± 0.1**	48.2± 0.2
TFS-ViT (ours)	T2T-ViT-14	21.5M	66.15± 0.31	**25.42± 0.04**	**56.11± 0.49**	**17.04± 0.21**	69.54± 0.18	55.75± 0.42	**48.34± 0.13**
ATFS-ViT (ours)	T2T-ViT-14	21.5M	66.31± 0.30	24.74± 0.51	55.44± 0.61	16.82± 0.81	69.08± 0.40	55.27± 0.24	47.94± 0.21

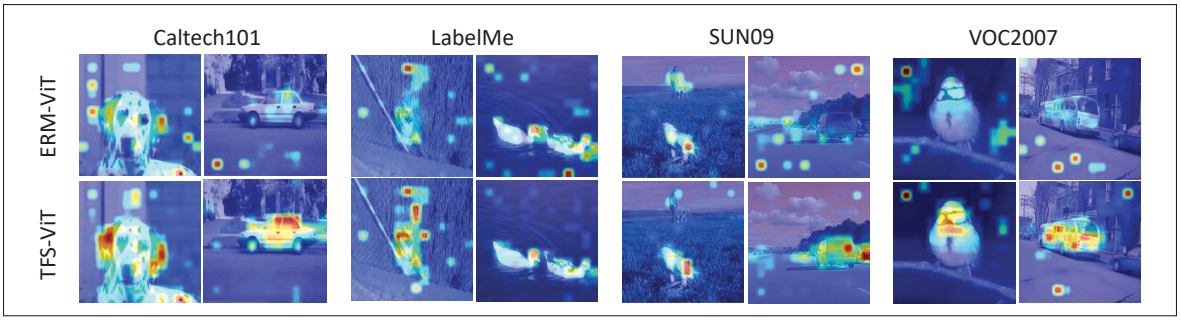

Figure-A I-3 Comparison of attention maps for the CLS token of the last layer generated by two models, ERM-ViT (baseline) and TFS-ViT (with DeiT-Small backbone), on various domains of the VLCS dataset as the unseen/target domain

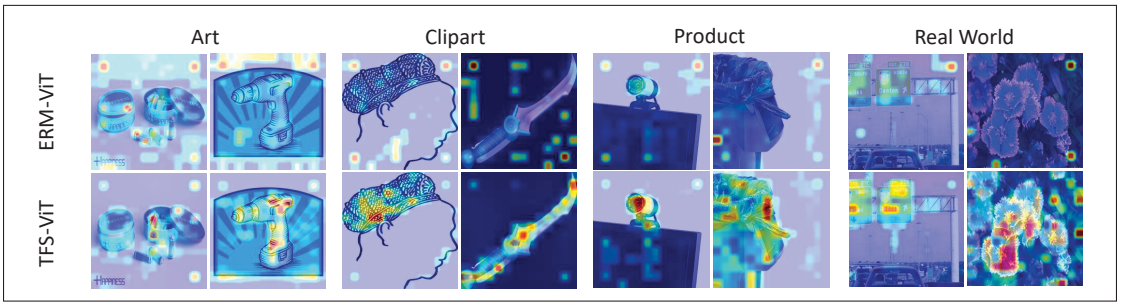

Figure-A I-4 Comparison of attention maps for the CLS token of the last layer generated by two models, ERM-ViT (baseline) and TFS-ViT (with DeiT-Small backbone), on various domains of the OfficeHome dataset as the unseen/target domain

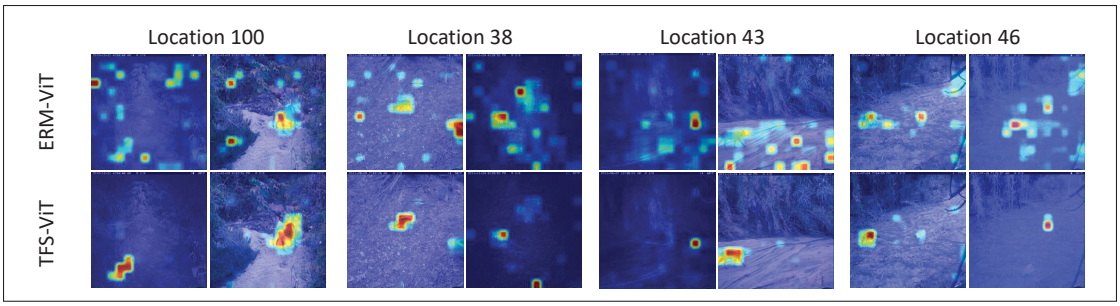

Figure-A I-5 Comparison of attention maps for the CLS token of the last layer generated by two models, ERM-ViT (baseline) and TFS-ViT (with DeiT-Small backbone), on various domains of the TerraIncognita dataset as the unseen/target domain

APPENDIX II

SUPPLEMENTARY MATERIAL FOR THE PAPER TITLED FDS: FEEDBACK-GUIDED DOMAIN SYNTHESIS WITH MULTI-SOURCE CONDITIONAL DIFFUSION MODELS FOR DOMAIN GENERALIZATION

1. Implementation

Our proposed FDS method is built using the Python language and the PyTorch framework. We utlized four NVIDIA A100 GPUs for all our experiments. For initializing our models, we utilize the original Stable Diffusion version 1.5 as our initial weight (Rombach *et al.*, 2022). The key hyperparameter configurations employed for training these diffusion models and generating new domains are detailed in Tables II-1 and II-2, respectively.

Furthermore, for classifier training, we adhere to the methodologies and parameter settings described by Cha et al. (Cha *et al.*, 2021), ensuring consistency and reproducibility in our experimental setup. The original implementation and instructions for reproducing our results are accessible via https://github.com/Mehrdad-Noori/FDS.

1.1 Choice of Generative Backbone

We chose diffusion models over GANs for several reasons specific to our setting. First, diffusion models have demonstrated consistently superior image quality and diversity compared to GANs across a wide range of benchmarks (Dhariwal & Nichol, 2021), making them a stronger foundation for generating realistic pseudo-domain samples. Second, our method requires conditioning a single generative model simultaneously on multiple domains and classes, a scenario where GANs are known to suffer from training instability and mode collapse (Goodfellow *et al.*, 2020). Diffusion models, by contrast, are considerably more stable under such complex multi-conditional training. Finally, our domain mixing strategy relies on meaningful interpolation in either the noise space or the condition space, which is well-suited to the iterative denoising process of diffusion models. GANs do not offer an analogous, principled mechanism for such interpolation.

2. Additional Ablation

Selection/Filtering. In this section, we provide visual examples to show the efficacy of our synthetic sample selection and filtering mechanism. As mentioned in the method section, this mechanism is intricately designed to scrutinize the generated images through two lenses: the alignment of the predicted class with the intended label, and the entropy indicating the prediction's uncertainty.

The Figures II-2, II-3, II-4 showcase a set of images generated from interpolations between two domains. Specifically, the diffusion model is trained on *"art"*, *"sketch"*, and *"photo"* of the PACS dataset, and the selected images, demonstrated in the first two rows, exemplify successful blends of domain characteristics, embodying a balanced mixture that enriches the training data with novel, domain-bridging examples. These images were chosen based on their ability to meet our criteria: correct class prediction aligned with high entropy scores. The third and fourth rows highlight the filtering aspect of our mechanism, displaying images not selected due to class mismatches and low entropy, respectively. This visual demonstration underlines the pivotal role of our selection/filtering process in refining the synthetic dataset, ensuring only the most challenging and domain-representative samples are utilized for model training. Through this approach, we aim to significantly bolster the model's capacity to generalize across diverse visual domains.

Inter-domain Transition. In this section, we demonstrate the model's ability to navigate between distinct visual domains, a capability enabled by adjusting the mix coefficient α. Trained on multiple source domains, our model can generate images that blend the unique attributes of each source domain. By varying α from 0.0 to 1.0, we enable smooth transitions between two source domains, where $\alpha = 0.0$ and $\alpha = 1.0$ correspond to generating pure images of the first and second domain, respectively. As an example, we illustrated this ability for our model trained on the PACS sources' *"art"*, *"sketch"*, and *"photo"*. These domain transitions are illustrated in the figures, showcasing transitions from *"photo"* to *"art"* domain in Figure II-5, *"sketch"* to *"art"* domain in Figure II-6, and *"sketch"* to *"photo"* domain in Figure II-7, respectively.

Table-A II-1 Hyperparameter Configuration for Training Diffusion Models

Config	Value
Number of GPUs	4
Learning rate	1e-4
Learning rate scheduler	LambdaLinear
Batch size	96 (24 per GPU)
Precision	FP16
Max training steps	10000
Denosing timesteps	1000
Sampler	DDPM (Ho *et al.*, 2020)
Autoencoder input size	256 x 256 x 3
Latent diffusion input size	32 x 32 x 4

Table-A II-2 Hyperparameter Configuration for Generating New Domains

Config	Value
Sampler	DDIM (Song *et al.*, 2020)
Denosing timesteps	50
Classifier-free guidance (CFG)	Randomly from [5, 6]
Mix coefficient α	Randomly from [0.3, 0.7]
Mix timestep T	Randomly from [20, 45]
Generated images (PACS)	32k per class
Generated images (VLCS)	32k per class
Generated images (OfficeHome)	16k per class

The examples provided highlight the effectiveness of our interpolation method in producing images that incorporate the distinctive features of the mixed domains, thus affirming the model's capability to generate novel and coherent visual content that bridges the attributes of its training domains. Note that in all of our generation experiments, we constrained α to the range of 0.3 to 0.7 to ensure the generated images optimally embody the characteristics of the two mixing domains, as detailed in Table II-2.

Number of Generated Domains The impact of varying the number of generated domains on model performance was rigorously evaluated, as summarized in Table II-3. This analysis aimed to understand how different combinations of augmented domains influence the overall accuracy

Table-A II-3 Analysis of the impact of utilizing different numbers/combinations of generated domains on final model performance across the PACS dataset domains (Leave-one-out accuracy). For definitions of each augmented domain (ID0, ID1, ID2), see Table II-4

Method	Augmented Domains	Accuracy (%)				
		Art	Cartoon	Photo	Sketch	Avg.
SWAD (reproduced)	—	89.49± 0.2	83.65± 0.4	97.25± 0.2	82.06± 1.0	88.11± 0.45
SWAD + FDS	ID0	91.03± 0.5	83.87± 0.6	97.75± 0.3	85.77± 0.4	89.61± 0.30
SWAD + FDS	ID1	91.01± 0.6	85.06± 1.3	97.90± 0.3	83.64± 0.4	89.40± 0.65
SWAD + FDS	ID2	91.46± 0.3	85.22± 0.8	97.88± 0.2	84.27± 0.3	89.71± 0.40
SWAD + FDS	ID0 + ID1	91.52± 0.0	85.87± 0.7	98.03± 0.3	85.70± 1.0	90.28± 0.50
SWAD + FDS	ID1 + ID2	91.62± 0.8	85.57± 0.4	98.20± 0.3	83.88± 0.6	89.82± 0.53
SWAD + FDS	ID0 + ID2	91.52± 0.1	84.54± 0.5	**98.28± 0.1**	**86.45± 0.8**	90.20± 0.38
SWAD + FDS	ID0 + ID1 + ID2	**91.80± 0.3**	**86.03± 0.8**	98.05± 0.2	86.11± 0.1	**90.50± 0.35**

Table-A II-4 Explanation of augmented domains ID definitions for each target domain of PACS dataset

Augmented Domains	Art	Cartoon	Photo	Sketch
ID0	Cartoon + Photo	Art + Photo	Art + Cartoon	Art + Cartoon
ID1	Cartoon + Sketch	Art + Sketch	Art + Sketch	Art + Photo
ID2	Photo + Sketch	Photo + Sketch	Cartoon + Sketch	Cartoon + Photo

across various dataset domains such as Art, Cartoon, Photo, and Sketch. By integrating diverse domain combinations, identified by IDs (as defined in Table II-4), we observed improvement gain when we add more generated domain of different combinations. Notably, all possible combinations of augmented domains (3 new domains for PACS, VLCS and OfficeHome) were utilized as the final method, leveraging the full spectrum of available data domains.

Stability Analysis. In this section, we demonstrate the performance of our model across different stages of training within two domains of the PACS dataset, depicted in Figure II-1. It is important to note that these test accuracies *were not used* in the selection of the best-performing model mentioned in earlier sections and all of our experiments follow leave-one-out settings

suggested by DomainBed. The results indicate that our model achieves higher stability and better mean accuracy with lower standard deviation compared to the ERM trained on original data. Note that we cannot plot the figures for SWAD since it is a WA of ERM and does not have individual training curves. These results demonstrate the robustness and stability of our model during training, which is crucial for domain generalization algorithms.

t-SNE Visualizations. This section presents a comprehensive t-SNE analysis for all classes in the PACS dataset, demonstrating the effectiveness of the FDS method in generating diverse, high-quality samples. The plots are provided in Figure II-8. Each t-SNE plot illustrates the distribution of both original and FDS-generated samples across different domains. The results shown here are based on our diffusion model trained on the “Art,” “Photo,” and “Sketch” source domains from the PACS dataset. To create these visualizations, we extracted features using the CLIP vision encoder (Radford *et al.*, 2021b). Each class in the PACS dataset is represented as distinct clusters, with “x” markers indicating the location of the average representation of each domains. These averages serve as a reference to assess how well the FDS-generated samples are compared with the original domains. These plots demonstrate how FDS enables smooth transitions between domains by interpolating between domain characteristics. This ability to generate synthetic data across a broad spectrum of domain representations improves the diversity of training data and enhances the model’s generalization ability. By covering a wider range of the domain space, FDS helps the model better handle unseen domains, making it more robust in real-world applications. These visualizations also suggest that the generated domains can be viewed as new pseudo-domains, as the FDS samples exhibit distributions distinct from their original sources domains. This additional diversity is critical for training models capable of generalizing beyond the source domains.

Visual Comparisons. This section visually compares the original images from the PACS dataset with the synthetic images generated by our FDS method, highlighting the ability of FDS to interpolate between domains. We provide examples for each pair of source domains used in training: "Art," "Photo," and "Sketch.". The visual comparisions are illustrated in Figures II-9, II-10, and II-11. Each figure contains three sections: the first section shows samples from

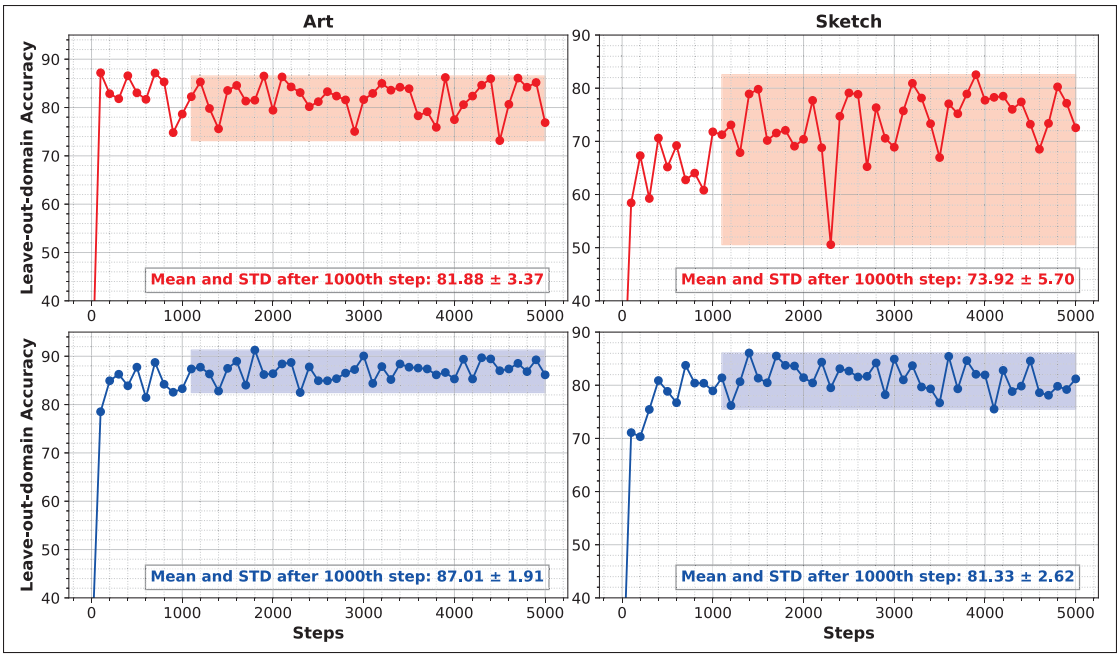

Figure-A II-1 Accuracy (%) across training steps: Comparison between ERM (top row) vs. FDS (bottom row) in "Art" and "Sketch" domains of PACS dataset

one original PACS domain, the middle section contains FDS-generated images combining the two selected domains, and the final section shows samples from the other original domain. These visual comparisons show that the FDS-generated images effectively blend domain-specific features, offering new pseudo-domain that can enrich the training set and enhance model generalization.

3. Oracle Results

In addition to leave-one-out setting, where the validation set is selected from the training domains, some studies also report the results of oracle (test-domain validation set). This can be particularly useful for understanding the potential of a method when domain knowledge is available. In this section, we compare our method (FDS+ERM) with the state-of-the-art results, as shown in Table II-5. It is important to note that no Weight Averaging (WA) methods reported their oracle results within the DomainBed framework for a fair comparison. Therefore, we only train and report our ERM results here. Our proposed method, FDS+ERM, demonstrates

Table-A II-5 Oracle (test-domain validation set) accuracy (%) results on the PACS, VLCS, and OfficeHome benchmarks. "**Aug.**" indicates whether advanced augmentation or domain mixing techniques are used. The **best results** and second-best results are highlighted

	Method	Aug.	PACS	VLCS	OfficeHome	Avg.
Standard Methods	ERM (*baseline*) (Gulrajani & Lopez-Paz, 2021)	✗	86.7± 0.3	77.6± 0.3	66.4± 0.5	76.9
	ERM (*reproduced*)	✗	86.6± 0.8	79.8± 0.4	68.4± 0.3	78.3
	IRM (Arjovsky *et al.*, 2019)	✗	84.5± 1.1	76.9± 0.6	63.0± 2.7	74.8
	GroupDRO (Sagawa *et al.*, 2019)	✗	87.1± 0.1	77.4± 0.5	66.2± 0.6	76.9
	Mixup (Yan *et al.*, 2020)	✓	86.8± 0.3	78.1± 0.3	68.0± 0.2	77.6
	CORAL (Sun & Saenko, 2016)	✗	87.1± 0.5	77.7± 0.2	68.4± 0.2	77.7
	MMD (Li *et al.*, 2018b)	✗	87.2± 0.1	77.9± 0.1	66.2± 0.3	77.1
	DANN (Ganin *et al.*, 2016)	✗	85.2± 0.2	79.7± 0.5	65.3± 0.8	76.7
	SagNet (Nam *et al.*, 2021a)	✓	86.4± 0.4	77.6± 0.1	67.5± 0.2	77.2
	RSC (Huang *et al.*, 2020)	✓	86.2± 0.5	—	66.5± 0.6	—
	SelfReg (Kim *et al.*, 2021b)	✓	86.7± 0.8	78.2± 0.1	68.1± 0.3	77.7
	Fishr (Rame, Dancette & Cord, 2022a)	✗	85.8± 0.6	78.2± 0.2	66.0± 2.9	76.7
	CDGA (Hemati *et al.*, 2023)	✓	89.6± 0.3	80.9± 0.1	68.8± 0.3	79.3
	ERM + FDS (ours)	✓	**89.7± 0.8**	**82.0± 0.1**	**71.8± 0.9**	**81.2**

superior performance across multiple benchmarks. Specifically, it achieves an average accuracy of 81.2%, outperforming all other methods. On the PACS dataset, FDS+ERM attains the highest accuracy of 89.7%, with significant improvements in the VLCS and OfficeHome datasets as well, achieving accuracies of 82.0% and 71.8% respectively. In addition to leave-one-out setting, these results also highlight the effectiveness of our approach in enhancing the performance under the oracle setting.

4. Detailed Results

Here we present the comprehensive tables containing all the detailed information that was summarized in the main paper. The leave-one-out performance (train-domain validation set) across different domains of PACS, VLCS, and OfficeHome datasets are detailed in the tables II-6, II-7, and II-8, respectively. Additionally, the oracle (test-domain validation set) accuracy

Table-A II-6 Leave-one-out accuracy (%) results on the PACS dataset. "**Aug.**" indicates whether advanced augmentation or domain mixing techniques are used. The **best results** and second-best results are highlighted

	Method	Aug.	Target Domains				
			Art	Cartoon	Photo	Sketch	Avg.
Standard Methods	ERM (baseline) (Gulrajani & Lopez-Paz, 2021)	✗	84.7± 0.4	80.8± 0.6	97.2± 0.3	79.3± 1.0	85.5± 0.2
	ERM (reproduced)	✗	86.9± 0.6	80.2± 0.7	96.6± 0.4	74.5± 2.9	84.3± 1.1
	IRM (Arjovsky *et al.*, 2019)	✗	84.8± 1.3	76.4± 1.1	96.7± 0.6	76.1± 1.0	83.5± 0.8
	GroupDRO (Sagawa *et al.*, 2019)	✗	83.5± 0.9	79.1± 0.6	96.7± 0.3	78.3± 2.0	84.4± 0.8
	Mixup (Yan *et al.*, 2020)	✓	86.1± 0.5	78.9± 0.8	**97.6± 0.1**	75.8± 1.8	84.6± 0.6
	CORAL (Sun & Saenko, 2016)	✗	88.3± 0.2	80.0± 0.5	97.5± 0.3	78.8± 1.3	86.2± 0.3
	MMD (Li *et al.*, 2018b)	✗	86.1± 1.4	79.4± 0.9	96.6± 0.2	76.5± 0.5	84.6± 0.5
	DANN (Ganin *et al.*, 2016)	✗	86.4± 0.8	77.4± 0.8	97.3± 0.4	73.5± 2.3	83.6± 0.4
	MLDG (Li *et al.*, 2018a)	✗	85.5± 1.4	80.1± 1.7	97.4± 0.3	76.6± 1.1	84.9± 1.1
	VREx (Krueger *et al.*, 2021)	✗	86.0± 1.6	79.1± 0.6	96.9± 0.5	77.7± 1.7	84.9± 1.1
	ARM (Zhang *et al.*, 2021b)	✗	86.8± 0.6	76.8± 0.5	97.4± 0.3	79.3± 1.2	85.1± 0.6
	SagNet (Nam *et al.*, 2021a)	✓	87.4± 1.0	80.7± 0.6	97.1± 0.1	80.0± 0.4	86.3± 0.2
	RSC (Huang *et al.*, 2020)	✓	85.4± 0.8	79.7± 1.8	97.6± 0.3	78.2± 1.2	85.2± 0.9
	Mixstyle (Zhou *et al.*, 2021)	✓	86.8± 0.5	79.0± 1.4	96.6± 0.1	78.5± 2.3	85.2± 0.3
	mDSDI (Bui *et al.*, 2021)	✗	87.7± 0.4	80.4± 0.7	98.1± 0.3	78.4± 1.2	86.2± 0.2
	SelfReg (Kim *et al.*, 2021b)	✓	87.9± 1.0	79.4± 1.4	96.8± 0.7	78.3± 1.2	85.6± 0.4
	Fishr (Rame *et al.*, 2022a)	✗	88.4± 0.2	78.7± 0.7	97.0± 0.1	77.8± 2.0	85.5± 0.5
	DCAug (Aminbeidokhti *et al.*, 2024b)	✓	88.5± 0.8	78.8± 1.5	96.3± 0.1	80.8± 0.5	86.1± 0.7
	DomainDiff (Miao *et al.*, 2024)	✓	84.9± 1.6	82.9± 0.0	95.5± 0.0	79.0± 0.9	85.6± 0.6
	DSI (Yu *et al.*, 2023a)	✓	84.6± 2.4	81.4± 1.6	96.8± 0.5	82.5± 1.0	86.9± 1.4
	CDGA (Hemati *et al.*, 2023)	✓	89.1± 1.0	82.5± 0.5	97.4± 0.2	**84.8± 0.9**	88.5± 0.5
	ERM + FDS (ours)	✓	**90.7± 0.9**	**84.2± 0.6**	97.2± 0.1	83.0± 0.4	**88.8± 0.1**
WA Methods	SWAD (baseline) (Cha *et al.*, 2021)	✗	89.3± 0.2	83.4± 0.6	97.3± 0.3	82.5± 0.5	88.1± 0.1
	SWAD (reproduced)	✗	89.5± 0.2	83.7± 0.4	97.3± 0.2	82.1± 0.1	88.1± 0.4
	SelfReg SWA (Kim *et al.*, 2021b)	✓	85.9± 0.6	81.9± 0.4	96.8± 0.1	81.4± 0.6	86.5± 0.3
	DNA (Chu *et al.*, 2022)	✗	89.8± 0.2	83.4± 0.4	97.7± 0.1	82.6± 0.2	88.4± 0.1
	DiWA (Rame *et al.*, 2022b)	✓	90.1± 0.6	83.3± 0.6	**98.2± 0.1**	83.4± 0.4	88.8± 0.4
	TeachDCAug (Aminbeidokhti *et al.*, 2024b)	✓	89.6± 0.0	81.8± 0.5	97.7± 0.0	84.5± 0.2	88.4± 0.2
	SWAD + FDS (ours)	✓	**91.8± 0.3**	**86.0± 0.8**	98.1± 0.2	**86.1± 0.1**	**90.5± 0.3**

results for the PACS, VLCS, and OfficeHome benchmarks are detailed in Table II-9, II-10, and II-11, respectively.

Table-A II-7 Leave-one-out accuracy (%) results on the VLCS dataset. "**Aug.**" indicates whether advanced augmentation or domain mixing techniques are used. The **best results** and second-best results are highlighted

	Method	Aug.	Target Domains				
			Caltech101	LabelMe	SUN09	VOC2007	Avg.
Standard Methods	ERM (baseline) (Gulrajani & Lopez-Paz, 2021)	✗	97.7± 0.4	64.3± 0.9	73.4± 0.5	74.6± 1.3	77.5± 0.4
	ERM (reproduced)	✗	96.9± 1.4	64.1± 1.4	71.1± 1.5	72.8± 0.9	76.2± 1.1
	IRM (Arjovsky *et al.*, 2019)	✗	98.6± 0.1	64.9± 0.9	73.4± 0.6	77.3± 0.9	78.5± 0.5
	GroupDRO (Sagawa *et al.*, 2019)	✗	97.3± 0.3	63.4± 0.9	69.5± 0.8	76.7± 0.7	76.7± 0.6
	Mixup (Yan *et al.*, 2020)	✓	98.3± 0.6	64.8± 1.0	72.1± 0.5	74.3± 0.8	77.4± 0.6
	CORAL (Sun & Saenko, 2016)	✗	98.3± 0.1	66.1± 1.2	73.4± 0.3	77.5± 1.2	78.8± 0.6
	MMD (Li *et al.*, 2018b)	✗	97.7± 0.1	64.0± 1.1	72.8± 0.2	75.3± 3.3	77.5± 0.9
	DANN (Ganin *et al.*, 2016)	✗	**99.0± 0.3**	65.1± 1.4	73.1± 0.3	77.2± 0.6	78.6± 0.4
	MLDG (Li *et al.*, 2018a)	✗	97.4± 0.2	65.2± 0.7	71.0± 1.4	75.3± 1.0	77.2± 0.8
	VREx (Krueger *et al.*, 2021)	✗	98.4± 0.3	64.4± 1.4	74.1± 0.4	76.2± 1.3	78.3± 0.8
	ARM (Zhang *et al.*, 2021b)	✗	98.7± 0.2	63.6± 0.7	71.3± 1.2	76.7± 0.6	77.6± 0.6
	SagNet (Nam *et al.*, 2021a)	✓	97.9± 0.4	64.5± 0.5	71.4± 1.3	77.5± 0.5	77.8± 0.5
	RSC (Huang *et al.*, 2020)	✓	97.9± 0.1	62.5± 0.7	72.3± 1.2	75.6± 0.8	77.1± 0.5
	Mixstyle (Zhou *et al.*, 2021)	✓	98.6± 0.3	64.5± 1.1	72.6± 0.5	75.7± 1.7	77.9± 0.5
	mDSDI (Bui *et al.*, 2021)	✗	97.6± 0.1	66.4± 0.4	74.0± 0.6	77.8± 0.7	79.0± 0.3
	SelfReg (Kim *et al.*, 2021b)	✓	96.7± 0.4	65.2± 1.2	73.1± 1.3	76.2± 0.7	77.8± 0.9
	Fishr (Rame *et al.*, 2022a)	✗	98.9± 0.3	64.0± 0.5	71.5± 0.2	76.8± 0.7	77.8± 0.5
	DCAug (Aminbeidokhti *et al.*, 2024b)	✓	98.3± 0.1	64.2± 0.4	74.4± 0.6	77.5± 0.3	78.6± 0.5
	CDGA (Hemati *et al.*, 2023)	✓	96.3± 0.7	**75.7± 1.0**	72.8± 1.3	73.7± 1.3	79.6± 0.9
	ERM + FDS (ours)	✓	98.8± 0.3	65.6± 0.9	**75.5± 0.9**	**79.3± 1.8**	**79.8± 0.5**
WA Methods	SWAD (baseline) (Cha *et al.*, 2021)	✗	98.8± 0.1	63.3± 0.3	75.3± 0.5	79.2± 0.6	79.1± 0.1
	SWAD (reproduced)	✗	98.7± 0.2	**63.9± 0.3**	74.3± 1.1	78.6± 0.6	78.9± 0.5
	SelfReg SWA (Kim *et al.*, 2021b)	✓	97.4± 0.4	63.5± 0.3	72.6± 0.1	76.7± 0.7	77.5± 0.0
	DNA (Chu *et al.*, 2022)	✗	98.8± 0.1	63.6± 0.2	74.1± 0.1	79.5± 0.4	79.0± 0.5
	DiWA (Rame *et al.*, 2022b)	✓	98.4± 0.1	63.4± 0.1	75.5± 0.3	78.9± 0.6	79.1± 0.2
	TeachDCAug (Aminbeidokhti *et al.*, 2024b)	✓	98.5± 0.1	63.7± 0.3	75.6± 0.5	77.0± 0.7	78.7± 0.5
	SWAD + FDS (ours)	✓	**99.5± 0.2**	62.9± 0.2	**76.9± 0.4**	**79.6± 1.3**	**79.7± 0.5**

Table-A II-8 Leave-one-out accuracy (%) results on the OfficeHome dataset. "**Aug.**" indicates whether advanced augmentation or domain mixing techniques are used. The **best results** and second-best results are highlighted

	Method	Aug.	Target Domains				
			Art	Clipart	Product	Real World	Avg.
Standard Methods	ERM (baseline) (Gulrajani & Lopez-Paz, 2021)	✗	61.3± 0.7	52.4± 0.3	75.8± 0.1	76.6± 0.3	66.5± 0.3
	ERM (reproduced)	✗	59.5± 2.1	51.3± 1.3	73.8± 0.8	73.8± 0.2	64.6± 1.1
	IRM (Arjovsky *et al.*, 2019)	✗	58.9± 2.3	52.2± 1.6	72.1± 2.9	74.0± 2.5	64.3± 2.2
	GroupDRO (Sagawa *et al.*, 2019)	✗	60.4± 0.7	52.7± 1.0	75.0± 0.7	76.0± 0.7	66.0± 0.7
	Mixup (Yan *et al.*, 2020)	✓	62.4± 0.8	54.8± 0.6	76.9± 0.3	78.3± 0.2	68.1± 0.3
	CORAL (Sun & Saenko, 2016)	✗	65.3± 0.4	54.4± 0.5	76.5± 0.1	78.4± 0.5	68.7± 0.3
	MMD (Li *et al.*, 2018b)	✗	60.4± 0.2	53.3± 0.3	74.3± 0.1	77.4± 0.6	66.3± 0.1
	DANN (Ganin *et al.*, 2016)	✗	59.9± 1.3	53.0± 0.3	73.6± 0.7	76.9± 0.5	65.9± 0.6
	MLDG (Li *et al.*, 2018a)	✗	61.5± 0.9	53.2± 0.6	75.0± 1.2	77.5± 0.4	66.8± 0.7
	VREx (Krueger *et al.*, 2021)	✗	60.7± 0.9	53.0± 0.9	75.3± 0.1	76.6± 0.5	66.4± 0.6
	ARM (Zhang *et al.*, 2021b)	✗	58.9± 0.8	51.0± 0.5	74.1± 0.1	75.2± 0.3	64.8± 0.4
	SagNet (Nam *et al.*, 2021a)	✓	63.4± 0.2	54.8± 0.4	75.8± 0.4	78.3± 0.3	68.1± 0.1
	RSC (Huang *et al.*, 2020)	✓	60.7± 1.4	51.4± 0.3	74.8± 1.1	75.1± 1.3	65.5± 0.9
	Mixstyle (Zhou *et al.*, 2021)	✓	51.1± 0.3	53.2± 0.4	68.2± 0.7	69.2± 0.6	60.4± 0.3
	mDSDI (Bui *et al.*, 2021)	✗	**68.1± 0.3**	52.1± 0.4	76.0± 0.2	80.4± 0.2	69.2± 0.4
	SelfReg (Kim *et al.*, 2021b)	✓	63.6± 1.4	53.1± 1.0	76.9± 0.4	78.1± 0.4	67.9± 0.7
	Fishr (Rame *et al.*, 2022a)	✗	62.4± 0.5	54.4± 0.4	76.2± 0.5	78.3± 0.1	67.8± 0.5
	DCAug (Aminbeidokhti *et al.*, 2024b)	✓	61.8± 0.6	55.4± 0.6	77.1± 0.3	78.9± 0.3	68.3± 0.4
	DomainDiff (Miao *et al.*, 2024)	✓	57.6± 0.4	49.2± 0.6	73.0± 0.6	75.2± 0.9	63.7± 0.6
	CDGA (Hemati *et al.*, 2023)	✓	60.1± 1.4	54.2± 0.5	78.2± 0.6	80.4± 0.1	68.2± 0.6
	ERM + FDS (ours)	✓	64.6± 0.2	**57.7± 0.1**	**80.2± 0.5**	**82.0± 0.4**	**71.1± 0.1**
WA Methods	SWAD (baseline) (Cha *et al.*, 2021)	✗	66.1± 0.4	57.7± 0.4	78.4± 0.1	80.2± 0.2	70.6± 0.2
	SWAD (reproduced)	✗	65.9± 0.9	56.8± 0.4	78.8± 0.3	80.0± 0.2	70.3± 0.4
	SelfReg SWA (Kim *et al.*, 2021b)	✓	64.9± 0.8	55.4± 0.6	78.4± 0.2	78.8± 0.1	69.4± 0.2
	DNA (Chu *et al.*, 2022)	✗	**67.7± 0.2**	57.7± 0.3	78.9± 0.2	80.5± 0.2	71.2± 0.1
	DiWA (Rame *et al.*, 2022b)	✓	67.3± 0.2	57.9± 0.2	79.0± 0.2	79.9± 0.1	71.0± 0.1
	TeachDCAug (Aminbeidokhti *et al.*, 2024b)	✓	66.2± 0.2	57.0± 0.3	78.3± 0.1	80.1± 0.0	70.4± 0.2
	SWAD + FDS (ours)	✓	67.3± 0.8	**60.5± 0.5**	**82.6± 0.1**	**83.6± 0.3**	**73.5± 0.4**

Table-A II-9 Oracle (test-domain validation set) accuracy (%) results on the PACS dataset. "**Aug.**" indicates whether advanced augmentation or domain mixing techniques are used. The **best results** and second-best results are highlighted

	Method	Aug.	Target Domains				
			Art	Cartoon	Photo	Sketch	Avg.
Standard Methods	ERM (baseline) (Gulrajani & Lopez-Paz, 2021)	✗	86.5± 1.0	81.3± 0.6	96.2± 0.3	82.7± 1.1	86.7± 0.8
	ERM (reproduced)	✗	88.6± 0.9	80.9± 1.9	**98.4± 0.4**	78.4± 1.2	86.6± 1.0
	IRM	✗	84.2± 0.9	79.7± 1.5	95.9± 0.4	78.3± 2.1	84.5± 1.2
	GroupDRO	✗	87.5± 0.5	82.9± 0.6	97.1± 0.3	81.1± 1.2	87.2± 0.7
	Mixup	✓	87.5± 0.4	81.6± 0.7	97.4± 0.2	80.8± 0.9	86.8± 0.6
	CORAL	✗	86.6± 0.8	81.8± 0.9	97.1± 0.5	82.7± 0.6	87.1± 0.7
	MMD	✗	88.1± 0.8	82.6± 0.7	97.1± 0.5	81.2± 1.2	87.3± 0.8
	DANN	✗	87.0± 0.4	80.3± 0.6	96.8± 0.3	76.9± 1.1	85.3± 0.6
	SagNet	✓	87.4± 0.5	81.2± 1.2	96.3± 0.8	80.7± 1.1	86.4± 0.9
	RSC	✓	86.0± 0.7	81.8± 0.9	96.8± 0.7	80.4± 0.5	86.3± 0.7
	Fishr	✗	87.9± 0.6	80.8± 0.5	97.9± 0.4	81.1± 0.8	86.9± 0.6
	SelfReg	✓	87.9± 0.5	80.6± 1.1	97.1± 0.4	81.1± 1.3	86.7± 0.8
	CDGA	✓	89.6± 0.8	**85.3± 0.7**	97.3± 0.3	**86.2± 0.5**	89.6± 0.6
	ERM + FDS (ours)	✓	**91.1± 0.3**	84.9± 0.7	97.3± 0.5	85.6± 2.3	**89.7± 0.8**

Table-A II-10 Oracle (test-domain validation set) accuracy (%) results on the VLCS dataset. "**Aug.**" indicates whether advanced augmentation or domain mixing techniques are used. The **best results** and second-best results are highlighted

	Method	Aug.	Target Domains				
			Caltech101	LabelMe	SUN09	VOC2007	Avg.
Standard Methods	ERM (baseline) (Gulrajani & Lopez-Paz, 2021)	✗	97.6± 0.3	67.9± 0.7	70.9± 0.2	74.0± 0.6	77.6± 0.5
	ERM (reproduced)	✗	98.6± 0.2	68.6± 0.7	73.6± 1.7	78.6± 1.2	79.8± 0.4
	IRM	✗	97.3± 0.2	66.7± 0.1	71.0± 2.3	72.8± 0.4	77.0± 0.8
	GroupDRO	✗	97.7± 0.2	65.9± 0.2	72.8± 0.8	73.4± 1.3	77.5± 0.6
	Mixup	✓	97.8± 0.4	67.2± 0.4	71.5± 0.2	75.7± 0.6	78.1± 0.4
	CORAL	✗	97.3± 0.2	67.5± 0.6	71.6± 0.6	74.5± 0.0	77.7± 0.4
	MMD	✗	98.8± 0.0	66.4± 0.4	70.8± 0.5	75.6± 0.4	77.9± 0.3
	DANN	✗	9.0± 0.2	66.3± 1.2	73.4± 1.4	80.1± 0.5	79.7± 0.8
	SagNet	✓	97.4± 0.3	66.4± 0.4	71.6± 0.1	75.0± 0.8	77.6± 0.4
	RSC	✓	98.0± 0.4	67.2± 0.3	70.3± 1.3	75.6± 0.4	77.8± 0.6
	Fishr	✗	97.6± 0.7	67.3± 0.5	72.2± 0.9	75.7± 0.3	78.2± 0.6
	SelfReg	✓	98.2± 0.3	63.9± 0.8	72.2± 0.1	75.5± 0.4	77.5± 0.2
	CDGA	✓	96.6± 0.7	**75.5± 1.9**	73.6± 1.1	77.8± 1.0	80.9± 1.2
	ERM + FDS (ours)	✓	**99.5± 0.1**	68.7± 0.3	**77.4± 0.7**	**82.6± 0.1**	**82.0± 0.1**

Table-A II-11 Oracle (test-domain validation set) accuracy (%) results on the OfficeHome dataset. "**Aug.**" indicates whether advanced augmentation or domain mixing techniques are used. The **best results** and second-best results are highlighted

	Method	Aug.	Target Domains				
			Art	Clipart	Product	Real World	Avg.
Standard Methods	ERM (baseline) (Gulrajani & Lopez-Paz, 2021)	✗	61.7± 0.7	53.4± 0.3	74.1± 0.4	76.2± 0.6	66.4± 0.5
	ERM (reproduced)	✗	64.0± 0.9	53.7± 1.1	77.1± 0.3	78.8± 0.4	68.4± 0.3
	IRM	✗	56.4± 3.2	51.2± 2.3	71.7± 2.7	72.7± 2.7	63.0± 2.7
	GroupDRO	✗	60.5± 1.6	53.1± 0.3	75.5± 0.3	75.9± 0.7	66.3± 0.7
	Mixup	✓	63.5± 0.2	54.6± 0.4	76.0± 0.3	78.0± 0.7	68.0± 0.4
	CORAL	✗	64.8± 0.8	54.1± 0.9	76.5± 0.4	78.2± 0.4	68.4± 0.6
	MMD	✗	60.4± 1.0	53.4± 0.5	74.9± 0.1	76.1± 0.7	66.2± 0.6
	DANN	✗	60.6± 1.4	51.8± 0.7	73.4± 0.5	75.5± 0.9	65.3± 0.9
	SagNet	✓	62.7± 0.5	53.6± 0.5	76.0± 0.3	77.8± 0.1	67.5± 0.4
	RSC	✓	61.7± 0.8	53.0± 0.9	74.8± 0.8	76.3± 0.5	66.5± 0.8
	Fishr	✗	63.4± 0.8	54.2± 0.3	76.4± 0.3	78.5± 0.2	68.1± 0.4
	SelfReg	✓	64.2± 0.6	53.6± 0.7	76.7± 0.3	77.9± 0.5	68.1± 0.3
	CDGA	✓	61.1± 1.1	55.9± 1.0	78.2± 0.8	79.8± 0.2	68.8± 0.8
	ERM + FDS (ours)	✓	**65.3± 0.8**	**58.4± 0.8**	**81.2± 0.2**	**82.4± 0.6**	**71.8± 0.9**

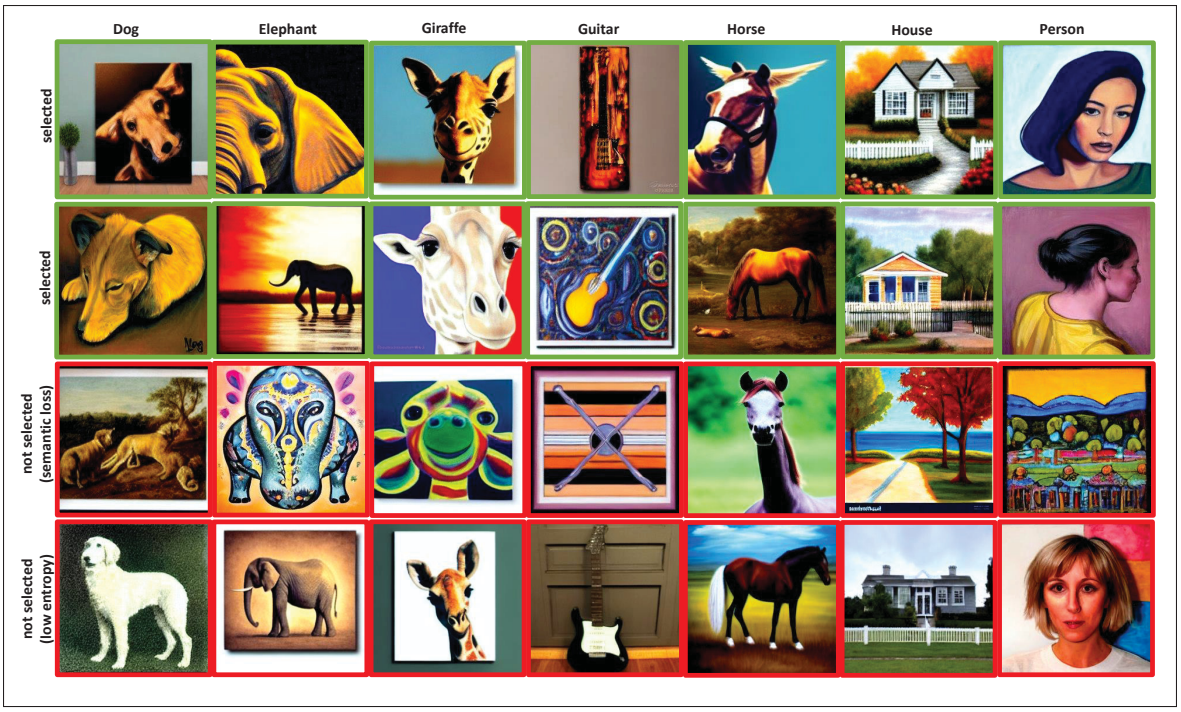

Figure-A II-2 Synthetic images from interpolating between *"art"* and *"photo"* domains of PACS, with selected images showcasing a blend of artistic and realistic features (top two rows) and non-selected images (bottom rows) due to class mismatches and low entropy

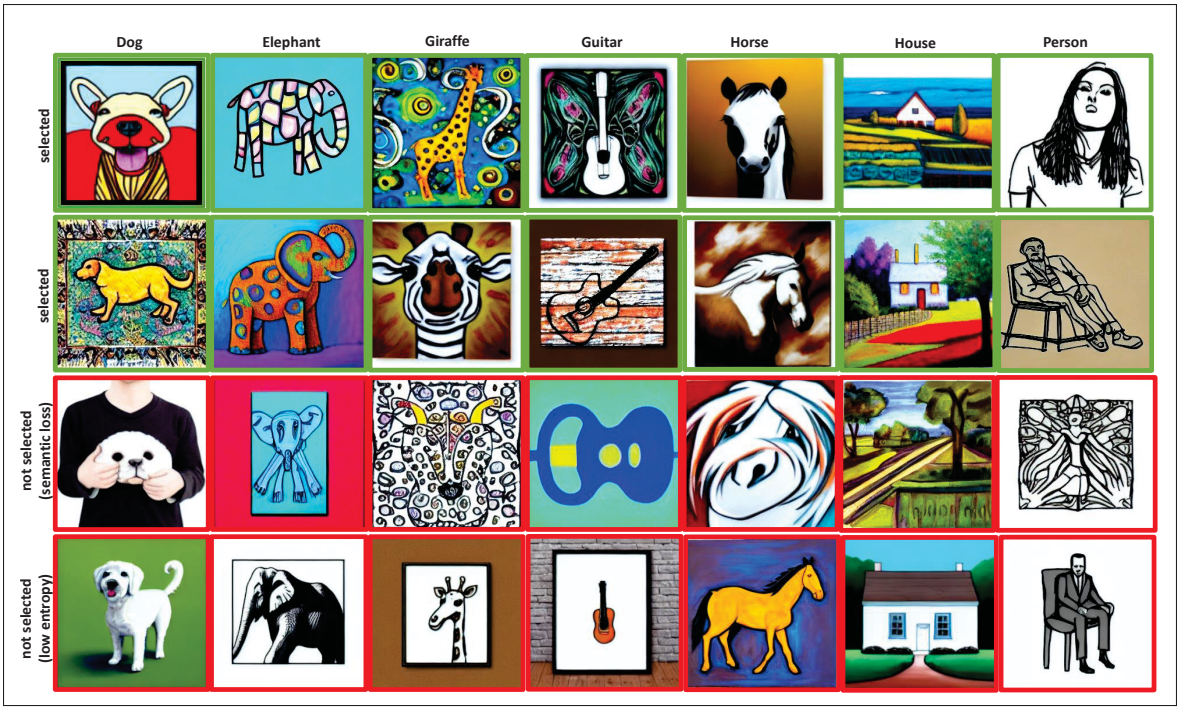

Figure-A II-3 Interpolation between *"art"* and *"sketch"* in PACS highlights selected images (top rows) merging textures and outlines, and non-selected images (bottom rows) for failing selection criteria

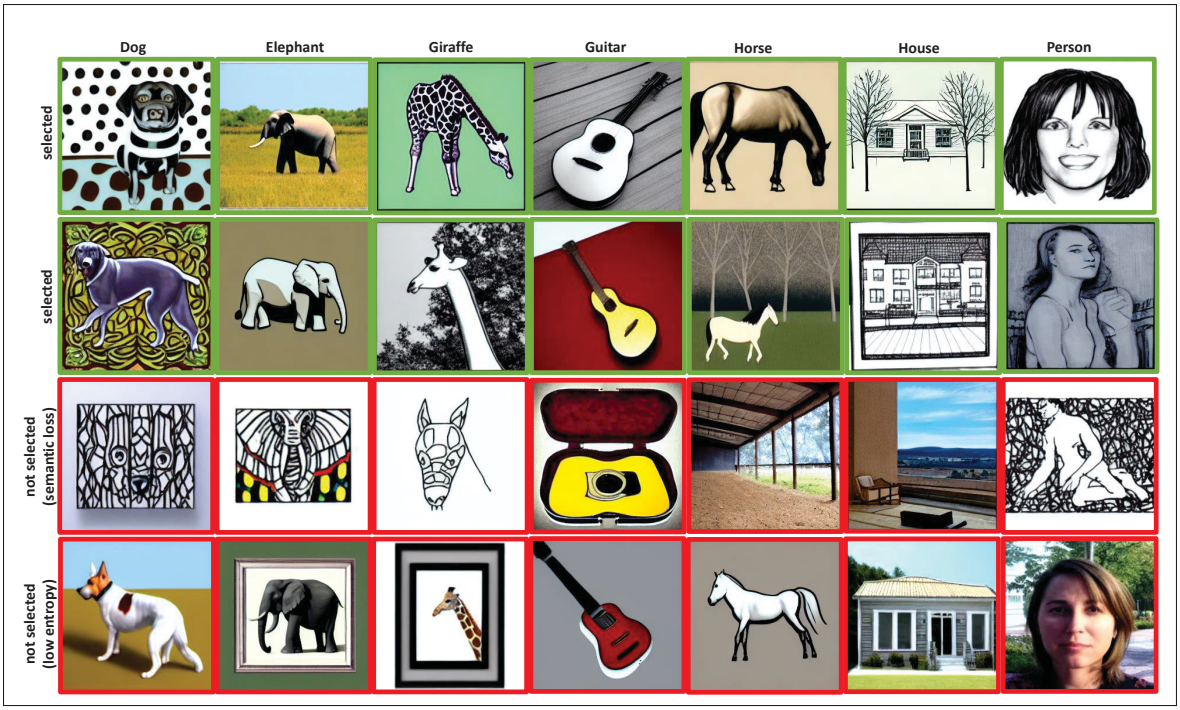

Figure-A II-4 Results from *"photo"* and *"sketch"* domain interpolation in PACS, with selected synthetic images (top rows) and non-selected due to predictability and class misalignment (bottom rows)

Figure-A II-5 Inter-domain transition from *"photo"* to *"art"*. This sequence illustrates how varying α from 0.0 (purely photorealistic images) to 1.0 (purely artistic representations) enables the model to seamlessly blend photographic realism with artistic expression, demonstrating a smooth progression from real-world imagery to stylized art

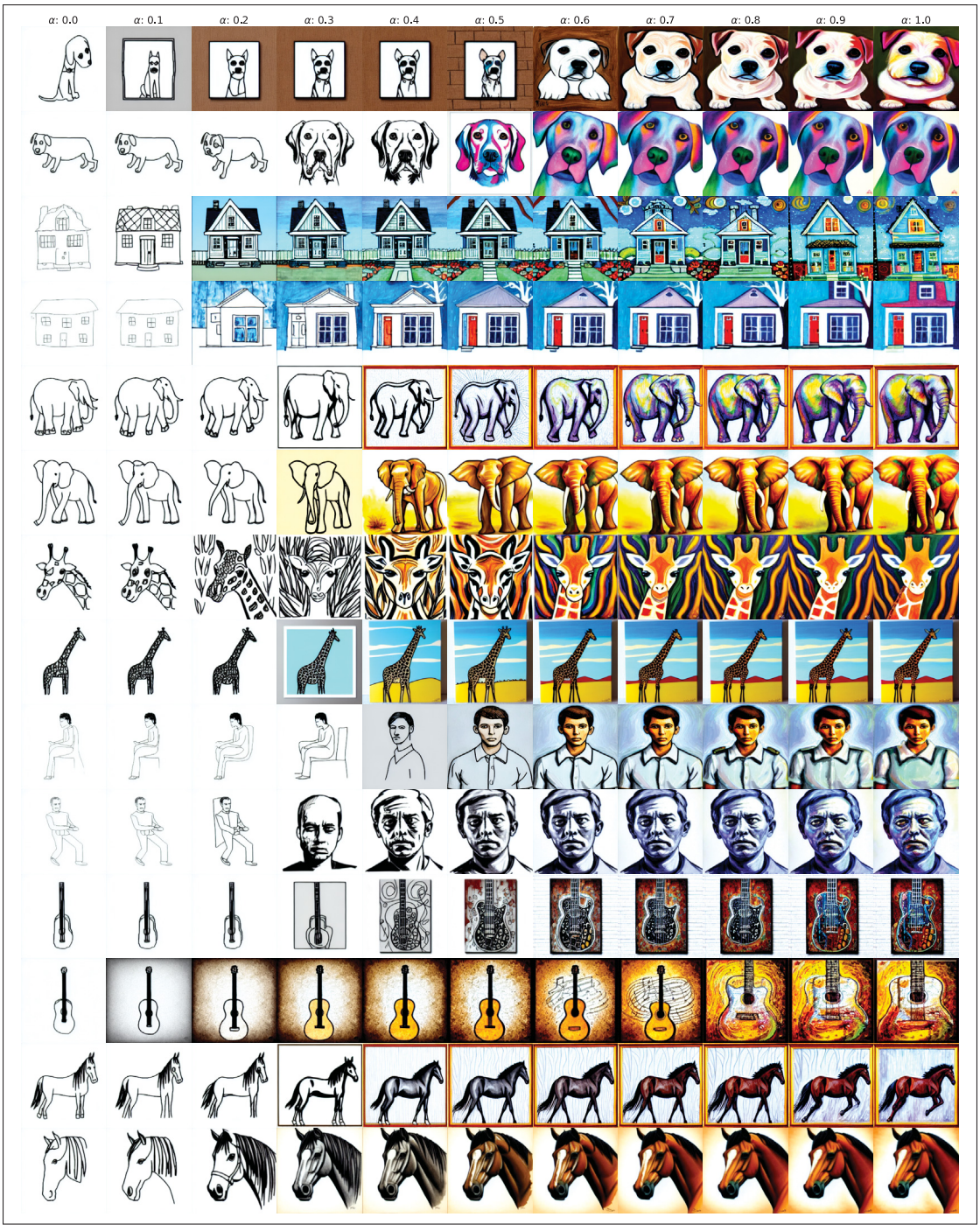

Figure-A II-6 Inter-domain transition from *"sketch"* to *"art"*. Displayed here is the transformation that occurs as α is adjusted, beginning with 0.0 (pure sketches) and moving towards 1.0 (fully art-inspired images). The model effectively infuses basic sketches with complex textures and colors, transitioning from minimalistic line art to detailed and vibrant artistic images

Figure-A II-7 Inter-domain transition from *"sketch"* to *"photo"*. This figure demonstrates the capability of the model to morph sketches into photorealistic images by altering α from 0.0 (entirely sketch-based) to 1.0 (completely photorealistic). The transition highlights the model's proficiency in enriching simple outlines with lifelike details and textures, bridging the gap between abstract sketches and reality

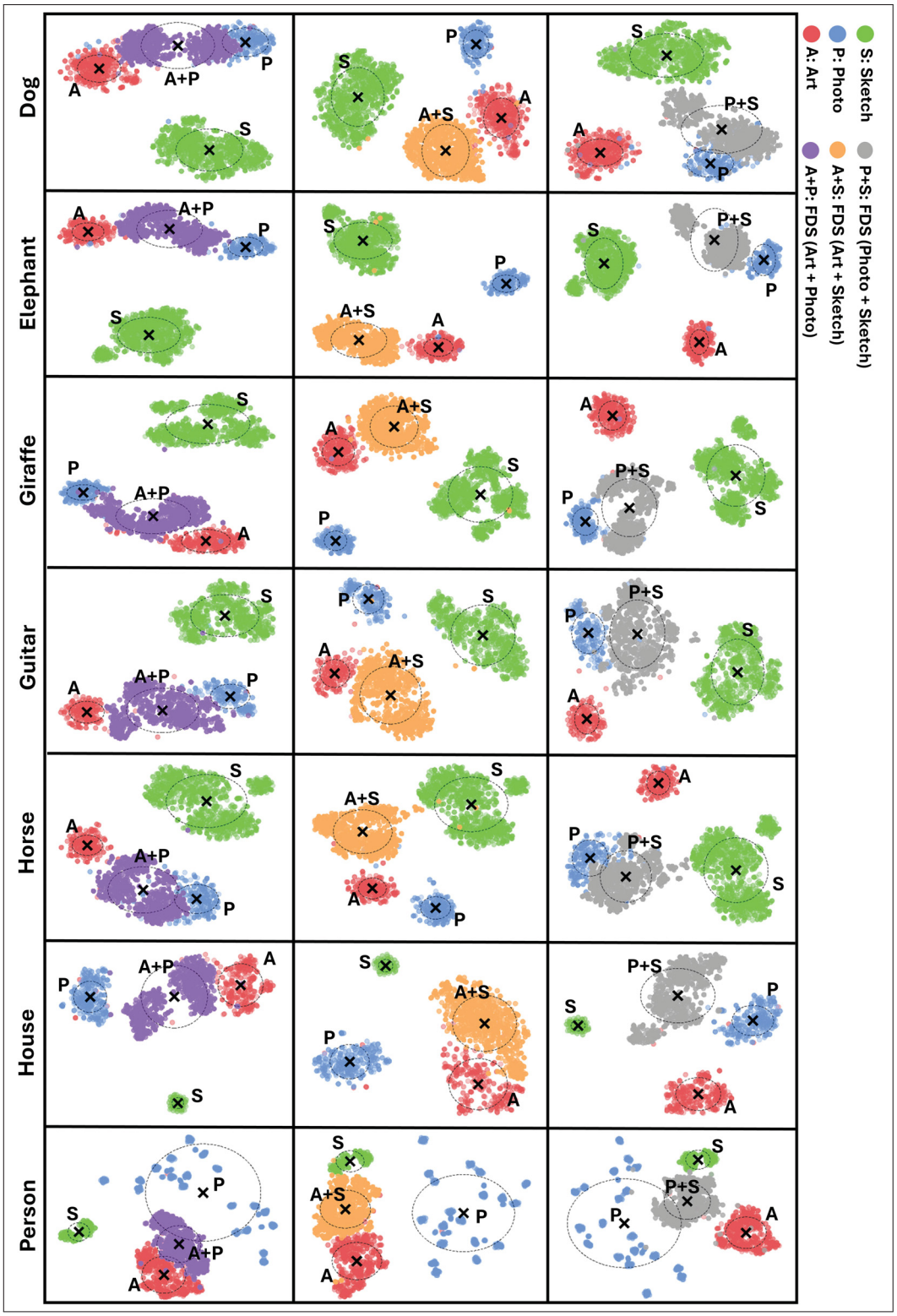

Figure-A II-8 t-SNE visualization of all classes from the PACS dataset, showing the distribution of original source domains (Art, Photo, Sketch) and FDS ones

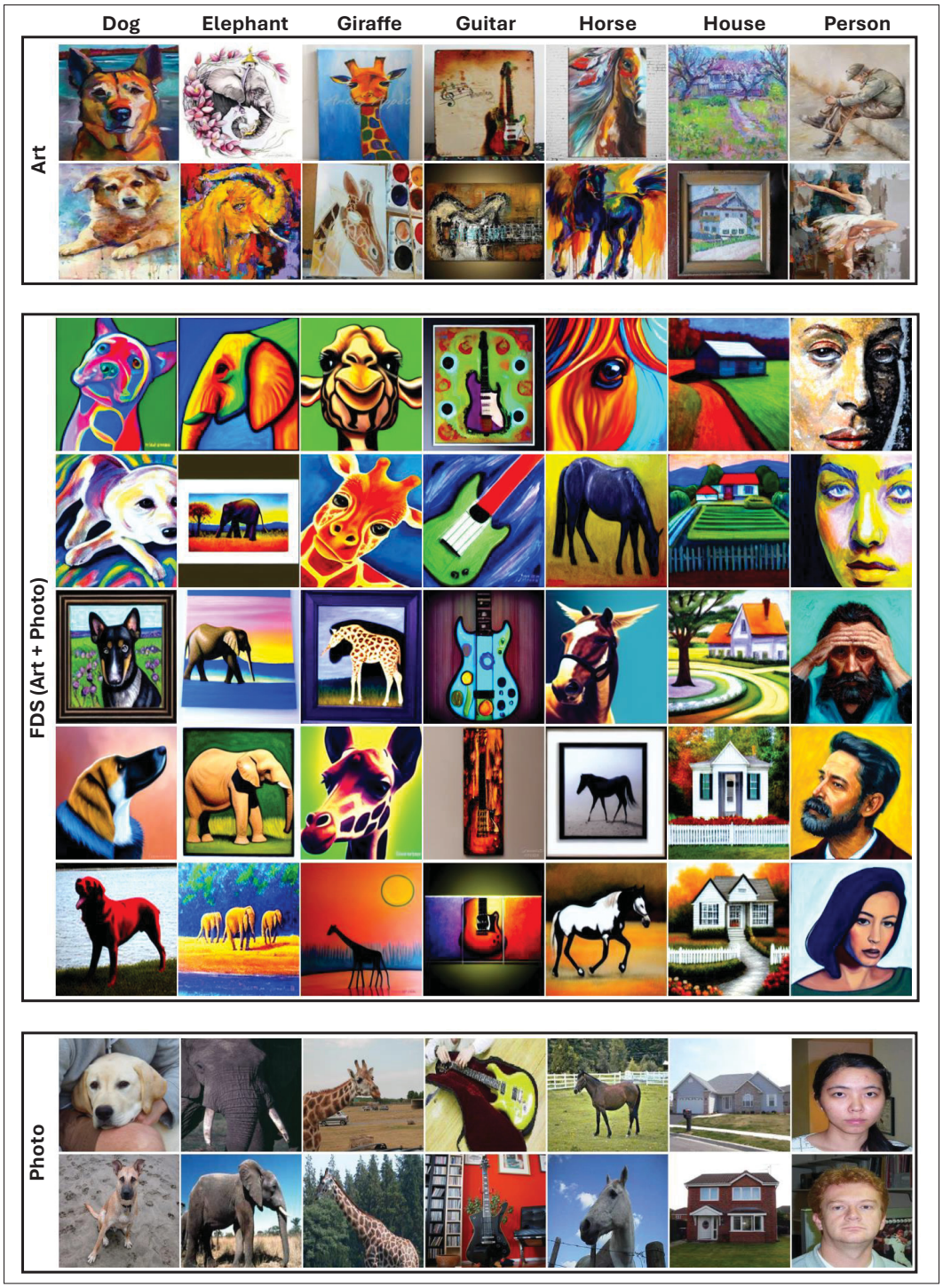

Figure-A II-9 Visual comparison of original “Art” and “Photo” samples from PACS with synthetic images generated by FDS (Art + Photo). The middle section illustrates how FDS combines visual elements from both domains, producing diverse, domain-bridging images

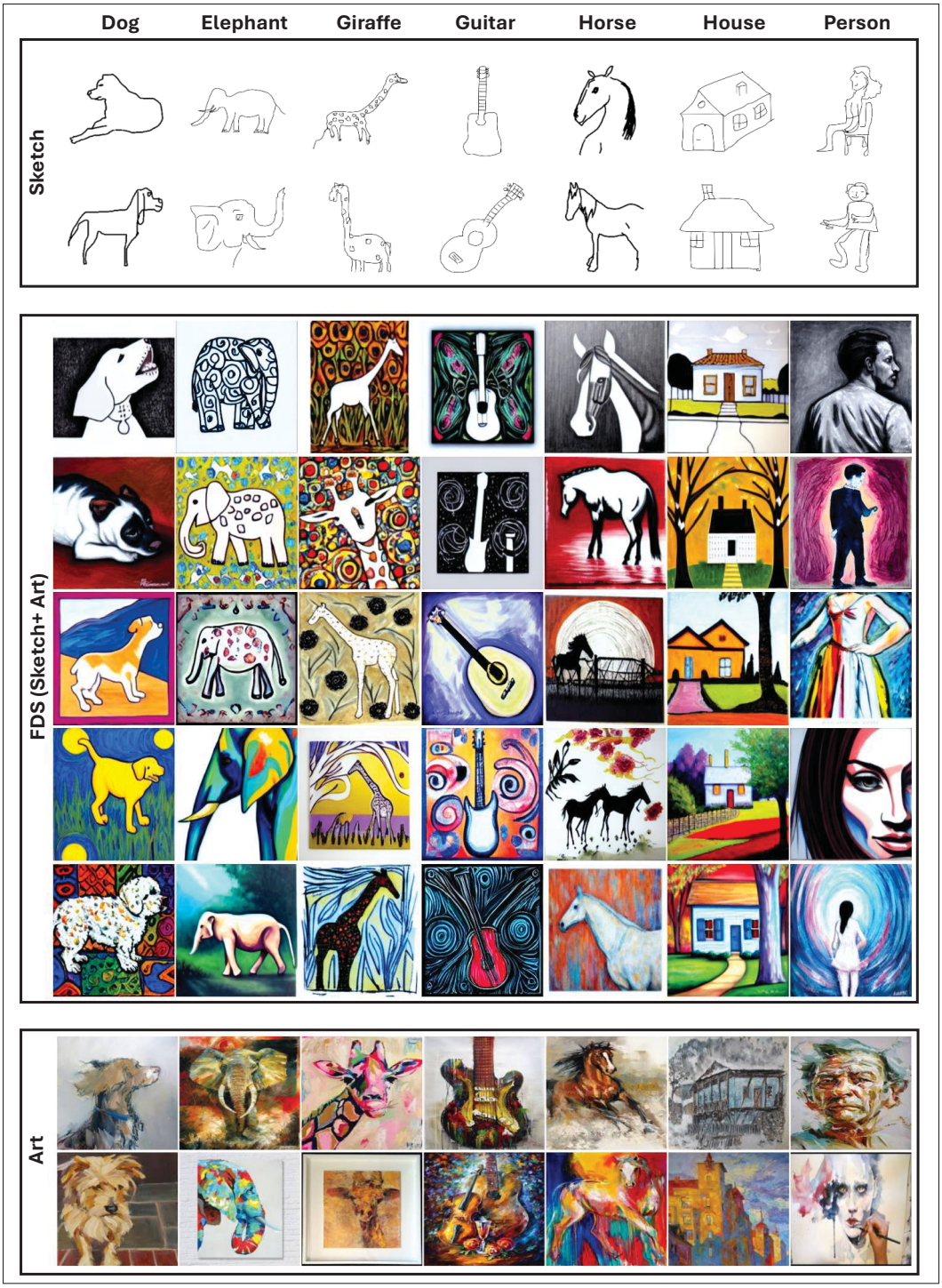

Figure-A II-10 Visual comparison of original “Sketch” and “Art” samples from PACS with synthetic images generated by FDS (Sketch + Art). The generated images in the middle section showcase a blend of artistic textures and sketched outlines

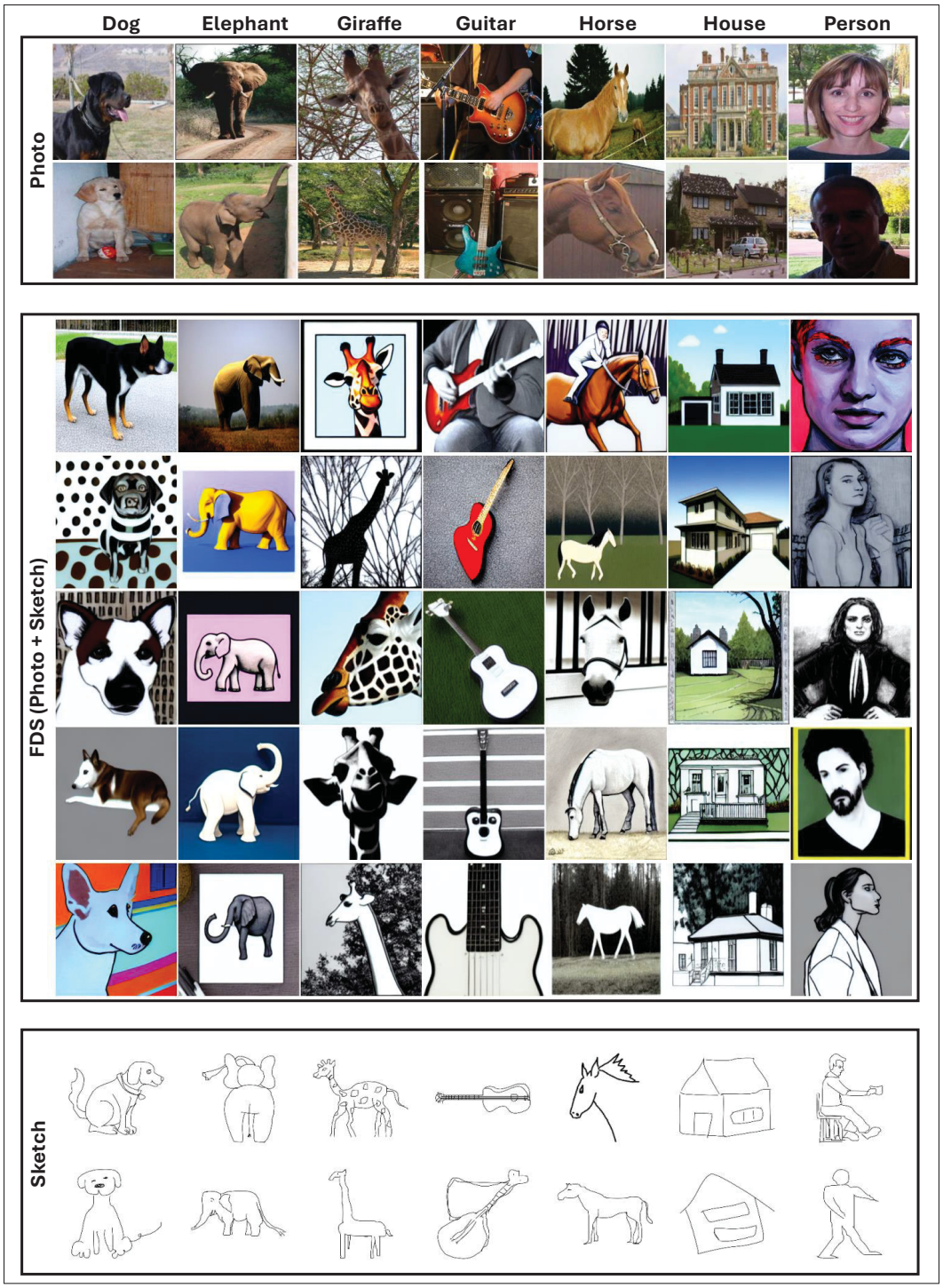

Figure-A II-11 Visual comparison of original “Photo” and “Sketch” samples from PACS with synthetic images generated by FDS (Photo + Sketch). The middle section demonstrates how FDS integrates the photorealistic details of the “Photo” domain with the elements of the “Sketch” domain

APPENDIX III

SUPPLEMENTARY MATERIAL FOR THE PAPER TITLED TEST-TIME ADAPTATION OF VISION-LANGUAGE MODELS FOR OPEN-VOCABULARY SEMANTIC SEGMENTATION

1. Proof of Proposition 1

Unbiasedness and Variance Bound Proposition 1 (Restated): Assume each per-template gradient $g_t(\theta) = \nabla_\theta[L_{\text{UAML}}(T_t) + L_{\text{ILE}}(T_t)]$ has variance bounded by σ^2. Then, the ensemble gradient, defined by $\nabla_\theta \mathcal{L}_{\text{final}} = \frac{1}{T}\sum_{t=1}^{T} g_t(\theta)$, is unbiased and satisfies the following variance bound:

$$\mathbb{E}[\nabla_\theta L_{\text{final}}] = \mathbb{E}[g_t(\theta)], \quad \text{Var}(\nabla_\theta L_{\text{final}}) = \frac{1}{T^2}\sum_{t=1}^{T} \text{Var}(g_t(\theta)) \leq \frac{\sigma^2}{T}.$$

Step 1: Unbiasedness. By linearity of expectation, we have:

$$\mathbb{E}\left[\nabla_\theta L_{\text{final}}\right] = \mathbb{E}\left[\frac{1}{T}\sum_{t=1}^{T} g_t(\theta)\right] = \frac{1}{T}\sum_{t=1}^{T} \mathbb{E}[g_t(\theta)] = \mathbb{E}[g_t(\theta)].$$

Hence, averaging the gradients from all T templates gives an unbiased estimate of the true gradient of the final loss.

Step 2: Variance Bound. Assuming independence with $\text{Var}(g_t) \leq \sigma^2$, we get

$$\text{Var}(\nabla_\theta L_{\text{final}}) = \text{Var}\left(\frac{1}{T}\sum_{t=1}^{T} g_t(\theta)\right) = \frac{1}{T^2}\sum_{t=1}^{T} \text{Var}(g_t(\theta)) \leq \frac{T\sigma^2}{T^2} = \frac{\sigma^2}{T}$$

Thus, the variance bound holds:

$$\text{Var}(\nabla_\theta L_{\text{final}}) \leq \frac{\sigma^2}{T}.$$

This completes the proof of Proposition 1.

2. Implementation Details

Unless otherwise noted, all experiments utilize NACLIP (Hajimiri *et al.*, 2025) with ViT-L/14 as the OVSS backbone. We adapt only the LayerNorm parameters within the vision encoder, amounting to approximately 0.02% of the model's total parameters. Our adaptation setup follows prior work in classification (Osowiechi *et al.*, 2024b; Hakim *et al.*, 2024), using the Adam optimizer with a fixed learning rate of 0.001 and 10 adaptation steps (iterations) across all experiments. Additionally we use batch of 2 images during adaptation. After each batch, model weights are explicitly reset to their original pre-adaptation values to ensure that each batch is adapted independently, without leveraging information from previously processed data. Following standard settings from (Radford *et al.*, 2021a), we use the default softmax temperature value of 100 in all experiments. All images are resized to 224×224 pixels. Due to the high resolution of images in the Cityscapes dataset, we split them into overlapping patches of size 224×224 pixels with an overlap of 112 pixels between patches. The segmentation predictions from these patches are aggregated to reconstruct the final, full-resolution segmentation maps. No image augmentation techniques are applied during either the adaptation or evaluation phases.

All experiments are conducted on NVIDIA V100 GPUs equipped with 32GB memory. We implement our approach using the PyTorch deep learning framework. To ensure statistical robustness and fairness in our comparisons, we repeat each experiment three times, reporting average performance along with standard deviation. We have provided detailed instructions and step-by-step scripts in our repository, clearly demonstrating how to generate datasets, perform the described test-time adaptations, and reproduce our reported results.

We adapt other baseline methods (TENT (Wang *et al.*, 2020a), TPT (Shu *et al.*, 2022), WATT (Osowiechi *et al.*, 2024b), CLIPArTT (Hakim *et al.*, 2024)) to work with spatial tokens. General adaptation hyperparameters (optimizer, learning rate, adaptation steps, and batch size) remain consistent with our setup. Method-specific hyperparameters or components are retained

as reported in their original implementations. Specifically, for WATT, we used the sequential version (WATT-S) with default values of $l = 2$ and $m = 5$, for CLIPArTT we used the default $k = 3$, and for TPT we used the 4 learnable tokens. Adapting these baseline methods to the segmentation task allowed us to systematically evaluate how various adaptation strategies—such as prompt tuning (TPT), pseudo-labeling (WATT, CLIPArTT), prompt refinement (CLIPArTT), and weight averaging (WATT)—perform in the context of open-vocabulary segmentation.

3. Template Details

Table III-1 lists the seven prompt templates used in our MLMP method. These templates were selected by the original CLIP authors and are general-purpose, not tailored to any specific image content. The CLIP repository also provides a full set of 80 prompt templates, which we use for larger-scale ablations.

More specifically, for the ablation in Figure 4.4 of the main paper, we vary the number of templates T as follows:

- $T = 1$: the default CLIP prompt "`a photo of a {class}`"
- $T = 3$: the first three templates $\{T^1, T^2, T^3\}$ in Table III-1
- $T = 7$: all seven templates $\{T^1, \ldots, T^7\}$ in Table III-1
- $T = 20$: the first 20 templates from the 80-template pool
- $T = 80$: the complete set of 80 CLIP templates

4. Dataset Details

This section details the datasets used in our experiments, along with the synthetic, real, and rendered domain shifts applied to evaluate robustness under distributional changes.

More specifically, we conduct experiments on a diverse set of segmentation benchmarks:

- **Pascal VOC (V20/V21)** (Everingham *et al.*, 2010a): Contains 20 foreground object classes (v20) with a background class (v21), widely used for benchmarking semantic segmentation tasks.

Table-A III-1 List of the seven prompt templates used in our MLMP method. These general-purpose templates, originally proposed by the CLIP authors, serve as diverse textual views of each class and are not tailored to specific datasets or domains

ID	Prompt template
T^1	`itap of a {class}`
T^2	`a bad photo of the {class}.`
T^3	`a origami {class}.`
T^4	`a photo of the large {class}.`
T^5	`a {class} in a video game.`
T^6	`art of the {class}.`
T^7	`a photo of the small {class}.`

- **Pascal Context (P59/P60)** (Mottaghi *et al.*, 2014): An extension of the Pascal VOC 2010 dataset, providing pixel-level annotations for more than 400 classes. Due to sparsity, a frequently used subset includes 59 object classes plus a background class, totaling 60 categories.
- **Cityscapes** (Cordts *et al.*, 2016): A large-scale dataset for semantic segmentation of urban street scenes. It comprises around 5,000 finely annotated images from 50 cities, recorded under various daylight conditions, featuring dynamic objects, varying layouts, and changing backgrounds, capturing significant natural domain shifts.
- **COCO-Stuff** (Caesar *et al.*, 2016): Extends the original COCO dataset by adding annotations for background categories ("stuff"), resulting in 171 classes—80 objects, and 91 stuff categories.
- **COCO-Object** (Lin *et al.*, 2014): A subset of the original COCO dataset, consisting exclusively of 80 object categories without background annotations.
- **ACDC** (Sakaridis *et al.*, 2021): Includes pixel-level annotations for 19 semantic classes, covering diverse driving scenes under fog, night, rain and snow conditions.
- **GTA-V** (Richter *et al.*, 2016): Contains 24,966 synthetic images with pixel-accurate semantic labels; one version aligns to 34/19 classes compatible with real-world segmentation benchmarks.

In addition to the original versions of each dataset—some of which already reflect natural domain shifts (e.g., Cityscapes)—we further assess the robustness of all methods by applying synthetic corruptions. Inspired by ImageNet-C (Hendrycks & Dietterich, 2019a), we apply **15 synthetic corruptions** to evaluate the robustness of segmentation models under various perturbations. These corruptions include Gaussian noise, shot noise, impulse noise, defocus blur, glass blur, motion blur, zoom blur, snow, frost, fog, brightness variations, contrast variations, elastic transformations, pixelation, and JPEG compression. Each corruption is applied at severity level 5, representing the most challenging scenario.

We resize images from all datasets to 224×224 pixels. Due to the high resolution of Cityscapes and ACDC images, we process them as overlapping patches of size 224×224 pixels (with overlaps of 112 pixels). Predictions for these patches are subsequently aggregated to produce the final segmentation maps.

This extended benchmark comprises a total of **87 distinct test scenarios**, encompassing synthetic, real, and rendered domain shifts. It provides a rigorous and comprehensive evaluation protocol that captures variations in resolution, scene diversity, object size, and semantic granularity.

5. Computational Complexity

In this section, we provide an exhaustive analysis of each test-time adaptation method's resource footprint by measuring: (i) latency, (ii) floating-point operations (FLOPs), (iii) peak GPU memory usage, and (iv) number of learnable parameters. For a fair comparison, all measurements were performed on a single test sample using the same NVIDIA V100 (32 GB) GPU. The results are summarized in Table III-2.

We compare TENT, TPT, CLIPArTT, WATT, and our MLMP across all four complexity metrics. In TENT, WATT, CLIPArTT, and MLMP, only the LayerNorm parameters of the vision encoder are updated during adaptation, whereas TPT introduces additional learnable tokens at the input of the text encoder. Additionally TENT, WATT, and MLMP use a fixed sets of text features without a need for recalculating them during adaptation/evaluation, so all prompt templates can

Table-A III-2 Computational-complexity comparison across methods. For GFLOPs, only the forward-pass cost in adaptation and evaluation is measured; by common practice, the cost of back-propagation in adaptation phase can be approximated as twice the forward cost. A ✓ in the second column indicates that all text information are encoded *once* and cached. A dagger (†) indicates that TPT adds additional parameters beyond the original network

Method	One-time Text Encoder	Time (sec.) ↓		GFLOPs ↓		Max Memory (MB) ↓	Learn. Params (Ratio) ↓
		Adapt	Eval	Adapt	Eval		
TENT	✓	0.462	0.018	79.1	79.1	2,068.4	102,400 (0.02%)
TPT	✗	0.445	0.031	275.6	275.6	2,583.1	3,072† (<0.01%)
CLIPArTT	✗	3.494	0.031	1,755.5	275.6	8,928.5	102,400 (0.02%)
WATT	✓	5.197	0.018	553.9	79.1	7,232.4	716,800 (0.17%)
MLMP (ours)	✓	0.541	0.041	82.4	82.9	2,093.9	102,400 (0.02%)

be encoded once (✓ in the "One-time Text Encoder" column) and then reused throughout both adaptation and evaluation phases. In contrast, TPT and CLIPArTT modify prompt embeddings or refine text templates during adaptation, requiring multiple forward passes through the text encoder.

In terms of latency, MLMP completes both adaptation and evaluation in just 0.582 ms, which is only marginally slower than TENT (0.480 ms) but substantially faster than WATT (5.215 ms) and CLIPArTT (3.525 ms). Despite leveraging multi-level fusion and multiple prompt templates, MLMP maintains a lightweight computational profile with only 82.4 GFLOPs and 82.9 GFLOPs for adaptation and evaluation, respectively—comparable to TENT and significantly lower than all other baselines. Furthermore, MLMP's peak memory usage is among the lowest at 2,093.9 MB, and like TENT, it updates only 0.02% of the model parameters. These results highlight MLMP's efficiency: it delivers rich representational capacity through multi-level and multi-prompt integration by boosting the results significantly (as shown in Table 4.5 of the main paper), yet remains almost as lightweight as the simplest baseline.

6. Performance with a Single Test Sample

As shown in Table III-3, while TENT yields improvements on the original dataset, it leads to a performance drop on V20-C. In contrast, our method, MLMP, consistently improves performance

over the no adaptation baseline, with gains of 8.77% on the otiginal data and 9.40% on the average across corruptions.

Table-A III-3 mIoU performance comparison when using a single test sample for V20 dataset

OVSS Backbone: NACLIP	Adaptation Method		
Dataset: V20	No Adapt.	TENT	MLMP
Original	75.91	76.20	**84.68**
Gaussian noise	62.89	62.59	**71.92**
Shot noise	66.26	65.45	**75.78**
Impulse noise	63.16	63.34	**72.35**
Defocus blur	72.59	71.37	**80.91**
Glass blur	71.44	69.95	**79.29**
Motion blur	73.10	72.55	**81.64**
Zoom blur	59.03	60.64	**69.99**
Snow	71.49	70.03	**80.64**
Frost	65.38	65.96	**74.33**
Fog	70.69	69.39	**80.77**
Brightness	74.95	74.73	**84.58**
Contrast	71.51	69.74	**79.78**
Elastic transform	62.86	65.09	**75.32**
Pixelate	77.28	75.84	**85.63**
JPEG compression	72.59	70.85	**83.20**
V20-C Average	69.01	68.50	**78.41**

7. Generalization Across Model Variants

To evaluate the robustness and generality of our method, we conduct a series of experiments using different model configurations within the segmentation pipeline. Specifically, we assess how MLMP performs when (i) changing the vision transformer backbone, (ii) switching between different open-vocabulary semantic segmentation (OVSS) formulations, and (iii) adopting an entirely different vision–language model. These experiments demonstrate that our method maintains consistent improvements across a wide range of architectural and algorithmic configurations, confirming its flexibility and transferability.

7.1 Comparison Across ViT Backbones

To evaluate whether our method generalizes across different vision transformer backbones, we replicate the main experiments using ViT-B/16 and ViT-B/32 in place of the default ViT-L/14 model. These backbones represent lighter configurations, with fewer parameters and larger patch (32 and 16). As shown in Tables III-4 and III-5, MLMP continues to provide substantial improvements over all baselines across both configurations, on both clean data (V20 Original) and under severe synthetic corruptions. These results confirm that the benefits of our multi-level and multi-prompt adaptation strategy are not tied to model scale or specific architectural configurations, and remain effective even in lower-resolution settings.

Table-A III-4 Performance comparison of test-time adaptation methods using the ViT-B/16 backbone with NaCLIP as the OVSS model. Results are reported as mIoU scores on the V20 dataset (original) and 15 corruption types. MLMP achieves the highest performance across all settings, demonstrating strong robustness even with a smaller backbone

OVSS Backbone: NACLIP	Adaptation Method					
Dataset: V20	No Adapt.	TENT	TPT	WATT	CLIPArTT	MLMP
Original	77.62	79.15± 0.06	77.63± 0.01	43.18± 0.07	72.70± 0.17	**84.18± 0.07**
Gaussian noise	48.00	52.04± 0.12	48.11± 0.00	18.83± 0.14	33.80± 0.52	**61.67± 0.00**
Shot noise	52.49	55.56± 0.16	52.41± 0.00	19.86± 0.12	37.62± 0.33	**64.97± 0.10**
Impulse noise	49.51	52.87± 0.11	49.41± 0.01	19.12± 0.23	35.92± 0.05	**61.15± 0.09**
Defocus blur	68.03	69.85± 0.04	67.88± 0.01	37.54± 0.17	56.77± 0.19	**76.71± 0.02**
Glass blur	62.17	65.14± 0.17	62.45± 0.00	31.12± 0.08	47.86± 0.15	**72.62± 0.04**
Motion blur	69.56	71.93± 0.02	69.54± 0.01	36.89± 0.15	58.64± 0.29	**77.08± 0.03**
Zoom blur	47.34	52.30± 0.19	47.38± 0.01	22.83± 0.12	33.52± 0.20	**59.05± 0.20**
Snow	60.88	64.38± 0.05	61.24± 0.00	27.68± 0.21	49.91± 0.38	**71.41± 0.15**
Frost	55.45	58.38± 0.14	55.44± 0.00	29.66± 0.23	46.94± 0.04	**67.42± 0.09**
Fog	67.07	70.01± 0.02	67.07± 0.00	35.84± 0.23	59.98± 0.09	**76.32± 0.02**
Brightness	73.33	75.14± 0.17	73.23± 0.00	40.87± 0.09	67.30± 0.33	**82.14± 0.03**
Contrast	60.30	63.30± 0.04	60.20± 0.00	29.68± 0.12	48.42± 0.55	**70.02± 0.04**
Elastic transform	50.14	54.83± 0.01	50.00± 0.01	23.32± 0.14	43.67± 0.00	**63.65± 0.07**
Pixelate	75.48	77.44± 0.03	75.31± 0.01	42.26± 0.08	67.33± 0.08	**83.29± 0.01**
JPEG compression	69.17	70.97± 0.07	69.15± 0.01	34.94± 0.07	60.30± 0.54	**79.35± 0.12**
V20-C Average	60.59	63.61	60.59	30.03	49.87	**71.12**

7.2 Comparison Across OVSS Methods

Our method is designed to be flexible and agnostic to the underlying open-vocabulary semantic segmentation (OVSS) formulation. While all main experiments in the paper use NaCLIP as the

Table-A III-5 Performance comparison of test-time adaptation methods using the ViT-B/32 backbone with NaCLIP as the OVSS model. Results are reported as mIoU scores on the V20 dataset (original) and 15 corruption types. MLMP achieves the highest performance across all settings, demonstrating strong robustness even with a smaller backbone and smaller patch size

OVSS Backbone: NACLIP	Adaptation Method					
Dataset: V20	No Adapt.	TENT	TPT	WATT	CLIPArTT	MLMP
Original	72.43	72.83 ± 0.01	72.52 ± 0.00	50.71 ± 0.10	67.67 ± 0.15	**79.95 ± 0.01**
Gaussian noise	47.59	49.30 ± 0.18	47.43 ± 0.00	29.26 ± 0.09	37.36 ± 0.17	**59.27 ± 0.16**
Shot noise	51.80	54.43 ± 0.21	51.83 ± 0.00	31.82 ± 0.15	41.07 ± 0.21	**63.73 ± 0.12**
Impulse noise	48.79	51.59 ± 0.11	48.79 ± 0.00	30.31 ± 0.04	37.65 ± 0.17	**58.81 ± 0.04**
Defocus blur	60.23	61.86 ± 0.07	60.17 ± 0.00	36.54 ± 0.20	48.97 ± 0.21	**66.70 ± 0.08**
Glass blur	54.59	56.73 ± 0.09	54.80 ± 0.00	28.70 ± 0.10	37.96 ± 0.26	**65.34 ± 0.03**
Motion blur	59.53	60.49 ± 0.02	59.65 ± 0.00	35.80 ± 0.06	47.61 ± 0.52	**66.71 ± 0.08**
Zoom blur	38.66	41.07 ± 0.10	38.62 ± 0.00	21.72 ± 0.14	24.39 ± 0.18	**49.53 ± 0.08**
Snow	49.17	51.01 ± 0.08	48.88 ± 0.01	27.55 ± 0.17	40.30 ± 0.00	**62.30 ± 0.09**
Frost	47.60	50.18 ± 0.09	47.53 ± 0.00	28.48 ± 0.15	39.23 ± 0.12	**60.22 ± 0.07**
Fog	56.25	59.47 ± 0.08	56.22 ± 0.00	35.76 ± 0.09	47.63 ± 0.29	**68.18 ± 0.04**
Brightness	68.03	68.94 ± 0.04	67.95 ± 0.00	46.11 ± 0.02	61.07 ± 0.25	**76.90 ± 0.09**
Contrast	48.79	50.67 ± 0.07	48.88 ± 0.02	29.54 ± 0.06	36.64 ± 0.05	**58.81 ± 0.11**
Elastic transform	52.73	55.59 ± 0.20	52.75 ± 0.00	29.86 ± 0.21	44.84 ± 0.18	**65.39 ± 0.09**
Pixelate	68.61	69.15 ± 0.01	68.65 ± 0.01	44.57 ± 0.17	60.30 ± 0.41	**77.28 ± 0.04**
JPEG compression	63.86	65.12 ± 0.05	63.87 ± 0.00	42.85 ± 0.11	55.44 ± 0.08	**74.40 ± 0.09**
V20-C Average	54.42	56.37	54.40	33.26	44.03	**64.97**

OVSS baseline—paired with No Adapt., TENT, TPT, CLIPArTT, WATT, and our MLMP—we further evaluate the generality of our approach by applying MLMP to two alternative OVSS methods.

Specifically, we consider the original CLIP (Radford *et al.*, 2021a), adapted for pixel-wise segmentation via patch-level similarity, and SCLIP (Wang *et al.*, 2025), which incorporates spatial priors into the vision-language matching process. As shown in Table III-6, applying MLMP on top of these OVSS baselines yields consistent improvements across both clean and corrupted settings.

7.3 Generalization to Emerging VLMs

To further assess the generality of our approach, we evaluate MLMP on SigLIP v2 (Tschannen *et al.*, 2025), one of the most recently introduced vision–language models. As shown in Table III-

Table-A III-6 Performance comparison of test-time adaptation methods using the original CLIP (Radford *et al.*, 2021a) and SCLIP (Wang *et al.*, 2025) as the OVSS model with a ViT-B/16 backbone. MLMP consistently improves performance across all settings, highlighting its strong generalization capabilities

OVSS: **CLIP** (Radford *et al.*, 2021a)	Adaptation Method		
Dataset: V20	No Adapt.	TENT	MLMP
Original	33.11	51.36 ± 0.00	**61.47 ± 0.01**
Gaussian noise	22.74	37.78 ± 0.11	**49.35 ± 0.02**
Shot noise	23.67	38.86 ± 0.20	**50.81 ± 0.05**
Impulse noise	22.36	36.63 ± 0.15	**47.81 ± 0.17**
Defocus blur	31.83	48.33 ± 0.13	**58.25 ± 0.04**
Glass blur	28.60	46.59 ± 0.29	**56.02 ± 0.02**
Motion blur	32.62	50.61 ± 0.07	**59.55 ± 0.31**
Zoom blur	23.42	39.43 ± 0.06	**47.65 ± 0.10**
Snow	28.38	48.05 ± 0.04	**55.89 ± 0.04**
Frost	26.20	45.33 ± 0.04	**53.56 ± 0.17**
Fog	28.66	46.98 ± 0.05	**56.62 ± 0.02**
Brightness	33.71	51.77 ± 0.24	**60.96 ± 0.08**
Contrast	26.06	42.86 ± 0.08	**53.55 ± 0.04**
Elastic transform	27.12	45.92 ± 0.11	**51.02 ± 0.21**
Pixelate	33.32	51.11 ± 0.00	**61.22 ± 0.09**
JPEG compression	31.64	49.76 ± 0.04	**59.79 ± 0.06**
V21-C Average	28.02	45.33	**54.80**

OVSS: **SCLIP** (Wang *et al.*, 2025)	Adaptation Method		
Dataset: V20	No Adapt.	TENT	MLMP
Original	78.20	79.12 ± 0.05	84.91 ± 0.01
Gaussian noise	45.65	49.72 ± 0.07	61.20 ± 0.09
Shot noise	50.21	54.09 ± 0.15	64.06 ± 0.09
Impulse noise	47.05	50.62 ± 0.05	59.73 ± 0.14
Defocus blur	66.40	66.44 ± 0.05	75.03 ± 0.07
Glass blur	62.01	64.79 ± 0.22	72.22 ± 0.27
Motion blur	68.95	70.81 ± 0.02	76.23 ± 0.10
Zoom blur	45.12	48.50 ± 0.11	57.84 ± 0.16
Snow	60.61	64.60 ± 0.20	71.70 ± 0.06
Frost	56.14	58.43 ± 0.03	68.57 ± 0.02
Fog	68.34	70.03 ± 0.17	76.57 ± 0.06
Brightness	74.03	74.62 ± 0.11	82.91 ± 0.04
Contrast	58.98	61.69 ± 0.08	69.57 ± 0.10
Elastic transform	52.29	56.45 ± 0.02	65.05 ± 0.14
Pixelate	75.56	76.66 ± 0.09	82.99 ± 0.00
JPEG compression	68.38	69.85 ± 0.12	78.69 ± 0.17
V21-C Average	59.98	62.49	**70.82**

7, MLMP continues to deliver consistent improvements across both natural and corrupted datasets, indicating that the core components of our method—multi-level and multi-prompt aggregation—generalize well beyond CLIP-based architectures.

8. Effect of Longer Prompts

Recent studies such as Long-CLIP (Zhang, Zhang, Dong, Zang & Wang, 2024a) and TULIP (Najdenkoska *et al.*, 2024) have explored extending the text length capability of vision–language models, suggesting that richer linguistic descriptions may improve alignment. Motivated by this, we investigated whether incorporating longer and more descriptive prompts could further enhance MLMP's performance.

To this end, we generated extended templates derived from our original seven class-agnostic prompts using ChatGPT, together with extended class names where each category was expressed in full natural language. For example, "a photo of the large {class}" becomes "a photograph of a very large {class}, where the size of the object dominates the frame or is shown in contrast to smaller elements," while "aeroplane" becomes "a powered flying vehicle with fixed wings and engines, designed to transport people or cargo through the air over long distances."

Table-A III-7 Performance comparison of test-time adaptation using SigLIP-2 as the OVSS model. Results are reported as mIoU scores on the V20 dataset. MLMP shows improvement over the baseline in most corruption types

OVSS Backbone: SigLIP-2	Adaptation Method	
Dataset: V20	No Adapt.	MLMP
Original	66.62	**67.88** ± **0.32**
Gaussian noise	39.74	**41.05** ± **0.34**
Shot noise	40.24	**43.56** ± **0.07**
Impulse noise	40.76	**42.43** ± **0.22**
Defocus blur	47.15	**50.41** ± **0.02**
Glass blur	38.64	**40.60** ± **0.55**
Motion blur	42.30	**42.98** ± **0.39**
Zoom blur	27.29	**29.20** ± **0.03**
Snow	3.85	**4.84** ± **0.03**
Frost	21.32	**23.12** ± **0.02**
Fog	70.73	**70.86** ± **0.04**
Brightness	18.45	**19.14** ± **0.27**
Contrast	70.88	**70.35** ± **0.06**
Elastic transform	38.41	**38.71** ± **0.05**
Pixelate	57.85	**59.14** ± **0.07**
JPEG compression	56.35	**58.55** ± **0.04**
V20-C Average	40.93	**42.33**

As summarized in Table III-8, all experiments were conducted using the Long-CLIP (Zhang *et al.*, 2024a) backbone. We evaluate three text variants: (1) the default short templates used throughout our main experiments, (2) *Extended Templates*, where each template is replaced with a longer and more descriptive sentence, and (3) *Extended Classes*, where category names are expanded into full natural-language descriptions. For each variant, we report results with and without our MLMP adaptation (+ *MLMP*). All scores correspond to mIoU.

Overall, these results indicate that MLMP's adaptation mechanism effectively handles both concise and extended textual inputs without overfitting to prompt verbosity. We include this analysis to provide a clearer picture of MLMP's behavior under longer or more descriptive prompts.

9. Effect of Adaptation Iterations

Table-A III-8 Effect of longer prompts and extended class descriptions on segmentation performance (mIoU). All experiments use Long-CLIP as the OVSS model. Each pair of columns shows results without and with MLMP adaptation

OVSS: Long-CLIP	Adaptation Setting					
Dataset	Default Templates		Extended Templates		Extended Classes	
	No Adapt.	+ MLMP	No Adapt.	+ MLMP	No Adapt.	+ MLMP
V20	39.64	54.45 ± 0.09	35.55	53.43 ± 0.06	25.66	43.92 ± 0.04
V20-C Average	29.36	48.71	26.92	48.90	21.09	39.01
V21	16.83	19.54 ± 0.02	15.87	19.15 ± 0.04	11.27	15.25 ± 0.03
V21-C Average	13.64	18.57	12.98	18.27	10.23	14.66

The number of adaptation iterations in TTA varies across the literature. While some studies evaluate performance under a single adaptation step, others perform multiple iterations to improve convergence. In our main experiments, we followed the common 10-iteration setup adopted in prior TTA works (Wang *et al.*, 2020a; Osowiechi *et al.*, 2024b; Hakim *et al.*, 2024), as it provides a consistent protocol for comparing vision–language adaptation methods such as CLIPArTT and WATT. For fairness, we used identical hyperparameters across all methods.

To further examine the influence of the iteration count, we also evaluated MLMP under a stricter *one-iteration* protocol, where only a single adaptation step is performed. As summarized in Table III-9, MLMP still improves over the baseline and outperforms TENT even with just one iteration, which demonstrates the effectiveness of our method even under a stricter setting.

Table-A III-9 Evaluation of MLMP under a single-iteration TTA protocol compared to standard baselines (mIoU). All experiments use NACLIP as the OVSS model

OVSS: NACLIP	Adaptation Method		
Dataset	No Adapt.	TENT	MLMP
V20	75.91	76.80 ± 0.01	**79.88 ± 0.02**
V20-C Average	69.01	70.15	**73.91**
V21	45.12	45.36 ± 0.00	**50.24 ± 0.01**
V21-C Average	40.75	41.02	**46.07**

10. Episodic vs. Online Adaptation

We further investigate the difference between *episodic* (reset-based) and *online* adaptation settings in test-time adaptation (TTA). In our main experiments, all methods—including MLMP and baselines—were evaluated under the episodic setup, where the model is reset to its initial weights after each batch. This protocol avoids cumulative error propagation and is widely used in prior TTA works (Wang *et al.*, 2020a; Hakim *et al.*, 2024; Osowiechi *et al.*, 2024b).

To analyze the impact of continuous updates, we also tested an *online* version of MLMP, where model parameters are updated sequentially without resets and carried over to the next batch. We used identical hyperparameters as in the episodic setting, and further experimented with lower learning rates and fewer adaptation steps for stability.

As shown in Table III-10, we observe that while the online variant improves over baselines, it still underperforms compared to the reset-based (episodic) setting. We believe this is primarily due to error accumulation and model drift, as documented in prior work showing that some recent classification TTA methods exhibit performance degradation in an online setting—even when the distribution is static (Zhao, Liu, Alahi & Lin, 2023). This issue is particularly pronounced in semantic segmentation, where dense, spatial predictions make the model more sensitive to incorrect updates, and entropy minimization can amplify early misclassifications. Over time, this leads the model to drift away from its well-initialized source parameters, reducing its ability to generalize across the target domain.

Episodic (reset) adaptation avoids the runaway effects of carrying over errors, which is why it tends to be safer [2]. In contrast, fully online adaptation—though attractive for its potential to continuously refine the model—must confront these challenges.

We believe that online adaptation in segmentation can be improved through several future directions: (1) introducing occasional resets or using exponential moving average (EMA) of model weights to limit drift and reduce accumulated errors; (2) filtering high-entropy samples or gradients, which is particularly important in segmentation due to its spatial granularity and sensitivity to local noise; and (3) incorporating an auxiliary self-supervised objective, such as rotation prediction [1], to provide a more reliable adaptation signal—though this may come

with increased computational cost. Notably, the fact that our method works effectively with very few test samples and does not rely on accumulating state makes it efficient for real-world applications.

Table-A III-10 Comparison of episodic (reset-based) and online adaptation settings for MLMP and TENT (mIoU). All experiments use NACLIP as the OVSS model

OVSS: NACLIP	Adaptation Setting				
Dataset	No Adapt.	Episodic: TENT	Episodic: MLMP	Online: TENT	Online: MLMP
V20	75.91	77.00 ± 0.04	**83.76 ± 0.00**	76.44 ± 0.21	79.19 ± 0.45
V20-C Average	69.01	69.33	**77.58**	69.32	71.77

11. Visualization of Layer-Wise Confidence Weights

In the main paper (Figure 4.3), we presented the layer-wise confidence weights for the V20, Cityscapes, and COCO-Stuff datasets. Here, we extend the analysis by visualizing the weights for four additional datasets: V21, P59, P60, and COCO-Object in Figure III-1. As with the earlier results, we plot the mean and standard deviation of the learned weights across layers under various corruption types.

The observed trends closely mirror those reported in the main paper. In particular, deeper layers consistently receive higher confidence scores, while lower and mid-level layers still contribute under corrupted conditions. This reweighting behavior reflects the adaptive nature of our fusion strategy, which dynamically emphasizes the most reliable representations depending on the dataset and corruption type. These results further support the robustness and generality of our layer-wise uncertainty-aware fusion mechanism across diverse segmentation benchmarks.

12. Visualization of Segmentation Maps

Figure III-2 presents qualitative segmentation results on the v20 dataset, comparing predictions from the non-adaptive baseline, TENT, and our MLMP method. MLMP demonstrates stronger spatial consistency and fewer semantic errors, particularly in challenging regions where both

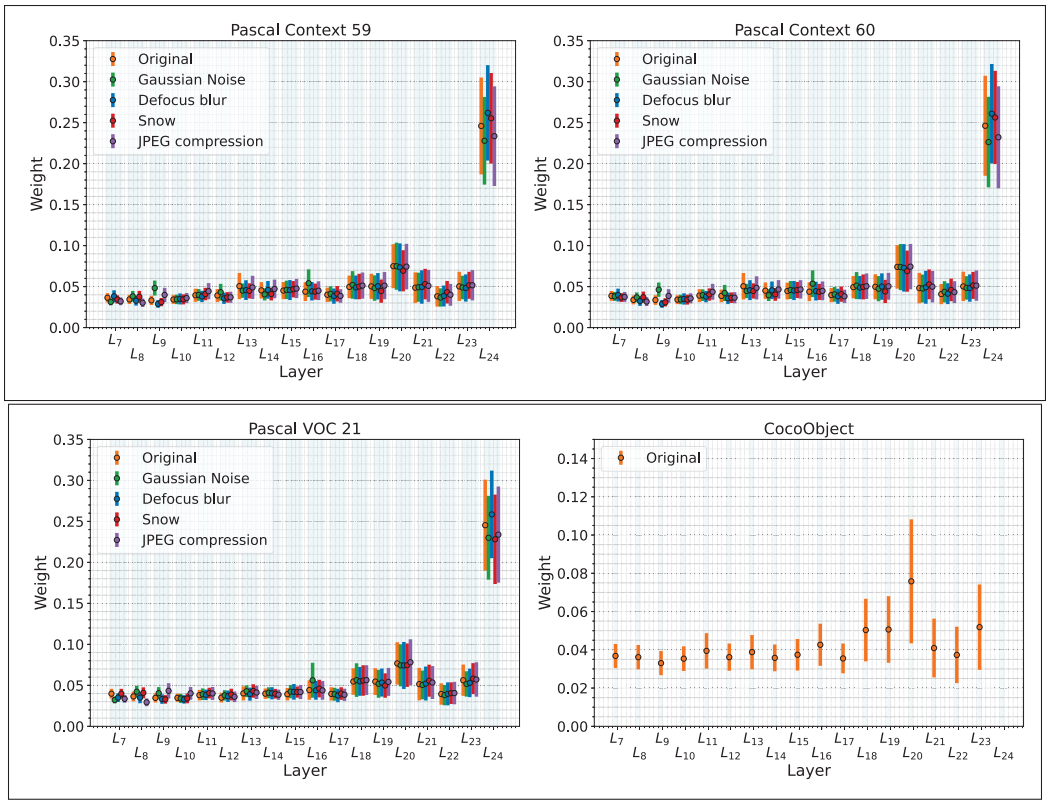

Figure-A III-1 Mean and standard deviation of weights of intermediate layers for several datasets

the baseline and TENT struggle. The integration of intermediate-layer supervision appears especially beneficial for refining small object boundaries and correcting fine-grained details.

In Figures III-3–III-11, we provide additional qualitative examples across a range of corruption types, using NaCLIP as the our OVSS backbone. These include both successful and failure cases under Gaussian noise, defocus blur, snow, and JPEG compression. Overall, the results illustrate the robustness of MLMP in mitigating noise-induced artifacts and improving prediction confidence, while also highlighting failure modes that warrant further investigation.

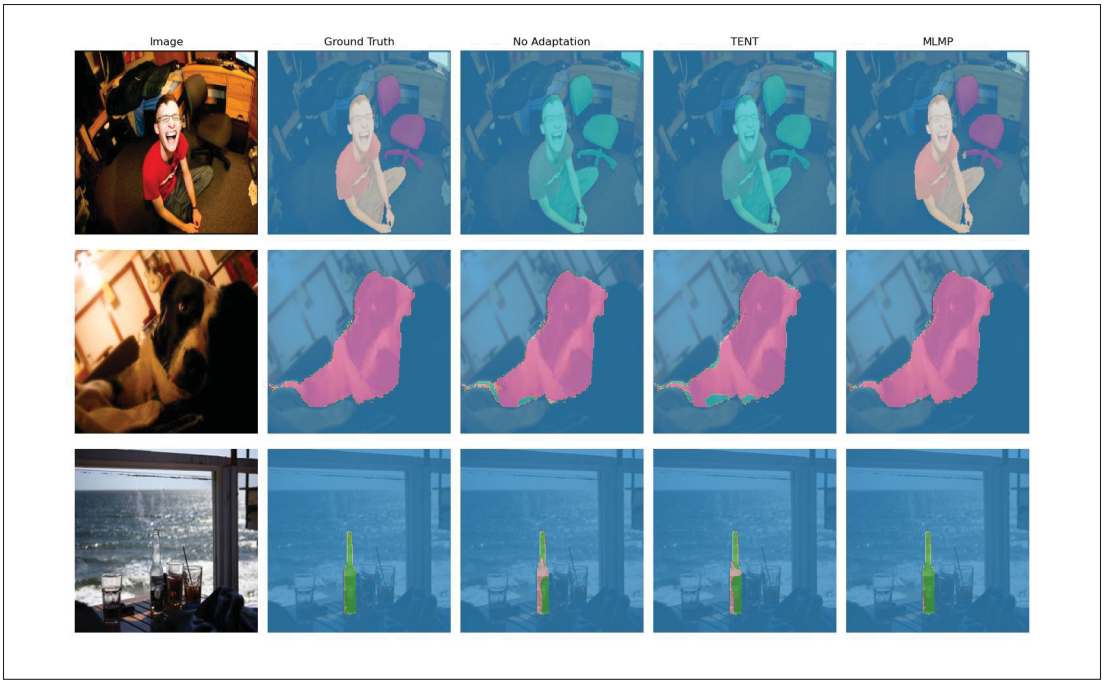

Figure-A III-2 Qualitative results for No Adapt., TENT and MLMP on V20 Original

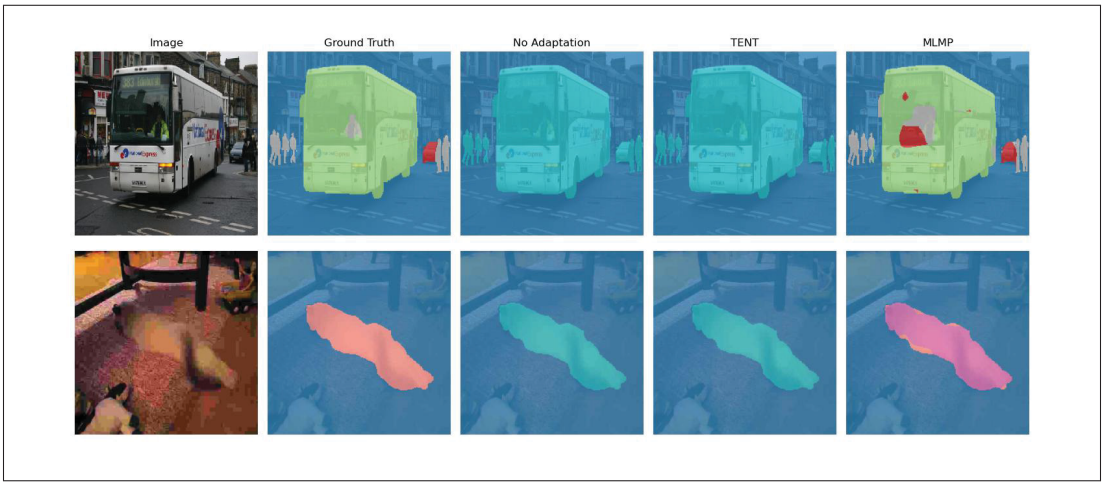

Figure-A III-3 Failed cases of MLMP on V20 Original

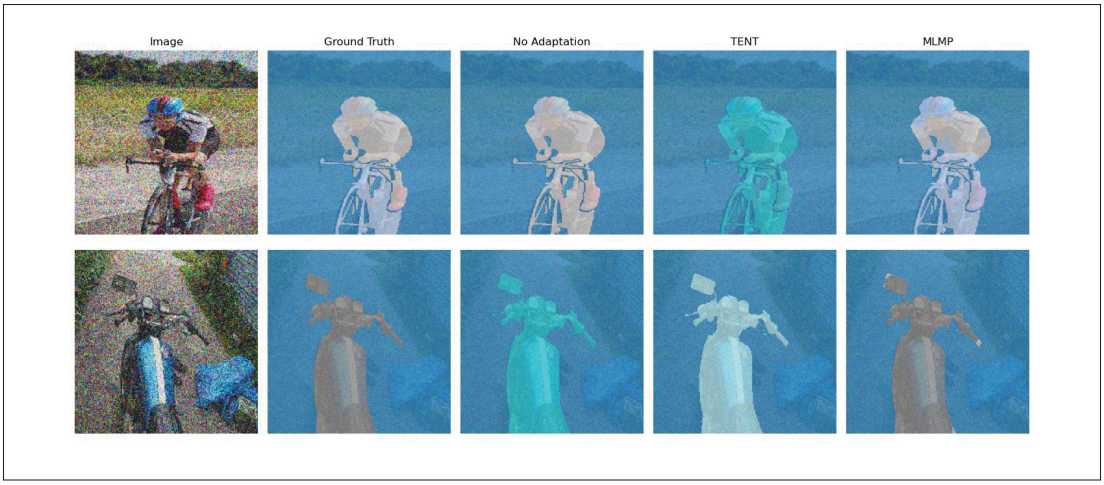

Figure-A III-4 Good cases of MLMP on V20 for V20 Gaussian noise corruption

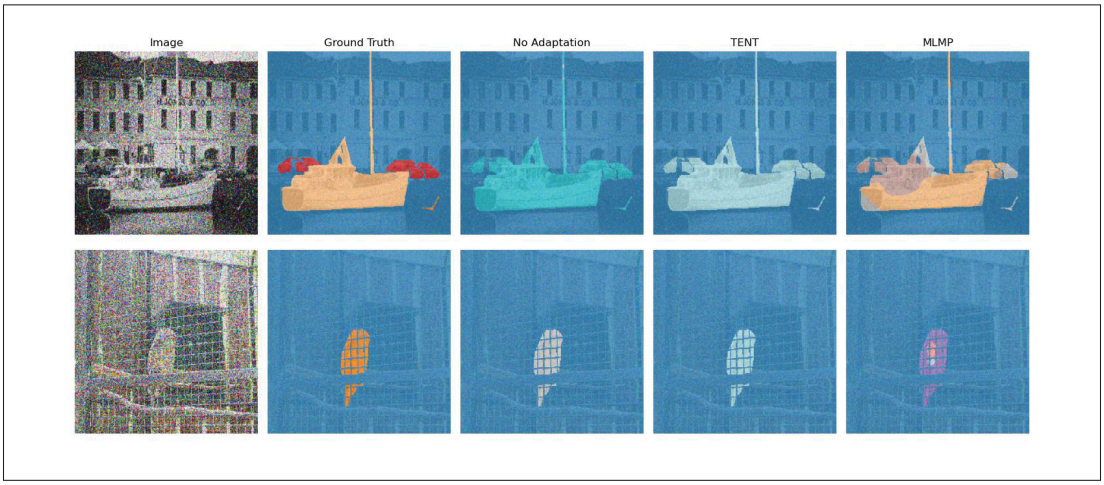

Figure-A III-5 Failed cases of MLMP on V20 Gaussian noise corruption.

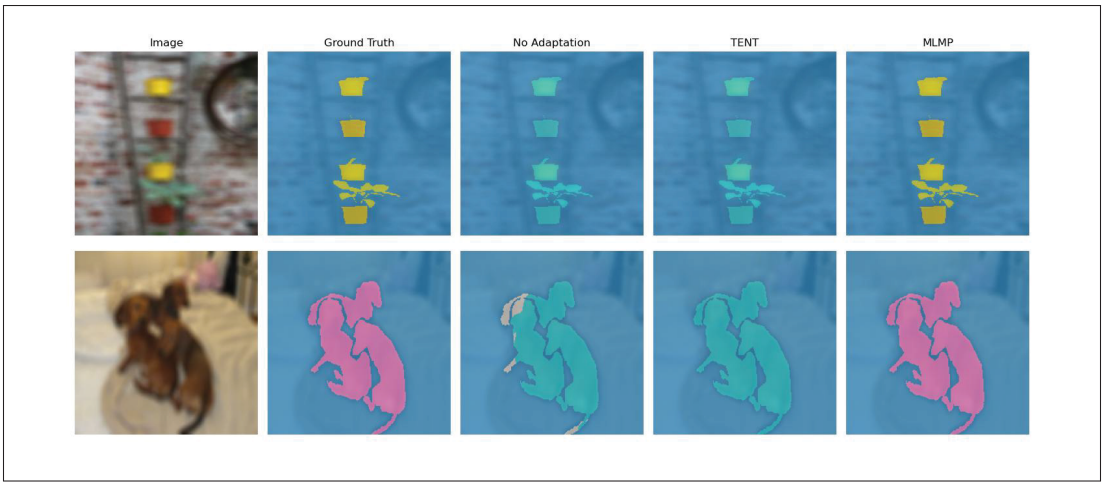

Figure-A III-6 Good cases of MLMP on V20 defocus blur corruption

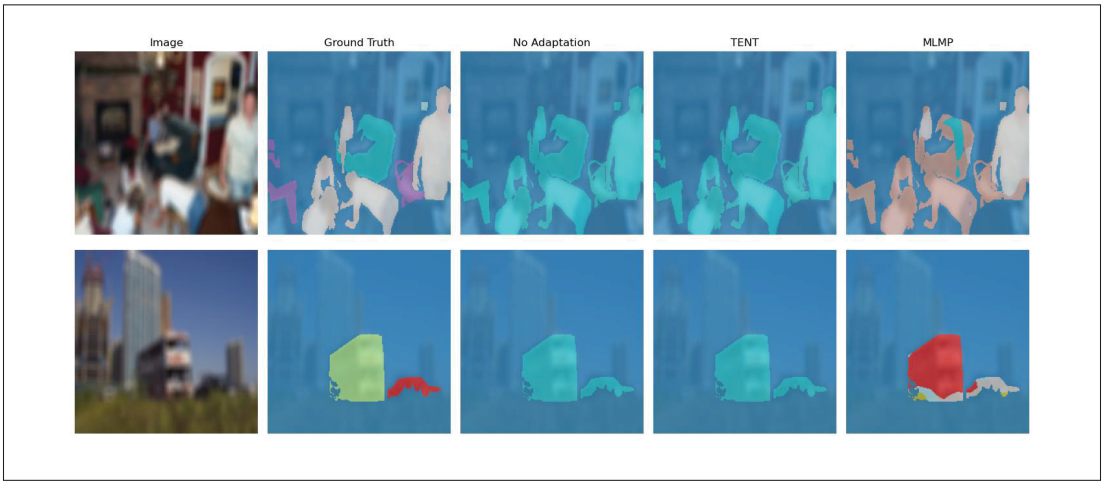

Figure-A III-7 Failed cases of MLMP on V20 defocus blur corruption

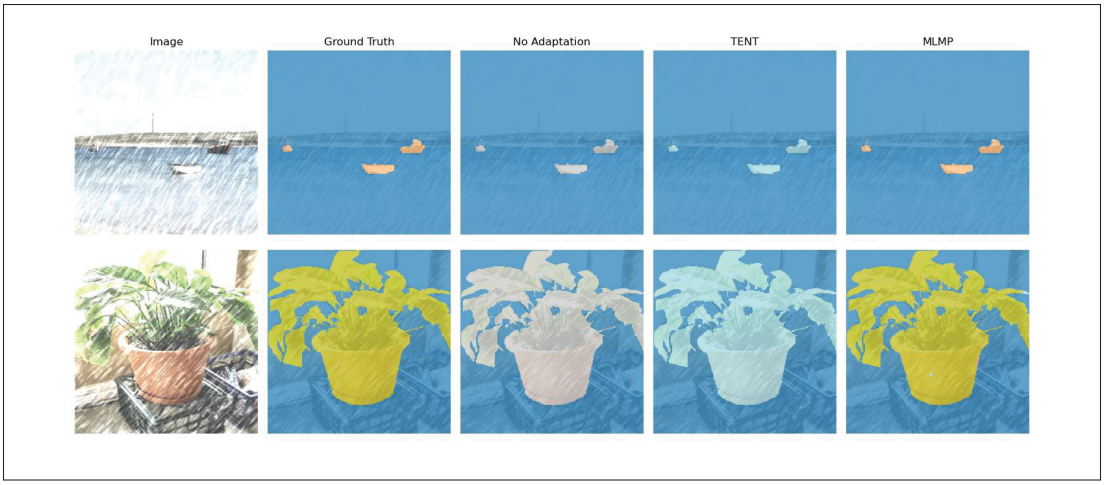

Figure-A III-8 Good cases of MLMP on V20 snow corruption

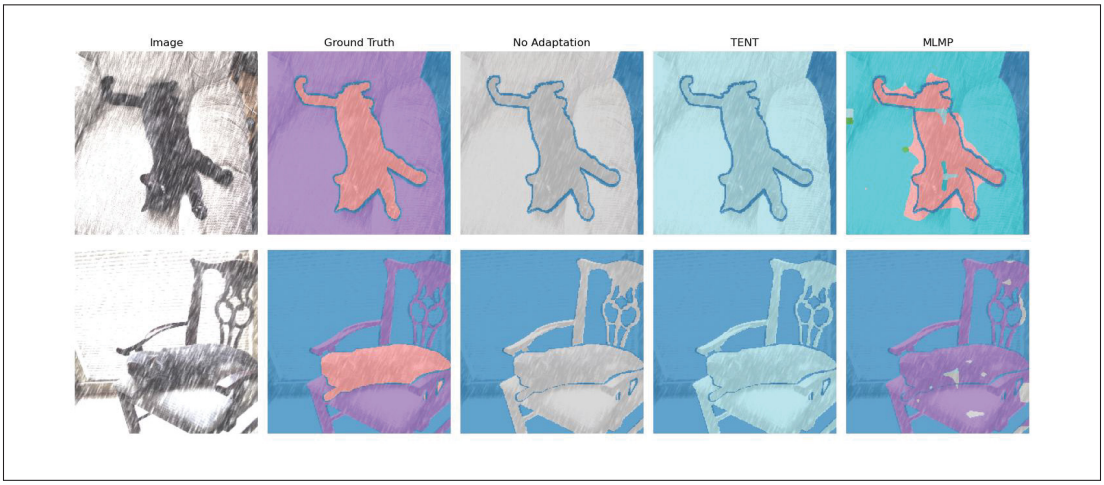

Figure-A III-9 Failed cases of MLMP on V20 snow corruption

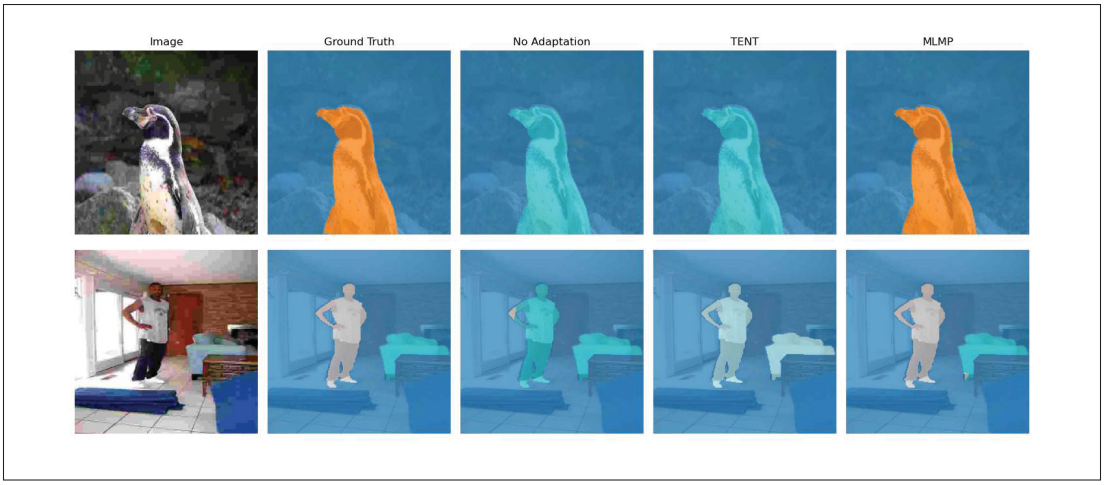

Figure-A III-10 Good cases of MLMP on V20 JPEG compression corruption

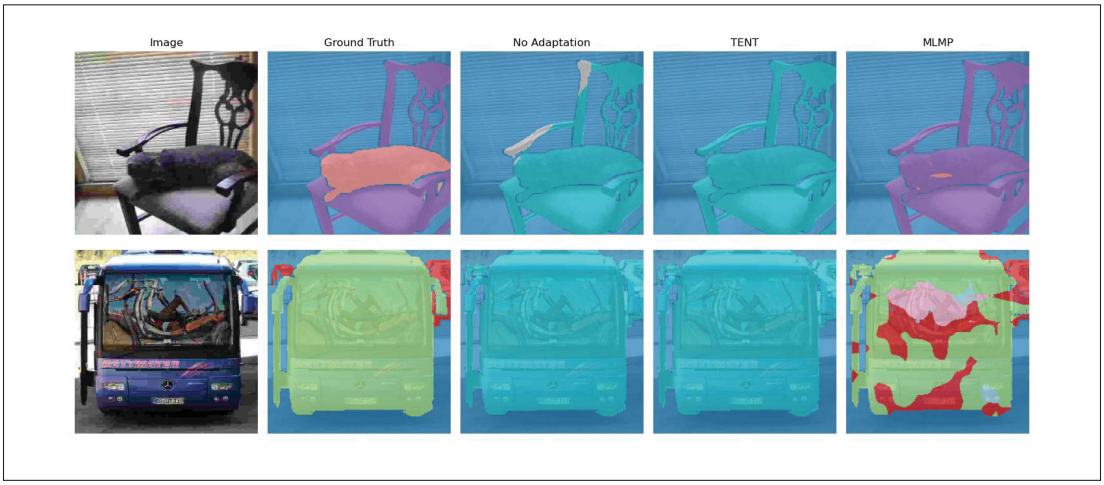

Figure-A III-11 Failed cases of MLMP on V20 JPEG compression corruption

13. Detailed Results of the Main Paper

This section provides complete versions of the experimental results that were summarized or partially reported in the main paper. It includes full tables for ablation studies and detailed comparisons with baseline adaptation methods across various datasets and evaluation settings. These results offer a more comprehensive view of the effectiveness and robustness of our proposed approach.

13.1 Ablation Studies

We provide here the detailed versions of the tables referenced in the ablation section of the main paper. Table III-11 shows the detailed results of using different layer ranges in the proposed multi-level adaptation. Table III-12 reports results for different strategies to integrate different prompt templates. Table III-13 presents a detailed analysis of the impact of the number of templates. Finally, Table III-14 presents results for the combination of all proposed components, showing individual and combined contributions.

13.2 Comparison with Alternative Adaptation Methods

This section presents comprehensive tables with the full experimental results corresponding to those summarized in the main paper. Table III-15 reports detailed results for the V21 dataset, Table III-16 for P59, Table III-17 for P60, and Table III-18 for the Cityscapes dataset. In addition to the datasets reported above, we also include detailed results on the **ACDC** (Sakaridis *et al.*, 2021) dataset, which is specifically designed to capture real-world adverse conditions such as fog, night, rain, and snow. It is worth mentioning that in ACDC, approximately half of the adverse-condition images have corresponding *reference (clean)* views captured at nearly the same locations. While these reference images are captured at approximately the same location as the shifted ones, the scene content may differ (e.g., different vehicles or objects present) due to real-world variability. Following our main protocol, we perform test-time adaptation independently on both reference and adverse views, and report the mIoU for each condition in

Table-A III-11 mIoU performance over different layer aggregation strategies. Maximum value in each row is highlighted

ViT-L/14 Layer Range	L_{24} (last)	$L_{23–24}$ (last two)	$L_{22–24}$ (last three)	$L_{19–24}$ (last 25%)	$L_{13–24}$ (last 50%)	$L_{7–24}$ (last 75%)	$L_{1–24}$ (all layers)
Original (V20)	77.00± 0.04	77.65± 0.02	77.66± 0.09	80.61± 0.05	80.50± 0.03	**81.67± 0.04**	78.79± 0.02
Gaussian noise	63.02± 0.06	64.41± 0.05	65.39± 0.13	66.88± 0.18	66.88± 0.02	**67.82± 0.01**	63.06± 0.09
Shot noise	65.88± 0.06	67.74± 0.11	68.43± 0.04	70.11± 0.09	70.40± 0.02	**70.52± 0.12**	64.88± 0.02
Impulse noise	64.17± 0.04	65.53± 0.09	66.19± 0.12	67.24± 0.17	67.15± 0.09	**68.10± 0.01**	62.09± 0.05
Defocus blur	72.06± 0.12	72.93± 0.19	72.84± 0.02	76.10± 0.16	76.37± 0.05	**78.78± 0.02**	77.56± 0.09
Glass blur	70.74± 0.07	71.73± 0.06	72.53± 0.15	74.20± 0.12	75.53± 0.05	**77.66± 0.05**	75.85± 0.02
Motion blur	73.50± 0.10	73.74± 0.08	74.62± 0.09	76.83± 0.07	77.50± 0.07	**79.39± 0.16**	77.22± 0.15
Zoom blur	61.36± 0.07	62.10± 0.01	62.56± 0.00	63.99± 0.20	64.59± 0.14	**66.50± 0.16**	64.79± 0.01
Snow	71.04± 0.05	72.09± 0.04	72.56± 0.02	74.47± 0.12	74.41± 0.01	**76.39± 0.02**	73.72± 0.07
Frost	67.01± 0.02	66.92± 0.02	67.23± 0.04	68.94± 0.01	70.03± 0.01	**71.17± 0.00**	68.23± 0.02
Fog	70.54± 0.07	71.44± 0.07	71.50± 0.05	74.50± 0.16	74.62± 0.08	**76.55± 0.06**	74.37± 0.03
Brightness	75.61± 0.02	76.27± 0.00	76.80± 0.04	79.16± 0.07	79.64± 0.16	**81.34± 0.04**	79.11± 0.05
Contrast	70.51± 0.04	71.72± 0.13	72.17± 0.00	74.58± 0.08	75.42± 0.09	**76.87± 0.06**	74.09± 0.00
Elastic transform	65.78± 0.05	66.08± 0.08	66.06± 0.06	68.62± 0.15	70.38± 0.09	**70.59± 0.14**	67.91± 0.12
Pixelate	76.95± 0.12	78.38± 0.02	78.46± 0.05	80.70± 0.05	81.11± 0.03	**83.02± 0.04**	81.53± 0.02
JPEG compression	71.84± 0.15	73.88± 0.11	74.40± 0.07	76.96± 0.02	77.67± 0.03	**78.73± 0.08**	75.87± 0.19
V20-C Average	69.33	70.33	70.78	72.89	73.45	**74.90**	72.02

Table III-19. The results show that MLMP consistently improves over TENT and the non-adapted baseline across all conditions, confirming its robustness to real, non-synthetic distribution shifts.

Table-A III-12 mIoU performance comparison for different strategies to integrate different prompt templates

Dataset: V20	Text	Params	Loss
Original	78.91± 0.07	74.46± 0.21	**79.70± 0.06**
Gaussian noise	66.27± 0.00	62.83± 0.04	**66.75± 0.01**
Shot noise	69.78± 0.10	67.10± 0.12	**70.03± 0.04**
Impulse noise	66.73± 0.03	64.57± 0.52	**67.88± 0.07**
Defocus blur	74.05± 0.10	70.28± 0.16	**74.31± 0.09**
Glass blur	73.31± 0.08	70.28± 0.34	**74.38± 0.36**
Motion blur	75.20± 0.09	71.56± 0.09	**75.72± 0.04**
Zoom blur	63.31± 0.11	61.29± 0.09	**64.61± 0.01**
Snow	73.78± 0.02	70.10± 0.30	**74.66± 0.01**
Frost	68.90± 0.06	66.42± 0.04	**69.50± 0.01**
Fog	72.60± 0.03	67.63± 0.07	**73.33± 0.03**
Brightness	78.16± 0.02	74.05± 0.03	**78.69± 0.02**
Contrast	73.75± 0.06	69.47± 0.24	**74.09± 0.01**
Elastic transform	68.41± 0.02	64.98± 0.09	**69.14± 0.05**
Pixelate	79.59± 0.06	75.54± 0.05	**80.09± 0.03**
JPEG compression	74.98± 0.05	70.55± 0.11	**75.56± 0.02**
V20-C Average	71.92	68.44	**72.58**

Table-A III-13 IoU performance of our method for different numbers of templates

Dataset: V20	1 Template	3 Templates	7 Templates	20 Templates	80 Templates
Original	77.00± 0.04	79.48± 0.08	**79.70± 0.06**	79.17± 0.01	79.25± 0.02
Gaussian noise	63.02± 0.06	66.51± 0.10	**66.75± 0.01**	65.94± 0.09	66.02± 0.02
Shot noise	65.88± 0.06	**70.20± 0.03**	70.03± 0.04	68.59± 0.01	69.00± 0.01
Impulse noise	64.17± 0.04	67.57± 0.01	**67.88± 0.07**	66.53± 0.06	66.91± 0.02
Defocus blur	72.06± 0.12	**74.66± 0.02**	74.31± 0.09	73.89± 0.12	74.09± 0.06
Glass blur	70.74± 0.07	**74.38± 0.12**	**74.38± 0.36**	73.78± 0.09	73.88± 0.07
Motion blur	73.50± 0.10	**75.80± 0.08**	75.72± 0.04	75.19± 0.04	75.19± 0.04
Zoom blur	61.36± 0.07	63.47± 0.02	**64.61± 0.01**	63.50± 0.03	63.84± 0.04
Snow	71.04± 0.05	74.09± 0.08	**74.66± 0.01**	73.78± 0.03	73.91± 0.01
Frost	67.01± 0.02	69.09± 0.01	**69.50± 0.01**	69.08± 0.00	69.10± 0.02
Fog	70.54± 0.07	**73.69± 0.01**	73.33± 0.03	72.87± 0.03	72.86± 0.05
Brightness	75.61± 0.02	78.49± 0.05	**78.69± 0.02**	77.86± 0.02	78.06± 0.02
Contrast	70.51± 0.04	74.01± 0.22	**74.09± 0.01**	73.48± 0.04	73.56± 0.09
Elastic transform	65.78± 0.05	68.09± 0.01	**69.14± 0.05**	67.94± 0.02	68.31± 0.09
Pixelate	76.95± 0.12	79.91± 0.04	**80.09± 0.03**	79.15± 0.06	79.41± 0.05
JPEG compression	71.84± 0.15	75.00± 0.05	**75.56± 0.02**	74.45± 0.09	74.75± 0.01
V20-C Average	69.33	72.33	**72.58**	71.74	71.93

Table-A III-14 Detailed mIoU comparison of MLMP components, showing individual and combined contributions

Multi-Level Fusion	✗	✓	✓	✗	✗	✓	✓	✓	✓	✗	✓	✓
Multi-Prompt Loss	✗	✗	✗	✓	✗	✓	✓	✗	✗	✓	✓	✓
Image-Level Entropy	✗	✗	✗	✗	✓	✗	✗	✓	✓	✓	✓	✓
Uncertainty-Aware Weight.	✗	✗	✓	✗	✗	✗	✓	✗	✓	✗	✗	✓
Original	77.00± 0.04	77.38± 0.01	81.67± 0.04	79.70± 0.06	78.74± 0.08	78.97± 0.03	83.00± 0.03	77.69± 0.01	82.70± 0.01	81.15± 0.02	79.13± 0.00	**83.76± 0.00**
Gaussian noise	63.02± 0.06	65.42± 0.04	67.82± 0.01	66.75± 0.01	65.66± 0.04	65.96± 0.04	69.13± 0.07	66.17± 0.04	69.00± 0.05	69.62± 0.01	67.35± 0.09	**71.13± 0.09**
Shot noise	65.88± 0.06	67.49± 0.01	70.52± 0.12	70.03± 0.04	68.97± 0.03	69.05± 0.03	72.31± 0.01	69.50± 0.06	73.22± 0.03	72.89± 0.02	71.02± 0.05	**75.02± 0.03**
Impulse noise	64.17± 0.04	65.11± 0.17	68.10± 0.01	67.88± 0.07	66.35± 0.12	65.39± 0.12	68.86± 0.13	65.16± 0.10	68.77± 0.15	70.31± 0.09	67.17± 0.09	**71.34± 0.11**
Defocus blur	72.06± 0.12	76.65± 0.23	78.78± 0.02	74.31± 0.09	75.00± 0.07	76.46± 0.15	78.78± 0.10	77.29± 0.10	79.78± 0.05	77.14± 0.05	77.79± 0.19	**80.36± 0.06**
Glass blur	70.74± 0.07	75.24± 0.05	77.66± 0.05	74.38± 0.36	73.54± 0.23	75.46± 0.09	77.48± 0.01	75.71± 0.10	78.09± 0.04	76.17± 0.08	76.87± 0.07	**78.84± 0.05**
Motion blur	73.50± 0.10	77.16± 0.13	79.39± 0.16	75.72± 0.04	76.09± 0.09	77.72± 0.09	79.97± 0.04	78.81± 0.03	81.49± 0.02	78.25± 0.05	79.00± 0.07	**81.41± 0.05**
Zoom blur	61.36± 0.07	64.22± 0.12	66.50± 0.16	64.61± 0.01	64.04± 0.02	66.38± 0.15	68.41± 0.13	65.12± 0.16	67.69± 0.08	68.32± 0.17	67.61± 0.05	**69.41± 0.12**
Snow	71.04± 0.05	72.64± 0.07	76.39± 0.02	74.66± 0.01	74.16± 0.02	73.25± 0.12	77.31± 0.11	74.05± 0.08	78.50± 0.16	77.20± 0.02	74.94± 0.06	**79.53± 0.05**
Frost	67.01± 0.02	67.30± 0.03	71.17± 0.00	69.50± 0.01	69.31± 0.00	67.57± 0.19	71.73± 0.21	68.31± 0.12	72.81± 0.08	71.34± 0.02	68.69± 0.11	**73.20± 0.07**
Fog	70.54± 0.07	72.73± 0.03	76.55± 0.06	73.33± 0.03	73.41± 0.00	75.06± 0.08	78.38± 0.03	73.48± 0.13	77.62± 0.05	75.98± 0.02	75.81± 0.02	**79.81± 0.06**
Brightness	75.61± 0.02	77.23± 0.07	81.34± 0.04	78.69± 0.02	77.34± 0.01	77.69± 0.02	82.00± 0.03	77.20± 0.05	82.16± 0.03	80.63± 0.05	78.27± 0.06	**83.51± 0.01**
Contrast	70.51± 0.04	73.36± 0.08	76.87± 0.06	74.09± 0.01	74.14± 0.14	73.57± 0.05	77.42± 0.09	74.09± 0.11	78.09± 0.08	76.97± 0.01	74.75± 0.06	**79.06± 0.16**
Elastic transform	65.78± 0.05	65.38± 0.12	70.59± 0.14	69.14± 0.05	67.92± 0.06	68.76± 0.04	72.96± 0.02	66.78± 0.03	71.80± 0.02	71.53± 0.03	69.77± 0.07	**74.03± 0.01**
Pixelate	76.95± 0.12	79.53± 0.03	83.02± 0.04	80.09± 0.03	79.09± 0.03	80.77± 0.05	83.93± 0.06	79.85± 0.08	83.90± 0.00	81.92± 0.12	81.34± 0.07	**84.97± 0.04**
JPEG compression	71.84± 0.15	74.38± 0.12	78.73± 0.08	75.56± 0.02	74.77± 0.11	76.77± 0.00	80.81± 0.09	74.61± 0.06	79.79± 0.02	77.98± 0.02	77.94± 0.07	**82.06± 0.01**
V20-C Average	69.33	71.59	74.90	72.58	71.99	72.66	75.97	72.41	76.18	75.08	73.89	**77.58**

Table-A III-15 mIoU comparison of MLMP and baseline methods on the V21 dataset, evaluated on both the original images and 15 corruption types

OVSS Backbone: NACLIP	Adaptation Method					
Dataset: V21	No Adapt.	TENT	TPT	WATT	CLIPArTT	MLMP
Original	45.12	45.65± 0.02	45.17± 0.00	28.58± 0.05	39.50± 0.04	**50.78± 0.02**
Gaussian noise	37.40	37.95± 0.00	37.34± 0.00	20.93± 0.07	30.05± 0.18	**43.59± 0.01**
Shot noise	39.33	39.17± 0.03	39.23± 0.00	22.06± 0.06	32.07± 0.17	**45.55± 0.02**
Impulse noise	37.81	37.73± 0.04	37.78± 0.00	19.95± 0.05	30.52± 0.08	**43.89± 0.01**
Defocus blur	41.46	41.46± 0.03	41.46± 0.00	24.79± 0.04	34.27± 0.05	**46.00± 0.00**
Glass blur	41.76	41.55± 0.01	41.82± 0.00	24.61± 0.03	34.55± 0.16	**46.83± 0.04**
Motion blur	42.65	42.81± 0.01	42.74± 0.00	25.66± 0.04	36.07± 0.11	**47.72± 0.04**
Zoom blur	34.46	34.25± 0.00	34.44± 0.00	20.74± 0.05	28.44± 0.07	**39.07± 0.02**
Snow	40.13	40.47± 0.00	40.23± 0.01	25.02± 0.06	33.98± 0.03	**46.30± 0.05**
Frost	40.70	41.83± 0.08	40.80± 0.00	23.43± 0.05	34.09± 0.01	**46.78± 0.01**
Fog	42.50	42.67± 0.00	42.47± 0.00	24.95± 0.05	36.37± 0.04	**47.61± 0.06**
Brightness	44.21	44.64± 0.03	44.27± 0.00	27.54± 0.07	38.44± 0.11	**49.89± 0.08**
Contrast	40.44	40.14± 0.03	40.44± 0.00	23.89± 0.04	33.68± 0.05	**45.22± 0.00**
Elastic transform	40.63	41.78± 0.01	40.67± 0.00	24.40± 0.05	35.38± 0.01	**46.87± 0.02**
Pixelate	44.70	44.95± 0.03	44.79± 0.00	27.74± 0.06	38.14± 0.06	**50.11± 0.01**
JPEG compression	43.05	42.87± 0.04	43.05± 0.00	26.04± 0.05	36.29± 0.10	**48.27± 0.02**
V21-C Average	40.75	40.95	40.77	24.12	34.16	**46.25**

Table-A III-16 mIoU comparison of MLMP and baseline methods on the P59 dataset, evaluated on both the original images and 15 corruption types

OVSS Backbone: NACLIP	Adaptation Method					
Dataset: P59	No Adapt.	TENT	TPT	WATT	CLIPArTT	MLMP
Original	28.23	28.73± 0.02	28.26± 0.01	16.55± 0.05	24.60± 0.00	**31.95± 0.02**
Gaussian noise	21.53	21.49± 0.01	21.59± 0.01	11.27± 0.04	16.42± 0.03	**24.84± 0.00**
Shot noise	22.35	22.31± 0.01	22.26± 0.00	11.80± 0.04	17.47± 0.00	**25.62± 0.01**
Impulse noise	21.74	21.59± 0.01	21.72± 0.00	11.24± 0.03	17.17± 0.00	**24.75± 0.01**
Defocus blur	25.42	25.14± 0.00	25.41± 0.00	14.31± 0.04	20.71± 0.00	**27.95± 0.03**
Glass blur	25.03	24.70± 0.01	25.01± 0.00	13.93± 0.03	20.63± 0.00	**27.66± 0.05**
Motion blur	26.11	26.00± 0.02	26.13± 0.01	15.13± 0.03	21.72± 0.00	**29.12± 0.03**
Zoom blur	19.20	19.49± 0.03	19.20± 0.00	10.86± 0.03	15.39± 0.00	**21.98± 0.02**
Snow	22.45	22.29± 0.02	22.45± 0.00	13.60± 0.04	19.17± 0.01	**25.47± 0.01**
Frost	21.95	21.94± 0.00	21.97± 0.00	12.83± 0.03	18.77± 0.04	**24.52± 0.01**
Fog	24.85	24.91± 0.03	24.86± 0.00	13.85± 0.05	20.66± 0.00	**27.84± 0.01**
Brightness	27.39	27.80± 0.01	27.42± 0.00	15.55± 0.04	23.56± 0.00	**30.63± 0.01**
Contrast	23.55	23.58± 0.01	23.54± 0.00	13.38± 0.05	19.12± 0.00	**26.95± 0.00**
Elastic transform	23.30	23.86± 0.02	23.30± 0.01	12.88± 0.04	20.28± 0.00	**27.16± 0.01**
Pixelate	27.61	27.55± 0.01	27.62± 0.00	15.84± 0.05	23.47± 0.00	**31.23± 0.01**
JPEG compression	25.75	25.51± 0.05	25.75± 0.00	14.09± 0.03	21.20± 0.00	**29.66± 0.02**
P59-C Average	23.88	23.88	23.88	13.37	19.72	**27.03**

Table-A III-17 mIoU comparison of MLMP and baseline methods on the P60 dataset, evaluated on both the original images and 15 corruption types

OVSS Backbone: NACLIP	Adaptation Method					
Dataset: P60	No Adapt.	TENT	TPT	WATT	CLIPArTT	MLMP
Original	24.95	25.29	24.98± 0.01	14.77± 0.04	21.88± 0.01	**27.99± 0.03**
Gaussian noise	19.31	19.17± 0.01	19.34± 0.01	10.29± 0.03	14.91± 0.01	**22.22± 0.02**
Shot noise	19.98	19.83± 0.02	19.91± 0.01	10.82± 0.03	15.82± 0.01	**22.81± 0.01**
Impulse noise	19.56	19.27± 0.01	19.54± 0.00	10.34± 0.03	15.58± 0.00	**22.26± 0.01**
Defocus blur	22.74	22.38± 0.01	22.72± 0.01	12.79± 0.04	18.67± 0.00	**24.92± 0.01**
Glass blur	22.49	22.05± 0.01	22.48± 0.01	12.64± 0.03	18.61± 0.00	**24.71± 0.05**
Motion blur	23.28	23.01± 0.03	23.30± 0.01	13.53± 0.03	19.51± 0.01	**25.71± 0.03**
Zoom blur	17.42	17.58± 0.03	17.42± 0.00	9.95± 0.03	14.07± 0.01	**19.70± 0.02**
Snow	20.06	19.80± 0.01	20.06± 0.01	12.29± 0.04	17.20± 0.02	**22.64± 0.02**
Frost	19.71	19.61± 0.01	19.73± 0.01	11.59± 0.03	17.02± 0.04	**22.02± 0.01**
Fog	22.20	22.09± 0.02	22.20± 0.01	12.50± 0.03	18.53± 0.01	**24.89± 0.00**
Brightness	24.35	24.56± 0.01	24.37± 0.01	13.87± 0.04	21.05± 0.01	**27.02± 0.00**
Contrast	21.09	21.00± 0.03	21.08± 0.01	12.04± 0.03	17.24± 0.01	**24.17± 0.02**
Elastic transform	21.17	21.46± 0.01	21.19± 0.01	11.70± 0.04	18.53± 0.01	**24.27± 0.02**
Pixelate	24.52	24.33± 0.01	24.54± 0.01	14.17± 0.04	21.01± 0.01	**27.48± 0.00**
JPEG compression	22.99	22.65± 0.02	24.54± 0.01	12.65± 0.03	19.05± 0.01	**26.24± 0.01**
P60-C Average	21.39	21.25	21.49	12.08	17.79	**24.07**

Table-A III-18 mIoU comparison of MLMP and baseline methods on the CityScapes dataset, evaluated on both the original images and 15 corruption types

OVSS Backbone: NACLIP	Adaptation Method				
Dataset: CityScapes	No Adapt.	TENT	TPT	WATT	MLMP
Original	29.49	30.54± 0.04	29.57± 0.01	20.77± 0.02	**33.35± 0.03**
Gaussian noise	**16.14**	15.00± 0.02	15.94± 0.01	10.19± 0.01	15.32± 0.00
Shot noise	20.18	19.38± 0.01	20.15± 0.01	12.41± 0.02	**20.57± 0.04**
Impulse noise	**16.37**	14.59± 0.02	16.35± 0.01	8.56± 0.02	15.76± 0.05
Defocus blur	23.34	23.50± 0.02	23.46± 0.01	15.63± 0.02	**24.86± 0.08**
Glass blur	24.41	24.04± 0.04	24.29± 0.00	15.44± 0.01	**25.70± 0.02**
Motion blur	24.73	24.98± 0.00	24.91± 0.01	14.21± 0.03	**26.02± 0.01**
Zoom blur	14.08	**14.57± 0.06**	14.08± 0.01	5.85± 0.02	14.04± 0.04
Snow	19.88	20.14± 0.01	19.90± 0.01	9.27± 0.03	**22.23± 0.03**
Frost	16.78	16.47± 0.02	16.78± 0.01	10.69± 0.01	**17.12± 0.07**
Fog	22.47	22.87± 0.05	22.25± 0.01	14.24± 0.01	**23.98± 0.03**
Brightness	28.45	29.30± 0.01	28.47± 0.01	19.99± 0.02	**31.88± 0.04**
Contrast	16.10	16.35± 0.03	16.07± 0.01	11.05± 0.01	**17.04± 0.00**
Elastic transform	29.14	30.16± 0.02	29.02± 0.01	19.99± 0.03	**32.86± 0.00**
Pixelate	28.67	29.59± 0.01	28.69± 0.01	18.30± 0.01	**32.72± 0.04**
JPEG compression	23.69	23.62± 0.01	23.67± 0.01	15.98± 0.02	**25.26± 0.03**
CityScapes-C Average	21.63	21.64	21.60	13.45	**23.02**

Table-A III-19 Detailed ACDC mIoU (reference vs. adverse views). All experiments use NACLIP as the OVSS model. MLMP consistently improves over TENT and the non-adapted baseline across all weather conditions

OVSS: NACLIP	Adaptation Method		
Condition	No Adapt.	TENT	MLMP
Fog (ref)	24.80	26.52± 0.03	**31.28± 0.03**
Fog	23.88	26.89± 0.04	**33.33± 0.04**
Night (ref)	24.95	26.54± 0.04	**28.81± 0.10**
Night	22.12	24.17± 0.00	**24.76± 0.03**
Rain (ref)	24.79	26.62± 0.00	**31.10± 0.03**
Rain	23.86	26.84± 0.04	**32.44± 0.04**
Snow (ref)	22.10	24.27± 0.04	**28.22± 0.02**
Snow	23.54	27.25± 0.05	**30.59± 0.03**
Average (all ref)	24.16	25.99	**29.85**
Average (all domains)	23.35	26.29	**30.28**

APPENDIX IV

SUPPLEMENTARY MATERIAL FOR THE PAPER TITLED HISTOPATH-C: TOWARDS REALISTIC DOMAIN SHIFTS FOR HISTOPATHOLOGY VISION-LANGUAGE ADAPTATION

1. Implementation Details

We conduct all experiments using the Quilt VLM (QuiltNet-B-32) Ikezogwo *et al.* (2024), a ViT-B/32–based model with 32×32 px patches and 224×224 px input resolution. We also evaluate on PathGen Sun *et al.* (2024), which uses a ViT-B/16 backbone with 16×16 px patches and a 224×224 px input resolution, trained on the large-scale PathGen pathology image–text dataset. Finally, we evaluate on the CONCH VLM Lu *et al.* (2024), which employs a ViT-B/16 backbone with 16×16 px patches and a 448×448 px input resolution, paired with a GPT-style text encoder for captioning and contrastive alignment. Following prior work Hakim *et al.* (2024), both our method (LATTE) and all baseline TTA approaches are fine-tuned at test time using the Adam optimizer with a fixed learning rate of 1×10^{-3} for 10 adaptation steps and a batch size of 128.

All runs use PyTorch on NVIDIA V100 GPUs with 32 GB of memory. To ensure statistical robustness, each experiment is repeated three times; we report the mean and standard deviation of each metric.

We have provided full reproduction scripts in https://github.com/Mehrdad-Noori/Histopath-C, which include on-the-fly corruption generation for Histopath-C, test-time adaptation routines for LATTE and all baselines, and evaluation code for computing zero-shot and post-adaptation metrics. For fairness, baselines use the same general hyperparameters above; any method-specific settings are set as recommended in their original publications.

2. Dataset Details

To assess the effectiveness and generalization capabilities of LATTE, we perform extensive evaluations across a suite of diverse and challenging histopathology datasets, encompassing various organs, cancer types, imaging conditions, and annotation granularities.

- **NCT-7K and NCT-100K** (Kather *et al.*, 2024) are large-scale colorectal cancer datasets composed of tissue patches extracted from whole-slide images. NCT-7K contains 7,180 non-overlapping patches, while NCT-100K extends this to 100,000 patches. Both datasets are annotated into nine tissue categories and can be grouped into broader classes of cancerous vs. normal tissues. These datasets serve as standard benchmarks for evaluating performance in colorectal cancer classification.
- **LC25000** (Borkowski *et al.*, 2019) comprises digitized histology images from two organs—lung and colon—and includes binary classification tasks distinguishing adenocarcinoma from normal tissue. We evaluate LATTE on the LC25K-Lung and LC25K-Colon subsets individually, as well as on a combined version (LC25K-All) to introduce additional heterogeneity and assess cross-organ generalization under stronger distributional shifts.
- **SkinCancer** (Kriegsmann *et al.*, 2022) is derived from the publicly available BCN20000 dataset and includes tissue patches extracted from skin biopsies. The dataset spans 12 anatomical compartments and 4 tumor types, such as basal cell carcinoma and melanoma, which we consolidate into the SkinTumor subset. This dataset introduces both intra-class variability and inter-class visual overlap, reflecting real-world diagnostic challenges in dermatopathology.
- **RenalCell** (Brummer *et al.*, 2022) includes histological images of clear cell renal cell carcinoma, annotated into five distinct tissue textures, including tumor, normal, and other microenvironmental structures. This dataset enables evaluation of LATTE on texture-centric classification tasks in renal oncology.
- **MHIST** (Wei *et al.*, 2021a) is a curated dataset of colorectal polyp images, specifically labeled as hyperplastic or sessile serrated adenomas (SSAs)—two classes with subtle morphological differences that pose a challenge even for expert pathologists. This binary classification task assesses model sensitivity to fine-grained structural features in colorectal screening.

For all datasets, we apply the ten corruption types defined in Histopath-C and refer to the resulting corrupted versions as *{DatasetName}-C* throughout the paper.

3. Text Templates

To account for the sensitivity of vision-language models to prompt phrasing, we evaluate our method using a diverse set of 25 text templates that describe each class from different linguistic perspectives. These templates range from generic image captions (e.g., "an image of class") to domain-specific phrasings that reference histopathological context and H&E staining.

Table IV-1 lists all templates used in our experiments. During adaptation, we employ loss-level ensembling across these prompts to improve prediction stability and mitigate prompt-induced variability.

4. Detailed Results of the Main Paper

This section provides complete versions of the experimental results that were summarized or partially reported in the main paper. The following tables report the detailed results when Quilt is used as the base VLM with different adaptation methods, TENT (Wang *et al.*, 2021a), LAME (Boudiaf *et al.*, 2022), TPT (Shu *et al.*, 2022), and CLIPArTT (Hakim *et al.*, 2024), across various datasets and evaluation settings. More specifically, Table IV-2 reports detailed results for the NCT100K dataset, Table IV-3 for LC25K-Lung, Table IV-4 for LC25K-Colon, Table IV-5 for LC25K-All, Table IV-6 for Skin, Table IV-7 for Renal, and Table IV-8 for MHIST. These results provide a more comprehensive assessment of the effectiveness and robustness of our approach, demonstrating that while LATTE may be outperformed by other methods on specific corruptions, it remains the most consistent overall across datasets and corruption types.

To further assess the generality of our approach, we conducted experiments on two additional vision–language models, PathGen (Sun *et al.*, 2024) and CONCH (Lu *et al.*, 2024). The detailed results, reported in Tables IV-9 - IV-24, consistently demonstrate that LATTE not only improves upon the source baseline but also frequently outperforms existing state-of-the-art methods by a

	Template
T^1	"a histopathology slide showing {class k}"
T^2	"histopathology image of {class k}"
T^3	"pathology tissue showing {class k}"
T^4	"presence of {class k} tissue on image"
T^5	"a photomicrograph showing {class k}"
T^6	"a photomicrograph of {class k}"
T^7	"an image of {class k}"
T^8	"an image showing {class k}"
T^9	"an example of {class k}"
T^{10}	"{class k} is shown"
T^{11}	"this is {class k}"
T^{12}	"there is {class k}"
T^{13}	"a histopathological image showing {class k}"
T^{14}	"a histopathological image of {class k}"
T^{15}	"a histopathological photograph of {class k}"
T^{16}	"a histopathological photograph showing {class k}"
T^{17}	"shows {class k}"
T^{18}	"presence of {class k}"
T^{19}	"{class k} is present"
T^{20}	"an H&E stained image of {class k}"
T^{21}	"an H&E stained image showing {class k}"
T^{22}	"an H&E image showing {class k}"
T^{23}	"an H&E image of {class k}"
T^{24}	"{class k}, H&E stain"
T^{25}	"{class k}, H&E"

Table-A IV-1 Full list of text templates used in LATTE. These templates provide diverse linguistic expressions of each class and are used during adaptation for loss-level ensembling

significant margin. These findings highlight the robustness and adaptability of LATTE across diverse model architectures and corruption scenarios.

VLM: Quilt Dataset		Source	TENT	LAME	TPT	CLIPArTT	LATTE (Ours)
NCT100K		55.98	41.42± 0.04	64.06± 0.21	52.83± 0.00	59.86± 0.01	**68.14± 0.03**
NCT100K-C	Stain-Light	47.35	37.53± 0.05	48.79± 0.07	44.74± 0.00	50.12± 0.05	**59.54± 0.00**
	Stain-Heavy	38.51	17.23± 0.02	35.33± 0.08	37.19± 0.01	52.30± 0.04	**58.61± 0.01**
	Dust	54.64	47.59± 0.14	60.69± 0.11	51.79± 0.04	61.33± 0.03	**64.47± 0.08**
	Air Bubble	57.78	53.44± 0.01	58.54± 0.17	56.79± 0.03	58.39± 0.03	**63.25± 0.07**
	Defocus Blur	50.02	35.88± 0.07	42.69± 0.09	48.67± 0.01	47.31± 0.05	**60.20± 0.09**
	Motion Blur	28.94	15.63± 0.02	26.47± 0.07	27.40± 0.06	40.79± 0.06	**44.05± 0.25**
	Gaussian Noise	32.12	16.01± 0.03	25.86± 0.03	30.64± 0.02	56.23± 0.03	**62.09± 0.11**
	Shot Noise	28.83	22.11± 0.12	19.37± 0.05	25.78± 0.01	48.69± 0.08	**55.65± 0.10**
	Brightness	37.27	22.43± 0.00	35.19± 0.11	37.20± 0.01	**57.39± 0.03**	52.24± 0.06
	Contrast	23.47	15.14± 0.00	19.03± 0.02	23.95± 0.00	39.22± 0.05	**41.18± 0.04**
	Mean	39.89	28.30	37.20	38.42	51.18	**56.13**

Table-A IV-2 Comparison of different adaptation methods on the NCT-100k dataset (8 classes) using Quilt (Ikezogwo *et al.*, 2024) as the base VLM

VLM: Quilt Dataset		Source	TENT	LAME	TPT	CLIPArTT	LATTE (Ours)
LC25K-Lung		82.87	70.34± 0.18	88.00± 0.18	83.03± 0.01	-	**89.41± 0.15**
LC25K-Lung-C	Stain-Light	73.79	45.57± 0.75	72.14± 0.31	73.85± 0.01	-	**78.70± 0.13**
	Stain-Heavy	57.88	33.33± 0.00	56.92± 0.18	59.55± 0.00	-	**75.57± 0.11**
	Dust	73.20	53.29± 0.16	71.85± 0.67	71.69± 0.09	-	**88.89± 0.02**
	Air Bubble	76.75	62.57± 0.01	68.75± 0.21	77.12± 0.03	-	**84.61± 0.31**
	Defocus Blur	73.40	34.47± 0.05	84.41± 0.57	70.44± 0.00	-	**91.19± 0.01**
	Motion Blur	77.89	64.18± 0.27	82.30± 0.14	77.13± 0.02	-	**86.75± 0.62**
	Gaussian Noise	80.54	70.64± 0.46	**85.86± 0.63**	78.99± 0.08	-	76.43± 0.20
	Shot Noise	78.47	62.49± 0.21	**83.85± 0.57**	77.31± 0.03	-	71.35± 0.56
	Brightness	77.19	66.63± 0.06	71.50± 0.03	76.57± 0.00	-	**91.61± 0.20**
	Contrast	56.32	33.33± 0.00	63.24± 0.39	57.95± 0.00	-	**42.91± 0.04**
	Mean	72.54	52.65	74.08	72.06	-	**78.80**

Table-A IV-3 Comparison of adaptation methods on LC25K-Lung (3 classes) using Quilt (Ikezogwo *et al.*, 2024) as the base VLM. CLIPArTT is not applicable due to fewer than three classes

VLM: Quilt Dataset		Source	TENT	LAME	TPT	CLIPArTT	LATTE (Ours)
LC25K-Colon		94.41	89.28± 0.54	98.70± 0.04	94.50± 0.03	-	**99.21± 0.03**
LC25K-C	Stain-Light	76.99	50.00± 0.00	82.77± 0.54	77.00± 0.00	-	**96.84± 0.07**
	Stain-Heavy	61.06	50.00± 0.00	53.37± 0.54	63.51± 0.01	-	**94.88± 0.04**
	Dust	90.42	58.76± 1.14	97.42± 0.60	90.08± 0.10	-	**98.04± 0.15**
	Air Bubble	89.24	50.95± 0.11	95.82± 0.47	89.22± 0.36	-	**96.38± 0.02**
	Gaussian Noise	75.78	50.00± 0.00	82.52± 0.16	76.56± 0.04	-	**99.00± 0.02**
	Shot Noise	60.23	50.00± 0.00	52.22± 0.32	59.54± 0.10	-	**97.83± 0.02**
	Defocus Blur	81.89	50.00± 0.00	94.22± 0.03	83.54± 0.00	-	**97.70± 0.15**
	Motion Blur	83.46	56.04± 0.19	**87.36± 0.03**	80.85± 0.25	-	61.63± 0.13
	Brightness	95.50	91.12± 0.33	98.99± 0.00	95.70± 0.00	-	**99.15± 0.02**
	Contrast	66.70	50.00± 0.00	53.86± 0.31	59.84± 0.01	-	**80.22± 0.00**
	Mean	78.13	55.69	79.86	77.58	-	**92.17**

Table-A IV-4 Comparison of adaptation methods on LC25K-Colon (2 classes) using Quilt (Ikezogwo *et al.*, 2024) as the base VLM. CLIPArTT is not applicable due to binary classification constraints

VLM: Quilt Dataset		Source	TENT	LAME	TPT	CLIPArTT	LATTE (Ours)
LC25K-All		79.28	71.39± 0.12	**87.13± 0.17**	79.22± 0.01	80.47± 0.01	86.97± 0.01
LC25K-C	Stain-Light	60.17	55.97± 0.16	61.49± 0.43	59.32± 0.00	**73.04± 0.05**	71.25± 0.07
	Stain-Heavy	46.36	24.53± 0.17	44.72± 0.11	45.48± 0.01	62.49± 0.04	**76.58± 0.07**
	Dust	68.65	59.92± 0.06	68.42± 0.32	68.18± 0.01	78.85± 0.03	**87.58± 0.01**
	Air Bubble	72.05	64.32± 0.06	77.97± 0.51	72.15± 0.15	77.15± 0.03	**80.82± 0.06**
	Gaussian Noise	57.59	47.81± 0.02	57.25± 0.32	57.51± 0.16	**81.22± 0.06**	80.85± 0.06
	Shot Noise	48.82	21.01± 0.04	46.19± 0.08	46.97± 0.09	**77.03± 0.05**	74.68± 0.17
	Defocus Blur	55.65	20.12± 0.01	56.07± 0.09	54.24± 0.00	48.99± 0.03	**70.31± 0.15**
	Motion Blur	56.26	20.77± 0.01	53.40± 0.52	55.15± 0.18	51.71± 0.04	**59.46± 0.48**
	Brightness	77.77	68.05± 0.07	79.69± 0.00	77.40± 0.00	82.26± 0.06	**89.30± 0.06**
	Contrast	28.01	20.05± 0.00	20.88± 0.00	**30.03± 0.02**	24.81± 0.03	25.93± 0.04
	Mean	57.13	40.26	56.61	56.64	65.76	**71.68**

Table-A IV-5 Comparison of adaptation methods on LC25K-All (5 classes) using Quilt (Ikezogwo *et al.*, 2024) as the base VLM

VLM: Quilt Dataset		Source	TENT	LAME	TPT	CLIPArTT	LATTE (Ours)
Skin		44.22	24.31± 0.13	40.09± 0.06	45.16± 0.00	46.42± 0.14	**50.62± 0.24**
Skin-C	Stain-Light	20.29	9.64± 0.05	11.75± 0.05	21.09± 0.00	23.96± 0.10	**33.28± 0.20**
	Stain-Heavy	13.68	3.28± 0.01	9.59± 0.18	13.72± 0.00	20.63± 0.14	**28.22± 0.06**
	Dust	40.07	21.34± 0.19	40.18± 0.09	42.31± 0.15	40.73± 0.07	**46.62± 0.05**
	Air Bubble	30.63	9.26± 0.04	23.13± 0.21	30.43± 0.02	38.40± 0.20	**44.29± 0.05**
	Gaussian Noise	14.15	2.57± 0.01	5.59± 0.00	14.02± 0.01	26.77± 0.06	**33.94± 0.09**
	Shot Noise	11.04	2.52± 0.01	4.30± 0.12	9.57± 0.07	20.99± 0.14	**25.95± 0.06**
	Defocus Blur	25.96	20.51± 0.01	17.97± 0.19	27.05± 0.01	**37.17± 0.03**	36.30± 0.09
	Motion Blur	26.44	6.90± 0.13	23.63± 0.03	26.43± 0.13	31.81± 0.09	**36.40± 0.06**
	Brightness	35.18	9.42± 0.10	34.03± 0.63	34.89± 0.00	**45.44± 0.16**	42.55± 0.13
	Contrast	4.64	2.39± 0.00	2.84± 0.02	5.17± 0.00	9.10± 0.07	**10.58± 0.04**
	Mean	22.21	8.78	17.30	22.47	29.50	**33.81**

Table-A IV-6 Comparison of adaptation methods on Skin (16 classes) using Quilt (Ikezogwo *et al.*, 2024) as the base VLM

VLM: Quilt Dataset		Source	TENT	LAME	TPT	CLIPArTT	LATTE (Ours)
Renal		49.76	43.19± 0.05	**50.77± 0.08**	50.29± 0.01	43.28± 0.12	46.14± 0.35
Renal-C	Stain-Light	26.78	17.86± 0.45	19.83± 0.13	28.50± 0.01	28.02± 0.25	**41.72± 0.13**
	Stain-Heavy	15.27	1.98± 0.02	2.73± 0.03	14.47± 0.02	23.48± 0.07	**44.35± 0.37**
	Dust	45.74	42.34± 0.00	**48.91± 0.10**	45.63± 0.10	39.02± 0.02	47.98± 0.31
	Air Bubble	38.91	42.37± 0.01	**43.31± 0.04**	39.19± 0.06	28.06± 0.02	36.01± 0.00
	Gaussian Noise	36.84	41.76± 0.02	41.64± 0.05	36.69± 0.02	33.43± 0.01	**45.63± 0.21**
	Shot Noise	34.23	41.86± 0.01	41.14± 0.08	33.37± 0.05	30.25± 0.17	**41.89± 0.01**
	Defocus Blur	25.70	16.14± 0.50	**31.02± 0.75**	24.54± 0.00	32.87± 0.23	25.68± 0.01
	Motion Blur	44.37	42.02± 0.01	43.19± 0.14	**44.43± 0.06**	30.40± 0.00	36.25± 0.19
	Brightness	24.81	12.21± 0.27	21.41± 0.08	23.31± 0.00	**32.14± 0.01**	31.76± 0.12
	Contrast	11.99	10.91± 0.01	10.78± 0.10	11.85± 0.00	16.05± 0.00	**31.63± 0.08**
	Mean	30.46	26.95	30.40	30.20	29.37	**38.29**

Table-A IV-7 Comparison of adaptation methods on Renal (5 classes) using Quilt (Ikezogwo *et al.*, 2024) as the base VLM

VLM: Quilt Dataset		Source	TENT	LAME	TPT	CLIPArTT	LATTE (Ours)
MHIST		62.95	63.15± 0.00	63.10± 0.56	61.51± 0.00	-	**64.02± 0.26**
MHIST-C	Stain-Light	47.19	36.85± 0.00	36.85± 0.00	47.49± 0.00	-	**62.90± 0.87**
	Stain-Heavy	42.78	36.85± 0.00	36.85± 0.00	41.56± 0.00	-	**60.85± 0.05**
	Dust	61.92	63.15± 0.00	62.69± 0.46	61.67± 0.15	-	**64.12± 0.97**
	Air Bubble	62.64	63.15± 0.00	63.15± 0.00	62.13± 0.00	-	**63.97± 0.61**
	Gaussian Noise	61.11	63.15± 0.00	**63.56± 1.94**	61.98± 0.26	-	62.74± 0.31
	Shot Noise	61.31	**63.15± 0.00**	59.67± 2.87	60.59± 0.00	-	62.64± 0.41
	Defocus Blur	63.66	63.15± 0.00	63.15± 0.00	**63.66± 0.00**	-	59.42± 0.36
	Motion Blur	66.33	63.15± 0.00	56.86± 0.46	**67.09± 0.46**	-	61.11± 0.92
	Brightness	54.76	36.85± 0.00	41.25± 0.51	53.89± 0.05	-	**63.41± 1.48**
	Contrast	55.78	**61.05± 0.77**	48.87± 0.77	55.89± 0.10	-	58.96± 0.20
	Mean	57.75	55.05	53.29	57.60	-	**62.01**

Table-A IV-8 Comparison of adaptation methods on MHIST (2 classes) using Quilt (Ikezogwo *et al.*, 2024) as the base VLM. CLIPArTT is not applicable due to binary classification constraints

VLM: PathGen Dataset		Source	TENT	LAME	TPT	CLIPArTT	LATTE (Ours)
NCT7K		64.74	73.02± 0.09	73.86± 0.15	64.42± 0.01	64.95± 0.02	**79.95± 0.36**
NCT7K-C	Stain-Light	50.97	60.02± 0.43	57.67± 0.01	50.64± 0.00	61.08± 0.05	**67.55± 0.47**
	Stain-Heavy	43.66	47.42± 0.16	45.26± 0.21	43.53± 0.00	56.90± 0.07	**68.81± 0.56**
	Dust	65.75	76.05± 0.11	75.17± 0.07	65.38± 0.13	71.83± 0.28	**82.05± 0.44**
	Air Bubble	48.79	52.91± 0.24	56.47± 0.60	49.09± 0.17	57.91± 0.04	**72.01± 0.37**
	Defocus Blur	45.03	43.64± 0.28	42.46± 0.47	45.63± 0.00	60.37± 0.11	**71.70± 0.53**
	Motion Blur	44.99	50.20± 0.11	45.45± 0.73	45.12± 0.34	63.97± 0.38	**76.37± 0.21**
	Gaussian Noise	23.09	11.22± 0.24	16.56± 0.46	22.49± 0.08	52.30± 0.38	**79.73± 0.14**
	Shot Noise	19.55	10.23± 0.02	13.76± 0.12	19.53± 0.06	42.89± 0.14	**76.44± 0.38**
	Brightness	56.53	62.16± 0.34	57.71± 0.42	56.13± 0.01	63.37± 0.14	**77.16± 0.12**
	Contrast	36.87	36.87± 0.03	34.09± 0.11	36.33± 0.01	44.52± 0.10	**68.72± 0.11**
	Mean	43.52	45.07	44.46	43.39	57.51	**74.05**

Table-A IV-9 Comparison of different adaptation methods on the NCT-7k dataset (8 classes) using PathGen (Sun *et al.*, 2024) as the base VLM

VLM: PathGen Dataset		Source	TENT	LAME	TPT	CLIPArTT	LATTE (Ours)
NCT100K		66.41	67.3± 0.02	68.73± 0.05	65.73± 0.0	69.84± 0.02	**81.73± 0.03**
NCT100K-C	Stain-Light	53.99	56.56± 0.03	63.4± 0.12	54.03± 0.0	66.37± 0.01	**73.29± 0.0**
	Stain-Heavy	39.40	25.89± 0.14	39.63± 0.15	39.36± 0.01	49.36± 0.02	**68.26± 0.05**
	Defocus Blur	67.46	70.89± 0.03	74.85± 0.22	66.93± 0.01	73.12± 0.07	**79.86± 0.08**
	Motion Blur	47.78	35.47± 0.04	51.29± 0.15	47.20± 0.01	65.98± 0.07	**69.37± 0.05**
	Brightness	50.66	42.33± 0.05	47.05± 0.16	49.29± 0.0	59.86± 0.05	**70.33± 0.06**
	Contrast	48.39	44.32± 0.09	43.75± 0.05	48.48± 0.06	58.15± 0.08	**69.13± 0.07**
	Gaussian Noise	29.91	19.62± 0.07	23.71± 0.0	29.64± 0.05	52.02± 0.07	**72.35± 0.11**
	Shot Noise	27.29	15.58± 0.01	20.88± 0.06	26.85± 0.02	44.19± 0.02	**68.77± 0.07**
	Dust	57.32	61.53± 0.01	61.83± 0.06	57.10± 0.0	65.09± 0.01	**77.24± 0.05**
	Air Bubble	36.86	30.26± 0.03	28.87± 0.08	35.23± 0.0	48.09± 0.01	**68.16± 0.06**
	Mean	45.91	40.25	45.53	45.41	58.22	**71.68**

Table-A IV-10 Comparison of different adaptation methods on the NCT-100k dataset (8 classes) using PathGen (Sun *et al.*, 2024) as the base VLM

VLM: PathGen Dataset		Source	TENT	LAME	TPT	CLIPArTT	LATTE (Ours)
LC25K-Lung		82.79	81.85± 0.01	77.74± 0.05	81.41± 0.0	-	**96.05± 0.07**
LC25K-Lung-C	Stain-Light	79.57	66.81± 0.07	71.75± 0.73	79.29± 0.0	-	**93.84± 0.06**
	Stain-Heavy	68.81	62.84± 0.22	66.31± 0.36	68.75± 0.01	-	**88.61± 0.07**
	Dust	75.79	71.03± 0.24	72.92± 0.36	75.92± 0.26	-	**94.76± 0.03**
	Air Bubble	62.75	64.39± 0.01	65.70± 0.24	61.69± 0.08	-	**91.63± 0.07**
	Defocus Blur	67.42	39.16± 0.17	54.98± 0.28	65.13± 0.0	-	**90.54± 0.15**
	Motion Blur	61.91	35.67± 0.37	45.56± 0.07	60.83± 0.08	-	**90.83± 0.01**
	Gaussian Noise	55.27	33.38± 0.01	49.51± 0.28	55.27± 0.01	-	**91.94± 0.07**
	Shot Noise	56.48	33.41± 0.01	60.81± 0.45	58.04± 0.03	-	**90.82± 0.14**
	Brightness	85.40	88.78± 0.14	90.06± 0.31	84.52± 0.0	-	**94.98± 0.10**
	Contrast	57.46	33.33± 0.0	61.81± 0.17	54.91± 0.03	-	**65.78± 0.13**
	Mean	67.09	52.88	63.94	66.44	-	**89.37**

Table-A IV-11 Comparison of adaptation methods on LC25K-Lung (3 classes) using PathGen (Sun *et al.*, 2024) as the base VLM. CLIPArTT is not applicable due to fewer than three classes

VLM: PathGen Dataset		Source	TENT	LAME	TPT	CLIPArTT	LATTE (Ours)
LC25K-Colon		95.92	97.69± 0.06	**98.98± 0.01**	95.82± 0.0	-	98.27± 0.07
LC25K-Colon-C	Stain-Light	89.14	94.94± 0.05	**97.57± 0.11**	89.16± 0.0	-	97.44± 0.03
	Stain-Heavy	83.11	79.81± 1.01	92.71± 0.13	82.96± 0.02	-	**95.07± 0.02**
	Dust	92.93	96.13± 0.16	**98.78± 0.01**	92.87± 0.08	-	96.60± 0.21
	Air Bubble	83.59	83.38± 0.63	**95.56± 0.23**	83.58± 0.08	-	90.66± 0.13
	Defocus Blur	57.57	50.00± 0.0	50.86± 0.42	56.50± 0.0	-	**92.51± 0.10**
	Motion Blur	56.28	50.00± 0.0	50.03± 0.03	55.69± 0.15	-	**94.46± 0.07**
	Gaussian Noise	66.16	50.04± 0.01	56.03± 0.42	66.10± 0.05	-	**98.02± 0.01**
	Shot Noise	54.89	50.00± 0.0	50.39± 0.39	55.70± 0.09	-	**96.84± 0.06**
	Brightness	95.58	97.45± 0.08	**98.76± 0.11**	95.52± 0.0	-	97.57± 0.12
	Contrast	50.34	50.00± 0.0	50.00± 0.0	50.09± 0.0	-	**84.70± 0.90**
	Mean	72.96	70.18	74.07	72.82	-	**94.39**

Table-A IV-12 Comparison of adaptation methods on LC25K-Colon (2 classes) using PathGen (Sun *et al.*, 2024) as the base VLM. CLIPArTT is not applicable due to binary classification constraints

VLM: PathGen Dataset		Source	TENT	LAME	TPT	CLIPArTT	LATTE (Ours)
LC25K-ALL		83.25	90.47± 0.13	83.74± 0.09	82.38± 0.01	87.40± 0.04	**93.27± 0.01**
LC25K-ALL-C	Stain-Light	73.17	80.43± 0.09	77.24± 0.35	73.25± 0.01	81.20± 0.05	**91.10± 0.09**
	Stain-Heavy	57.83	54.13± 0.03	62.67± 0.09	57.83± 0.0	75.37± 0.07	**82.59± 0.16**
	Dust	77.90	84.48± 0.11	80.58± 0.44	77.67± 0.19	84.00± 0.06	**91.30± 0.07**
	Air Bubble	60.99	60.13± 0.04	65.34± 0.54	59.91± 0.07	69.98± 0.11	**82.70± 0.13**
	Defocus Blur	45.27	38.73± 0.0	40.18± 0.07	44.24± 0.0	54.64± 0.09	**80.48± 0.03**
	Motion Blur	40.34	24.41± 0.05	38.64± 0.12	39.61± 0.08	64.98± 0.02	**83.98± 0.11**
	Gaussian Noise	50.92	28.10± 0.09	48.11± 0.22	51.08± 0.08	78.68± 0.04	**87.64± 0.10**
	Shot Noise	45.25	20.02± 0.01	46.20± 0.02	45.92± 0.07	67.94± 0.06	**83.99± 0.18**
	Brightness	84.35	89.58± 0.10	90.50± 0.51	84.21± 0.0	85.37± 0.01	**91.82± 0.08**
	Contrast	20.56	20.00± 0.0	20.00± 0.0	20.16± 0.0	43.00± 0.07	**45.23± 0.19**
	Mean	55.66	50.00	56.95	55.39	70.52	**82.08**

Table-A IV-13 Comparison of adaptation methods on LC25K-All (5 classes) using PathGen (Sun *et al.*, 2024) as the base VLM

	VLM: PathGen Dataset	Source	TENT	LAME	TPT	CLIPArTT	LATTE (Ours)
Skin		55.26	56.02± 0.02	63.52± 0.06	54.71± 0.0	63.22± 0.13	**68.34± 0.12**
Skin-C	Stain-Light	27.44	13.55± 0.06	20.91± 0.47	27.33± 0.0	49.33± 0.02	**61.42± 0.21**
	Stain-Heavy	14.69	2.95± 0.01	5.92± 0.02	15.62± 0.0	33.27± 0.07	**52.58± 0.03**
	Dust	52.72	56.19± 0.07	56.98± 0.12	52.62± 0.16	60.01± 0.09	**66.81± 0.12**
	Air Bubble	35.77	21.48± 0.03	37.05± 0.15	35.25± 0.14	47.24± 0.18	**56.69± 0.14**
	Defocus Blur	26.16	21.16± 0.10	16.11± 0.22	25.07± 0.0	37.34± 0.06	**48.42± 0.13**
	Motion Blur	26.94	20.82± 0.09	24.10± 0.34	26.44± 0.0	38.33± 0.19	**52.20± 0.02**
	Gaussian Noise	9.92	2.55± 0.07	4.05± 0.06	10.14± 0.08	22.10± 0.06	**31.26± 0.02**
	Shot Noise	6.02	1.61± 0.02	1.76± 0.02	5.81± 0.07	16.82± 0.07	**30.14± 0.02**
	Brightness	46.97	45.31± 0.11	52.13± 0.02	46.67± 0.0	52.01± 0.03	**63.23± 0.16**
	Contrast	15.61	3.56± 0.05	18.05± 0.12	16.83± 0.0	24.27± 0.0	**43.73± 0.17**
	Mean	26.22	18.92	23.71	26.18	38.07	**50.65**

Table-A IV-14 Comparison of adaptation methods on Skin (16 classes) using PathGen (Sun *et al.*, 2024) as the base VLM

	VLM: PathGen Dataset	Source	TENT	LAME	TPT	CLIPArTT	LATTE (Ours)
Renal		48.61	49.66± 0.02	50.28± 0.06	48.67± 0.0	50.42± 0.01	**56.81± 0.17**
Renal-C	Stain-Light	42.56	45.01± 0.04	45.98± 0.09	41.40± 0.0	**47.94± 0.17**	44.97± 0.06
	Stain-Heavy	40.55	44.75± 0.01	42.62± 0.20	40.50± 0.0	**49.56± 0.04**	39.45± 0.11
	Dust	49.33	47.42± 0.02	47.22± 0.04	48.76± 0.07	51.07± 0.04	**55.18± 0.15**
	Air Bubble	35.00	7.29± 0.05	38.46± 0.15	35.94± 0.09	**46.26± 0.04**	44.53± 0.02
	Defocus Blur	6.54	1.96± 0.0	2.14± 0.01	6.36± 0.0	20.73± 0.06	**49.45± 0.29**
	Motion Blur	11.33	2.25± 0.0	2.61± 0.06	10.63± 0.08	21.49± 0.03	**45.18± 0.25**
	Gaussian Noise	9.35	1.95± 0.0	1.98± 0.01	9.54± 0.01	28.78± 0.04	**39.98± 0.05**
	Shot Noise	3.54	1.95± 0.0	1.95± 0.0	3.32± 0.01	14.26± 0.11	**38.66± 0.13**
	Brightness	25.10	2.77± 0.03	8.98± 0.07	25.42± 0.01	42.75± 0.17	**45.10± 0.27**
	Contrast	7.13	2.71± 0.02	3.93± 0.06	6.32± 0.0	24.00± 0.09	**38.32± 0.22**
	Mean	23.04	15.81	19.59	22.82	34.68	**44.08**

Table-A IV-15 Comparison of adaptation methods on Renal (5 classes) using PathGen (Sun *et al.*, 2024) as the base VLM

VLM: PathGen Dataset		Source	TENT	LAME	TPT	CLIPArTT	LATTE (Ours)
MHIST		57.11	39.00± 0.31	46.21± 0.97	57.22± 0.0	-	**60.49± 0.31**
MHIST-C	Stain-Light	55.99	36.95± 0.0	41.10± 0.05	56.29± 0.0	-	**62.69± 0.46**
	Stain-Heavy	56.50	37.26± 0.0	41.86± 0.72	56.50± 0.0	-	**61.98± 0.05**
	Dust	58.55	45.50± 1.38	47.80± 0.72	58.19± 0.26	-	**59.83± 0.56**
	Air Bubble	54.45	**63.15± 0.0**	56.14± 0.67	52.87± 0.05	-	44.93± 0.92
	Defocus Blur	57.32	**63.15± 0.0**	**63.15± 0.0**	57.63± 0.0	-	51.38± 0.41
	Motion Blur	**45.14**	36.85± 0.0	42.17± 1.23	43.19± 0.20	-	43.60± 0.10
	Gaussian Noise	53.22	37.31± 0.05	42.48± 0.92	52.71± 0.10	-	**60.18± 0.20**
	Shot Noise	57.01	40.63± 1.74	53.02± 1.64	57.83± 0.61	-	**60.95± 0.05**
	Brightness	62.85	**63.15± 0.0**	62.79± 0.77	62.69± 0.05	-	59.67± 0.31
	Contrast	51.89	36.85± 0.0	44.93± 1.64	51.48± 0.0	-	**56.96± 0.15**
	Mean	55.29	46.08	49.54	54.94	-	**56.22**

Table-A IV-16 Comparison of adaptation methods on MHIST (2 classes) using PathGen (Sun *et al.*, 2024) as the base VLM. CLIPArTT is not applicable due to binary classification constraints

VLM: CONCH Dataset		Source	TENT	LAME	TPT	CLIPArTT	LATTE (Ours)
NCT7K		67.55	68.66± 0.06	73.59± 0.29	67.41± 0.0	58.28± 0.14	**82.12± 0.24**
NCT7K-C	Stain-Light	56.80	58.36± 0.09	58.81± 0.13	56.64± 0.0	56.53± 0.05	**71.36± 0.18**
	Stain-Heavy	28.30	29.88± 0.39	29.31± 0.0	28.14± 0.0	37.83± 0.12	**68.77± 0.01**
	Dust	66.92	67.74± 0.14	72.92± 0.03	66.97± 0.19	60.86± 0.09	**79.89± 0.01**
	Air Bubble	52.50	51.83± 0.24	58.56± 0.69	52.06± 0.02	42.69± 0.34	**77.39± 0.02**
	Defocus Blur	40.27	36.86± 0.07	42.21± 0.03	40.23± 0.0	49.01± 0.05	**67.86± 0.04**
	Motion Blur	40.85	37.11± 0.33	42.13± 0.41	40.97± 0.02	44.80± 0.09	**77.02± 0.13**
	Gaussian Noise	12.38	18.47± 0.09	8.68± 0.24	12.50± 0.09	13.12± 0.19	**65.05± 0.09**
	Shot Noise	4.01	9.75± 0.13	1.18± 0.03	3.93± 0.06	11.21± 0.02	**55.82± 0.19**
	Brightness	49.31	45.29± 0.08	48.27± 0.32	49.30± 0.0	48.51± 0.05	**81.98± 0.16**
	Contrast	22.19	26.65± 0.03	24.50± 0.04	22.14± 0.0	26.30± 0.15	**33.96± 0.11**
	Mean	37.35	38.19	38.66	37.29	39.09	**67.91**

Table-A IV-17 Comparison of different adaptation methods on the NCT-7k dataset (8 classes) using CONCH (Lu *et al.*, 2024) as the base VLM

VLM: CONCH Dataset		Source	TENT	LAME	TPT	CLIPArTT	LATTE (Ours)
NCT100K		63.79	65.00± 0.02	71.82± 0.04	63.73± 0.0	59.66± 0.08	**82.76± 0.01**
NCT100K-C	Stain-Light	55.71	56.17± 0.02	60.76± 0.02	55.62± 0.0	56.33± 0.01	**73.41± 0.02**
	Stain-Heavy	22.37	18.54± 0.02	21.21± 0.04	22.37± 0.0	31.37± 0.05	**63.92± 0.02**
	Dust	61.15	62.41± 0.09	68.23± 0.04	61.56± 0.01	57.15± 0.02	**79.90± 0.03**
	Air Bubble	49.11	49.63± 0.05	57.76± 0.29	49.02± 0.09	50.25± 0.06	**71.74± 0.02**
	Defocus Blur	27.14	18.79± 0.03	25.30± 0.07	26.82± 0.0	40.16± 0.03	**72.14± 0.09**
	Motion Blur	28.95	21.09± 0.03	27.96± 0.04	28.44± 0.05	38.13± 0.02	**70.86± 0.01**
	Gaussian Noise	11.17	14.25± 0.05	4.55± 0.03	11.45± 0.05	19.72± 0.0	**57.66± 0.11**
	Shot Noise	3.97	12.05± 0.02	1.06± 0.07	3.94± 0.02	13.37± 0.0	**47.19± 0.09**
	Brightness	20.04	50.11± 0.03	55.42± 0.03	50.02± 0.0	47.96± 0.02	**78.88± 0.11**
	Contrast	**50.19**	20.16± 0.0	20.11± 0.0	20.20± 0.0	15.63± 0.04	19.63± 0.02
	Mean	32.98	32.32	34.24	32.94	37.01	**63.53**

Table-A IV-18 Comparison of different adaptation methods on the NCT-100k dataset (8 classes) using CONCH (Lu *et al.*, 2024) as the base VLM

VLM: CONCH Dataset		Source	TENT	LAME	TPT	CLIPArTT	LATTE (Ours)
LC25K-Lung		87.95	90.22± 0.06	92.67± 0.01	87.45± 0.01	-	**97.02± 0.12**
LC25K-Lung-C	Stain-Light	77.00	78.00± 0.01	79.78± 0.16	77.18± 0.0	-	**94.94± 0.08**
	Stain-Heavy	51.31	40.34± 0.09	49.37± 0.24	51.31± 0.0	-	**82.68± 0.19**
	Dust	75.19	75.75± 0.01	78.88± 0.12	75.52± 0.01	-	**96.19± 0.02**
	Air Bubble	52.51	38.82± 0.04	48.03± 0.07	52.73± 0.11	-	**94.73± 0.01**
	Defocus Blur	53.69	42.55± 0.14	52.66± 0.07	53.61± 0.0	-	**93.20± 0.04**
	Motion Blur	60.34	54.04± 0.49	57.32± 0.0	60.08± 0.09	-	**93.15± 0.13**
	Gaussian Noise	34.33	33.68± 0.03	33.54± 0.05	34.32± 0.02	-	**82.47± 0.13**
	Shot Noise	33.43	33.39± 0.0	33.34± 0.0	33.43± 0.0	-	**76.15± 0.18**
	Brightness	87.69	90.08± 0.18	93.08± 0.10	87.71± 0.0	-	**96.46± 0.04**
	Contrast	**35.83**	33.33± 0.0	33.53± 0.09	35.80± 0.0	-	27.24± 0.18
	Mean	56.13	52.00	55.95	56.17	-	**83.72**

Table-A IV-19 Comparison of adaptation methods on LC25K-Lung (3 classes) using CONCH (Lu *et al.*, 2024) as the base VLM. CLIPArTT is not applicable due to fewer than three classes

VLM: CONCH Dataset		Source	TENT	LAME	TPT	CLIPArTT	LATTE (Ours)
LC25K-Colon		95.31	96.32± 0.07	97.46± 0.02	95.15± 0.0	-	**99.18± 0.08**
LC25K-Colon-C	Stain-Light	89.76	92.27± 0.06	93.40± 0.03	89.75± 0.0	-	**99.20± 0.08**
	Stain-Heavy	58.07	53.16± 0.30	55.89± 0.14	58.07± 0.0	-	**96.52± 0.20**
	Dust	93.92	95.73± 0.10	97.05± 0.02	93.81± 0.07	-	**99.02± 0.04**
	Air Bubble	89.04	92.28± 0.44	93.03± 0.24	88.92± 0.20	-	**99.00± 0.00**
	Defocus Blur	59.32	53.89± 0.19	57.08± 0.30	59.34± 0.0	-	**99.19± 0.06**
	Motion Blur	57.64	55.14± 0.33	56.01± 0.16	57.36± 0.31	-	**99.22± 0.02**
	Gaussian Noise	65.46	59.96± 0.03	64.48± 0.08	65.16± 0.14	-	**97.45± 0.06**
	Shot Noise	53.54	51.07± 0.08	51.80± 0.09	53.46± 0.09	-	**90.66± 0.13**
	Brightness	95.16	95.78± 0.07	96.96± 0.09	95.10± 0.0	-	**99.35± 0.05**
	Contrast	50.00	50.00± 0.0	50.00± 0.0	50.00± 0.0	-	**94.57± 0.16**
	Mean	71.19	69.93	71.57	71.10	-	**97.42**

Table-A IV-20 Comparison of adaptation methods on LC25K-Colon (2 classes) using CONCH (Lu *et al.*, 2024) as the base VLM. CLIPArTT is not applicable due to binary classification constraints

VLM: CONCH Dataset		Source	TENT	LAME	TPT	CLIPArTT	LATTE (Ours)
LC25K-All		86.58	87.98± 0.06	90.55± 0.01	86.17± 0.0	89.49± 0.04	**95.81± 0.08**
LC25K-All-C	Stain-Light	69.46	73.28± 0.04	74.47± 0.03	69.51± 0.0	43.98± 0.07	**88.90± 0.03**
	Stain-Heavy	37.96	25.34± 0.13	36.37± 0.07	37.93± 0.0	29.42± 0.04	**73.85± 0.02**
	Dust	80.01	83.83± 0.16	85.02± 0.15	80.13± 0.16	81.19± 0.03	**95.16± 0.05**
	Air Bubble	60.73	59.55± 0.02	61.20± 0.08	60.96± 0.03	56.05± 0.18	**93.13± 0.11**
	Defocus Blur	50.35	49.89± 0.22	50.17± 0.0	50.29± 0.0	46.68± 0.13	**89.77± 0.20**
	Motion Blur	53.15	51.13± 0.09	51.56± 0.0	52.99± 0.12	65.31± 0.01	**91.89± 0.08**
	Gaussian Noise	31.41	24.25± 0.07	30.01± 0.02	31.56± 0.01	25.31± 0.10	**82.57± 0.10**
	Shot Noise	25.41	20.10± 0.0	22.46± 0.10	25.31± 0.02	27.31± 0.03	**63.39± 0.19**
	Brightness	83.22	85.23± 0.06	88.57± 0.03	83.06± 0.0	79.61± 0.0	**94.17± 0.09**
	Contrast	19.98	20.01± 0.0	20.00± 0.0	19.97± 0.0	25.74± 0.02	**30.98± 0.31**
	Mean	51.17	49.26	51.98	51.17	48.06	**80.38**

Table-A IV-21 Comparison of adaptation methods on LC25K-All (5 classes) using CONCH (Lu *et al.*, 2024) as the base VLM

VLM: CONCH Dataset		Source	TENT	LAME	TPT	CLIPArTT	LATTE (Ours)
Skin		34.56	33.34± 0.14	35.98± 0.12	34.11± 0.0	31.39± 0.15	**64.32± 0.01**
Skin-C	Stain-Light	26.86	27.29± 0.01	27.48± 0.0	26.88± 0.0	23.28± 0.07	**48.99± 0.02**
	Stain-Heavy	18.02	15.98± 0.04	17.76± 0.13	18.04± 0.0	15.10± 0.03	**40.54± 0.07**
	Dust	30.98	29.67± 0.01	30.24± 0.07	30.96± 0.01	32.19± 0.09	**62.41± 0.20**
	Air Bubble	22.00	20.42± 0.10	21.69± 0.03	21.73± 0.02	25.10± 0.07	**52.94± 0.15**
	Defocus Blur	23.48	20.22± 0.07	23.06± 0.07	23.47± 0.0	22.20± 0.03	**54.91± 0.09**
	Motion Blur	20.75	15.59± 0.31	19.99± 0.21	20.61± 0.0	25.37± 0.09	**59.95± 0.26**
	Gaussian Noise	19.02	21.14± 0.05	21.24± 0.02	18.96± 0.05	20.85± 0.01	**37.14± 0.08**
	Shot Noise	15.31	10.62± 0.02	16.29± 0.16	15.13± 0.07	15.31± 0.09	**30.76± 0.12**
	Brightness	33.44	31.23± 0.02	32.27± 0.03	33.36± 0.0	36.16± 0.16	**62.89± 0.15**
	Contrast	5.31	4.28± 0.0	5.06± 0.02	5.30± 0.0	2.96± 0.01	**25.59± 0.30**
	Mean	21.52	19.64	21.51	21.44	21.85	**47.61**

Table-A IV-22 Comparison of adaptation methods on Skin (16 classes) using CONCH (Lu *et al.*, 2024) as the base VLM

VLM: CONCH Dataset		Source	TENT	LAME	TPT	CLIPArTT	LATTE (Ours)
Renal		46.16	48.74± 0.22	51.26± 0.20	45.65± 0.0	52.87± 0.06	**57.22± 0.22**
Renal-C	Stain-Light	28.20	22.84± 0.01	27.28± 0.04	28.36± 0.0	34.60± 0.09	**51.95± 0.06**
	Stain-Heavy	7.11	2.69± 0.03	3.23± 0.06	6.85± 0.0	10.46± 0.12	**42.67± 0.01**
	Dust	43.43	42.31± 0.07	45.87± 0.07	43.58± 0.04	46.88± 0.18	**59.11± 0.07**
	Air Bubble	35.22	35.25± 0.17	39.93± 0.24	35.35± 0.03	32.86± 0.0	**53.90± 0.09**
	Defocus Blur	27.20	26.24± 0.11	27.54± 0.10	27.32± 0.0	29.02± 0.01	**51.10± 0.12**
	Motion Blur	15.76	12.31± 0.11	14.47± 0.02	15.62± 0.09	19.12± 0.27	**44.94± 0.03**
	Gaussian Noise	3.54	2.12± 0.0	2.02± 0.0	3.23± 0.02	9.14± 0.03	**49.01± 0.20**
	Shot Noise	3.26	1.96± 0.0	2.00± 0.0	3.17± 0.01	7.64± 0.09	**46.63± 0.09**
	Brightness	19.47	11.27± 0.04	15.24± 0.04	19.30± 0.0	19.82± 0.05	**53.48± 0.09**
	Contrast	23.01	24.56± 0.06	24.47± 0.14	23.01± 0.0	16.88± 0.05	**37.12± 0.03**
	Mean	20.62	18.16	20.21	20.58	22.64	**48.99**

Table-A IV-23 Comparison of adaptation methods on Renal (5 classes) using CONCH (Lu *et al.*, 2024) as the base VLM

VLM: CONCH Dataset		Source	TENT	LAME	TPT	CLIPArTT	LATTE (Ours)
MHIST		59.98	62.59± 0.56	60.39± 1.02	59.72± 0.05	-	**63.00± 0.36**
MHIST-C	Stain-Light	57.11	49.18± 0.77	55.22± 1.59	57.11± 0.0	-	**63.36± 0.51**
	Stain-Heavy	53.94	46.42± 2.41	50.46± 0.51	53.84± 0.0	-	**56.65± 0.15**
	Dust	58.24	**63.15± 0.0**	61.51± 0.51	57.47± 0.56	-	62.64± 1.23
	Air Bubble	60.70	63.15± 0.0	62.95± 0.20	62.33± 0.10	-	**64.59± 1.33**
	Defocus Blur	51.28	**63.15± 0.0**	58.96± 0.41	52.05± 0.05	-	56.81± 0.51
	Motion Blur	61.82	**63.15± 0.0**	63.00± 0.26	61.72± 0.51	-	56.91± 0.61
	Gaussian Noise	61.72	**63.15± 0.0**	**63.15± 0.0**	61.67± 0.05	-	50.31± 0.77
	Shot Noise	62.03	**63.15± 0.0**	**63.15± 0.0**	62.03± 0.20	-	51.59± 0.51
	Brightness	63.25	63.15± 0.0	**64.07± 0.10**	63.25± 0.0	-	58.03± 1.43
	Contrast	48.82	36.85± 0.0	45.34± 0.31	**49.13± 0.0**	-	45.04± 0.41
	Mean	57.89	57.45	**58.78**	58.06	-	56.59

Table-A IV-24 Comparison of adaptation methods on MHIST (2 classes) using CONCH (Lu *et al.*, 2024) as the base VLM. CLIPArTT is not applicable due to binary classification constraints

BIBLIOGRAPHY

Albuquerque, I., Naik, N., Li, J., Keskar, N. & Socher, R. (2020). Improving out-of-distribution generalization via multi-task self-supervised pretraining. *arXiv preprint arXiv:2003.13525*.

Alpher, F. (2002). Frobnication. *IEEE TPAMI*, 12(1), 234–778.

Alpher, F. & Fotheringham-Smythe, F. (2003). Frobnication revisited. *Journal of Foo*, 13(1), 234–778.

Alpher, F. & Gamow, F. (2005). Can a computer frobnicate? *CVPR*, pp. 234–778.

Alpher, F., Fotheringham-Smythe, F. & Gamow, F. (2004). Can a machine frobnicate? *Journal of Foo*, 14(1), 234–778.

Aminbeidokhti, M., Guerrero-Pena, F. A., Medeiros, H. R., Dubail, T., Granger, E. & Pedersoli, M. (2024a). Domain Generalization by Rejecting Extreme Augmentations. *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*. Retrieved from: https://github.com/Masseeh/DCAug.

Aminbeidokhti, M., Pena, F. A. G., Medeiros, H. R., Dubail, T., Granger, E. & Pedersoli, M. (2024b). Domain Generalization by Rejecting Extreme Augmentations. *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 2215–2225.

Angelopoulos, A. N. & Bates, S. (2021). A gentle introduction to conformal prediction and distribution-free uncertainty quantification. *arXiv preprint arXiv:2107.07511*.

Anonymous. [ECCV submission ID 00324, supplied as supplemental material `00324.pdf`]. (2024a). The frobnicatable foo filter.

Anonymous. [Supplied as supplemental material `tr.pdf`]. (2024b). Frobnication tutorial.

Aresta, G., Araújo, T., Kwok, S., Chennamsetty, S. S., Safwan, M., Alex, V., Marami, B., Prastawa, M., Chan, M., Donovan, M. et al. (2019). Bach: Grand challenge on breast cancer histology images. *Medical image analysis*, 56, 122–139.

Arjovsky, M., Bottou, L., Gulrajani, I. & Lopez-Paz, D. (2019). Invariant risk minimization. *arXiv preprint arXiv:1907.02893*.

Arunachalam, H. B., Mishra, R., Daescu, O., Cederberg, K., Rakheja, D., Sengupta, A., Leonard, D., Hallac, R. & Leavey, P. (2019). Viable and necrotic tumor assessment from whole slide images of osteosarcoma using machine-learning and deep-learning models. *PloS one*, 14(4), e0210706.

Aubreville, M., Bertram, C., Veta, M., Klopfleisch, R., Stathonikos, N., Breininger, K., ter Hoeve, N., Ciompi, F. & Maier, A. (2021). Quantifying the scanner-induced domain gap in mitosis detection. *arXiv preprint arXiv:2103.16515*.

Author, N. N. (2021). Suppressed for Anonymity.

Avidan, S., Brostow, G., Cissé, M., Farinella, G. M. & Hassner, T. (Eds.). (2022). *Computer Vision – ECCV 2022*. Springer. doi: 10.1007/978-3-031-19769-7.

Azizi, S., Kornblith, S., Saharia, C., Norouzi, M. & Fleet, D. J. (2023). Synthetic data from diffusion models improves imagenet classification. *arXiv preprint arXiv:2304.08466*.

Balaji, Y., Sankaranarayanan, S. & Chellappa, R. (2018). Metareg: Towards domain generalization using meta-regularization. *Advances in neural information processing systems*, 31.

Barsellotti, D., Amoroso, R. & Caputo, B. (2023). Enhancing Open-Vocabulary Semantic Segmentation with Prototype Retrieval. *Pattern Recognition. ICPR International Workshops and Challenges*, pp. 243–258.

Barsellotti, D., Amoroso, R. & Caputo, B. (2024). FOSSIL: Free Open-Vocabulary Semantic Segmentation through Synthetic References Retrieval. *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 1234–1243.

Beery, S., Van Horn, G. & Perona, P. (2018). Recognition in terra incognita. *Proceedings of the European conference on computer vision (ECCV)*, pp. 456–473.

Ben-David, S., Blitzer, J., Crammer, K., Kulesza, A., Pereira, F. & Vaughan, J. W. (2010). A theory of learning from different domains. *Machine learning*, 79, 151–175.

Blanchard, G., Lee, G. & Scott, C. (2011). Generalizing from several related classification tasks to a new unlabeled sample. *Advances in neural information processing systems*, 24.

Blanchard, G., Deshmukh, A. A., Dogan, U., Lee, G. & Scott, C. (2017). Domain generalization by marginal transfer learning. *arXiv preprint arXiv:1711.07910*.

Blanchard, G., Deshmukh, A. A., Dogan, Ü., Lee, G. & Scott, C. (2021). Domain generalization by marginal transfer learning. *The Journal of Machine Learning Research*, 22(1), 46–100.

Bolhasani, H., Amjadi, E., Tabatabaeian, M. & Jassbi, S. J. (2020). A histopathological image dataset for grading breast invasive ductal carcinomas. *Informatics in Medicine Unlocked*, 19, 100341.

Borkowski, A. A., Bui, M. M., Thomas, L. B., Wilson, C. P., DeLand, L. A. & Mastorides, S. M. (2019). Lung and colon cancer histopathological image dataset (lc25000). *arXiv preprint arXiv:1912.12142*.

Borlino, F. C., D'Innocente, A. & Tommasi, T. (2021). Rethinking domain generalization baselines. *2020 25th International Conference on Pattern Recognition (ICPR)*, pp. 9227–9233.

Boudiaf, M., Mueller, R., Ben Ayed, I. & Bertinetto, L. (2022). Parameter-free Online Test-time Adaptation. *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8344–8353.

Bousselham, W., Bursuc, A., Alldieck, T. & Pérez, P. (2024). Grounding Everything: Emerging Localization Properties in Vision-Language Transformers. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5678–5687.

Brummer, O., Pölönen, P., Mustjoki, S. & Brück, O. (2022). Integrative Analysis of Histological Textures and Lymphocyte Infiltration in Renal Cell Carcinoma using Deep Learning. *bioRxiv*, 2022–08.

Bucci, S., D'Innocente, A., Liao, Y., Carlucci, F. M., Caputo, B. & Tommasi, T. (2021). Self-supervised learning across domains. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(9), 5516–5528.

Bui, M.-H., Tran, T., Tran, A. & Phung, D. (2021). Exploiting domain-specific features to enhance domain generalization. *Advances in Neural Information Processing Systems*, 34, 21189–21201.

Caesar, H., Uijlings, J. R. R. & Ferrari, V. (2016). COCO-Stuff: Thing and Stuff Classes in Context. *CoRR*, abs/1612.03716. Retrieved from: http://arxiv.org/abs/1612.03716.

Cai, K., Ren, P., Zhu, Y., Xu, H., Liu, J., Li, C., Wang, G. & Liang, X. (2023). Mixreorg: Cross-modal mixed patch reorganization is a good mask learner for open-world semantic segmentation. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1196–1205.

Canny, J. (1986). A computational approach to edge detection. *IEEE Transactions on pattern analysis and machine intelligence*, (6), 679–698.

Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A. & Zagoruyko, S. (2020). End-to-end object detection with transformers. *European conference on computer vision*, pp. 213–229.

Carlucci, F. M., D'Innocente, A., Bucci, S., Caputo, B. & Tommasi, T. (2019a). Domain generalization by solving jigsaw puzzles. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 2229–2238.

Carlucci, F. M., Russo, P., Tommasi, T. & Caputo, B. (2019b). Hallucinating agnostic images to generalize across domains. *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, pp. 3227–3234.

Cha, J., Chun, S., Lee, K., Cho, H.-C., Park, S., Lee, Y. & Park, S. (2021). Swad: Domain generalization by seeking flat minima. *Advances in Neural Information Processing Systems*, 34, 22405–22418.

Cha, J., Mun, J. & Roh, B. (2023). Learning to generate text-grounded mask for open-world semantic segmentation from only image-text pairs. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11165–11174.

Chapelle, O., Weston, J., Bottou, L. & Vapnik, V. (2000). Vicinal risk minimization. *Advances in neural information processing systems*, 13.

Chapelle, O., Scholkopf, B. & Zien, A. (2006). Semi-supervised learning. 2006. *Cambridge, Massachusettes: The MIT Press View Article*, 2.

Chattopadhyay, P., Balaji, Y. & Hoffman, J. (2020). Learning to balance specificity and invariance for in and out of domain generalization. *European Conference on Computer Vision*, pp. 301–318.

Chen, D., Wang, D., Darrell, T. & Ebrahimi, S. (2022). Contrastive test-time adaptation. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 295–305.

Chen, Y., Li, Y., Zhang, Z. & Liu, Y. (2023). Exploring Open-Vocabulary Semantic Segmentation from CLIP Vision Encoder Distillation Only. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 10010–10019.

Cheraghalikhani, M., Noori, M., Osowiechi, D., Hakim, G. A. V., Ayed, I. B. & Desrosiers, C. (2024). Structure-aware feature stylization for domain generalization. *Computer Vision and Image Understanding*, 244, 104016.

Chidlovskii, B., Clinchant, S. & Csurka, G. (2016). Domain adaptation in the absence of source domain data. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 451–460.

Choi, H. K., Choi, J. & Kim, H. J. (2022). TokenMixup: Efficient Attention-guided Token-level Data Augmentation for Transformers. *arXiv preprint arXiv:2210.07562*.

Choi, Y., Choi, M., Kim, M., Ha, J.-W., Kim, S. & Choo, J. (2018). Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 8789–8797.

Chu, X., Jin, Y., Zhu, W., Wang, Y., Wang, X., Zhang, S. & Mei, H. (2022). DNA: Domain Generalization with Diversified Neural Averaging. *International Conference on Machine Learning*, pp. 4010–4034.

Colussi, M., Mascetti, S., Dolz, J. & Desrosiers, C. (2024). ReC-TTT: Contrastive Feature Reconstruction for Test-Time Training.

Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S. & Schiele, B. (2016). The Cityscapes Dataset for Semantic Urban Scene Understanding. *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Corradini, A., Bianchi, F., Nozza, D. & Hovy, D. (2024). FreeSeg-Diff: Training-Free Open-Vocabulary Segmentation with Diffusion Models. *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 567–576.

Dai, X., Chen, Y., Yang, J., Zhang, P., Yuan, L. & Zhang, L. (2021, October). Dynamic DETR: End-to-End Object Detection With Dynamic Attention. *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 2988-2997.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K. & Fei-Fei, L. (2009a). Imagenet: A large-scale hierarchical image database. *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K. & Fei-Fei, L. (2009b). Imagenet: A large-scale hierarchical image database. *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255.

Dhariwal, P. & Nichol, A. (2021). Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34, 8780–8794.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S. et al. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

Dou, Q., Coelho de Castro, D., Kamnitsas, K. & Glocker, B. (2019). Domain generalization via model-agnostic learning of semantic features. *Advances in Neural Information Processing Systems*, 32.

Duda, R. O., Hart, P. E. & Stork, D. G. (2000). *Pattern Classification* (ed. 2nd). John Wiley and Sons.

Dunlap, L., Umino, A., Zhang, H., Yang, J., Gonzalez, J. E. & Darrell, T. (2023). Diversify your vision datasets with automatic diffusion-based augmentation. *Advances in neural information processing systems*, 36, 79024–79034.

D'Innocente, A. & Caputo, B. (2018). Domain generalization with domain-specific aggregation modules. *German Conference on Pattern Recognition*, pp. 187–198.

Everingham, M., Gool, L. V., Williams, C. K. I., Winn, J. M. & Zisserman, A. (2010a). The Pascal Visual Object Classes (VOC) Challenge. *Int. J. Comput. Vis.*, 88(2), 303-338. Retrieved from: http://dblp.uni-trier.de/db/journals/ijcv/ijcv88.html#EveringhamGWWZ10.

Everingham, M., Van Gool, L., Williams, C. K., Winn, J. & Zisserman, A. (2010b). The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88, 303–338.

Fan, X., Wang, Q., Ke, J., Yang, F., Gong, B. & Zhou, M. (2021). Adversarially adaptive normalization for single domain generalization. *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*, pp. 8208–8217.

Fang, C., Xu, Y. & Rockmore, D. N. (2013). Unbiased metric learning: On the utilization of multiple datasets and web images for softening bias. *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1657–1664.

Fei-Fei, L., Fergus, R. & Perona, P. (2004). Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. *2004 conference on computer vision and pattern recognition workshop*, pp. 178–178.

Feng, C.-M., Yu, K., Liu, Y., Khan, S. & Zuo, W. (2023, October). Diverse Data Augmentation with Diffusions for Effective Test-time Prompt Tuning. *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 2704-2714.

Finn, C., Abbeel, P. & Levine, S. (2017). Model-agnostic meta-learning for fast adaptation of deep networks. *International conference on machine learning*, pp. 1126–1135.

Gan, C., Yang, T. & Gong, B. (2016). Learning attributes equals multi-source domain generalization. *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 87–97.

Gandelsman, Y., Sun, Y., Chen, X. & Efros, A. A. (2022). Test-Time Training with Masked Autoencoders. *Advances in Neural Information Processing Systems*. Retrieved from: https://openreview.net/forum?id=SHMi1b7sjXk.

Ganin, Y. & Lempitsky, V. (2015). Unsupervised domain adaptation by backpropagation. *International conference on machine learning*, pp. 1180–1189.

Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M. & Lempitsky, V. (2016). Domain-adversarial training of neural networks. *The journal of machine learning research*, 17(1), 2096–2030.

Garipov, T., Izmailov, P., Podoprikhin, D., Vetrov, D. P. & Wilson, A. G. (2018). Loss Surfaces, Mode Connectivity, and Fast Ensembling of DNNs. *Advances in Neural Information Processing Systems*, 31. Retrieved from: https://proceedings.neurips.cc/paper_files/paper/2018/file/be3087e74e9100d4bc4c6268cdbe8456-Paper.pdf.

Ghiasi, G., Cui, Y., Srinivas, A., Qian, R., Lin, T.-Y., Cubuk, E. D., Le, Q. V. & Zoph, B. (2022). Scaling Open-Vocabulary Image Segmentation with Image-Level Labels. *European Conference on Computer Vision*, pp. 343–360.

Ghifary, M., Bastiaan Kleijn, W., Zhang, M. & Balduzzi, D. (2015). Domain generalization for object recognition with multi-task autoencoders. *Proceedings of the IEEE international conference on computer vision*, pp. 2551–2559.

Gidaris, S., Singh, P. & Komodakis, N. (2018). Unsupervised representation learning by predicting image rotations. *arXiv preprint arXiv:1803.07728*.

Gong, B., Shi, Y., Sha, F. & Grauman, K. (2012). Geodesic flow kernel for unsupervised domain adaptation. *2012 IEEE conference on computer vision and pattern recognition*, pp. 2066–2073.

Gong, T., Kim, Y., Lee, T., Chottananurak, S. & Lee, S.-J. (2024). SoTTA: Robust Test-Time Adaptation on Noisy Data Streams. *Advances in Neural Information Processing Systems*, 36.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. & Bengio, Y. (2020). Generative adversarial networks. *Communications of the ACM*, 63(11), 139–144.

Goyal, S., Sun, M., Raghunanthan, A. & Kolter, Z. (2022). Test-time adaptation via conjugate pseudo-labels. *Advances in Neural Information Processing Systems*.

Grill, J.-B., Strub, F., Altché, F., Tallec, C., Richemond, P., Buchatskaya, E., Doersch, C., Avila Pires, B., Guo, Z., Gheshlaghi Azar, M. et al. (2020). Bootstrap your own latent-a new approach to self-supervised learning. *Advances in Neural Information Processing Systems*, 33, 21271–21284.

Gulrajani, I. & Lopez-Paz, D. (2020). In search of lost domain generalization. *arXiv preprint arXiv:2007.01434*.

Gulrajani, I. & Lopez-Paz, D. (2021). In Search of Lost Domain Generalization. *ArXiv*, abs/2007.01434.

Guzhov, A., Raue, F., Hees, J. & Dengel, A. (2022a). Audioclip: Extending clip to image, text and audio. *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 976–980.

Guzhov, A., Raue, F., Hees, J. & Dengel, A. (2022b). Audioclip: Extending clip to image, text and audio. *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 976–980.

Hajimiri, S., Ben Ayed, I. & Dolz, J. (2025). Pay Attention to Your Neighbours: Training-Free Open-Vocabulary Semantic Segmentation. *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*.

Hakim, G. A. V., Osowiechi, D., Noori, M., Cheraghalikhani, M., Bahri, A., Ben Ayed, I. & Desrosiers, C. (2023). ClusT3: Information Invariant Test-Time Training. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 6136–6145.

Hakim, G. A. V., Osowiechi, D., Noori, M., Cheraghalikhani, M., Bahri, A., Yazdanpanah, M., Ayed, I. B. & Desrosiers, C. (2024). CLIPArTT: Light-weight Adaptation of CLIP to New Domains at Test Time. *arXiv preprint arXiv:2405.00754*. Retrieved from: https://arxiv.org/abs/2405.00754.

He, J., Zhang, Z., Wang, Z. & Huang, Y. (2021). Autoencoder based test-time adaptation for semantic segmentation. *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 998–1007.

He, K., Zhang, X., Ren, S. & Sun, J. (2016). Deep residual learning for image recognition. *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778.

He, K., Chen, X., Xie, S., Li, Y., Dollár, P. & Girshick, R. (2022). Masked autoencoders are scalable vision learners. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16000–16009.

Hemati, S., Beitollahi, M., Estiri, A. H., Omari, B. A., Chen, X. & Zhang, G. (2023). Cross Domain Generative Augmentation: Domain Generalization with Latent Diffusion Models. *arXiv preprint arXiv:2312.05387*.

Hendrycks, D. & Dietterich, T. (2019a). Benchmarking neural network robustness to common corruptions and perturbations.

Hendrycks, D. & Dietterich, T. (2019b). Benchmarking neural network robustness to common corruptions and perturbations. *arXiv preprint arXiv:1903.12261*.

Hendrycks, D. & Dietterich, T. G. (2019c). Benchmarking Neural Network Robustness to Common Corruptions and Perturbations. *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*.

Hinton, G. E. & Roweis, S. (2002). Stochastic neighbor embedding. *Advances in neural information processing systems*, 15.

Ho, J., Jain, A. & Abbeel, P. (2020). Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33, 6840–6851.

Hoque, M. Z., Keskinarkaus, A., Nyberg, P. & Seppänen, T. (2024). Stain normalization methods for histopathology image analysis: A comprehensive review and experimental comparison. *Information Fusion*, 102, 101997. doi: https://doi.org/10.1016/j.inffus.2023.101997.

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., Chen, W. et al. (2022). Lora: Low-rank adaptation of large language models. *ICLR*, 1(2), 3.

Hu, S., Zhang, K., Chen, Z. & Chan, L. (2020). Domain generalization via multidomain discriminant analysis. *Uncertainty in Artificial Intelligence*, pp. 292–302.

Huang, X. & Belongie, S. (2017). Arbitrary style transfer in real-time with adaptive instance normalization. *Proceedings of the IEEE international conference on computer vision*, pp. 1501–1510.

Huang, Z., Wang, H., Xing, E. P. & Huang, D. (2020). Self-challenging improves cross-domain generalization. *European Conference on Computer Vision*, pp. 124–140.

Huang, Z., Bianchi, F., Yuksekgonul, M., Montine, T. J. & Zou, J. (2023). A visual–language foundation model for pathology image analysis using medical twitter. *Nature medicine*, 29(9), 2307–2316.

Hussein, N., Shamshad, F., Naseer, M. & Nandakumar, K. (2024). Promptsmooth: Certifying robustness of medical vision-language models via prompt learning. *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 698–708.

Ikezogwo, W., Seyfioglu, S., Ghezloo, F., Geva, D., Sheikh Mohammed, F., Anand, P. K., Krishna, R. & Shapiro, L. (2024). Quilt-1m: One million image-text pairs for histopathology. *Advances in neural information processing systems*, 36.

Ilse, M., Tomczak, J. M., Louizos, C. & Welling, M. (2020). Diva: Domain invariant variational autoencoders. *Medical Imaging with Deep Learning*, pp. 322–348.

Iscen, A., Tolias, G., Avrithis, Y. & Chum, O. (2019). Label propagation for deep semi-supervised learning. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 5070–5079.

Izmailov, P., Podoprikhin, D., Garipov, T., Vetrov, D. & Wilson, A. G. (2018a). Averaging weights leads to wider optima and better generalization. *arXiv preprint arXiv:1803.05407*.

Izmailov, P., Podoprikhin, D., Garipov, T., Vetrov, D. & Wilson, A. G. (2018b). Averaging weights leads to wider optima and better generalization. *arXiv preprint arXiv:1803.05407*.

Jang, M., Chung, S.-Y. & Chung, H. W. (2022). Test-time adaptation via self-training with nearest neighbor information. *arXiv preprint arXiv:2207.10792*.

Jeon, S., Hong, K., Lee, P., Lee, J. & Byun, H. (2021). Feature stylization and domain-aware contrastive learning for domain generalization. *Proceedings of the 29th ACM International Conference on Multimedia*, pp. 22–31.

Jia, C., Yang, Y., Xia, Y., Chen, Y.-T., Parekh, Z., Pham, H., Le, Q., Sung, Y.-H., Li, Z. & Duerig, T. (2021). Scaling up visual and vision-language representation learning with noisy text supervision. *International conference on machine learning*, pp. 4904–4916.

Jiang, C., Liao, J., Dong, P., Ma, Z., Cai, D., Zheng, G., Liu, Y., Bu, H. & Yao, J. (2020). Blind deblurring for microscopic pathology images using deep learning networks. *arXiv preprint arXiv:2011.11879*.

Jin, X., Lan, C., Zeng, W. & Chen, Z. (2020). Feature alignment and restoration for domain generalization and adaptation. *arXiv preprint arXiv:2006.12009*.

Jurgas, A., Wodzinski, M., D'Amato, M., Van Der Laak, J., Atzori, M. & Müller, H. (2024). Improving quality control of whole slide images by explicit artifact augmentation. *Scientific Reports*, 14(1), 17847. doi: 10.1038/s41598-024-68667-2.

Karazija, L., Bousselham, W., Bursuc, A., Alldieck, T. & Pérez, P. (2023). Diffusion Models for Zero-Shot Open-Vocabulary Segmentation. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1010–1020.

Karmanov, A., Guan, D., Lu, S., El Saddik, A. & Xing, E. (2024, June). Efficient Test-Time Adaptation of Vision-Language Models. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14162-14171.

Kather, J. N., Halama, N. & Marx, A. (2024). 100,000 histological images of human colorectal cancer and healthy tissue, April 2018. *URL https://doi. org/10.5281/zenodo*, 1214456.

Kearns, M. J. (1989). *Computational Complexity of Machine Learning*. (Ph.D. thesis, Department of Computer Science, Harvard University).

Kim, D., Yoo, Y., Park, S., Kim, J. & Lee, J. (2021a, October). SelfReg: Self-Supervised Contrastive Regularization for Domain Generalization. *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9619-9628.

Kim, D., Yoo, Y., Park, S., Kim, J. & Lee, J. (2021b). Selfreg: Self-supervised contrastive regularization for domain generalization. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9619–9628.

Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A. C., Lo, W.-Y. et al. (2023). Segment anything. *arXiv preprint arXiv:2304.02643*.

Kohlberger, T., Liu, Y., Moran, M., Chen, P.-H. C., Brown, T., Hipp, J. D., Mermel, C. H. & Stumpe, M. C. (2019a). Whole-Slide Image Focus Quality: Automatic Assessment and Impact on AI Cancer Detection. *Journal of Pathology Informatics*, 10(1), 39. doi: https://doi.org/10.4103/jpi.jpi_11_19.

Kohlberger, T., Liu, Y., Moran, M., Chen, P.-H. C., Brown, T., Hipp, J. D., Mermel, C. H. & Stumpe, M. C. (2019b). Whole-slide image focus quality: Automatic assessment and impact on ai cancer detection. *Journal of pathology informatics*, 10(1), 39.

Kriegsmann, K., Lobers, F., Zgorzelski, C., Kriegsmann, J., Janßen, C., Meliß, R. R., Muley, T., Sack, U., Steinbuss, G. & Kriegsmann, M. (2022). Deep learning for the detection of anatomical tissue structures and neoplasms of the skin on scanned histopathological tissue sections. *Frontiers in Oncology*, 12, 1022967.

Krizhevsky, A., Sutskever, I. & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25.

Krueger, D., Caballero, E., Jacobsen, J.-H., Zhang, A., Binas, J., Zhang, D., Le Priol, R. & Courville, A. (2021). Out-of-distribution generalization via risk extrapolation (rex). *International Conference on Machine Learning*, pp. 5815–5826.

Kulis, B., Saenko, K. & Darrell, T. (2011). What you saw is not what you get: Domain adaptation using asymmetric kernel transforms. *CVPR 2011*, pp. 1785–1792.

Lai, Z., Vesdapunt, N., Zhou, N., Wu, J., Huynh, C. P., Li, X., Fu, K. K. & Chuah, C.-N. (2023, October). PADCLIP: Pseudo-labeling with Adaptive Debiasing in CLIP for Unsupervised Domain Adaptation. *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 16155-16165.

Laine, R. F., Jacquemet, G. & Krull, A. (2021). Imaging in focus: An introduction to denoising bioimages in the era of deep learning. *The International Journal of Biochemistry and Cell Biology*, 140, 106077. doi: https://doi.org/10.1016/j.biocel.2021.106077.

Lampert, C. H., Nickisch, H. & Harmeling, S. (2013). Attribute-based classification for zero-shot visual object categorization. *IEEE transactions on pattern analysis and machine intelligence*, 36(3), 453–465.

Langley, P. (2000). Crafting Papers on Machine Learning. *Proceedings of the 17th International Conference on Machine Learning (ICML 2000)*, pp. 1207–1216.

LeCun, Y., Bottou, L., Bengio, Y. & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11), 2278–2324.

LeCun, Y., Bengio, Y. & Hinton, G. (2015). Deep learning. *nature*, 521(7553), 436–444.

Lee, J., Das, D., Choo, J. & Choi, S. (2023, October). Towards Open-Set Test-Time Adaptation Utilizing the Wisdom of Crowds in Entropy Minimization. *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 16380-16389.

Li, B., Wang, Y., Zhang, S., Li, D., Keutzer, K., Darrell, T. & Zhao, H. (2021). Learning invariant representations and risks for semi-supervised domain adaptation. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1104–1113.

Li, D., Yang, Y., Song, Y.-Z. & Hospedales, T. M. (2017). Deeper, broader and artier domain generalization. *Proceedings of the IEEE international conference on computer vision*, pp. 5542–5550.

Li, D., Yang, Y., Song, Y.-Z. & Hospedales, T. (2018a). Learning to generalize: Meta-learning for domain generalization. *Proceedings of the AAAI conference on artificial intelligence*, 32(1).

Li, D., Zhang, J., Yang, Y., Liu, C., Song, Y.-Z. & Hospedales, T. M. (2019a). Episodic training for domain generalization. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1446–1455.

Li, H., Pan, S. J., Wang, S. & Kot, A. C. (2018b). Domain generalization with adversarial feature learning. *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5400–5409.

Li, H., Wang, Y., Wan, R., Wang, S., Li, T.-Q. & Kot, A. (2020). Domain generalization for medical imaging classification with linear-dependency regularization. *Advances in Neural Information Processing Systems*, 33, 3118–3129.

Li, Y., Tian, X., Gong, M., Liu, Y., Liu, T., Zhang, K. & Tao, D. (2018c). Deep domain generalization via conditional invariant adversarial networks. *Proceedings of the European conference on computer vision (ECCV)*, pp. 624–639.

Li, Y., Yang, Y., Zhou, W. & Hospedales, T. (2019b). Feature-critic networks for heterogeneous domain generalization. *International Conference on Machine Learning*, pp. 3915–3924.

Liang, F., Wu, B., Dai, X., Li, K., Zhao, Y., Zhang, H., Zhang, P., Vajda, P. & Marculescu, D. (2023). Open-vocabulary semantic segmentation with mask-adapted clip. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7061–7070.

Liang, J., Hu, D. & Feng, J. (2020). Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation. *International conference on machine learning*, pp. 6028–6039.

Lin, G., Lai, H., Pan, Y. & Yin, J. (2023). Improving Entropy-Based Test-Time Adaptation from a Clustering View. *arXiv preprint arXiv:2310.20327*.

Lin, T., Maire, M., Belongie, S. J., Bourdev, L. D., Girshick, R. B., Hays, J., Perona, P., Ramanan, D., Dollár, P. & Zitnick, C. L. (2014). Microsoft COCO: Common Objects in Context. *CoRR*, abs/1405.0312. Retrieved from: http://arxiv.org/abs/1405.0312.

Lin, Z., Geng, S., Zhang, R., Gao, P., de Melo, G., Wang, X., Dai, J., Qiao, Y. & Li, H. (2022a). Frozen clip models are efficient video learners. *European Conference on Computer Vision*, pp. 388–404.

Lin, Z., Geng, S., Zhang, R., Gao, P., De Melo, G., Wang, X., Dai, J., Qiao, Y. & Li, H. (2022b). Frozen clip models are efficient video learners. *European Conference on Computer Vision*, pp. 388–404.

Liu, H., Kang, G., You, S., Bao, J., Li, M. & Zhang, Y. (2021a). Source-free domain adaptation for semantic segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Liu, J., Zhang, Y., Chen, J.-N., Xiao, J., Lu, Y., A Landman, B., Yuan, Y., Yuille, A., Tang, Y. & Zhou, Z. (2023a). Clip-driven universal model for organ segmentation and tumor detection. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 21152–21164.

Liu, J., Zhang, Y., Chen, J.-N., Xiao, J., Lu, Y., A Landman, B., Yuan, Y., Yuille, A., Tang, Y. & Zhou, Z. (2023b). Clip-driven universal model for organ segmentation and tumor detection. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 21152–21164.

Liu, Q., Dou, Q. & Heng, P.-A. (2020a). Shape-aware meta-learning for generalizing prostate MRI segmentation to unseen domains. *Medical Image Computing and Computer Assisted Intervention–MICCAI 2020: 23rd International Conference, Lima, Peru, October 4–8, 2020, Proceedings, Part II 23*, pp. 475–485.

Liu, Q., Dou, Q., Yu, L. & Heng, P. A. (2020b). MS-Net: multi-site network for improving prostate segmentation with heterogeneous MRI data. *IEEE transactions on medical imaging*, 39(9), 2713–2724.

Liu, Y., Kothari, P., van Delft, B., Bellot-Gurlet, B., Mordan, T. & Alahi, A. (2021b). TTT++: When Does Self-Supervised Test-Time Training Fail or Thrive? *Neural Information Processing Systems (NeurIPS)*.

Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S. & Guo, B. (2021c). Swin transformer: Hierarchical vision transformer using shifted windows. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 10012–10022.

Liu, Z., Miao, Z., Pan, X., Zhan, X., Lin, D., Yu, S. X. & Gong, B. (2020c). Open compound domain adaptation. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12406–12415.

Long, M., Cao, Y., Wang, J. & Jordan, M. (2015). Learning transferable features with deep adaptation networks. *International conference on machine learning*, pp. 97–105.

Lu, M. Y., Chen, B., Williamson, D. F., Chen, R. J., Liang, I., Ding, T., Jaume, G., Odintsov, I., Le, L. P., Gerber, G. et al. (2024). A visual-language foundation model for computational pathology. *Nature Medicine*, 30(3), 863–874.

Lu, Z., Yang, Y., Zhu, X., Liu, C., Song, Y.-Z. & Xiang, T. (2020). Stochastic classifiers for unsupervised domain adaptation. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9111–9120.

Lu, Z., He, S., Zhu, X., Zhang, L., Song, Y.-Z. & Xiang, T. (2021). Simpler is better: Few-shot semantic segmentation with classifier weight transformer. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 8741–8750.

Ma, X., Zhang, J., Guo, S. & Xu, W. (2024). Swapprompt: Test-time prompt adaptation for vision-language models. *Advances in Neural Information Processing Systems*, 36.

Mahajan, D., Tople, S. & Sharma, A. (2021). Domain generalization using causal matching. *International Conference on Machine Learning*, pp. 7313–7324.

Majzoub, R. A., Malik, H., Naseer, M., Zaheer, Z., Mahmood, T., Khan, S. & Khan, F. (2025). How good is my histopathology vision-language foundation model? a holistic benchmark. *arXiv preprint arXiv:2503.12990*.

Mallya, A., Davis, D. & Lazebnik, S. (2018). Piggyback: Adapting a single network to multiple tasks by learning to mask weights. *Proceedings of the European conference on computer vision (ECCV)*, pp. 67–82.

Mancini, M., Bulò, S. R., Caputo, B. & Ricci, E. (2018). Best sources forward: domain generalization through source-specific nets. *2018 25th IEEE International Conference on Image Processing (ICIP)*, pp. 1353–1357.

Mancini, M., Akata, Z., Ricci, E. & Caputo, B. (2020). Towards recognizing unseen categories in unseen domains. *European Conference on Computer Vision*, pp. 466–483.

Marza, P., Fillioux, L., Boutaj, S., Mahatha, K., Desrosiers, C., Piantanida, P., Dolz, J., Christodoulidis, S. & Vakalopoulou, M. (2025). THUNDER: Tile-level Histopathology image UNDERstanding benchmark. *arXiv preprint arXiv:2507.07860*.

Miao, Q., Yuan, J., Zhang, S., Wu, F. & Kuang, K. (2024). Domaindiff: Boost out-of-Distribution Generalization with Synthetic Data. *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5640–5644.

Michalski, R. S., Carbonell, J. G. & Mitchell, T. M. (Eds.). (1983). *Machine Learning: An Artificial Intelligence Approach, Vol. I.* Palo Alto, CA: Tioga.

Mitchell, T. M. (1980). *The Need for Biases in Learning Generalizations*. New Brunswick, MA.

Moreno-Torres, J. G., Raeder, T., Alaiz-Rodríguez, R., Chawla, N. V. & Herrera, F. (2012). A unifying view on dataset shift in classification. *Pattern recognition*, 45(1), 521–530.

Motiian, S., Piccirilli, M., Adjeroh, D. A. & Doretto, G. (2017). Unified deep supervised domain adaptation and generalization. *Proceedings of the IEEE international conference on computer vision*, pp. 5715–5725.

Mottaghi, R., Chen, X., Liu, X., Cho, N.-G., Lee, S.-W., Fidler, S., Urtasun, R. & Yuille, A. (2014). The Role of Context for Object Detection and Semantic Segmentation in the Wild. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Muandet, K., Balduzzi, D. & Schölkopf, B. (2013). Domain generalization via invariant feature representation. *International conference on machine learning*, pp. 10–18.

Müller, S. G. & Hutter, F. (2021). Trivialaugment: Tuning-free yet state-of-the-art data augmentation. *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 774–782.

Mummadi, C. D., Arens, M. & Brox, T. (2021). Test-time adaptation for continual semantic segmentation. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11828–11837.

Nado, Z., Padhy, S., Sculley, D., D'Amour, A., Lakshminarayanan, B. & Snoek, J. (2020). Evaluating prediction-time batch normalization for robustness under covariate shift. *arXiv preprint arXiv:2006.10963*.

Nado, Z., Padhy, S., Sculley, D., D'Amour, A., Lakshminarayanan, B. & Snoek, J. (2021). Evaluating prediction-time batch normalization for robustness under covariate shift. *arXiv:2006.10963 [cs, stat]*. Retrieved from: http://arxiv.org/abs/2006.10963. arXiv: 2006.10963.

Najdenkoska, I., Derakhshani, M. M., Asano, Y. M., Van Noord, N., Worring, M. & Snoek, C. G. (2024). Tulip: Token-length upgraded clip. *arXiv preprint arXiv:2410.10034*.

Nam, H., Lee, H., Park, J., Yoon, W. & Yoo, D. (2021a, June). Reducing Domain Gap by Reducing Style Bias. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8690-8699.

Nam, H., Lee, H., Park, J., Yoon, W. & Yoo, D. (2021b). Reducing domain gap by reducing style bias. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8690–8699.

Neary-Zajiczek, L., Beresna, L., Razavi, B., Pawar, V., Shaw, M. & Stoyanov, D. (2023). Minimum resolution requirements of digital pathology images for accurate classification. *Medical Image Analysis*, 89, 102891.

Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B. & Ng, A. Y. (2011). Reading digits in natural images with unsupervised feature learning.

Newell, A. & Rosenbloom, P. S. (1981). Mechanisms of Skill Acquisition and the Law of Practice. In Anderson, J. R. (Ed.), *Cognitive Skills and Their Acquisition* (ch. 1, pp. 1–51). Hillsdale, NJ: Lawrence Erlbaum Associates, Inc.

Niu, L., Li, W. & Xu, D. (2015). Multi-view domain generalization for visual recognition. *Proceedings of the IEEE international conference on computer vision*, pp. 4193–4201.

Niu, S., Wu, J., Zhang, Y., Chen, Y., Zheng, S., Zhao, P. & Tan, M. (2022, 17–23 Jul). Efficient Test-Time Model Adaptation without Forgetting. *Proceedings of the 39th International Conference on Machine Learning*, 162(Proceedings of Machine Learning Research), 16888–16905.

Niu, S., Wu, J., Zhang, Y., Wen, Z., Chen, Y., Zhao, P. & Tan, M. (2023). Towards Stable Test-Time Adaptation in Dynamic Wild World. Retrieved from: https://arxiv.org/abs/2302.12400.

Noori, M., Cheraghalikhani, M., Bahri, A., Hakim, G. A. V., Osowiechi, D., Ayed, I. B. & Desrosiers, C. (2024a). TFS-ViT: Token-level feature stylization for domain generalization. *Pattern Recognition*, 149, 110213.

Noori, M., Cheraghalikhani, M., Bahri, A., Hakim, G. A. V., Osowiechi, D., Ayed, I. B. & Desrosiers, C. (2024b). Tfs-vit: Token-level feature stylization for domain generalization. *Pattern Recognition*, 149, 110213.

Noori, M., Cheraghalikhani, M., Bahri, A., Hakim, G. A. V., Osowiechi, D., Yazdanpanah, M., Ayed, I. B. & Desrosiers, C. (2024c). FDS: Feedback-guided Domain Synthesis with Multi-Source Conditional Diffusion Models for Domain Generalization. *arXiv preprint arXiv:2407.03588*.

Noori, M., Cheraghalikhani, M., Bahri, A., A Vargas Hakim, G., Osowiechi, D., Yazdanpanah, M., Ben Ayed, I. & Desrosiers, C. (2025a, February). FDS: Feedback-Guided Domain Synthesis with Multi-Source Conditional Diffusion Models for Domain Generalization. *Proceedings of the Winter Conference on Applications of Computer Vision (WACV)*, pp. 8493-8503.

Noori, M., Osowiechi, D., Vargas Hakim, G., Bahri, A., Yazdanpanah, M., Dastani, S., Beizaee, F., Ayed, I. & Desrosiers, C. (2025b). Test-Time Adaptation of Vision-Language Models for Open-Vocabulary Semantic Segmentation. *Advances in Neural Information Processing Systems*. Retrieved from: https://openreview.net/forum?id=CH76rSKWZr.

Noori, M., Hakim, G. A. V., Osowiechi, D., Shakeri, F., Bahri, A., Yazdanpanah, M., Dastani, S., Ben Ayed, I. & Desrosiers, C. (2026). Histopath-C: Towards Realistic Domain Shifts for Histopathology Vision-Language Adaptation. *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 4890–4900.

Nuriel, O., Benaim, S. & Wolf, L. (2021). Permuted AdaIN: reducing the bias towards global statistics in image classification. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9482–9491.

Ochi, M., Komura, D., Onoyama, T., Shinbo, K., Endo, H., Odaka, H., Kakiuchi, M., Katoh, H., Ushiku, T. & Ishikawa, S. (2024a). Registered multi-device/staining histology image dataset for domain-agnostic machine learning models. *Scientific Data*, 11(1), 330. doi: 10.1038/s41597-024-03122-5.

Ochi, M., Komura, D., Onoyama, T., Shinbo, K., Endo, H., Odaka, H., Kakiuchi, M., Katoh, H., Ushiku, T. & Ishikawa, S. (2024b). Registered multi-device/staining histology image dataset for domain-agnostic machine learning models. *Scientific Data*, 11(1), 330. doi: 10.1038/s41597-024-03122-5.

Osowiechi, D., Hakim, G. A. V., Noori, M., Cheraghalikhani, M., Ben Ayed, I. & Desrosiers, C. (2023a). TTTFlow: Unsupervised Test-Time Training with Normalizing Flow. *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 2126–2134.

Osowiechi, D., Vargas Hakim, G. A., Noori, M., Cheraghalikhani, M., Ayed, I. & Desrosiers, C. (2023b, jan). TTTFlow: Unsupervised Test-Time Training with Normalizing Flow. *2023 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 2125-2126. doi: 10.1109/WACV56688.2023.00216.

Osowiechi, D., Hakim, G. A. V., Noori, M., Cheraghalikhani, M., Bahri, A., Yazdanpanah, M., Ben Ayed, I. & Desrosiers, C. (2024a, June). NC-TTT: A Noise Constrastive Approach for Test-Time Training. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6078-6086.

Osowiechi, D., Noori, M., Vargas Hakim, G., Yazdanpanah, M., Bahri, A., Cheraghalikhani, M., Dastani, S., Beizaee, F., Ayed, I. & Desrosiers, C. (2024b). WATT: Weight Average Test Time Adaptation of CLIP. *Advances in Neural Information Processing Systems*, 37, 48015–48044. Retrieved from: https://proceedings.neurips.cc/paper_files/paper/2024/file/55d16334943f8728073e17139e5baa3d-Paper-Conference.pdf.

Pan, S. J. & Yang, Q. (2009). A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10), 1345–1359.

Pandey, P., Raman, M., Varambally, S. & AP, P. (2021). Domain generalization via inference-time label-preserving target projections. *arXiv preprint arXiv:2103.01134*.

Peng, X., Usman, B., Kaushik, N., Wang, D., Hoffman, J. & Saenko, K. (2018, June). VisDA: A Synthetic-to-Real Benchmark for Visual Domain Adaptation. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*.

Peng, X., Bai, Q., Xia, X., Huang, Z., Saenko, K. & Wang, B. (2019). Moment matching for multi-source domain adaptation. *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 1406–1415.

Podell, D., English, Z., Lacey, K., Blattmann, A., Dockhorn, T., Müller, J., Penna, J. & Rombach, R. (2023). Sdxl: Improving latent diffusion models for high-resolution image synthesis. *arXiv preprint arXiv:2307.01952*.

Prabhakara, S., Golden, D., Pilgrim, R., Eswaran, K. & Sellergren, A. ELIXR: Towards a general purpose X-ray artificial intelligence system through alignment of large language models and radiology vision encoders.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J. et al. (2021a). Learning transferable visual models from natural language supervision. *International conference on machine learning*, pp. 8748–8763.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J. et al. (2021b). Learning transferable visual models from natural language supervision. *International conference on machine learning*, pp. 8748–8763.

Rame, A., Dancette, C. & Cord, M. (2022a). Fishr: Invariant gradient variances for out-of-distribution generalization. *International Conference on Machine Learning*, pp. 18347–18377.

Rame, A., Kirchmeyer, M., Rahier, T., Rakotomamonjy, A., Gallinari, P. & Cord, M. (2022b). Diverse weight averaging for out-of-distribution generalization. *Advances in Neural Information Processing Systems*, 35, 10821–10836.

Recht, B., Roelofs, R., Schmidt, L. & Shankar, V. (2019). Do imagenet classifiers generalize to imagenet? *International conference on machine learning*, pp. 5389–5400.

Richter, S. R., Vineet, V., Roth, S. & Koltun, V. (2016). Playing for data: Ground truth from computer games. *European conference on computer vision*, pp. 102–118.

Rombach, R., Blattmann, A., Lorenz, D., Esser, P. & Ommer, B. (2022). High-resolution image synthesis with latent diffusion models. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10684–10695.

Ronneberger, O., Fischer, P. & Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, pp. 234–241.

Ruifrok, A. C., Johnston, D. A. et al. (2001). Quantification of histochemical staining by color deconvolution. *Analytical and quantitative cytology and histology*, 23(4), 291–299.

Russell, B. C., Torralba, A., Murphy, K. P. & Freeman, W. T. (2008). LabelMe: a database and web-based tool for image annotation. *International journal of computer vision*, 77, 157–173.

Saenko, K., Kulis, B., Fritz, M. & Darrell, T. (2010). Adapting visual category models to new domains. *Computer Vision–ECCV 2010: 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part IV 11*, pp. 213–226.

Sagawa, S., Koh, P. W., Hashimoto, T. B. & Liang, P. (2019). Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. *arXiv preprint arXiv:1911.08731*.

Sain, S. R. (1996). The nature of statistical learning theory. Taylor & Francis.

Saito, K., Watanabe, K., Ushiku, Y. & Harada, T. (2018). Maximum classifier discrepancy for unsupervised domain adaptation. *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3723–3732.

Sakaridis, C., Dai, D. & Van Gool, L. (2021). ACDC: The adverse conditions dataset with correspondences for semantic driving scene understanding. *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 10765–10775.

Samuel, A. L. (1959). Some Studies in Machine Learning Using the Game of Checkers. *IBM Journal of Research and Development*, 3(3), 211–229.

Satapute, M., P, S. & Gu, K. (2020). Artifacts: A menace in histopathology. *International Journal of Clinical and Diagnostic Pathology*, 3(1), 290–292. doi: 10.33545/pathol.2020.v3.i1e.185.

Scalbert, M., Vakalopoulou, M. & Couzinié-Devy, F. (2022). Test-Time Image-to-Image Translation Ensembling Improves Out-of-Distribution Generalization in Histopathology. *Medical Image Computing and Computer Assisted Intervention – MICCAI 2022*, pp. 120–129.

Senaras, C., Niazi, M. K. K., Lozanski, G. & Gurcan, M. N. (2018). DeepFocus: detection of out-of-focus regions in whole slide digital images using deep learning. *PloS one*, 13(10), e0205387.

Seo, S., Suh, Y., Kim, D., Kim, G., Han, J. & Han, B. (2020). Learning to optimize domain specific normalization for domain generalization. *European Conference on Computer Vision*, pp. 68–83.

Shafer, G. & Vovk, V. (2008). A Tutorial on Conformal Prediction. *Journal of Machine Learning Research*, 9(3).

Shakeri, F., Huang, Y., Silva-Rodríguez, J., Bahig, H., Tang, A., Dolz, J. & Ben Ayed, I. (2024). Few-shot adaptation of medical vision-language models. *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 553–563.

Shankar, S., Piratla, V., Chakrabarti, S., Chaudhuri, S., Jyothi, P. & Sarawagi, S. (2018). Generalizing across domains via cross-gradient training. *arXiv preprint arXiv:1804.10745*.

Sharifi-Noghabi, H., Asghari, H., Mehrasa, N. & Ester, M. (2020). Domain generalization via semi-supervised meta learning. *arXiv preprint arXiv:2009.12658*.

Shi, B., Zhang, D., Dai, Q., Zhu, Z., Mu, Y. & Wang, J. (2020a). Informative dropout for robust representation learning: A shape-bias perspective. *International Conference on Machine Learning*, pp. 8828–8839.

Shi, Y., Yu, X., Sohn, K., Chandraker, M. & Jain, A. K. (2020b). Towards universal representation learning for deep face recognition. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6817–6826.

Shu, M., Nie, W., Huang, D.-A., Yu, Z., Goldstein, T., Anandkumar, A. & Xiao, C. (2022). Test-time prompt tuning for zero-shot generalization in vision-language models. *Advances in Neural Information Processing Systems*, 35, 14274–14289.

Shu, Y., Cao, Z., Wang, C., Wang, J. & Long, M. (2021). Open domain generalization with domain-augmented meta-learning. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9624–9633.

Silva-Rodríguez, J., Colomer, A., Sales, M. A., Molina, R. & Naranjo, V. (2020). Going deeper through the Gleason scoring scale: An automatic end-to-end system for histology prostate grading and cribriform pattern detection. *Computer methods and programs in biomedicine*, 195, 105637.

Silva-Rodríguez, J., Chakor, H., Kobbi, R., Dolz, J. & Ben Ayed, I. (2025). A Foundation Language-Image Model of the Retina (FLAIR): encoding expert knowledge in text supervision. *Medical Image Analysis*, 99, 103357.

Somavarapu, N., Ma, C.-Y. & Kira, Z. (2020). Frustratingly simple domain generalization via image stylization. *arXiv preprint arXiv:2006.11207*.

Song, J., Meng, C. & Ermon, S. (2020). Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*.

Strudel, R., Garcia, R., Laptev, I. & Schmid, C. (2021). Segmenter: Transformer for semantic segmentation. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 7262–7272.

Sudre, C. H., Li, W., Vercauteren, T., Ourselin, S. & Jorge Cardoso, M. (2017). Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations. In *Deep learning in medical image analysis and multimodal learning for clinical decision support* (pp. 240–248). Springer.

Sultana, M., Naseer, M., Khan, M. H., Khan, S. & Khan, F. S. (2022). Self-Distilled Vision Transformer for Domain Generalization. *arXiv preprint arXiv:2207.12392*.

Sun, B. & Saenko, K. (2016). Deep coral: Correlation alignment for deep domain adaptation. *Computer Vision–ECCV 2016 Workshops: Amsterdam, The Netherlands, October 8-10 and 15-16, 2016, Proceedings, Part III 14*, pp. 443–450.

Sun, Y., Wang, X., Liu, Z., Miller, J., Efros, A. A. & Hardt, M. (2020). Test-time training with self-supervision for generalization under distribution shifts. *International Conference on Machine Learning (ICML)*.

Sun, Y., Zhang, Y., Si, Y., Zhu, C., Shui, Z., Zhang, K., Li, J., Lyu, X., Lin, T. & Yang, L. (2024). PathGen-1.6M: 1.6 Million Pathology Image-text Pairs Generation through Multi-agent Collaboration. Retrieved from: https://arxiv.org/abs/2407.00203.

Tak, Y.-O., Park, A., Choi, J., Eom, J., Kwon, H.-S. & Eom, J. B. (2020). Simple shading correction method for brightfield whole slide imaging. *Sensors*, 20(11), 3084.

Taori, R., Dave, A., Shankar, V., Carlini, N., Recht, B. & Schmidt, L. (2020). Measuring robustness to natural distribution shifts in image classification. *Advances in Neural Information Processing Systems*, 33, 18583–18599.

Taqi, S. A., Sami, S. A., Sami, L. B. & Zaki, S. A. (2018). A review of artifacts in histopathology. *Journal of Oral and Maxillofacial Pathology*, 22(2), 279–279. doi: 10.4103/jomfp.JOMFP_125_15.

Tellez, D., Balkenhol, M., Otte-Höller, I., van de Loo, R., Vogels, R., Bult, P., Wauters, C., Vreuls, W., Mol, S., Karssemeijer, N., Litjens, G., van der Laak, J. & Ciompi, F. (2018). Whole-Slide Mitosis Detection in H&E Breast Histology Using PHH3 as a Reference to Train Distilled Stain-Invariant Convolutional Networks. *IEEE Transactions on Medical Imaging*, 37(9), 2126-2136.

Torralba, A. & Efros, A. A. (2011). Unbiased look at dataset bias. *CVPR 2011*, pp. 1521–1528.

Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A. & Jégou, H. (2021). Training data-efficient image transformers & distillation through attention. *International Conference on Machine Learning*, pp. 10347–10357.

Tsai, C.-C., Chen, Y.-C. & Lu, C.-S. (2025). Test-Time Stain Adaptation with Diffusion Models for Histopathology Image Classification. *Computer Vision – ECCV 2024*, pp. 257–275.

Tschannen, M., Gritsenko, A., Wang, X., Naeem, M. F., Alabdulmohsin, I., Parthasarathy, N., Evans, T., Beyer, L., Xia, Y., Mustafa, B. et al. (2025). Siglip 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features. *arXiv preprint arXiv:2502.14786*.

Valvano, G., Leo, A., Saito, K. & Tommasi, T. (2023). Learning multi-modal self-supervision for test-time adaptation. *NeurIPS*.

Vapnik, V. N. (1998). *Statistical Learning Theory*. Wiley-Interscience.

Vargas Hakim, G. A., Osowiechi, D., Noori, M., Cheraghalikhani, M., Bahri, A., Ben Ayed, I. & Desrosiers, C. (2023). ClusT3: Information Invariant Test-Time Training. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 6136–6145.

Veeling, B. S., Linmans, J., Winkens, J., Cohen, T. & Welling, M. (2018). Rotation equivariant CNNs for digital pathology. *Medical Image Computing and Computer Assisted Intervention–MICCAI 2018: 21st International Conference, Granada, Spain, September 16-20, 2018, Proceedings, Part II 11*, pp. 210–218.

Venkateswara, H., Eusebio, J., Chakraborty, S. & Panchanathan, S. (2017). Deep hashing network for unsupervised domain adaptation. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5018–5027.

Volpi, R. & Murino, V. (2019). Addressing model vulnerability to distributional shifts over image transformation sets. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 7980–7989.

Volpi, R., Namkoong, H., Sener, O., Duchi, J. C., Murino, V. & Savarese, S. (2018). Generalizing to unseen domains via adversarial data augmentation. *Advances in Neural Information Processing Systems*, pp. 5334–5344.

Volpi, R., Larlus, D. & Rogez, G. (2021). Continual adaptation of visual representations via domain randomization and meta-learning. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4443–4453.

Wang, D., Shelhamer, E., Liu, S., Olshausen, B. & Darrell, T. (2020a). Tent: Fully test-time adaptation by entropy minimization. *arXiv preprint arXiv:2006.10726*.

Wang, D., Shelhamer, E., Liu, S., Olshausen, B. A. & Darrell, T. (2021a). Tent: Fully Test-Time Adaptation by Entropy Minimization. *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*.

Wang, F., Mei, J. & Yuille, A. (2025). SCLIP: Rethinking self-attention for dense vision-language inference. *European Conference on Computer Vision*, pp. 315–332.

Wang, J., Lan, C., Liu, C., Ouyang, Y., Qin, T., Lu, W., Chen, Y., Zeng, W. & Yu, P. S. (2022a). Generalizing to unseen domains: A survey on domain generalization. *IEEE transactions on knowledge and data engineering*, 35(8), 8052–8072.

Wang, K., Liew, J. H., Zou, Y., Zhou, D. & Feng, J. (2019). Panet: Few-shot image semantic segmentation with prototype alignment. *proceedings of the IEEE/CVF international conference on computer vision*, pp. 9197–9206.

Wang, Q., Fink, O., Van Gool, L. & Dai, D. (2022b). Continual test-time domain adaptation. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7201–7211.

Wang, S., Yu, L., Li, C., Fu, C.-W. & Heng, P.-A. (2020b). Learning from extrinsic and intrinsic supervisions for domain generalization. *European Conference on Computer Vision*, pp. 159–176.

Wang, Y., Li, H., Chau, L.-p. & Kot, A. C. (2021b). Embracing the Dark Knowledge: Domain Generalization Using Regularized Knowledge Distillation. *Proceedings of the 29th ACM International Conference on Multimedia*, pp. 2595–2604.

Wang, Z., Zhang, Y., Yu, K. & Zhang, H. (2023). Active test-time adaptation for semantic segmentation. *arXiv preprint arXiv:2312.01835*.

Wang, Z., Wu, Z., Agarwal, D. & Sun, J. (2022c). Medclip: Contrastive learning from unpaired medical images and text. *Proceedings of the Conference on Empirical Methods in Natural Language Processing. Conference on Empirical Methods in Natural Language Processing*, 2022, 3876.

Wang, Z., Loog, M. & Van Gemert, J. (2021c). Respecting domain relations: Hypothesis invariance for domain generalization. *2020 25th International Conference on Pattern Recognition (ICPR)*, pp. 9756–9763.

Wei, J., Suriawinata, A., Ren, B., Liu, X., Lisovsky, M., Vaickus, L., Brown, C., Baker, M., Tomita, N., Torresani, L. et al. (2021a). A petri dish for histopathology image analysis. *Artificial Intelligence in Medicine: 19th International Conference on Artificial Intelligence in Medicine, AIME 2021, Virtual Event, June 15–18, 2021, Proceedings*, pp. 11–24.

Wei, J., Suriawinata, A., Ren, B., Liu, X., Lisovsky, M., Vaickus, L., Brown, C., Baker, M., Tomita, N., Torresani, L. et al. (2021b). A petri dish for histopathology image analysis. *Artificial Intelligence in Medicine: 19th International Conference on Artificial Intelligence in Medicine, AIME 2021, Virtual Event, June 15–18, 2021, Proceedings*, pp. 11–24.

Wilson, G. & Cook, D. J. (2020). A survey of unsupervised deep domain adaptation. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 11(5), 1–46.

Wu, H., Xiao, B., Codella, N., Liu, M., Dai, X., Yuan, L. & Zhang, L. (2021a). Cvt: Introducing convolutions to vision transformers. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 22–31.

Wu, H., Phan, J. H., Bhatia, A. K., Cundiff, C. A., Shehata, B. M. & Wang, M. D. (2015a). Detection of blur artifacts in histopathological whole-slide images of endomyocardial biopsies. *Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Annual International Conference*, 2015, 727—730. doi: 10.1109/embc.2015.7318465.

Wu, H., Phan, J. H., Bhatia, A. K., Cundiff, C. A., Shehata, B. M. & Wang, M. D. (2015b). Detection of blur artifacts in histopathological whole-slide images of endomyocardial biopsies. *2015 37th annual international Conference of the IEEE Engineering in Medicine and biology society (EMBC)*, pp. 727–730.

Wu, Q., Yue, X. & Sangiovanni-Vincentelli, A. (2021b). Domain-agnostic test-time adaptation by prototypical training with auxiliary data. *NeurIPS 2021 Workshop on Distribution Shifts: Connecting Methods and Applications*.

Xian, Y., Lampert, C. H., Schiele, B. & Akata, Z. (2018). Zero-shot learning—a comprehensive evaluation of the good, the bad and the ugly. *IEEE transactions on pattern analysis and machine intelligence*, 41(9), 2251–2265.

Xiao, J., Hays, J., Ehinger, K. A., Oliva, A. & Torralba, A. (2010). Sun database: Large-scale scene recognition from abbey to zoo. *2010 IEEE computer society conference on computer vision and pattern recognition*, pp. 3485–3492.

Xie, S. & Tu, Z. (2015). Holistically-nested edge detection. *Proceedings of the IEEE international conference on computer vision*, pp. 1395–1403.

Xu, C., Sun, Y., Zhang, Y., Liu, T., Wang, X., Hu, D., Huang, S., Li, J., Zhang, F. & Li, G. (2025). Stain Normalization of Histopathological Images Based on Deep Learning: A Review. *Diagnostics*, 15(8). doi: 10.3390/diagnostics15081032.

Xu, J., De Mello, S., Liu, S. & Wang, X. (2022). GroupViT: Semantic Segmentation Emerges from Text Supervision. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18134–18144.

Xu, Q., Zhang, R., Zhang, Y., Wang, Y. & Tian, Q. (2021). A Fourier-based framework for domain generalization. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14383–14392.

Xu, Q., Zhang, R., Fan, Z., Wang, Y., Wu, Y.-Y. & Zhang, Y. (2023). Fourier-based augmentation with applications to domain generalization. *Pattern Recognition*, 139, 109474.

Yan, S., Song, H., Li, N., Zou, L. & Ren, L. (2020). Improve unsupervised domain adaptation with mixup training. *arXiv preprint arXiv:2001.00677*.

Yang, F.-E., Cheng, Y.-C., Shiau, Z.-Y. & Wang, Y.-C. F. (2021). Adversarial teacher-student representation learning for domain generalization. *Advances in Neural Information Processing Systems*, 34, 19448–19460.

Yang, J., Zhou, K., Li, Y. & Liu, Z. (2024). Generalized out-of-distribution detection: A survey. *International Journal of Computer Vision*, 132(12), 5635–5662.

Yang, Y. & Hospedales, T. (2016). Deep multi-task representation learning: A tensor factorisation approach. *arXiv preprint arXiv:1605.06391*.

Ye, N., Li, K., Bai, H., Yu, R., Hong, L., Zhou, F., Li, Z. & Zhu, J. (2022). Ood-bench: Quantifying and understanding two dimensions of out-of-distribution generalization. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7947–7958.

Yu, R., Liu, S., Yang, X. & Wang, X. (2023a). Distribution Shift Inversion for Out-of-Distribution Prediction. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3592–3602.

Yu, R., Liu, S., Yang, X. & Wang, X. (2023b). Distribution Shift Inversion for Out-of-Distribution Prediction. *arXiv preprint arXiv:2306.08328*. Retrieved from: https://arxiv.org/abs/2306.08328. National University of Singapore.

Yu, Y., Sheng, L., He, R. & Liang, J. (2024). STAMP: Outlier-Aware Test-Time Adaptation with Stable Memory Replay. *arXiv:2407.15773*.

Yuan, L., Chen, Y., Wang, T., Yu, W., Shi, Y., Jiang, Z.-H., Tay, F. E., Feng, J. & Yan, S. (2021). Tokens-to-token vit: Training vision transformers from scratch on Imagenet. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 558–567.

Yue, X., Zhang, Y., Zhao, S., Sangiovanni-Vincentelli, A., Keutzer, K. & Gong, B. (2019). Domain randomization and pyramid consistency: Simulation-to-real generalization without accessing target domain data. *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 2100–2110.

Zhang, B., Zhang, P., Dong, X., Zang, Y. & Wang, J. (2024a). Long-clip: Unlocking the long-text capability of clip. *European conference on computer vision*, pp. 310–325.

Zhang, C., Zhang, M., Zhang, S., Jin, D., Zhou, Q., Cai, Z., Zhao, H., Yi, S., Liu, X. & Liu, Z. (2021a). Delving deep into the generalization of vision transformers under distribution shifts. *arXiv preprint arXiv:2106.07617*.

Zhang, H., Cisse, M., Dauphin, Y. N. & Lopez-Paz, D. (2017). mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*.

Zhang, J., Liu, S., Song, J., Zhu, T., Xu, Z. & Song, M. (2024b). Lookaround Optimizer: k steps around, 1 step average. *Advances in Neural Information Processing Systems*, 36.

Zhang, J., Zhang, X.-Y., Wang, C. & Liu, C.-L. (2023a). Deep Representation Learning for Domain Generalization with Information Bottleneck Principle. *Pattern Recognition*, 109737.

Zhang, L., Wang, X., Yang, D., Sanford, T., Harmon, S., Turkbey, B., Wood, B. J., Roth, H., Myronenko, A., Xu, D. et al. (2020). Generalizing deep learning for medical image segmentation to unseen domains via deep stacked transformation. *IEEE transactions on medical imaging*, 39(7), 2531–2540.

Zhang, L., Rao, A. & Agrawala, M. (2023b). Adding conditional control to text-to-image diffusion models. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 3836–3847.

Zhang, M., Marklund, H., Dhawan, N., Gupta, A., Levine, S. & Finn, C. (2021b). Adaptive risk minimization: Learning to adapt to domain shift. *Advances in Neural Information Processing Systems*, 34, 23664–23678.

Zhang, M., Levine, S. & Finn, C. (2022). Memo: Test time robustness via adaptation and augmentation. *Advances in neural information processing systems*, 35, 38629–38642.

Zhang, S., Xu, Y., Usuyama, N., Bagga, J., Tinn, R., Preston, S., Rao, R., Wei, M., Valluri, N., Wong, C. et al. Large-scale domain-specific pretraining for biomedical vision-language processing (2023). *arXiv preprint arXiv:2303.00915*, 2.

Zhang, X., Cui, P., Xu, R., Zhou, L., He, Y. & Shen, Z. (2021c). Deep stable learning for out-of-distribution generalization. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5372–5382.

Zhao, H., Liu, Y., Alahi, A. & Lin, T. (2023). On pitfalls of test-time adaptation. *arXiv preprint arXiv:2306.03536*.

Zheng, S., Lu, J., Zhao, H., Zhu, X., Luo, Z., Wang, Y., Fu, Y., Feng, J., Xiang, T., Torr, P. H. et al. (2021). Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 6881–6890.

Zhou, C., Loy, C. C. & Dai, B. (2022a). Extract Free Dense Labels from CLIP. *European Conference on Computer Vision*, pp. 288–304.

Zhou, K., Yang, Y., Hospedales, T. & Xiang, T. (2020a). Deep domain-adversarial image generation for domain generalisation. *Proceedings of the AAAI conference on artificial intelligence*, 34(07), 13025–13032.

Zhou, K., Yang, Y., Hospedales, T. & Xiang, T. (2020b). Learning to generate novel domains for domain generalization. *European conference on computer vision*, pp. 561–578.

Zhou, K., Yang, Y., Qiao, Y. & Xiang, T. (2021). Domain generalization with mixstyle. *arXiv preprint arXiv:2104.02008*.

Zhou, K., Liu, Z., Qiao, Y., Xiang, T. & Loy, C. C. (2022b). Domain generalization: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(4), 4396–4415.

Zhou, K., Loy, C. C. & Liu, Z. (2023). Semi-supervised domain generalization with stochastic stylematch. *International Journal of Computer Vision*, 131(9), 2377–2387.

Zhou, K., Yang, Y., Qiao, Y. & Xiang, T. (2024a). Mixstyle neural networks for domain generalization and adaptation. *International Journal of Computer Vision*, 132(3), 822–836.

Zhou, S., Xiong, Z. & Wu, F. (2024b). Test-Time Adaptation via Style and Structure Guidance for Histological Image Registration. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(7), 7677-7685.

Zhu, J.-Y., Park, T., Isola, P. & Efros, A. A. (2017). Unpaired image-to-image translation using cycle-consistent adversarial networks. *Proceedings of the IEEE international conference on computer vision*, pp. 2223–2232.