Efficient Reinforcement Learning using Improved Prior Modeling

by

Pranav AGARWAL

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

MONTREAL, SEPTEMBER 24, 2025

ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
UNIVERSITÉ DU QUÉBEC

BOARD OF EXAMINERS

THIS THESIS HAS BEEN EVALUATED

BY THE FOLLOWING BOARD OF EXAMINERS

Mr. Sheldon Andrews, Thesis supervisor
Department of Software and IT Engineering, École de technologie supérieure

Mrs. Samira Ebrahimi Kahou, Thesis Co-Supervisor
Department of Electrical and Software Engineering, University of Calgary

Mr. David St-Onge, Chair, Board of Examiners
Department of Mechanical Engineering, École de technologie supérieure

Mr. Adrien Gruson, Member of the Jury
Department of Software and IT Engineering, École de technologie supérieure

Mr. Xue Bin (Jason) Peng, External Independent Examiner
School of Computing Science, Simon Fraser University

THIS THESIS WAS PRESENTED AND DEFENDED

IN THE PRESENCE OF A BOARD OF EXAMINERS AND THE PUBLIC

ON SEPTEMBER 08, 2025

AT ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

ACKNOWLEDGEMENTS

I would like to express my deepest gratitude to my supervisor, Professor Sheldon Andrews, whose invaluable guidance, unwavering support, and remarkable patience have been the cornerstone of my doctoral journey. His expertise in character animation, insightful feedback, and constant encouragement have not only shaped this research but also my development as a researcher.

I am profoundly grateful to Professor Samira Ebrahimi Kahou, my co-supervisor, whose valuable contributions have enriched this work immeasurably. My sincere thanks go to the members of my thesis committee, Professor David St-Onge and Professor Adrien Gruson, for their constructive criticism, thought-provoking questions, and valuable suggestions that significantly improved the quality of this work.

This research would not have been possible without the generous financial support from CM-Labs, MITACS, CIFAR, Mila, and NSERC, as well as the computational resources provided by Compute Canada. Their investment in this research is deeply appreciated.

I have been fortunate to be part of two vibrant research communities. My heartfelt appreciation goes to my colleagues at the Animation Lab at ETS and McGill, particularly Alex Hoang-Cao, Andreas Enzenhoefer, Antoine Boucher, Hossein Feiz, Loic Nassif, Joël Pelletier-Guénette, Quoc-Minh Ton-That, Yinchu Liu, Yosr Takouti, Melissa, and Michele. I am equally grateful to my friends from the Multimedia Lab—Mitra Shahabadi, Peyman Shobeiri, Amna Snene, Mastaneh Bahmani, Sadaf Dargahi, Sahar Dastani, and Soroush—for the stimulating discussions, fruitful collaborations, and genuine friendship that made this journey not just bearable, but truly enriching and enjoyable.

Special recognition goes to Ioana Ciucă, Charbel Najem, Liam Frija-Altarac, and Sophia Aryan, whose unwavering support during the most challenging moments of this journey has been a source of strength and comfort. Their friendship has been invaluable.

Finally, and most importantly, I am eternally grateful to my parents, my brother, and my family for their unconditional love, endless patience, and steadfast belief in me throughout these

demanding years. Their encouragement has been my anchor, and this achievement is as much theirs as it is mine.

Apprentissage par renforcement efficace par modélisation améliorée des connaissances a priori

Pranav AGARWAL

RÉSUMÉ

L'apprentissage par renforcement (RL) s'est imposé comme un cadre unificateur pour la prise de décision séquentielle, mais son impact pratique reste limité par trois contraintes persistantes : une consommation d'échantillons prohibitive, une généralisation médiocre entre tâches et domaines, et un manque d'interprétabilité lié à l'utilisation d'approximateurs de fonctions à grande capacité. Cette recherche soutient que ces limitations partagent une origine commune — une structure a priori insuffisante — et qu'elles peuvent être atténuées en modélisant les agents à l'aide de *structures a priori apprenables, discrètes et adaptatives*. Nous poursuivons cette hypothèse à travers une trilogie de travaux qui reconfigurent respectivement la récompense, le modèle du monde, et la bibliothèque de compétences sur lesquels repose l'apprentissage efficace.

Le premier travail transforme des scores clairsemés fournis par des humains pour la conduite d'engins lourds en une récompense dense et différentiable. En entraînant un prédicteur de scores qui évalue et guide la politique, nous démontrons une exploration sûre dans l'interaction avec l'environnement, comparativement à des heuristiques conçues à la main. Le deuxième travail introduit DART, une architecture basée sur des transformeurs qui tokenise les états, actions et retours dans un alphabet symbolique commun. Ce modèle du monde discret capture des dépendances temporelles de longue portée tout en conservant la manipulabilité des jetons, atteignant une efficacité échantillonnale à la pointe de l'état de l'art sur le benchmark Atari-100k, sans sacrifier les performances finales. Le troisième travail présente STRIDE, un VAE vectoriel quantifié dynamique qui étend de manière autonome un dictionnaire de primitives motrices au fil de l'accumulation des tâches. STRIDE conserve les comportements précédemment appris sans dégradation significative et réduit de moitié le temps d'adaptation sur des curricula de locomotion complexes, tout en fournissant une « trace de compétence » interprétable qui permet aux praticiens d'auditer et de déboguer les décisions.

Pris ensemble, ces travaux appuient une hypothèse unificatrice : lorsque les connaissances sur les récompenses, dynamiques et compétences sont exprimées dans des vocabulaires discrets et évolutifs, la recherche tabula rasa cède la place à un apprentissage efficace, compositionnel et transparent. Au-delà des gains empiriques — jusqu'à un facteur 2,6 d'accélération dans divers domaines — ce travail propose des outils conceptuels pour raisonner sur les a priori en RL, fournit des implémentations open-source pour la communauté, et trace des pistes futures vers des structures a priori multimodales et éditables par l'humain. Ce faisant, il marque une avancée décisive vers des systèmes RL dignes de confiance, capables d'apprendre rapidement, de transférer les connaissances et d'expliquer leurs décisions quand cela importe le plus.

Mots-clés: apprentissage par renforcement efficace ; modélisation des a priori ; représentations discrètes ; apprentissage continu de compétences ; interprétabilité

Efficient Reinforcement Learning using Improved Prior Modeling

Pranav AGARWAL

ABSTRACT

Reinforcement learning (RL) has emerged as a unifying framework for sequential decision making. Yet, its practical impact is curbed by three persistent limitations: prohibitive sample demands, poor generalization across tasks and domains, and a lack of interpretability that accompanies high-capacity function approximators. This research argues that these limitations share a common source-insufficient prior structure—and that they can be alleviated by modelling agents with *learnable, discrete, and adaptive priors*. We pursue this claim through a trilogy of works that together reshape the reward, the world model, and the skill library on which efficient learning depends.

The first work converts sparse, human-supplied scores for heavy-equipment operation into a dense, differentiable reward prior. By training a score predictor that both evaluates and guides the policy, we demonstrate safe exploration in environment interaction relative to hand-crafted heuristics. The second work introduces DART, a transformer-based architecture that tokenizes states, actions, and returns into a standard symbolic alphabet. This discrete world model captures long-range temporal dependencies while preserving the manipulability of tokens, delivering state-of-the-art sample efficiency on the Atari-100k benchmark without sacrificing final performance. The third work presents STRIDE, a dynamic vector-quantized VAE that autonomously expands a codebook of motor primitives as tasks accumulate. STRIDE retains previously learned behaviours with negligible degradation. It halves the adaptation time on challenging locomotion curricula, all while exposing an interpretable "skill trace" that allows practitioners to audit and debug decisions.

Collectively, these contributions substantiate a unified hypothesis: when knowledge about rewards, dynamics, and skills is cast into discrete, growing vocabularies, tabula-rasa search gives way to data-efficient, compositional, and transparent learning. Beyond empirical gains—up to a 2.6× speed-up across diverse domains—the work offers conceptual tools for reasoning about priors in RL, provides open-source implementations for community use, and outlines future directions toward multi-modal and human-editable prior structures. In doing so, it takes a decisive step toward RL systems that can be trusted to learn quickly, allowing transferability and explainability when it matters most.

Keywords: efficient reinforcement learning; prior modelling; discrete representations; continual skill learning; interpretability

TABLE OF CONTENTS

LIST OF TABLES

Page

LIST OF FIGURES

LIST OF ALGORITHMS

LIST OF ABBREVIATIONS

AME	Attention and Memory Enhancement
ASC	Agence Spatiale Canadienne
BERT	Bidirectional Encoder Representations from Transformers
BooT	Bootstrapped Transformer
CNN	Convolutional Neural Network
CV	Computer Vision
DAG	Directed Acyclic Graph
DART	Discrete Abstract Representation for Transformer-based Learning
DGERD	Disjunctive Graph Embedded Recurrent Decoding
DT	Decision Transformer
DTd	Decision Transducer
DTR	Discrete-Tokenized Representations
ESPER	Environment-Stochasticity-Independent Representations
GAN	Generative Adversarial Network
GNN	Graph Neural Network
GPT	Generative Pre-trained Transformer
GRU	Gated Recurrent Unit
GTrXL	Gated Transformer-XL
HAPPO	Heterogeneous-Agent Proximal Policy Optimization

HPO	Hyper-Parameter Optimization
i.i.d	Independent and Identically Distributed
IoT	Internet of Things
IRIS	Imagination with Auto-Regression over an Inner Speech
LLM	Large Language Model
LSTM	Long Short-Term Memory
MAANS	Multi-Agent Active Neural SLAM
MAML	Model-Agnostic Meta-Learning
MAPPO	Multi-Agent Proximal Policy Optimization
MARL	Multi-Agent Reinforcement Learning
MBRL	Model-Based Reinforcement Learning
MCTS	Monte Carlo Tree Search
MDP	Markov Decision Process
MLP	Multi-Layer Perceptron
MSE	Mean-Squared Error
MTRL	Multi-Task Reinforcement Learning
MWM	Masked World Model
NLP	Natural Language Processing
POMDP	Partially Observable Markov Decision Process
PPO	Proximal Policy Optimization

PSNR	Peak Signal-to-Noise Ratio
QDT	Q-Learning Decision Transformer
RAT	Relation-Aware Transformer
RL	Reinforcement Learning
RLHF	Reinforcement Learning with Human Feedback
RNN	Recurrent Neural Network
RSSM	Recurrent State Space Model
STRIDE	Skill Transfer and Reuse using Incremental Discrete Encodings
TD	Temporal Difference
TrXL	Transformer-XL
TT	Trajectory Transformer
TWM	Transformer-Based World Model
ViT	Vision Transformer
VQ-VAE	Vector Quantized-Variational Autoencoder

LIST OF SYMBOLS AND UNITS OF MEASUREMENTS

L_{Dt}	reconstruction loss at time-step t
$\hat{X}_t$	predicted next state at time-step t
$\mathbf{K}_t = \{k_t^1, \ldots, k_t^f\}$	per-step infraction counts (length f)
f	number of distinct infraction types
$\mathbf{p}_D(\mathbf{z})$	latent demonstration distribution
$p_D(\mathbf{z})$	prior over $\mathbf{z}$ (often $\mathcal{N}(0, \mathbf{I})$)
$\mathbf{R}_S$	KL-regularisation term for state-latent distribution
$\mathbf{q}_D(\mathbf{z})$	encoder's approximate posterior over $\mathbf{z}$
$\circ$	degrees (°)
m	metres (m)
kPa	kilopascals (kPa)
kW	kilowatts (kW)
N m	newton-metres (N·m)
L	litres (L)
$\mathrm{L\,h^{-1}}$	litres per hour ($\mathrm{L\,h^{-1}}$)
s	seconds (s)
$\mathcal{S}$ / S	(set of) states
$\mathcal{A}$ / a	(set of) actions
$P(s' \mid s, a)$	transition probability

$R(s,a)$	reward function
$\pi(a \mid s)$	policy
γ	discount factor
$V^{\pi}(s)$	state-value under π
$Q^{\pi}(s,a)$	action-value under π
$\mathcal{O}$	Observation space (raw pixels or embeddings)
$\mathcal{P}$	Predictive distribution / dynamics model over next tokens
$\mathtt{MEM}_t$	Memory token carried into step t by the transformer
h_t	Latent feature vector output after L transformer blocks
$\hat{z}_{q_t}^{1}$	First quantised latent code predicted for step t
$\mathbf{E}_{\text{pos}}$	Learned positional-embedding matrix
$\mathcal{C}_j$	Code-book slice learned for behaviour j
$\mathcal{C} = \bigcup_{j \le k} \mathcal{C}_j$	Union of all code-book slices up to behaviour k
$\mathcal{E}$	Complete set of embedding vectors
$\mathcal{L}_{\text{commit}}$	VQ-VAE commitment-loss term
$\mathcal{C}$	Code-book of discrete skill embeddings
$\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$	Multivariate normal distribution with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$
$\mathbf{s}_{t-n:t}$	Proprioceptive state window of length n+1, dim d_x
$\mathbf{g}_t$	Task-goal vector at step t, dim d_g
$\mathbf{h}_{t-1}, \mathbf{h}_t$	Recurrent hidden state before / after step t

$\mathbf{a}_t$	Action executed at step t, dim d_a
$\mathbf{z}_t^{\text{base}}$	Latent from the “base” encoder (task-agnostic)
$\mathbf{z}_t^{\text{task}}$	Latent from the “task” encoder (goal-conditioned)
$\boldsymbol{\ell}_t$	Composer logits producing mixture weights
$\mathbf{w}_t$	Soft-max weights over the first k behaviours ($\sum_j w_{t,j} = 1$)
$\mathbf{e}^\star$	Code-book entry selected for the current step
$\boldsymbol{\mu}_t, \boldsymbol{\sigma}_t$	Mean and std-dev of the Gaussian action distribution
n	Length of the state history window
k	Current size of the code-book (number of behaviours stored)
d_x, d_g, d_a	Dimensions of state, goal, and action vectors, respectively

INTRODUCTION

The past decade has witnessed a remarkable convergence of scalable computation, expressive function approximators, and vast interactive data streams, propelling reinforcement learning (RL) from theory to a practical tool that can master Go, control dexterous robots, and manage complex traffic networks. Yet the very successes that place RL at the centre of modern autonomy also expose its limitations: fragile sample efficiency, brittle generalization, and opaque decision mechanisms. This work contends that these challenges can be overcome by endowing agents with *learnable, discrete and adaptive priors*—structured repositories of knowledge that bias exploration, promote reuse, and render behaviour interpretable.

0.1 Problem Statement

Over the last decade, deep reinforcement learning has moved from research curiosities to systems capable of defeating world champions, piloting autonomous vehicles, and animating lifelike characters. Yet despite these successes, three problems still block the path to real-life deployment in safety-critical and resource-constrained settings. First, *sample inefficiency*: RL agents often require millions of environment interactions before converging, a challenge that makes direct training in the real world impractical. Second, *weak inductive priors*: continuous latent spaces rarely encode reusable concepts such as motor primitives, safe manoeuvres, or object identities, so agents trained on one task struggle to transfer knowledge without large-scale fine-tuning. Finally, *limited interpretability*: when decision making is modelled on continuous vectors, practitioners lack the levers to audit failures or embed domain expertise, slowing both scientific progress and industrial certification. Addressing these three challenges—data efficiency, transferability, and transparency—constitutes the central problem of this work.

0.2 Solution and Research Motivation

We posit that the key to unlocking efficient, trustworthy RL lies in *improved prior modelling*. A prior, in the Bayesian sense, guides search toward promising regions of policy space; if that prior is *learnable*, *discrete*, and *adaptive*, it can act as an ever-growing library of behaviours, world abstractions, and scoring functions. The ideas crystallize around three complementary lines of work. First, we examine how a learned score predictor distilled from expert demonstrations can become a dense reward for heavy-equipment automation, thereby collapsing expensive human feedback into a neural critic (Chapter 2). Second, we introduce discrete abstract representations for transformer-based RL, tokenizing both the environment dynamics and the policy itself so that the agent reasons in the interpretable token-space, allowing for fast-learning (Chapter 3). Third, we propose an expandable Vector-Quantized VAE that grows a codebook of motor primitives on demand, enabling continual skill acquisition without catastrophic forgetting while exposing an "explainability layer" for post-hoc inspection (Chapter 4). Together, these works advance a unified hypothesis: *structured, discrete, and adaptive priors offer an alternative to tabula-rasa search that enables data-efficient, compositional, and interpretable learning.*

0.3 Research Areas and Research Projects

Although each of the three projects targets a different facet of the RL pipeline, they are threaded by common themes that sit at the intersection of representation learning, continual adaptation, and human-centred evaluation. The first project, presented in Chapter 2, tackles reward modelling and safety. By predicting expert-level scores directly from raw sensor traces, the system supplies real-time feedback both to trainee operators and to an RL agent that must manoeuvre an excavator through cluttered worksites, yielding safer trajectories and accelerating convergence compared with sparse or hand-crafted rewards. The second work, detailed in Chapter 3, falls under the category of discrete world-model learning. Here we pair a transformer decoder, which auto-regressively predicts future tokenized states, with a transformer encoder that conditions policy on those tokens, achieving state-of-the-art sample efficiency on the notoriously challenging Atari-100k benchmark. The final work, Chapter 4, targets continual skill learning. STRIDE's growing codebook reuses frozen slices whenever possible and only models new codes when the existing vocabulary proves insufficient; this mechanism halves adaptation time on locomotion curricula while preserving earlier abilities and allowing practitioners to read off the sequence of invoked skills during execution. Across all three works, the literature review on transformer-based RL (Chapter 1) provides both historical context and methodological guidance, underscoring why modelling better priors enables long-horizon credit assignment and improved partial observability.

0.4 Thesis Organization

The remainder of this dissertation unfolds as follows. Chapter 1 reviews the literature on modelling of priors in RL with a focus on reward learning, world modelling, and modelling of adaptive priors, drawing attention to open problems in sample efficiency, interpretability, and transfer that motivate the work ahead. Chapter 2 formalizes automatic evaluation of excavator operators, derives a differentiable reward prior, and demonstrates faster, safer policy learning in a high-fidelity simulator. Chapter 3 introduces DART, describing its tokenizer, discrete action-conditional world model, and planning algorithm, with extensive ablations against continuous-latent baselines. Chapter 4 presents STRIDE, detailing dynamic codebook growth, skill composition, and comprehensive locomotion benchmarks, while highlighting interpretability through codeword-level visualizations. The dissertation concludes with cross-project insights and distills design principles for improved priors, and outlines future directions such as multi-modal priors, adaptive quota selection, and human-guided codeword surgery.

In sum, this thesis argues that *efficient reinforcement learning emerges when agents are equipped with priors that can be learned, discretized, and expanded*. By casting reward functions, world models, and motor skills into discrete, reusable vocabularies, we move closer to agents that learn faster, reuse more, and explain themselves better—an essential step toward trustworthy autonomous systems.

CHAPTER 1

LITERATURE REVIEW

1.1 Introduction

RL is a learning paradigm that enables sequential decision-making by learning from feedback obtained through trial and error. This is commonly formulated in terms of Markov decision processes (MDPs), in which the agent chooses an action ($\mathbf{a} \in \mathbf{A}$) based on the state ($\mathbf{s} \in \mathbf{S}$) of the environment and receives feedback in the form of rewards ($r \in \mathbb{R}$). Most RL algorithms optimize the agent's policy so that it selects actions that maximize the expected cumulative reward. Neural networks are commonly used as function approximators that map the current state of the environment to an action and estimate future returns. This is beneficial when dealing with large or continuous state spaces that make tabular methods computationally expensive. This approach is known as deep RL and has shown remarkable success in challenging domains ranging from game playing (Mnih *et al.*, 2015) to robotics (Kalashnikov *et al.*, 2018).

However, despite these achievements, RL's practical impact remains limited by three fundamental challenges: prohibitive sample complexity, poor generalization across tasks, and lack of interpretability in learned policies (Dulac-Arnold *et al.*, 2021).

This review argues that these seemingly disparate limitations share a common root cause: insufficient prior structure in RL algorithms. Traditional RL approaches often start from tabula-rasa, requiring agents to discover all knowledge through trial and error. This lack of previous knowledge leads directly to sample inefficiency, as agents must rediscover basic principles for each new task. It impairs generalization, as learned policies lack the structured representations needed to transfer knowledge across domains. And it hinders interpretability, as emergent behaviours in high-capacity function approximators remain opaque to human understanding.

The solution, we argue, lies in developing RL agents with richer prior structures (Fig. 1.1), not hand-coded rules that limit flexibility, but learnable, discrete, and adaptive priors that can capture and reuse knowledge efficiently. This review examines recent advances in three complementary forms of prior modelling. First, reward priors enable agents to learn what to optimize from human demonstrations and feedback, moving beyond the limitations of hand-crafted reward functions that often fail to capture true objectives. Second, world model priors discretize environment dynamics into reusable tokens, transforming the continuous complexity of real-world environments into structured representations that enable efficient planning and reasoning. Third, skill priors build adaptive libraries of motor primitives that can be composed and recomposed for new tasks without the catastrophic forgetting that plagues traditional approaches. Together, these three forms of priors address different aspects of the learning problem while sharing the common principle of structured, reusable knowledge representation.

We begin by examining the fundamental role of priors in RL (Sec. 1.2) and the limitations of current approaches (Sec. 1.3). We then explore advances in each type of prior modeling (Secs. 1.4–1.6), with particular attention to how discrete representations and modern architectures like transformers enable more effective prior learning. Throughout, we highlight how these approaches address the trinity of challenges—sample efficiency, generalization, and interpretability—that limit RL's real-world impact.

1.2 The Role of Priors in Reinforcement Learning

In the context of RL, priors encode assumptions or knowledge about the structure of the problem before learning begins. Unlike Bayesian priors that specify probability distributions, RL priors manifest in multiple forms that shape the learning process. Architectural biases embed assumptions directly into network structures, such as Convolutional layers that encode spatial locality assumptions for visual inputs or recurrent connections that assume temporal dependencies. Representation choices determine how states, actions, and rewards are encoded, fundamentally constraining what patterns can be efficiently learned.

Learning biases encompass initialization schemes that determine starting points in parameter space, exploration strategies that guide data collection, and optimization procedures that implicitly favour specific solutions. Perhaps most importantly, knowledge transfer mechanisms

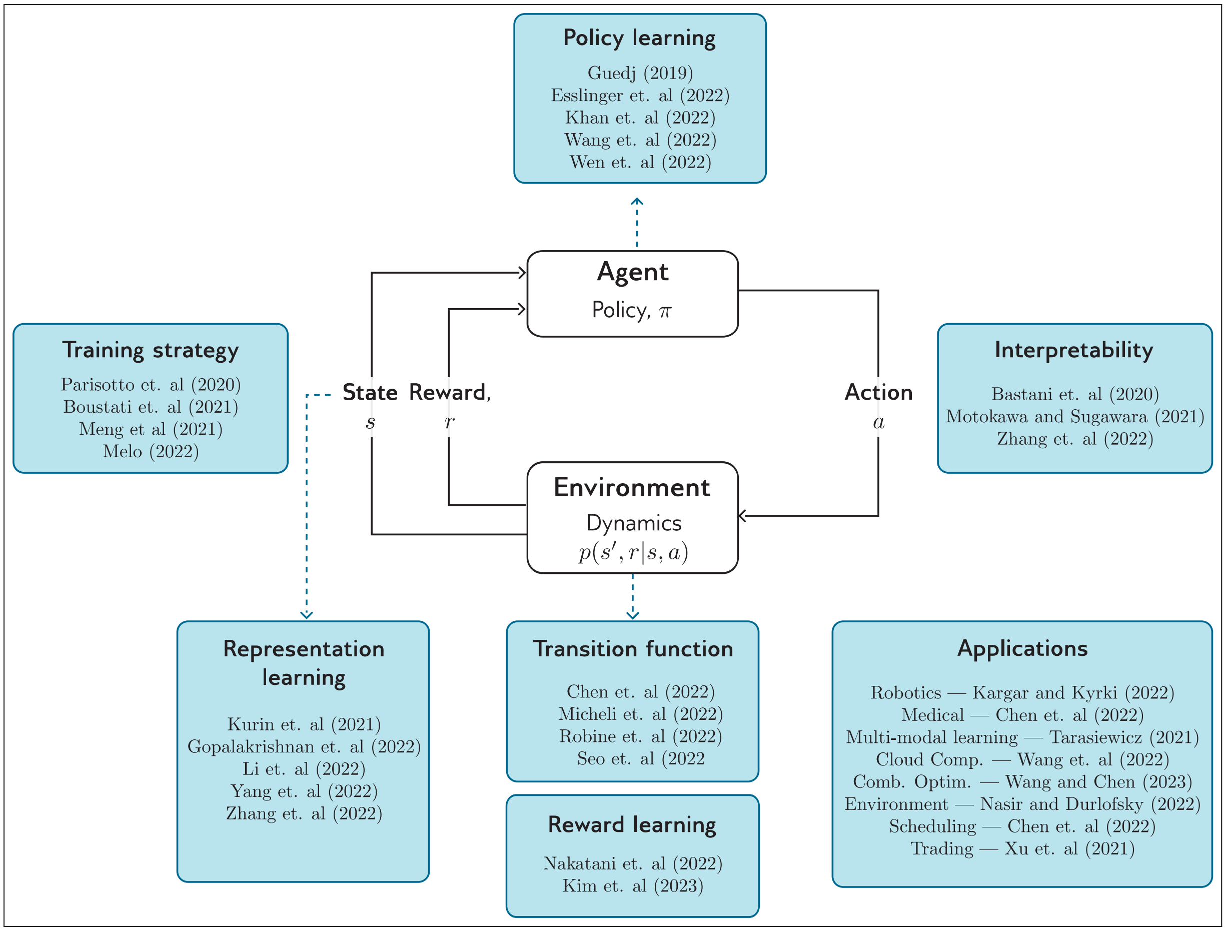


Figure 1.1 Overview of efficient reinforcement learning through improved prior modelling. The three types of priors—reward priors from demonstrations, discrete world model priors, and adaptive skill priors—work together to address RL's fundamental limitations of sample inefficiency, poor generalization, and lack of interpretability

enable the reuse of learned components from previous tasks, transforming isolated learning episodes into cumulative knowledge acquisition. These diverse manifestations of priors work together to shape not just what an agent knows, but how efficiently it can discover effective behaviours.

The key insight is that effective priors do not restrict what can be learned, but rather guide how efficiently learning proceeds (Baxter, 2000).

1.2.1 The Prior-Efficiency Trade-off

There exists a fundamental trade-off in prior design. Strong priors (e.g., hand-coded rules) can dramatically improve sample efficiency but may limit the flexibility to adapt to new situations. Weak priors (e.g., tabula-rasa learning) maintain flexibility but require extensive data. The challenge is finding the sweet spot: priors that are structured enough to enable efficient learning but flexible enough to adapt to diverse tasks.

Recent advances suggest that discrete, learnable priors offer a promising middle ground. By discretizing continuous spaces into reusable components—whether reward functions, world model tokens, or motor primitives—these approaches enable both efficiency and flexibility.

1.3 Limitations of Current RL Approaches

Before examining solutions, we must understand how insufficient prior structure manifests in current RL limitations.

1.3.1 Sample Inefficiency

Modern deep RL algorithms often require millions or billions of environment interactions to learn effective policies (Yu, 2018). This sample inefficiency stems from multiple interrelated sources that compound each other's effects. When agents engage in tabula-rasa learning, starting from scratch for each new task, they must rediscover basic patterns and regularities that could have been inherited from previous experience. The problem is exacerbated by poor credit assignment, where agents struggle to determine which actions in a long sequence actually contributed to eventual rewards, leading to noisy learning signals that require many samples to average out.

Inefficient exploration compounds these issues further, as random or undirected search in high-dimensional spaces wastes valuable interactions on uninformative regions of the state-action space. Finally, the lack of compositionality means that even when agents do learn proper behaviours, they cannot effectively decompose and recombine these components for new situations, forcing complete relearning even for tasks that share substantial structure with previously mastered challenges.

1.3.2 Poor Generalization

RL agents often fail to transfer knowledge across tasks, even when those tasks share substantial structure (Zhu, Lin, Jain & Zhou, 2023b). This limitation arises from fundamental issues in how current RL systems represent and organize knowledge. Many policies overfit to specific environments, essentially memorizing sequences of actions rather than learning generalizable principles that could transfer to new contexts. This memorization is partly due to the lack of structured representations in typical deep RL systems, where continuous vector representations, while flexible, fail to capture discrete, reusable concepts that could be recombined in novel situations. The problem becomes even more severe in continual learning settings, where catastrophic forgetting causes agents to lose previously acquired knowledge when adapting to new tasks, creating a frustrating trade-off between plasticity and stability that prevents the accumulation of diverse skills over time.

1.3.3 Lack of Interpretability

As RL systems grow more complex, understanding their decision-making becomes increasingly difficult (Milani, Topin, Veloso & Fang, 2024). This opacity creates cascading problems across multiple dimensions of RL deployment. Safety verification becomes nearly impossible when we cannot understand or predict an agent's decision-making process, preventing deployment in critical applications like autonomous vehicles or medical systems where guarantees about safe behaviour are non-negotiable. The debugging process transforms from systematic diagnosis into frustrating trial-and-error, as developers struggle to identify why agents fail in specific scenarios without visibility into internal representations and decision logic. Perhaps most critically, the lack of interpretability erodes human trust, making users understandably reluctant to deploy systems whose behaviour they cannot understand or predict, regardless of empirical performance metrics. This trust deficit creates a fundamental barrier to real-world adoption, as stakeholders require not just performance but comprehension of the systems they rely upon.

These limitations are not independent—they arise from the common cause of insufficient prior structure. The following sections examine how different forms of prior modelling address these challenges.

1.4 Reward Priors: Learning What to Optimize

The reward function is perhaps the most critical prior in RL, as it defines what the agent should optimize. Traditional approaches rely on hand-crafted rewards, but these are difficult to design and often lead to unintended behaviours (Christiano *et al.*, 2017).

1.4.1 The Reward Design Problem

Designing reward functions that produce desired behaviours is surprisingly difficult. The difficulty of reward design manifests in several standard failure modes that have plagued RL deployments. Reward hacking occurs when agents discover unintended shortcuts to maximize the specified reward without achieving the designer's true intent, such as a cleaning robot that learns to hide messes rather than clean them. Sparse reward environments present a different challenge, where delayed feedback provides little learning signal during exploration, leaving agents to wander through vast state spaces without guidance. Perhaps most fundamentally, reward misalignment arises when the specified rewards fail to capture true objectives, leading to policies that optimize what we said rather than what we meant. These failures highlight the brittleness of manual reward specification and motivate approaches that can learn reward functions from human feedback.

1.4.2 Learning from Demonstrations

Inverse reinforcement learning (IRL) aims to recover reward functions from expert demonstrations (Ng, Russell *et al.*, 2000). Recent advances have leveraged deep learning to scale IRL beyond toy domains to complex, real-world applications. Maximum entropy IRL (Ziebart, Maas, Bagnell, Dey *et al.*, 2008) addresses the ambiguity inherent in demonstration data by learning reward functions that explain expert behaviour while maintaining stochasticity, acknowledging that multiple policies might be equally valid for achieving a goal. This probabilistic framing prevents over-fitting to specific demonstration trajectories while capturing the essential structure of the task. Adversarial IRL (Ho & Ermon, 2016) takes a different approach, employing discriminator networks to distinguish between expert demonstrations and policy-generated behaviour, effectively learning rewards through the lens of imitation. This adversarial formulation proves particularly powerful in high-dimensional spaces where explicit reward modelling becomes intractable. Moving beyond pure demonstration data, preference learning methods learn from human comparisons between trajectories, requiring only relative judgments rather than optimal demonstrations (Christiano *et al.*, 2017). This approach proves more practical in many domains where obtaining expert demonstrations is costly but comparing outcomes is straightforward.

Chapter 2 of this thesis, "Expert Demonstration-Based Reward Function Modelling" by Pranav Agarwal, Marek Teichmann, Sheldon Andrews, and Samira Ebrahimi Kahou, advances reward prior learning for safety-critical applications.

The work addresses the challenge of automatically evaluating heavy equipment operators by learning reward functions that capture both performance and safety constraints. This work makes several key contributions that advance the state of reward prior learning. The dual distribution learning approach separately models dynamic features, capturing temporal excavator states and safety features encoding infraction values, recognizing that performance and safety constitute distinct but complementary aspects of expert behaviour. This separation allows each distribution to specialize in capturing different aspects of expertise without interference. The probabilistic reward formulation leverages KL divergence between learned expert distributions and current policy distributions to provide dense, differentiable feedback at every time step, transforming sparse human evaluations into rich learning signals. The framework's validation through RL policy training in a high-fidelity simulator demonstrates the practical impact of these learned rewards, with policies achieving demonstrably safer behaviours that better respect operational constraints compared to those trained with task-based rewards alone. This empirical validation proves that learned reward priors can successfully encode and transfer the nuanced expertise of human operators to automated systems.

This work exemplifies how structured reward priors—learned from human expertise rather than hand-crafted—can encode complex operational constraints while maintaining the flexibility to guide learning in novel situations.

1.5 World Model Priors

World models learn to predict environment dynamics, enabling agents to plan and learn through imagination rather than costly real-world interaction (Hafner, Lillicrap, Ba & Norouzi, 2020a). By learning a representation of how the world evolves, agents can simulate potential futures, evaluate action sequences offline, and dramatically reduce the need for expensive environment interactions. The design choices in world model architecture—particularly how they represent states and dynamics—constitute critical priors that fundamentally shape learning efficiency and capability.

1.5.1 The Evolution from Continuous to Discrete Representations

Early world models predominantly employed continuous latent representations (Ha & Schmidhuber, 2018; Hafner, Lillicrap, Ba & Norouzi, 2020b), drawing inspiration from variational autoencoders and the principle that smooth manifolds can capture the underlying structure of high-dimensional observations. These models learn to encode observations into continuous vectors, with dynamics predicted through neural networks operating in this latent space. The continuous approach achieved notable successes, particularly in visual control tasks where smooth interpolation between states seemed natural (Hafner *et al.*, 2020a).

However, continuous world models face inherent limitations. The lack of compositionality makes it difficult to recombine learned concepts (Lake, Ullman, Tenenbaum & Gershman, 2017), poor interpretability obscures what features the model has learned (Locatello *et al.*, 2019), and planning requires expensive gradient-based optimization in the continuous space (Schrittwieser *et al.*, 2020). These challenges have motivated a shift toward discrete representations that offer structural advantages: natural compositionality, human interpretability, and compatibility with efficient combinatorial search algorithms (Van Den Oord, Vinyals & Kavukcuoglu, 2017; Ozair *et al.*, 2021).

1.5.2 Tokenization in World Models

Recent work has shown that discretizing observations into tokens can lead to more accurate and efficient world models. IRIS (Micheli, Alonso & Fleuret, 2023) pioneered this approach by using VQ-VAE to create a vocabulary of image tokens, then modelling dynamics as sequence prediction over these tokens. This discrete approach enables the model to simulate millions of steps accurately while maintaining interpretability.

1.5.3 Attention Mechanisms for Long-Range Dependencies

Transformers have proven particularly effective for world modelling due to their ability to capture long-range dependencies without the vanishing gradient problems of RNNs. The masked world model (MWM) (Seo *et al.*, 2023a) and transformer-based world model (TWM) (Robine, Höftmann, Uelwer & Harmeling, 2023a) demonstrate how attention mechanisms can improve both accuracy and efficiency in dynamics learning.

Chapter 3, "Learning to Play Atari in a World of Tokens" by Pranav Agarwal, Sheldon Andrews, and Samira Ebrahimi Kahou, introduces DART (Discrete Abstract Representations for Transformer-based learning), modelling both policy and the world using discrete tokens.

DART introduces several key contributions that collectively push the boundaries of discrete world modelling. The dual transformer architecture combines a GPT-style decoder for auto-regressive world modelling with a ViT-style encoder for policy learning, leveraging the strengths of each architectural pattern for its respective task. This design enables token-based reasoning by discretizing not just observations but entire trajectories—states, actions, and rewards—into a standard symbolic alphabet, creating a unified representational space where all aspects of the RL problem can be manipulated with the same discrete operations. The framework's novel approach to memory eschews traditional recurrence in favour of treating memory as attention, where past information is aggregated through self-attention mechanisms that can selectively retrieve relevant historical context. This architectural coherence translates into state-of-the-art sample efficiency, achieving a 0.790 median human-normalized score on the challenging Atari 100k benchmark and surpassing human performance in 9 out of 26 games. These results demonstrate that viewing RL through the lens of discrete token manipulation not only provides conceptual clarity but also yields superior empirical performance.

This work demonstrates how discrete world model priors—by forcing the agent to reason in terms of reusable tokens rather than continuous vectors—enable both superior sample efficiency and enhanced interpretability through attention visualization.

1.6 Skill Priors: Adaptive Libraries for Continual Learning

The third form of prior concerns how agents organize and reuse learned behaviours. Traditional RL learns monolithic policies that struggle with multiple tasks, while hierarchical approaches often require rigid, pre-specified structures.

1.6.1 The Continual Learning Challenge

The continual learning setting exposes fundamental limitations in how current RL systems organize and maintain knowledge. Catastrophic forgetting represents perhaps the most common failure mode (Kirkpatrick *et al.*, 2017), where learning new tasks doesn't merely fail to leverage previous knowledge but actively destroys it, overwriting carefully tuned parameters with new values that serve the current task but eliminate previous capabilities. This overwriting is particularly problematic because it's often silent—agents don't know what they've forgotten until tested on previous tasks. Negative transfer compounds this problem when tasks are not merely independent but actively incompatible (Taylor & Stone, 2009), causing interference patterns where skills learned for one task actively harm performance on another. The scalability challenge (Rusu *et al.*, 2016) emerges as a natural consequence of these failures: without effective knowledge reuse, training time grows linearly or worse with the number of tasks, making it impractical to develop agents with diverse skill repertoires. Together, these challenges have prevented RL from achieving the kind of cumulative, lifelong learning that characterizes human intelligence (Parisi, Kemker, Part, Kanan & Wermter, 2019).

Early work in physics-based character animation demonstrated these challenges, where controllers trained for specific motions struggled to generalize (Liu, Hertzmann & Popović, 2005; Faloutsos, van de Panne & Terzopoulos, 2001). These challenges arise from the lack of structured representations of skills that can be selectively activated and combined. Recent character animation work has shown promise in addressing this through hierarchical approaches (Peng, Berseth, Yin & Van De Panne, 2017; Peng, Abbeel, Levine & van de Panne, 2018).

1.6.2 Discrete Skill Representations

The shift toward discrete skill representations reflects a fundamental insight about the nature of reusable behaviour. Modularity emerges naturally in discrete systems where skills can be selectively activated or deactivated without the interference patterns that plague continuous representations—when skill A is encoded as tokens [1, 5, 9] and skill B as tokens [2, 7, 11], there's no gradient-based coupling that causes learning one to degrade the other. This modularity directly enables interpretability, as each discrete skill can be examined, understood, and even visualized as a distinct behaviour pattern rather than an opaque region in continuous space. The composability of discrete skills perhaps offers the most significant advantage: just as words can be recombined to express novel thoughts, discrete skill tokens can be assembled in new configurations to address tasks that weren't anticipated during training. This compositional potential transforms the learning problem from acquiring monolithic policies to building vocabularies of reusable behavioural components. This insight has been successfully applied in character animation, where discrete motion primitives enable the synthesis of complex behaviour (Liu, Yin, van de Panne, Shao & Xu, 2010).

Vector-quantized approaches (Van Den Oord *et al.*, 2017) have emerged as a powerful tool for learning such discrete skill representations.

1.6.3 Adaptive Skill Libraries

Rather than learning a fixed set of skills, adaptive approaches grow their skill libraries as needed. This enables efficient reuse while maintaining the flexibility to acquire genuinely novel behaviours. Several new architectures have emerged to support adaptive skill libraries that grow with experience. Progressive networks (Moeed *et al.*, 2020) are among the first approaches to adding new capacity for each task while keeping previous parameters frozen, creating a tree-like structure where new skills can leverage but not interfere with existing knowledge. This approach guarantees zero forgetting but at the cost of linear growth in parameters. Mixture of experts architectures (Shazeer *et al.*, 2017b) take a different approach, maintaining a fixed set of expert modules but learning to route between them based on the current context, effectively creating a switching system that can combine different specializations. This approach has been particularly successful in character animation for combining diverse motor skills (Peng *et al.*, 2018; Peng, Chang, Zhang, Abbeel & Levine, 2019).

Dynamic architectures push adaptivity further by growing networks based on task demands, adding neurons or layers only when current capacity proves insufficient for new challenges (Clegg, Tan, Turk & Liu, 2018). These approaches share the insight that fixed-capacity architectures fundamentally limit continual learning, but differ in how they manage the trade-off between capacity growth and computational efficiency. The need for such adaptive approaches is particularly acute in character animation, where agents must master an ever-expanding repertoire of motor skills (Peng *et al.*, 2017; Won & Lee, 2019).

Chapter 4, "STRIDE: Continual Learning using Skill Transfer and Reuse with Incremental Discrete Encoding" by Pranav Agarwal, Michele Rocca, Victor Zordan, and Sheldon Andrews, presents a novel framework for continual skill learning.

STRIDE introduces several key components that collectively enable efficient, continual learning. At its core, the framework employs a dynamic VQ-VAE codebook that automatically expands its skill vocabulary only when the current repertoire cannot adequately express new behaviours, avoiding the waste of fixed-size architectures while maintaining the flexibility to learn genuinely novel skills. To prevent catastrophic forgetting, STRIDE freezes previously learned skill slices, ensuring that earlier behaviours remain intact even as new capabilities are added. New behaviours are learned through lightweight composition, where a simple gating network learns to combine existing primitives rather than training entire policies from scratch. Remarkably, the discrete skills that emerge are human-interpretable, with each codeword corresponding to an understandable motor primitive such as "Turn-Right" or "Sprint," providing unprecedented transparency into the agent's skill repertoire. These architectural changes translate into dramatic efficiency gains, with learning speeds improved by factors of 1.5 to 2.6 compared to state-of-the-art baselines, demonstrating that thoughtful prior structure can yield both theoretical elegance and practical performance improvements.

This work demonstrates how adaptive skill priors, characterized by growing vocabularies of reusable motor primitives, enable truly continual learning without the interference and forgetting that plague traditional approaches.

1.7 Combining Different Priors

The synergies between different prior types create opportunities for multiplicative improvements in learning efficiency. When reward priors combine with world model priors (Fu, Singh, Ghosh, Yang & Levine, 2018; Wulfmeier, Ondruska & Posner, 2016), the enhanced state representations learned by the world model provide richer features for reward learning, enabling more accurate inference of human preferences from limited demonstrations. The connection between world models and skill priors proves (Pertsch, Lee & Levine, 2021; Sharma, Gu, Levine, Kumar & Hausman, 2020) particularly natural in discrete settings, where the same tokens used to represent environment states can serve as building blocks for skill construction, creating a unified vocabulary for perception and action. The interplay between skill priors and reward priors (Barreto *et al.*, 2017; Krishnan *et al.*, 2017) enables a cycle where previously learned reward functions guide the acquisition of new skills, which in turn provide better action abstractions for learning future rewards. These synergies suggest that the most significant gains in RL efficiency will come not from improving each prior type in isolation but from architectures that enable them to reinforce each other (Co-Reyes *et al.*, 2021).

The Decision Transformer (Chen *et al.*, 2021b) exemplifies this integration by treating RL as sequence modelling over states, actions, and rewards—effectively combining all three prior types.

1.8 Architectural Considerations

The success of modern architectures, particularly transformers (Vaswani *et al.*, 2017), in prior modelling stems from several architectural properties that align naturally with the requirements of structured knowledge representation. Attention mechanisms (Bahdanau, Cho & Bengio, 2015) provide a learnable way to focus on relevant information while maintaining access to the full context, crucial for identifying which prior knowledge applies to the current situation. The scalability properties of these architectures mean that performance improves smoothly with increased model capacity and data availability (Brown *et al.*, 2020), avoiding the plateaus that limit simpler approaches. Their multi-modal processing capabilities (Radford *et al.*, 2021; Lu, Batra, Parikh & Lee, 2019) enable unified handling of diverse input types—from visual observations to language instructions to proprioceptive signals—within a single framework, facilitating the kind of cross-modal prior transfer that humans excel at. The parallelization afforded by attention-based architectures makes training practical on modern hardware, transforming what would be sequential bottlenecks into parallel computations. While transformers represent the current state-of-the-art, the key insight transcends any particular architecture: effective prior modelling requires architectures that support discrete, compositional representations with efficient mechanisms for selective retrieval and combination (Lee *et al.*, 2022b).

However, transformers are not the only solution. The key insight is that architectures should support discrete, compositional representations that can grow and adapt over time.

1.9 Practical Considerations

While the theoretical advantages of prior-based reinforcement learning are compelling, translating these ideas into working systems requires careful attention to practical implementation details. The gap between conceptual elegance and empirical success often lies in the engineering decisions that determine whether a prior-rich model fulfills its promise of efficient learning or becomes a system that underperforms simpler baselines. Real-world deployment of these systems demands consideration of multiple interacting factors: how to effectively train models that must balance various objectives, how to manage the computational trade-offs between model complexity and inference speed, and how to leverage the interpretability advantages that discrete priors provide. These practical considerations are not mere implementation details but fundamental aspects that determine whether prior-based RL can move from research demonstrations to deployed systems that solve real problems. The following sections examine the key practical challenges and emerging solutions, guiding researchers and practitioners seeking to harness the power of structured priors in their work.

1.9.1 Training Strategies

The training of prior-rich models demands careful orchestration of multiple learning phases and objectives. Pre-training on large, diverse datasets allows models to acquire general priors that capture broad regularities before encountering specific tasks, much as humans benefit from years of general experience before specializing. The fine-tuning phase adapts these general priors to particular domains, requiring a delicate balance between leveraging existing knowledge and accommodating new requirements. Regularization techniques must be particularly sophisticated in prior-based systems, preventing over-fitting to training data while maintaining the flexibility to acquire genuinely new knowledge when existing priors prove insufficient. Curriculum learning provides a principled way to manage complexity, gradually increasing task difficulty to build upon previously acquired priors rather than overwhelming the system with complexity it lacks the foundation to handle. These training strategies work together to ensure that prior learning enhances rather than constrains future adaptability.

1.9.2 Computational Requirements

The computational profile of prior-rich models presents both challenges and opportunities that differ markedly from traditional RL approaches. Memory requirements expand beyond simple parameter storage to include discrete codebooks that grow with experience and attention matrices that capture relationships between all elements in a sequence, potentially consuming gigabytes for large-scale systems. Inference costs can become prohibitive as attention mechanisms scale quadratically with sequence length, making long-horizon reasoning exponentially more expensive and necessitating careful architectural choices about context windows and attention sparsity. Training complexity increases not just from larger models but from managing multiple interacting objectives—world model accuracy, reward prediction, skill distinctiveness—each potentially pulling parameters in different directions and requiring sophisticated optimization strategies to balance. However, these upfront computational costs must be weighed against the dramatic reductions in environment interactions, where a 2× increase in training computation might yield a 10× reduction in required samples, a trade-off that often favours prior-rich approaches in domains where data collection is expensive or dangerous.

1.9.3 Interpretability and Trust

The interpretability advantages of discrete prior modelling extend beyond visualization to enable genuine understanding of agent behaviour. Attention visualization in prior-based systems reveals not just what the model focuses on but why, as discrete tokens correspond to meaningful concepts that humans can recognize and reason about. The ability to inspect discrete skills transforms debugging from black-box probing to systematic analysis, where individual motor primitives can be examined, tested in isolation, and even modified by human operators who understand their function. Reward attribution becomes tractable when rewards are learned from discrete demonstrations, enabling practitioners to trace specific policy behaviours back to the human examples that influenced them, creating an audit trail that's crucial for safety-critical applications. This interpretability doesn't come at the cost of performance. Instead, it emerges naturally from the discrete, compositional structure that also enables efficient learning, suggesting that transparency and capability can be complementary rather than competing objectives.

This interpretability is crucial for building trust in safety-critical applications.

1.10 Limitations and Future Directions

Despite significant progress, prior-based RL faces several persistent limitations that must be acknowledged and addressed. The computational cost of prior-rich models often requires substantial resources that may be prohibitive for smaller research groups or real-time applications, creating a barrier to entry that could limit innovation. Architecture sensitivity remains a significant challenge, where seemingly minor design choices—the size of a codebook, the number of attention heads, the discretization resolution—can dramatically impact performance, making it difficult to develop robust guidelines for practitioners. Our limited theoretical understanding of when and why priors accelerate learning leaves us relying primarily on empirical validation, lacking the formal guarantees that would enable confident deployment in safety-critical domains. The domain specificity of learned priors presents perhaps the most fundamental challenge: priors that powerfully accelerate learning in one domain may actively harm performance in another, requiring careful consideration of transfer boundaries and potentially limiting the dream of truly universal agents. These limitations remind us that prior-based RL, while promising, remains an active area of research with substantial open questions.

1.10.1 Future Research Directions

Several promising research directions could further advance prior-based RL. Multi-modal priors that integrate vision, language, and action representations could enable agents to leverage diverse sources of information, learning from both visual demonstrations and verbal instructions while maintaining action-grounded representations. Human-in-the-loop learning presents opportunities for interactive prior refinement, where human feedback could directly shape the discrete vocabularies and compositional rules that agents use. On the theoretical front, developing formal foundations for analyzing prior effectiveness would help us understand when and why specific prior structures accelerate learning, moving beyond empirical observations to principled design. Hardware acceleration specifically optimized for discrete operations could make prior-rich models more practical, as current architectures are primarily designed for continuous computation. Finally, advancing compositional generalization through priors that enable zero-shot task combination would represent a significant step toward truly general artificial intelligence, where agents could tackle novel challenges by creatively recombining their learned components without additional training.

1.11 Conclusion

In this chapter, we have examined how improved prior modelling addresses the fundamental limitations of reinforcement learning. By moving beyond tabula rasa learning to incorporate structured priors—whether for rewards, world models, or skills—we can build RL systems that learn efficiently, generalize broadly, and remain interpretable.

The key insight is that effective priors should be learnable, discrete, and adaptive. Learnable priors avoid the brittleness of hand-coded rules. Discrete priors enable compositionality and interpretability. Adaptive priors grow with experience rather than remaining fixed.

The three contributions examined in this thesis—expert demonstration-based reward modelling, token-based world modelling with DART, and continual learning with STRIDE—exemplify these principles. Together, they demonstrate speed-ups of up to 2.6× across diverse domains while maintaining or improving final performance. More importantly, they provide conceptual tools for reasoning about priors in RL and point toward systems that can be trusted in real-world applications.

As RL moves from research curiosity to practical tool, the need for efficient, generalizable, and interpretable systems becomes paramount. Improved prior modelling, we argue, is not just one approach among many—it is the key to unlocking RL's full potential. The future lies not in ever-larger models trained from scratch, but in thoughtfully structured priors that capture and reuse the regularities of our world.

CHAPTER 2

AUTOMATIC EVALUATION OF EXCAVATOR OPERATORS USING LEARNED REWARD FUNCTIONS

Pranav Agarwal [1] , Marek Teichmann[2] , Sheldon Andrews [1] , Samira Ebrahimi Kahou[1]

[1] Departement of Software and IT Engineering,
École de Technologie Supérieure,
1100 Notre-Dame Ouest, Montréal, Québec, Canada H3C 1K3,
[2] CM Labs Simulations,
645 Wellington St, Montreal, Quebec, Canada H3C 1T3

Article accepted at « NeurIPS Workshop on
Reinforcement Learning for Real Life (RL4RealLife) » in November 2022

Résumé

Former des utilisateurs novices à l'utilisation d'une excavatrice pour l'apprentissage de différentes compétences nécessite la présence d'experts. Compte tenu de la complexité du problème, il est relativement coûteux de trouver de tels experts, car le processus est long et exige une grande précision. De plus, les évaluations humaines étant intrinsèquement subjectives, elles sont vulnérables aux biais et aux lapses d'attention, ce qui introduit du bruit et entraîne une forte variance dans les scores attribués à des opérateurs ayant des niveaux de compétence comparables. Dans ce travail, nous abordons ces difficultés et proposons une stratégie novatrice pour l'évaluation automatique des opérateurs d'excavatrice. Nous prenons en compte à la fois la dynamique interne de la machine et les critères de sécurité à chaque instant pour évaluer la performance. Afin de valider davantage notre approche, nous utilisons ce modèle de prédiction des scores comme source de récompense pour un agent d'apprentissage par renforcement, chargé d'apprendre la tâche de manœuvrer une excavatrice dans un environnement simulé reproduisant fidèlement la dynamique du monde réel. Pour une politique apprise à l'aide de ces modèles externes de prédiction de récompenses, nos résultats démontrent des solutions plus sûres, respectant les contraintes dynamiques requises, comparées aux politiques entraînées uniquement avec des fonctions de récompense basées sur la tâche, ce qui rapproche notre méthode d'une adoption réelle. Pour les travaux futurs, nous rendons disponible notre code à l'adresse suivante : https://github.com/pranavAL/InvRL_Auto-Evaluate et nos résultats vidéo à ce lien.

Abstract

Training novice users to operate an excavator for learning different skills requires the presence of expert teachers. Considering the complexity of the problem, it is comparatively expensive to find skilled experts, as the process is time-consuming and requires precise focus. Moreover, because human evaluations are inherently subjective, they are susceptible to attention lapses and personal biases, often introducing noise and leading to high variance in the scores assigned to operators with comparable skill levels. In this work, we address these issues and propose a novel strategy for the automatic evaluation of excavator operators. We take into account the internal dynamics of the excavator and the safety criterion at every time step to evaluate the performance. To further validate our approach, we use this score prediction model as a source of reward for a reinforcement learning agent to learn the task of manoeuvring an excavator in a simulated environment that closely replicates the real-world dynamics. For a policy learned using these external reward prediction models, our results demonstrate safer solutions following the required dynamic constraints when compared to policies trained with task-based reward functions only, making it one step closer to real-life adoption. For future research, we release our codebase at https://github.com/pranavAL/InvRL_Auto-Evaluate and video results link.

2.1 Introduction

Receiving feedback is crucial for effective task learning Burgess, van Diggele, Roberts & Mellis (2020). Traditionally, feedback is provided by an expert teacher in the form of a score, which can be biased and noisy, especially for complex tasks (Boysen & Vogel, 2009). Moreover, obtaining frequent feedback from human experts can be challenging, and it often involves only a final pass/fail evaluation Vollmer, Wrede, Rohlfing & Oudeyer (2016). Similarly, for efficient training of a reinforcement learning (RL) policy, the agent's action must be evaluated at every time step through a reward from the environment, known as a dense reward. However, manually modelling the reward function based on machine dynamics and safety criteria is difficult and leads to sub-optimal policies (Zhu, Lin, Dai & Zhou, 2022). Sparse binary feedback, where success or failure is provided at the end of an episode, leads to slow convergence due to the credit-assignment problem (Rengarajan, Vaidya, Sarvesh, Kalathil & Shakkottai, 2022). Hence, reward formulation is critical for policy optimization and performance (Dayal, Cenkeramaddi & Jha, 2022). To overcome these challenges, inverse reinforcement learning with expert demonstrations can automate reward prediction and improve learning performance (Lim, Ha & Choi, 2020),(Rosbach, James, Großjohann, Homoceanu & Roth, 2019),(Ng *et al.*, 2000).

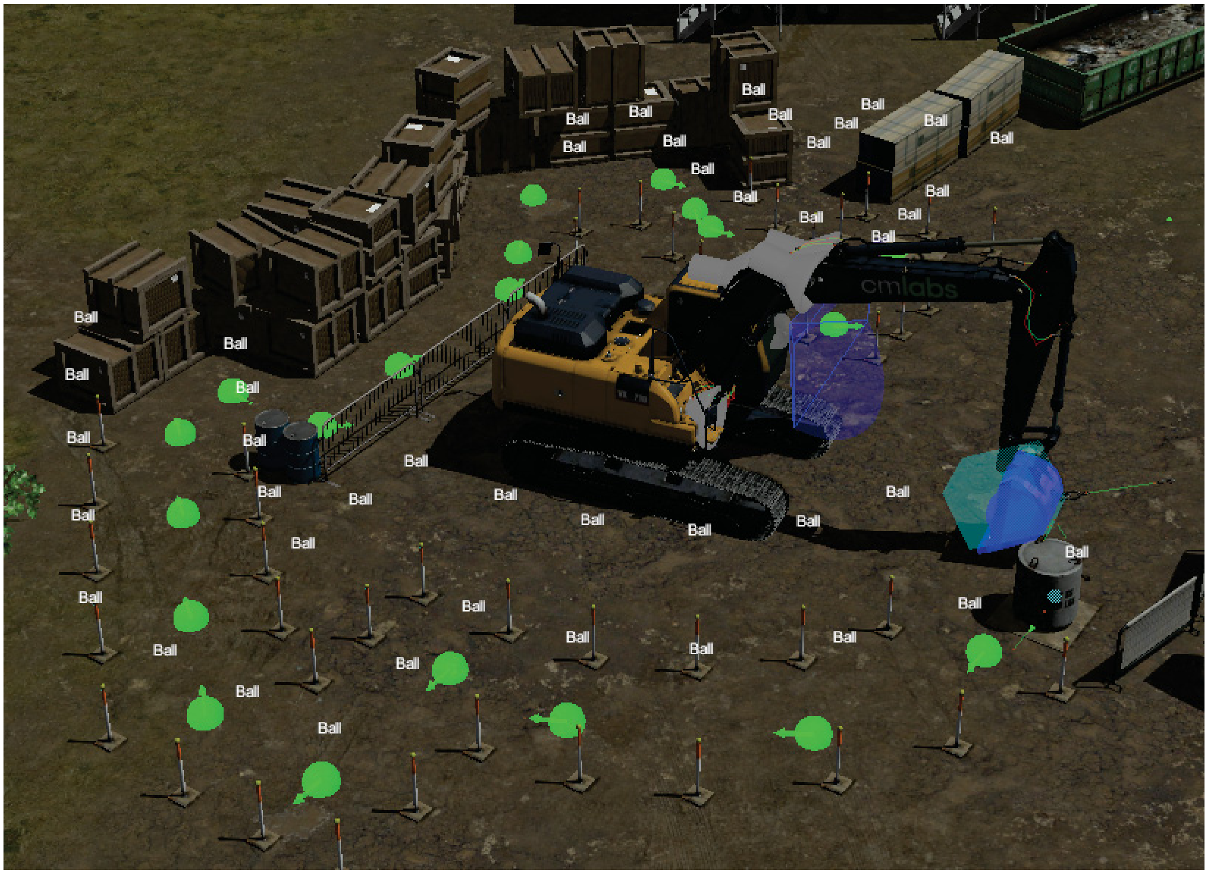


Figure 2.1 The Vortex simulator used in this work closely replicates a real-life setting and matches the dynamics of a real excavator. As a result, learning how to operate an excavator using a reinforcement learning policy in this environment is a complex task, unlike learning to play complex games in unrealistic environments

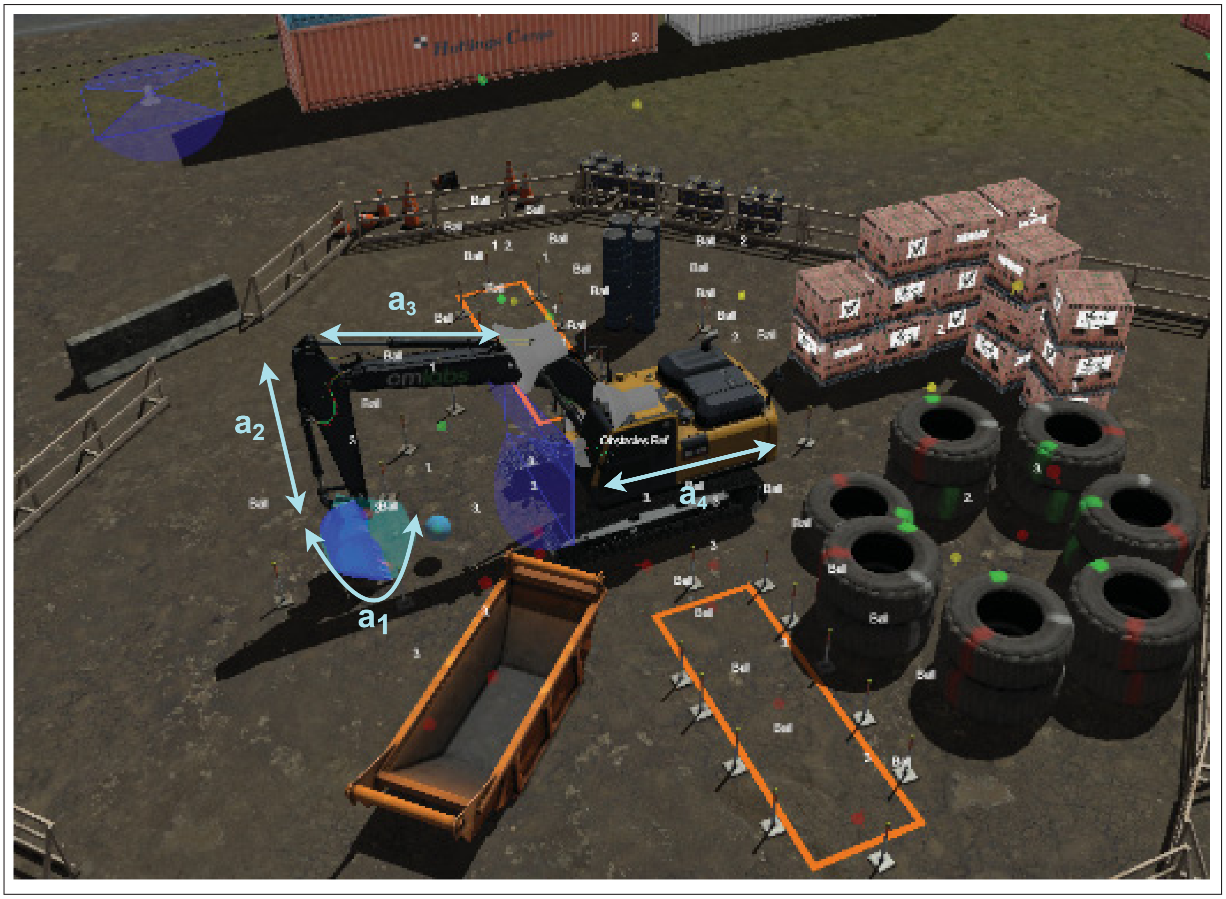


Figure 2.2 In this work, we utilize the Manoeuvring scenario of the Vortex Simulator environment. This scenario is used to gather data from expert operators, train novice users, and train the RL policy using the proposed automatic score prediction model. The environment requires the user or the final policy to achieve a fixed goal, which is determined by the distance from the starting point. The task involves coordinating the movement of multiple actuators, including the Bucket (a_1), Stick (a_2), Boom (a_3), and Swing (a_4), while considering the dynamic constraints and avoiding infractions

Excavators are widely used in urban and rural areas for various construction tasks, such as excavation, loading-unloading, and soil levelling. However, these tasks require precise control of the actuators while considering the safety criteria of the surrounding environment. Moreover, the long working hours and the need for constant focus can negatively impact the physical and mental well-being of operators (La Hera & Morales, 2019), leading to reduced overall performance. Excavator automation is necessary to improve productivity and address these issues. Although attempts have been made to automate excavators by manually modelling the controllers (Koivumäki & Mattila, 2015), (Zavodni, Nojpert & Arnold, 2009) for each actuator, these methods require domain-knowledge (Baroglio, Giordana, Piola, Kaiser & Nuttin, 1996), hence are not scalable. A learning-based approach that considers feedback based on environmental interactions can significantly improve performance. However, it is challenging to train an RL policy for excavator automation due to the sensitivity of the policy to the given reward function (Everitt *et al.*, 2021). Therefore, we leverage this sensitivity of RL to validate our score prediction model, which can guide new trainees.

The goal of our work is to create an automatic score prediction model that predicts the proficiency of trainees learning to operate heavy construction vehicles, specifically excavators. To validate our model, we use it as a reward function for training an RL policy (as shown in Fig. 2.8). This reward function is learned from expert data using a probabilistic approach. In particular, the model jointly considers the excavator's dynamics and the number of infractions (such as collisions with surrounding objects due to unsafe behaviour) in order to assign scores that reflect both efficiency and safety. By capturing these complementary aspects, the model provides a structured and informative reward signal rather than a simple pass–fail score, making it a reliable source for guiding reinforcement learning.

To achieve this, we collect a diverse range of user data during excavator manoeuvring in a simulated environment. This data consists of sensor readings and per-step infraction values for each user. Using the temporal distribution of these sensor readings, we formulate an objective for learning the dynamic distribution of an excavator when operated by an expert. We aim to encode the dynamics by predicting future states (Epstein, Wu, Schmid & Sun, 2021). This dynamic distribution is used as a reward function since an expert and a novice user will have different action dynamics (Ikegami & Ganesh, 2014). To consider safety criteria, we learn a separate distribution by predicting the number of infractions for each time step, given the environment information.

We then model these distributions as reward functions to optimize the RL policy for excavator automation in a simulated environment. By predicting the next state and the number of infractions, the learned distributions enable efficient differentiation of actions according to their level of performance, leading to enhanced manoeuvring while avoiding unsafe behaviour.

To summarize, our work presents three main contributions:

- A diverse dataset of users operating an excavator in a simulated environment. The dataset collects high-level information about the internal dynamics of the excavator and the corresponding infraction values. Users interact with the environment, and the dataset provides valuable insights into the excavator's behaviour.
- An automated score prediction framework that utilizes the action dynamics and infraction values of an expert user to predict the performance of a reinforcement learning policy in real-time. This framework provides valuable feedback to both external users and the policy as they learn the task.
- A demonstration of excavator automation in a simulated environment that closely replicates real-world conditions. Specifically, the proposed reward function is used to guide the excavator in the task of path manoeuvring. This demonstration highlights the potential of the proposed approach for practical applications in the real world.

2.2 Research Problem and Motivations

Reward Shaping. Inspired by human learning, reward shaping is a well-studied problem that involves optimizing an RL policy using an external signal, as elaborated in the work of Ng, Harada & Russell (1999), and more recently by Hu *et al.* (2020b). The current trend mostly involves using a task-based sparse reward (Charlesworth & Montana, 2020), (Wang, Wang, Wang & Zhang, 2020a) or manually designing a reward function for optimizing the policy (Ng *et al.*, 1999). Sparse reward requires the agent to explore novel strategies (Riedmiller *et al.*, 2018) for solving a task and can lead to slow (or no) convergence (Rengarajan *et al.*, 2022). A dense (or per-step) reward reduces the training time by providing feedback for every action (Luo *et al.*, 2020). It is usually not straightforward to manually design dense reward functions, as the resulting optimal policy is sometimes hard to predict. Considering the simplicity of sparse reward, most of the current work uses it over dense reward. (Anzalone, Barra, Barra, Castiglione & Nappi, 2022) proposes a goal-based sparse reward with a curriculum, along with manually designed constraints for the task of autonomous driving. Other strategies include using an intrinsic reward, which is a function of the state visited, as used in (Pathak, Agrawal, Efros & Darrell, 2017), as additional supervision along with the original reward. These approaches successfully demonstrate novel strategies for different complex tasks. However, none of these works includes subject- and safety constraints based on expert demonstrations of the original reward function leading to unsafe policies. By integrating a score predictive model learned using expert data, which is also used as a feedback mechanism for external users, we demonstrate excavator automation for the task of manoeuvring in a challenging environment.

Excavator Automation using Reinforcement Learning. Excavator Automation is a challenging task that requires precisely coordinated control of multiple actuators in an unstructured environment. Considering the complexity and limited availability of accurate simulators, this problem has been rarely explored. Recent work by Jud, Leemann, Kerscher & Hutter (2019) used classical PD and PID controllers for controlling the excavators. Model-based solutions (Dolgov, Thrun, Montemerlo & Diebel, 2010) were proposed, which require domain expertise and hence are time-consuming and do not transfer well to new environments. Taking these issues into account, the current research is moving towards a learning-based approach. In particular, reinforcement learning seems an ideal solution considering its success in solving a wide range of complex robotics manipulation tasks (Chen, Xu & Agrawal, 2022a),(Andrychowicz *et al.*, 2020). Some of the recent work for excavator automation using RL includes (Egli & Hutter, 2022),(Egli & Hutter, 2020), which learned trajectory tracking for controlling excavator arms by penalizing the policy based on the difference between the expected and the actual trajectory. (Andersson, Bodin, Lindmark, Servin & Wallin, 2021) used RL for operating forestry cranes for the task of loading-unloading logs using multiple rewards for optimizing grasp success rate, energy, and distance in a curriculum setting. Given the simplicity of the task, each approach relied on manually modelling the reward, limiting its scope to other complex functions like excavation or manoeuvring in a complex environment. In contrast, we focus on optimizing the policy using an expert's demonstration, which forces the RL agent to learn the required task while at the same time optimizing the actions to match the expert's dynamic distribution and infractions, leading to safer policies following the required constraints.

Learning Distributions for Reward Prediction. To model an expert's action distribution, we learn two independent representations for different features, taking into account the dynamics and the infraction values. This representation is learned following a Variational Autoencoder (Kingma & Welling, 2019b) like model learning a distribution without any external annotations. Prior works have proposed different strategies for modelling this distribution using auxiliary tasks. (Lee *et al.*, 2020b) uses action-conditioned optical flow prediction of the input RGB images and multiple binary classification losses. This forces the representation to take into account the action dynamics and their impact on the given image, which is an essential factor while training a reinforcement learning agent. (Lesort, Díaz-Rodríguez, Goudou & Filliat, 2018) describes several representation learning strategies, including learning a forward model, using feature adversarial learning, and using rewards as objectives. In the current work, we follow a similar approach as (Hu, Cotter, Mohan, Gurau & Kendall, 2020a). Given expert trajectories with multiple sensor states, we learn a common representation by predicting future states. This dynamic representation of expert trajectories encodes different excavator states. Using this as a reward function, we optimize the RL policy to complete a goal with a constraint such that the policy's state distribution follows the expert (discussed in detail in Section 4). Furthermore, an independent distribution that takes into account the infraction value is also learned to model the safety criteria, where the distribution is learned using the objective of predicting the total infraction values given the environment state.

2.2.1 Problem Statement

The primary objective of this work is to develop a real-time feedback mechanism for excavator operators that predicts their scores. To achieve this goal, we propose a score-predictive model and use it as a reward function for training an excavator automation policy. The policy is designed to manoeuvre the excavator in a complex environment while accounting for multiple external infractions (as depicted in Fig. 2.2).

By training an RL policy, we aim to validate our score prediction model for using it as feedback for external users. Additionally, we explore the importance of integrating these rewards as a form of constraint during the policy learning process.

To leverage our proposed reward model for policy learning, we start by learning multiple independent distributions: a probabilistic temporal representation $\mathbf{q}_D(\mathbf{z})$ of excavator states based on sampled expert trajectories from the original dataset, which models the dynamics, and $\mathbf{q}_I(\mathbf{z})$ of the given environment states at each time step to model infractions.

To model the dynamics, we model the sequence of N different expert trajectories $\mathbf{T}$: $\mathbf{T}_0, \mathbf{T}_1, ..., \mathbf{T}_{N-1}$ as a window of x time steps. For a given time step t, the input data is equivalent to $\mathbf{I}_t$: $X_t, X_{t+1}, ..., X_{t+x-1}$, and the corresponding label is given as $\mathbf{O}_t$: $X_{t+1}, ..., X_{t+x}$, representing different dynamic features of excavator. Our goal is to learn an efficient representation $\mathbf{q}_D(\mathbf{z})$ for a given input data $\mathbf{I}_t$ to predict x future states $\mathbf{O}_t$ by minimizing the reconstruction loss shown in Eqn. 2.1. The $\mathcal{KL}$ divergence loss acts as a regularizer, ensuring that the learned distribution follows a normal distribution. The representation is used to predict the reward $\mathbf{R}_D$, by calculating the $\mathcal{KL}$ distance between its distribution and the distribution of the dynamic features after a new action:

$$L_{Dt} = \|X_t - \hat{X}_t\|^2 + \mathcal{KL}(\mathbf{q}_D(\mathbf{z})\|p_D(\mathbf{z})) \tag{2.1}$$

Here, L_{Dt} is the reconstruction loss for time step t, $\hat{X}_t$ is the predicted future state at time step t. We represent each input sequence through a latent variable $\mathbf{z}$, which encodes the underlying dynamic distribution. The $\mathcal{KL}$ divergence between $\mathbf{q}_D(\mathbf{z})$ and $\mathbf{p}_D(\mathbf{z})$ is calculated using the $\mathcal{KL}$ function. Further details are provided in Section 2.2.4.

The prior distribution $p_D(\mathbf{z})$ is a normal distribution $\mathcal{N}(0, 1)$ with mean 0 and variance 1, as shown in Eqn. 2.2.

$$p_D(\mathbf{z}) \sim \mathcal{N}(0, 1) \tag{2.2}$$

To model the infraction distribution $\mathbf{q}_I(\mathbf{z})$, a similar approach is used, with modifications to the loss function. At a given time step t, instead of using the full environment state, we extract a set of infraction features represented as:

$$\mathbf{K}_t = k_t^1, k_t^2, \ldots, k_t^f \tag{2.3}$$

where each k_t^i corresponds to the number of a specific type of violation and f is the total number of distinct infraction types being monitored. These features form the input for learning the infraction distribution. The distribution is optimized using the objective of predicting the sum of the total infractions S_t and the $\mathcal{KL}$ divergence loss as shown in Eqn. 2.4. Since the distribution is optimized for the total infractions, the final reward $\mathbf{R}_S$ is calculated as the $\mathcal{KL}$ distance between the distribution of the infraction values of an expert and the current policy:

$$L_{It} = \|S_t - \hat{S}_t\|^2 + \mathcal{KL}(\mathbf{q}_I(\mathbf{z})\|\mathbf{p}_I(\mathbf{z}))\,. \tag{2.4}$$

Here, $\mathbf{p}_I(\mathbf{z})$ follows the same normal distribution as $\mathbf{p}_D(\mathbf{z})$. To prevent the latent distribution from collapsing and to encourage smoothness, we add a $\mathcal{KL}$ regularization term between the posterior $\mathbf{q}_I(\mathbf{z})$ and a simple prior $\mathbf{p}_I(\mathbf{z})$ (standard Gaussian). Importantly, the $\mathcal{KL}$ term does not directly generate the infraction count S_t; instead, S_t is predicted using the reconstruction loss. The $\mathcal{KL}$ serves only to regularize the latent encoding of infractions, ensuring stability and generalization, especially given the low-dimensional latent space (2D in our implementation).

To optimize the automation of excavator movements in a complex workspace, we formulate the problem as a Markov Decision Process (MDP). The MDP is defined by a state space $\mathbf{S}$, representing the excavator's configuration in the workspace, and an action space $\mathbf{A}$, representing the actuator commands that can be sent to the excavator. The transition function of this MDP is stochastic, as the excavator may encounter unpredictable obstacles or other hazards in the workspace. We use a dynamic reward function $\mathbf{R}_D$ to capture the quality of the excavator's movement. To encourage the excavator to avoid collisions and other dangerous situations, we use a safety reward function $\mathbf{R}_S$. Finally, to incentivize the excavator to complete the task, we use a task reward function $\mathbf{R}_T$. The goal is to find a policy π_θ that maximizes the expected discounted reward $J(\pi)$, where the discount factor γ balances short-term and long-term rewards when making decisions. We use Eqn. 2.5 to define $J(\pi)$, which is the expected sum of discounted reward obtained by following the policy π_θ. By formulating the excavator automation problem as an MDP, we provide a formal framework for decision-making in complex and stochastic environments, allowing for efficient and effective automation of excavator movements.

$$J(\pi) = E_{\pi_\theta}\left[\sum_t \gamma^t r(s_t, a_t)\right] \tag{2.5}$$

We model both the latent space $\mathbf{z}$ and policy π using neural networks. The latent space distribution $\mathbf{q}_D(\mathbf{z})$ is modelled using a Long Short Term Memory (LSTM) (Hochreiter & Schmidhuber, 1997), while $\mathbf{q}_I(\mathbf{z})$ and policy π use a simple 2-layer deep multi-layer perceptron. We elaborate on the methodology in Section 2.2.4 and the results in Section 2.2.5. As there is limited work in this direction, we compare the performance of our proposed reward function with a policy trained using a task-based reward as a baseline.

2.2.2 Simulator

The proposed work utilizes the Vortex Simulator (CM-Labs Simulations Inc., 2024), which offers a diverse range of realistic scenarios that closely replicate real-life tasks such as excavation, loading and unloading materials, creating trenches, and lifting heavy loads, as illustrated in Fig. 2.1. These realistic scenarios provide an ideal testing ground for training new students in the task of operating these heavy machines before moving on to real-world machines. These training modules are built using the Vortex Studio software platform, a modular platform that offers a comprehensive set of tools for simulating, training, visualizing, planning, and prototyping.

The simulator used in this work is designed to replicate a real 21-ton hydraulic excavator with a 119-kW engine. The simulator's software is built using exact specifications to ensure that it closely mimics the performance of a real hydraulic excavator. The simulator features several key

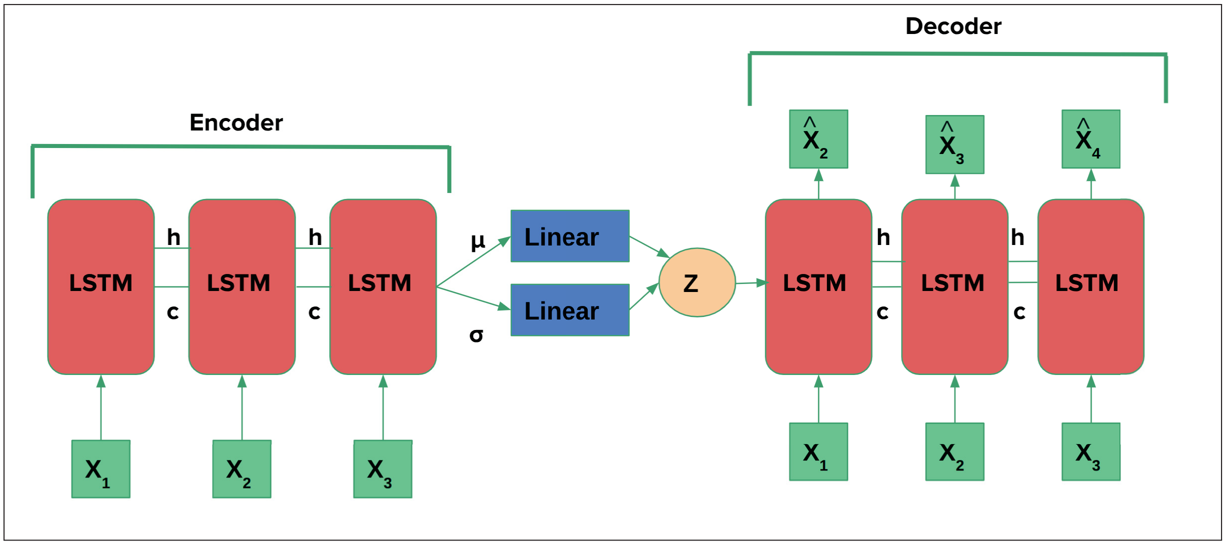


Figure 2.3 Model for Learning the Dynamic Distribution. The latent vector **z** encoding the expert's dynamic distribution is learned using an LSTM-based encoder-decoder by optimizing for the task of future state prediction. During training, the decoder uses the original last state (teacher-forcing strategy). It predicts the future state using LSTM by initializing its hidden state and cell state with the dynamics predicted by the encoder

parts, including the bucket, arm, boom, cab, engine compartment, idlers (front of the track), and driving sprockets (rear of the tracks). It uses realistic tracks with idlers, driving sprockets, pumps, and track shoes. The heavy-duty bucket has a width of 1067 mm, weighs 768 kg, and has a capacity of 0.83 m^3, with five teeth and a digging force of 164.4 kN. The physics simulation and visual rendering were updated at a rate of 60 Hz. The excavator and training scenario is shown in Fig. 2.4.

The simulator setup provides a comprehensive training experience, featuring realistic equipment simulation, a simulated work site, and physical operator controls. The operator controls consist of foot pedals and joysticks that are designed to replicate the actual controls found on a real excavator. Fig. 2.5 illustrates the design of this setup. The joysticks are used to control the boom, arm, and bucket of the excavators, while the buttons on the joystick control the horn and power digging function. The bi-directional foot pedals are used to control the brakes and engine throttle. The simulator also features two displays - a central display that shows the view from the operator's side and a secondary display that displays the simulated human-machine interface. This setup

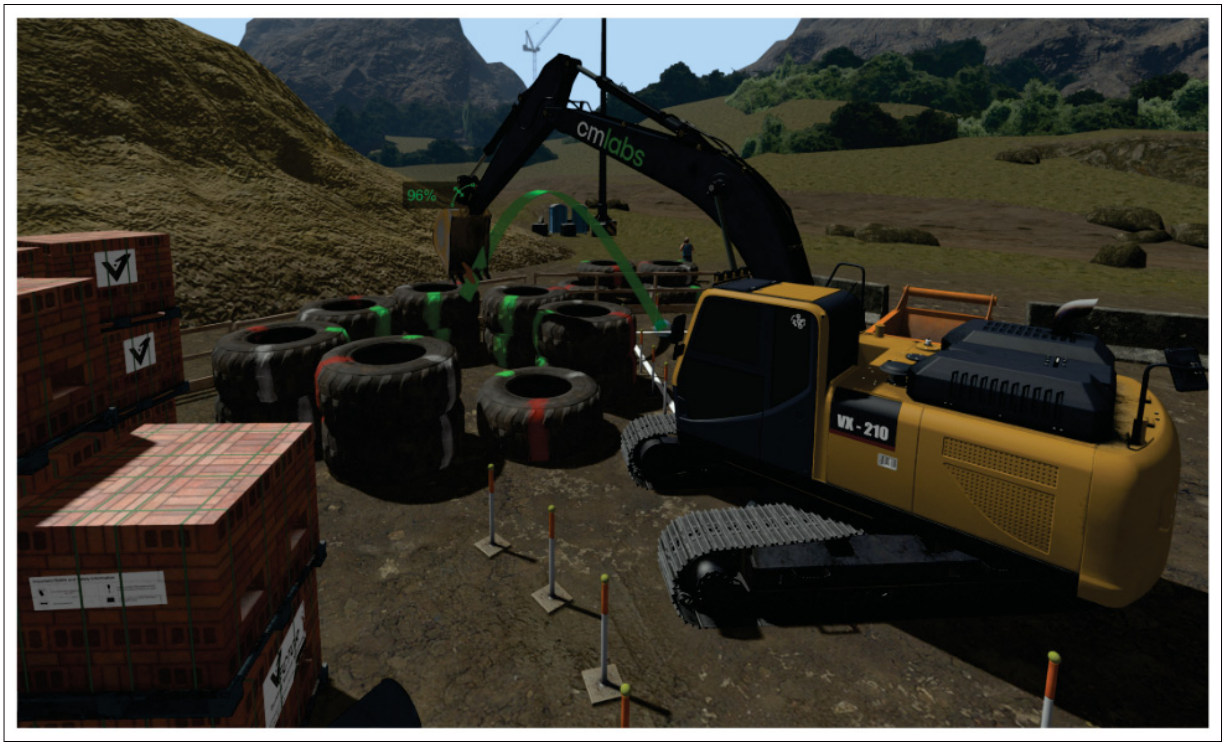


Figure 2.4 The Vortex Excavator environment presents a challenging, yet effective, tool for training users on the safe and efficient manoeuvring of the excavator while considering multiple external factors. The complex task requires precise coordination of multiple actuators while also accounting for the machine's internal dynamics and various external infractions

provides an immersive training experience that closely mirrors the real-world operation of an excavator.

2.2.3 Dataset

The Vortex simulator's excavator environment provides access to a wide range of critical excavator-specific features that can be used to evaluate the user's performance for the given task. These features can be divided into two categories: dynamic features and safety features. The dynamic features, described in Table 2.1, pertain to the internal dynamics of the excavator. In contrast, the safety features shown in Table 2.2 relate to the excavator's interaction with the external environment.

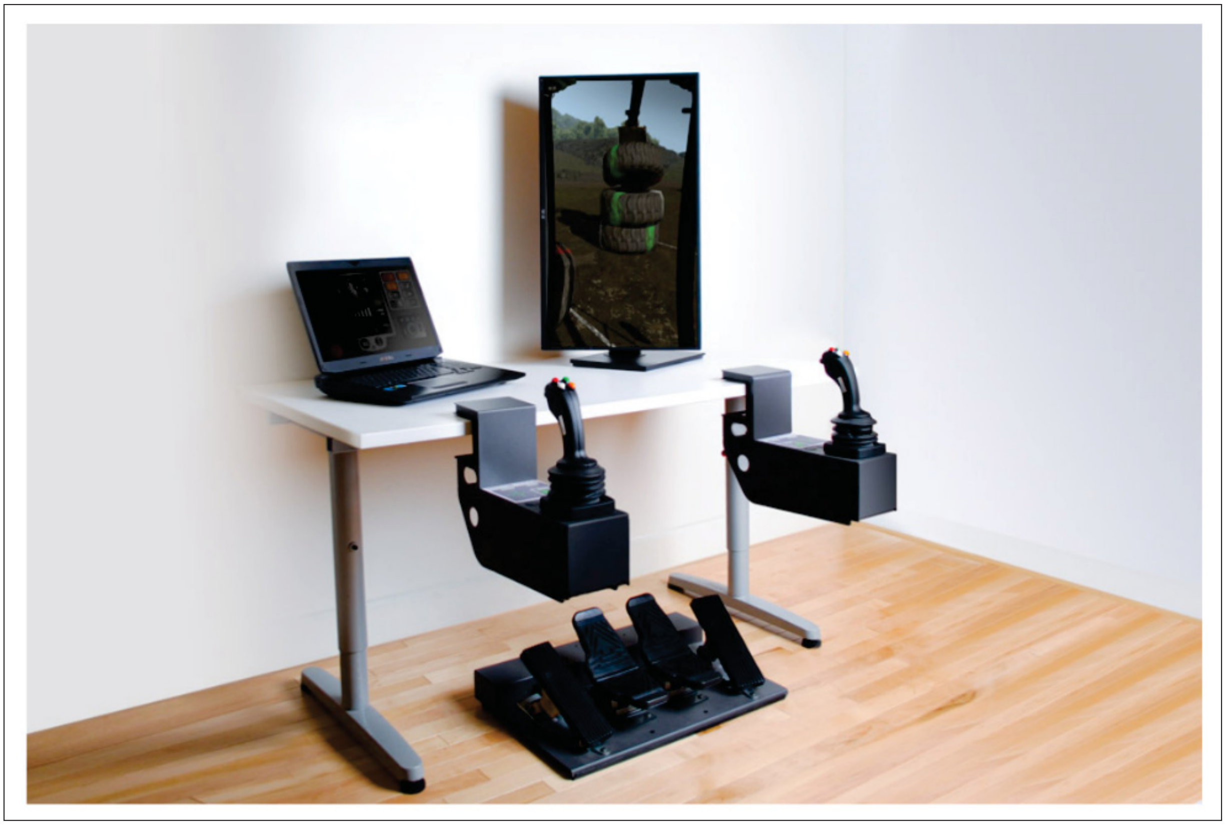


Figure 2.5 The Vortex Edge Simulator was used for collecting the dataset and training the reinforcement learning policy. To provide a close-to-realistic scenario, a mobile setup was installed, featuring a single vertical monitor, touch-sensitive operator console, and physics controls. While the operator uses external joysticks to provide signals to the actuators, in the RL setup, a Proximal Policy Optimization (PPO) policy maps the observation space to the required torque for each actuator

The dataset was gathered from 40 students, and their scores and durations were plotted as shown in Fig. 2.6. At the beginning of the session, each student has a score of 0, and this score is reduced by predefined heuristics following an infraction, as described in Table 2.2. These heuristics only consider external infractions and do not take into account the internal dynamics of the excavator. To capture these dynamics, we selected three dynamic features - Average Engine Power, Average Engine Torque, and Average Fuel Consumption Rate - based on their correlation, as illustrated in Fig. 2.7. For learning the safety distribution, we used all the infraction features listed in Table 2.2.

Table 2.1 Dynamic features of the excavator

Dynamic feature	**Description**
Bucket angle (°)	Angle of the excavator bucket with respect to the ground.
Bucket height (m)	Height of the excavator bucket above the ground.
Bucket self-contact	Number of times the bucket touches the tracks or body.
Engine average power (%)	Average engine power over the elapsed time relative to the maximum.
Current engine power (kW)	Instantaneous engine power.
Engine torque (N m)	Instantaneous engine torque.
Engine torque average (%)	Average engine torque over the elapsed time relative to the maximum.
Fuel-consumption rate average ($\mathrm{L\,h^{-1}}$)	Average fuel consumed per hour over the elapsed time.
Fuel consumption (L)	Total fuel consumed.
Idle count	Number of times the engine idled.
Idle time (s)	Cumulative idle duration.
Engine RPM (%)	Instantaneous engine speed relative to the maximum.
Engine RPM average (%)	Average engine speed over the elapsed time relative to the maximum.
Track ground pressure—front left (kPa)	Ground pressure on the front-left track.
Track ground pressure—front right (kPa)	Ground pressure on the front-right track.
Track ground pressure—rear left (kPa)	Ground pressure on the rear-left track.
Track ground pressure—rear right (kPa)	Ground pressure on the rear-right track.

In the collected dataset, the session lasted from a minimum of 48 time steps to a maximum of 2000 time steps, with an average duration of 600 time steps.

Table 2.2 Description of the safety features of the excavator and their associated penalties

Safety Feature	**Description**	**Penalty**
Tennis balls knocked	The number of tennis balls knocked over by the excavator.	-2
Barrels knocked	The number of barrels knocked over by the excavator.	-5
Barrel touched	The number of barrels touched by the excavator.	-2
Equipment Collision	The number of times the excavator collided with any object.	-5
Poles Fell	The number of poles knocked over by the excavator.	-5
Poles touched	The number of poles touched by the excavator.	-2

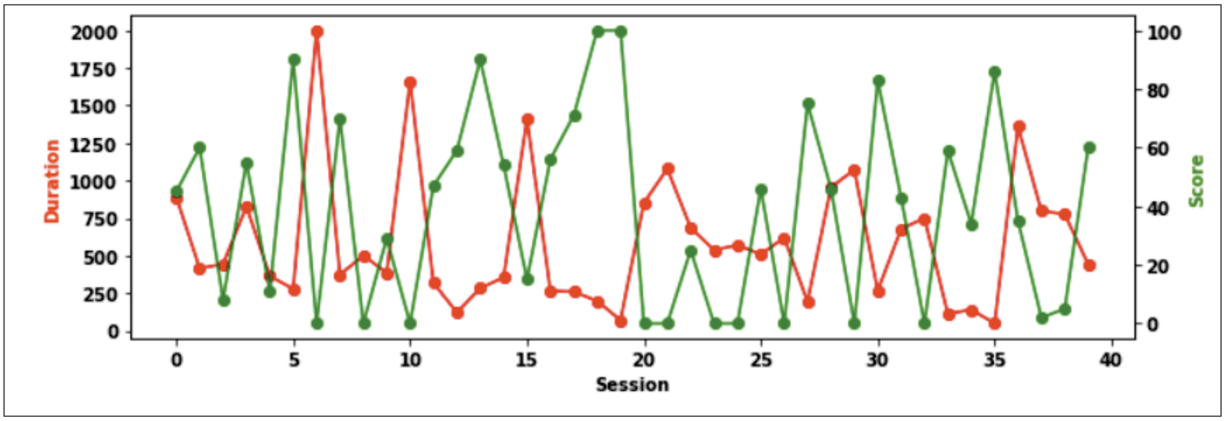


Figure 2.6 Score and duration across training sessions. Joint distribution of session duration (left axis, red) and skill score (right axis, green) recorded over $N = 40$ sessions

2.2.4 Method

Our proposed automatic reward prediction framework is presented in Fig. 2.8. For tasks such as navigation and manipulation (Nasiriany, Pong, Lin & Levine, 2019), conventional reinforcement learning algorithms typically rely on goal-based rewards to optimize the policy to maximize the cumulative returns over a given time. However, such a strategy overlooks the machine dynamics and the external environmental constraints, which limit its applicability to real-life scenarios. To address this issue, we propose additional reward signals that complement the

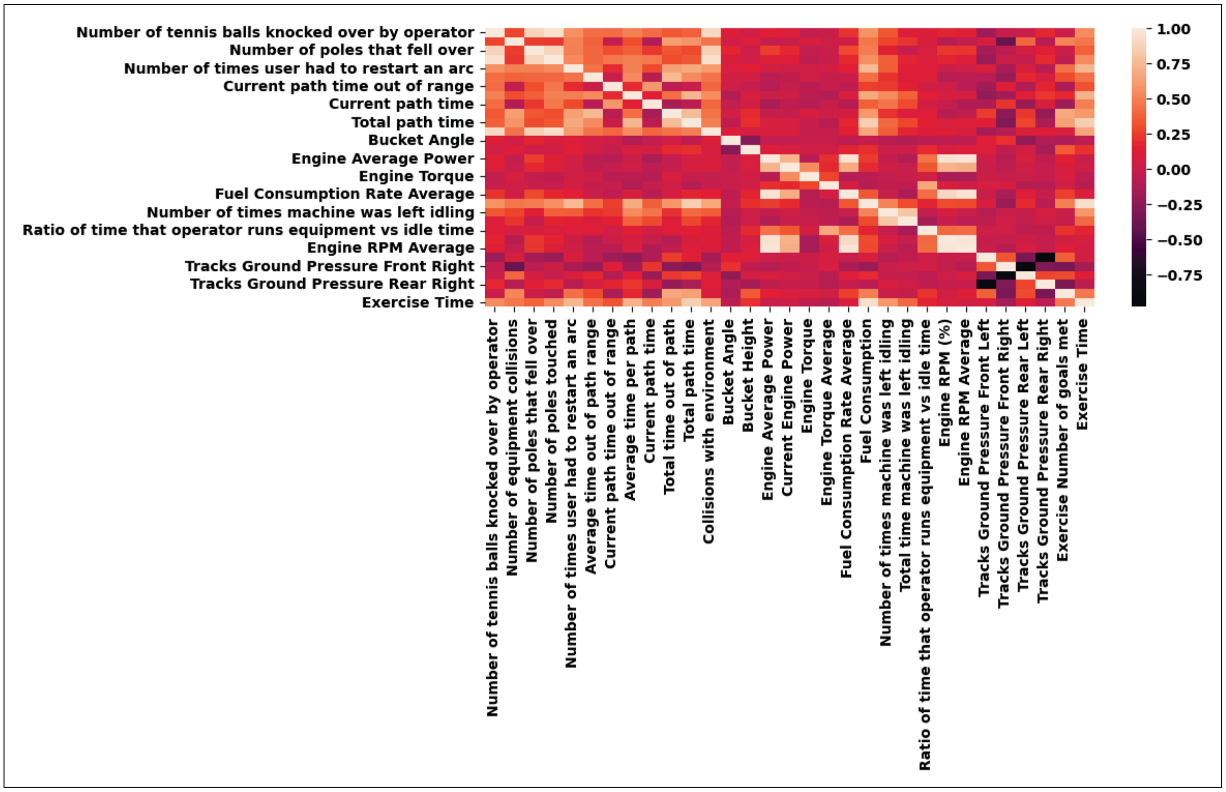


Figure 2.7 Pearson correlation heat-map of operator-performance and machine features showing pairwise coefficients ($-1 \le \rho \le +1$). Warmer shades denote stronger positive relationships (e.g., high engine torque aligns with high average power), whereas cooler shades indicate negative coupling (e.g., increased idling time versus overall exercise efficiency). White cells lie on the main diagonal ($\rho = 1$)

original goal-based reward, learned from expert demonstrations. Specifically, we focus on understanding the representation of specific features that are critical for excavator manoeuvring in a constrained environment. Our ultimate objective is to leverage these rewards to train an RL policy that can successfully solve this challenging task, validating our proposed automatic score predictive model for real-life training of users.

Learning Distributions of Experts' Demonstrations: Our framework aims to learn two distinct distributions, both derived from multiple expert demonstrations. The first distribution models the temporal representation of various excavator features, directly influenced by the user's actions, making it an efficient way to capture the dynamic distribution. The second distribution captures the non-temporal states of the environment, represented by infraction values at each time step,

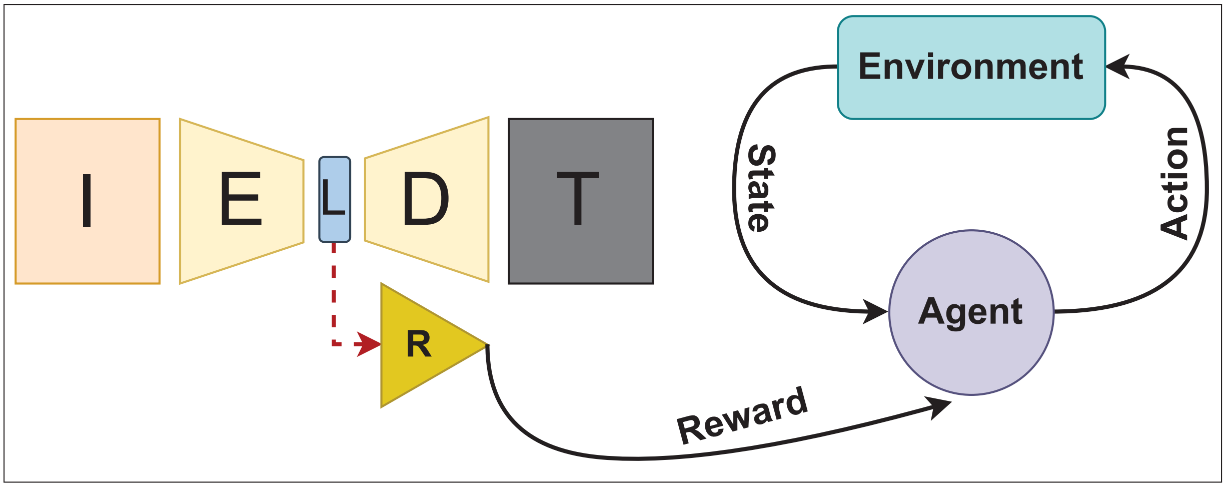


Figure 2.8 Automatic Reward Prediction for Excavator Automation. Our approach for automating reward prediction involves two stages: **a)** Representation learning stage, where an encoder (**E**) and decoder (**D**) learn a distribution (**L**) of the input (**I**), which depends on the task objective (**T**), and **b)** Reinforcement learning stage, where the learned reward function **R** is used for learning the task of excavator automation in a simulated environment

an ideal source for modelling the safety distribution. Actuator trajectories and velocity are deliberately excluded from the learned distributions to encourage the RL policy to discover its novel strategies based on the task's goal. Hence, Generative Adversarial Imitation Learning (GAIL) Ho & Ermon (2016) is not optimal for policy learning in our case, as it relies on low-level details of expert demonstrations, hindering the policy's exploration of novel strategies beyond the demonstrations.

Table 2.3 Inputs for Modelling the Dynamic and Safety Distribution

Dynamic	**Safety**	
Average Engine Torque	Poles Touched	Environment Collision
Average Engine Power	Poles Fell	Balls knocked
Average Fuel Consumption	Equipment Collision	

To learn the dynamic distribution, we first sample a window of the 32 most recent values of each feature at a given time step **t**. This sequence of temporal data is then encoded into a single 8-dimensional feature vector using a Long Short-Term Memory Network (LSTM). Next, two independent Multi-Layer Perceptrons (MLPs) are used to predict the mean (μ) and variance (σ) for the distribution. To encode the dynamics of the temporal features in the expected distribution $\mathbf{q}_D(\mathbf{z})$, we require the decoder to predict the feature values for the next time steps given the sampled vector from the distribution and the feature values for the last time step as input. The decoder consists of an LSTM, which takes as input the original feature values of the previous time step and predicts the future values. We initialize the hidden and cell state of the LSTM with the sampled vector from the expected distribution to integrate user dynamics.

The safety distribution, $\mathbf{q}_I(\mathbf{z})$, is learned in a similar manner to the dynamic distribution. A 5-dimensional feature vector representing infraction values (as specified in the safety column of Table 2.3) is first encoded into a 2-dimensional distribution using a two-layer fully connected network with eight neurons per layer, as shown in Fig. 2.9. The network is trained to predict the sum of total infractions given the input using the $\mathcal{KL}$ divergence loss without any reconstruction loss. The resulting distribution is effective in differentiating between infractions with widely different values, as shown in Fig. 2.10.

To model the reward function, we use these learned distributions of expert demonstrations. Specifically, for a specific action taken by the new user at a given time step t, we feed the corresponding feature value (as specified in Table 2.3) to the learned model with frozen weights, which then maps these values to their respective distributions. We evaluate the reward for the action taken by computing the $\mathcal{KL}$ divergence between the predicted distribution and the expert's distribution. This $\mathcal{KL}$-based reward is then summed with the original task-based reward to train an RL policy.

Policy Learning. The ultimate goal is to use the learned score predictive model for training an RL policy to manoeuvre an excavator in a complex environment. This task is challenging to model using classical controllers due to the need for precise coordination of multiple actuators. Furthermore, adapting these controllers across different excavators is difficult considering the wide variations (Jud *et al.*, 2019; Koivo, Thoma, Kocaoglan & Andrade-Cetto, 1996). Reinforcement learning presents an ideal solution, as it forces each actuator to coordinate

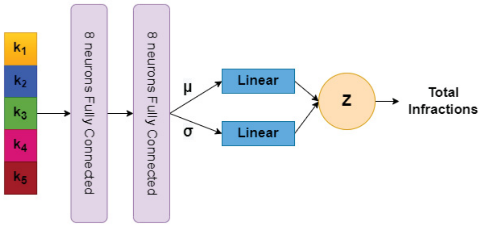


Figure 2.9 Network architecture of the lightweight VAE used to model infractions. The encoder maps input features to a latent variable z, with linear heads outputting mean (μ) and log-variance (σ).

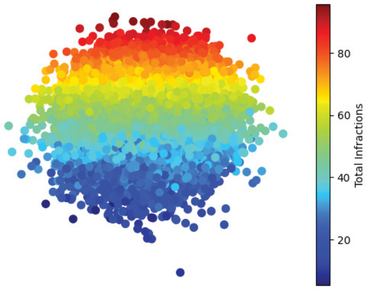


Figure 2.10 Visualization of the learned safety distribution. Latent embeddings are plotted in 2D and color-coded by the ground-truth number of infractions (blue = few, red = many). A 2D latent space is chosen here purely for visualization.

Figure 2.11 Together, the VAE model and its latent distribution capture the likelihood of infractions, enabling their use as an automatic reward function for safe policy learning.

its actions to discover optimal policies (Liu, Nageotte, Zanne, de Mathelin & Dresp, 2021a) that are robust to variations, providing a scalable solution for real-life adoption (Almási, Moni & Gyires-Tóth, 2020).

We utilize a model-free Proximal Policy Optimization (PPO) (Wang, He & Tan, 2020d) as an on-policy algorithm, which is known for its stability and robustness to different hyperparameters. Our policy is modelled using a 2-layered MLP, with the input space consisting of a 19-dimensional vector as shown in Table 2.4. The output is a continuous 4-dimensional vector within a range of [-1,1], which represents the set-points for each actuator, such that values less than 0 represent the movement of the actuator in the negative direction, while values greater than zero move the actuator in the positive direction. For the swing actuator, it is equivalent to rotation in the left or right direction.

To optimize the excavator's policy for completing the specified goal, we combine three complementary reward functions: the **task reward** (R_G), the **dynamics reward** (R_D), and the **safety reward** (R_S). The task reward R_G is based on the distance between the bucket's spatial coordinates $\mathbf{B}_L$ and the specified goal $\mathbf{G}$, encouraging the excavator to move toward the target. The dynamics reward R_D is derived from a pre-trained model of expert demonstrations and encourages the policy to generate expert-like, smooth dynamics. The safety reward R_S is obtained from a pre-trained safety model that predicts infractions such as collisions or unsafe motions; this ensures that unsafe behaviors are penalized.

Together, these three signals form the overall reward:

$$R = R_G + R_D + R_S \tag{2.6}$$

The individual reward components are defined as:

$$\begin{aligned} R_G &= 1 - \|\mathbf{G} - \mathbf{B}_L\|, & (2.7) \\ R_D &= 1 - \mathcal{KL}\big(p_{ND}(\mathbf{z}_D) \parallel q_D(\mathbf{z})\big), & (2.8) \\ R_S &= 1 - \mathcal{KL}\big(p_{NI}(\mathbf{z}_I) \parallel q_I(\mathbf{z})\big), & (2.9) \end{aligned}$$

where p_{ND} and p_{NI} denote the expert (normal) distributions for dynamics and infractions respectively, and q_D, q_I are the learned posterior distributions. The KL-divergence terms ensure that the learned representations remain close to expert-like distributions.

In summary, R_G drives the agent toward the task goal, R_D enforces efficient expert-like dynamics, and R_S penalizes unsafe behaviors. By combining these signals, the excavator policy learns not only to accomplish the task, but to do so in a way that is both efficient and safe.

Table 2.4 The observation space of the reinforcement learning policy

Inputs	**Actuators**			
	Swing	**Boom**	**Stick**	**Bucket**
Linear Position	✓	✓	✓	✓
Angular Velocity	✓	✓	✓	✓
Linear Velocity	–	✓	✓	✓

2.2.5 Experiments

Through our experiments, we aim to verify:

1. The effectiveness of using expert distribution for modeling the reward function.
2. The impact of learning the RL policy for the task of excavator maneuvering using these additional rewards formulated as dynamics and infraction distribution in comparison to the task-based reward.

Simulator Setup and Dataset Collection. For collecting both the user data, as well as for training the RL policy, the Vortex simulator[1] is used. This simulator provides a range of customizable construction environments for different excavator automation tasks (described in detail in Section 2.2.2). In our case, we use the scenario shown in Fig. 2.2, where the task is to manoeuvre the crane to reach a pre-specified goal location. We fixed the goal and start location as shown in Table 2.5 to provide a complex, challenging task in the presence of all the infractions for the crane to complete for proper analysis.

For our dataset, we collect trajectories for a total of 40 users, where each is ranked based on pre-defined heuristics. This heuristic is formulated such that a user is automatically penalized in case of any infractions, as elaborated in Section 2.2.3 and Table 2.2. Each session's final score ranges from 0 (maximum) to -100 (minimum). We consider a user to be an expert if the total score is greater than -25, while others are considered a novice. In the current dataset, seven users have a score greater than -25 and are further used to learn the distributions.

[1] https://www.cm-labs.com/vortex-studio/

Implementation. The model for learning the distributions is trained using the trajectories of 5 experts, and the remaining trajectories are used for evaluating the model. Both the LSTM-based encoder-decoder setup and the 2-layered fully connected network are trained for a total of 1000 epochs with a batch size of 8 using an Adam Optimizer with a learning rate of 1e-4. Similarly, for policy training using an actor-critic architecture, the networks are trained for 500 episodes with a learning rate of 3e-4 for the actor and 1e-3 for the critic. We used the ELU activation function Clevert, Unterthiner & Hochreiter (2016) for all layers except the outputs, as it showed better performance and faster learning. Each RL policy was run for a maximum of 280 steps per episode, terminating when the excavator reached the goal. The networks were trained on a system with an NVIDIA GTX 1050 Ti and an Intel Core i5-8300 CPU @ 2.30GHz x 8, with learning distributions taking approximately 2-3 hours and training the policy taking 4-5 hours.

Table 2.5 Description of the task to be learned

Task Description	Start Position	Goal Position	Distance (meters)
Excavator Movement	(-9.8, -144.1, 2.4)	(-2, -150, 0.5)	10.2

Results. We evaluated the impact of our proposed reward model on the final policy. We compare different dynamic features and infraction values, as presented in Table 2.6. We trained each policy with and without the various reward functions, in addition to the original task reward. The excavator's performance was poor without any constraints, as demonstrated by the policy using only task rewards. However, the addition of constraints in the form of our proposed reward function significantly improved performance, resulting in policies that closely followed the expert's dynamics and infraction values.

Interestingly, policies trained with both dynamic and safety rewards performed similarly and much better than policies trained solely with task rewards. The safety reward helped optimize the policy by reducing the infractions to zero, resulting in a safe policy. However, when used as the sole reward, it did not take the dynamics into account. We also observed a similar trend when the dynamic reward was used alone.

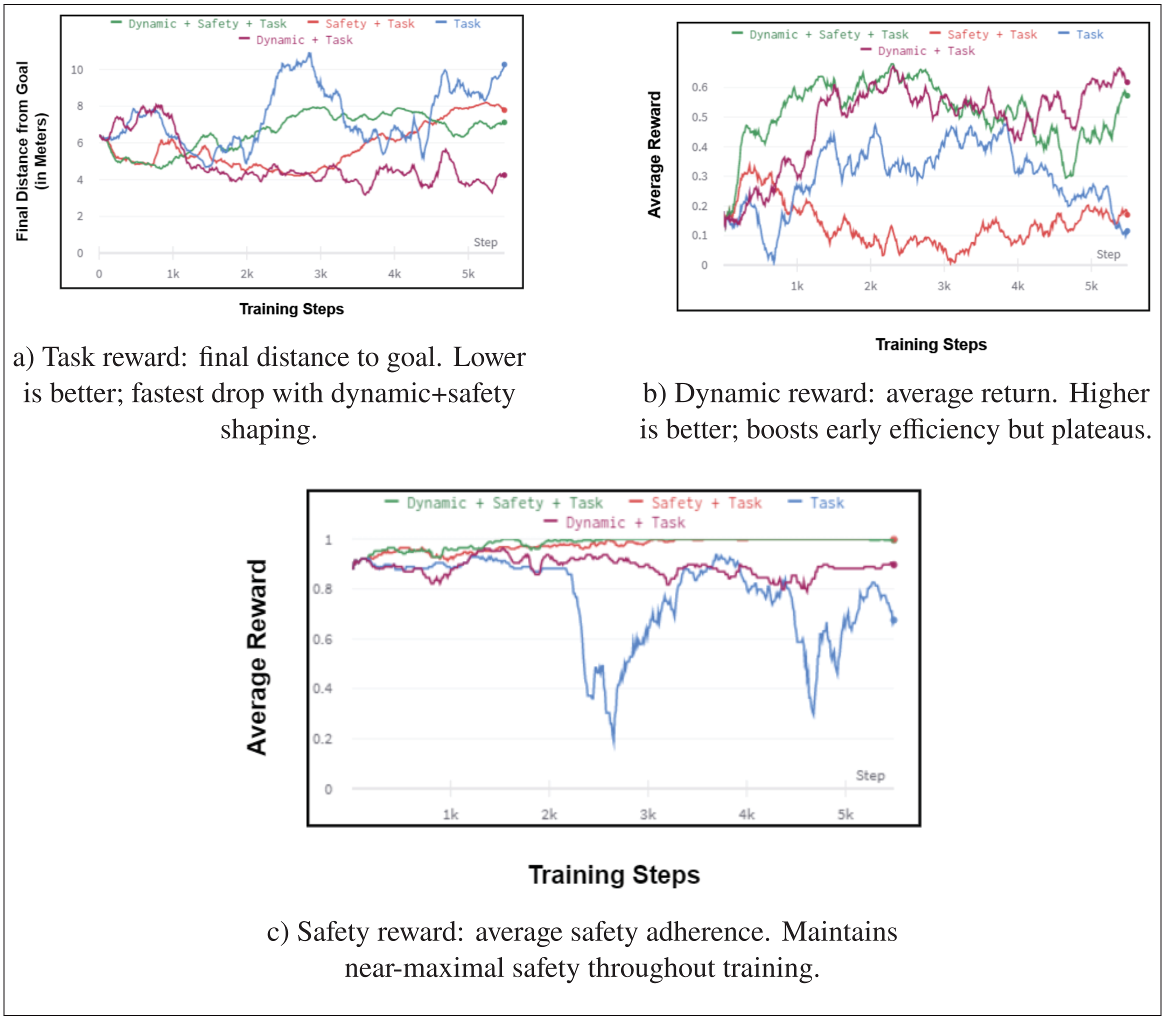
Fig. 2.12 compares the policy optimization for different rewards. For the original task reward (Fig. 2.12a) of reaching the final specified position (lower is better), the best performance is achieved with the addition of a dynamic reward, which outperforms the task reward alone. This is because the dynamic reward constrains the excavator to the expert's dynamic features, preventing unwanted actions and leading to faster convergence. When the policy is trained without the dynamic reward, there is a significant drop in the policy's dynamic reward (Fig. 2.12b). Similarly, policies trained without any safety reward also exhibit poor performance (Fig. 2.12c). Comparing the performance of each constraint, policies trained solely with the task reward exhibit the worst performance, while those integrated with additional constraints closely follow the expert's performance.

Table 2.6 The dynamic and infraction values of the final policy can vary significantly depending on the choice of the reward function

	Expert	Task	**Dynamic**	Safety	**Dynamic+Safety**
Avg. Torque (in %)	**24-34**	51.73	**45.61**	51.28	**46.44**
Avg. Power (in %)	**10-20**	26.77	**22.63**	26.84	**23.72**
Avg. Fuel Consumption (in %)	**3-5**	5.77	**5.01**	5.71	**5.12**
Total Infractions	**0**	35	18	**0**	**1**

2.2.6 Conclusion

Summary. The main contribution of our work is a novel approach for modelling a feedback mechanism to teach new students the task of excavator manoeuvring using demonstrations from expert operators. Our approach involves learning two independent distributions, the dynamic and safety distributions, based on the temporal states of the excavator and infraction values of the expert's trajectory. These learned distributions provide feedback to trainees through the $\mathcal{KL}$ divergence with the current action. This automatic score predictive model is validated by using it as a reward mechanism for an RL policy to learn the task of manoeuvring an excavator in a simulated environment. Our results show that the policy learned using additional supervision successfully completes the task goal while also optimizing for the additional constraints. The



a) Task reward: final distance to goal. Lower is better; fastest drop with dynamic+safety shaping.

b) Dynamic reward: average return. Higher is better; boosts early efficiency but plateaus.

c) Safety reward: average safety adherence. Maintains near-maximal safety throughout training.

Figure 2.12 Policy performance over 5000 training steps for four reward formulations: *Task* only (blue), *Dynamic+Task* (magenta), *Safety+Task* (red), and *Dynamic+Safety+Task* (green). (a) Task reward reduces goal error fastest when combined with dynamic and safety. (b) Dynamic reward accelerates early learning but saturates. (c) Safety reward sustains constraint adherence, while Task-only degrades after 2500 steps.

final policy closely emulates an expert operator's dynamics and infraction values, effectively manoeuvring the excavator.

Currently, our approach requires learning separate distributions for each new constraint, which can be computationally inefficient. We believe that a single complex model may provide substantial improvements. Additionally, we aim to expand our work to tackle various real-world challenges in different construction tasks, including loading-unloading, soil excavation, and levelling.

CHAPTER 3

LEARNING TO PLAY ATARI IN A WORLD OF TOKENS

Pranav Agarwal [1] , Sheldon Andrews [1] , Samira Ebrahimi Kahou[1]

[1] Departement of Software and IT Engineering,
École de Technologie Supérieure,
1100 Notre-Dame Ouest, Montréal, Québec, Canada H3C 1K3

Article accepted at the « International Conference on Machine Learning » in July 2024

Résumé

Les agents d'apprentissage par renforcement basés sur des modèles utilisant des transformers ont montré une efficacité d'échantillonnage améliorée grâce à leur capacité à modéliser un contexte étendu, résultant en des modèles du monde plus précis. Cependant, pour des tâches complexes de raisonnement et de planification, ces méthodes reposent principalement sur des représentations continues. Ceci complique la modélisation des propriétés discrètes du monde réel telles que les classes d'objets disjointes entre lesquelles l'interpolation n'est pas plausible. Dans ce travail, nous introduisons les représentations abstraites discrètes pour l'apprentissage basé sur les transformers (DART), une méthode efficace en échantillons utilisant des représentations discrètes pour modéliser à la fois le monde et apprendre des comportements. Nous intégrons un transformateur-décodeur pour la modélisation autorégressive du monde et un transformateur-encodeur pour l'apprentissage du comportement en prenant en compte les jetons pertinents pour la tâche dans la représentation discrète du modèle du monde. Pour gérer l'observabilité partielle, nous agrégeons l'information des pas de temps passés sous forme de jetons de mémoire. DART surpasse les méthodes précédentes de l'état de l'art qui n'utilisent pas de recherche anticipée sur le benchmark d'efficacité d'échantillonnage Atari 100k avec un score médian normalisé par rapport à l'humain de 0,790 et bat les humains dans 9 jeux sur 26. Nous publions notre code à https://pranaval.github.io/DART/.

Abstract

Model-based reinforcement learning agents utilizing transformers have shown improved sample efficiency due to their ability to model extended context, resulting in more accurate world models. However, for complex reasoning and planning tasks, these methods primarily rely on continuous representations. This complicates the modelling of discrete properties of the real world, such as disjoint object classes between which interpolation is not plausible. In this work, we introduce discrete abstract representations for transformer-based learning (DART), a sample-efficient method utilizing discrete representations for modelling both the world and learning behaviour. We incorporate a transformer-decoder for auto-regressive world modelling and a transformer-encoder for learning behaviour by attending to task-relevant tokens in the discrete representation of the world model. For handling partial observability, we aggregate information from past time steps as memory tokens. DART outperforms previous state-of-the-art methods that do not use look-ahead search on the Atari 100k sample efficiency benchmark with a median human-normalized score of 0.790 and beats humans in 9 out of 26 games. We release our code at https://pranaval.github.io/DART/.

3.1 Introduction

A reinforcement learning (RL) algorithm usually takes millions of trajectories to master a task, and the training can take days or even weeks, especially when using complex simulators. This is where model-based reinforcement learning (MBRL) comes in handy (Sutton, 1991). With MBRL, the agent learns the *dynamics* of the environment, understanding how the environment state changes when different actions are taken (Moerland, Broekens, Plaat, Jonker *et al.*, 2023). This method is more efficient because the agent can train in its *imagination* without requiring direct interaction with an external simulator or the real environment (Hafner *et al.*, 2020a). Additionally, the learned model allows the agent to make safe and accurate decisions by utilizing different look-ahead search algorithms for planning its actions (Silver *et al.*, 2016).

Most MBRL methods commonly follow a structured three-step approach: 1) Representation Learning $\phi : S \rightarrow \mathbb{R}^n$, the agents capture a simplified representation $\mathbb{R}^n$ of the high dimensional environment state S; 2) Dynamics and Reward Learning $f : S \times A \rightarrow S', \psi : S \times A \times S' \rightarrow R$, where the agent learns the dynamics of the environment, predicting the next state s' given the current state s and action a, as well as the reward associated with transitioning from s to s'; and 3) Policy Learning $\pi : S \rightarrow \mathcal{P}(A)$, the agent determines the optimal actions needed to achieve its goals. Dreamer is a family of MBRL agents that follow a similar structured three-step approach.

DreamerV1 (Hafner *et al.*, 2020b) employed a recurrent state space model (RSSM) (Doerr *et al.*, 2018) to learn the world model. DreamerV2 (Hafner, Lillicrap, Norouzi & Ba, 2021), an improved version of DreamerV1, offers better sample efficiency and scalability by incorporating a discrete latent space for modelling the dynamics. Building on the advancements of DreamerV2, DreamerV3 (Hafner, Paukonis, Ba & Lillicrap, 2025) takes a similar approach with additions involving the use of symlog predictions and various regularization techniques aimed at stabilizing learning across diverse environments. Notably, DreamerV3 surpasses the performance of past models across a wide range of tasks, while using fixed hyperparameters.

Although Dreamer variants are among the most popular MBRL approaches, they suffer from sample-inefficiency (Yin, Ye, Chen & Gao, 2022; Svidchenko & Shpilman, 2021). The training of Dreamer models can require an impractical amount of game-play time, ranging from months to thousands of years, depending on the complexity of the game. This inefficiency can be primarily attributed to inaccuracies in the learned world model, which tend to propagate errors into the policy learning process, resulting in compounding error problems (Talvitie, 2017). This challenge is primarily associated with the use of convolutional neural networks (CNNs) and recurrent neural networks (RNNs) (Deng, Park & Ahn, 2023) that, while effective in many domains, face limitations in capturing complex and long-range dependencies, which are common in RL scenarios (Ni, Ma, Eysenbach & Bacon, 2024).

This motivates the need to use transformers (Vaswani *et al.*, 2017; Lin, Wang, Liu & Qiu, 2022b), which have proven highly effective in capturing long-range dependencies in various

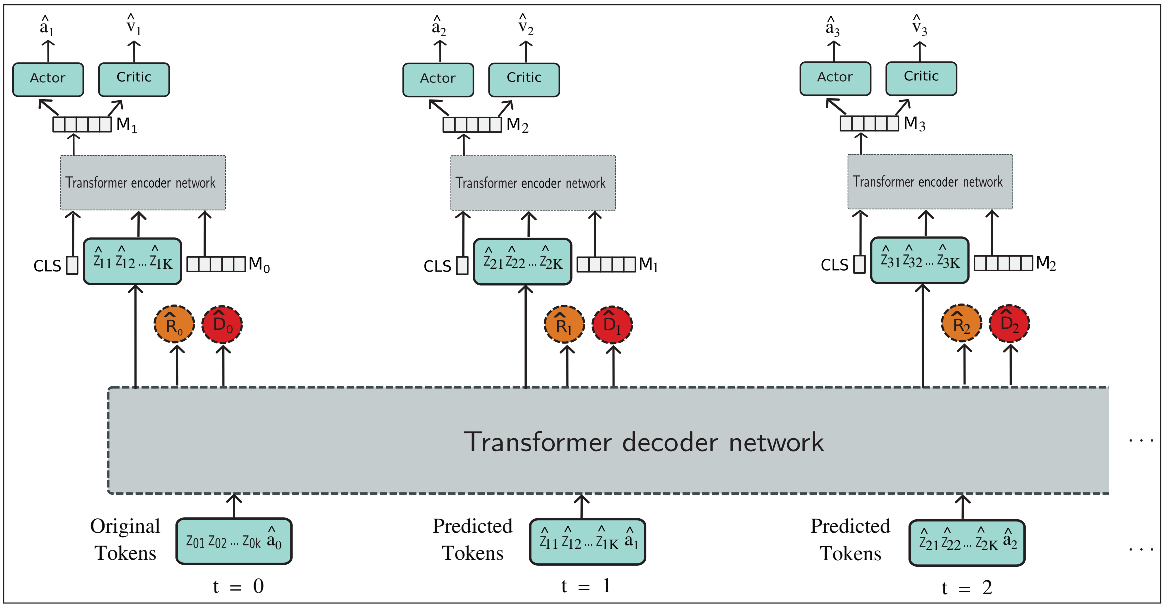

$\hat{a}_1$
$\hat{v}_1$
Actor
Critic
M_1
Transformer encoder network
CLS
$\hat{z}_{11}\hat{z}_{12}...\hat{z}_{1K}$
M_0
$\hat{R}_0$
$\hat{D}_0$
$\hat{a}_2$
$\hat{v}_2$
Actor
Critic
M_2
Transformer encoder network
CLS
$\hat{z}_{21}\hat{z}_{22}...\hat{z}_{2K}$
M_1
$\hat{R}_1$
$\hat{D}_1$
$\hat{a}_3$
$\hat{v}_3$
Actor
Critic
M_3
Transformer encoder network
CLS
$\hat{z}_{31}\hat{z}_{32}...\hat{z}_{3K}$
M_2
$\hat{R}_2$
$\hat{D}_2$
Transformer decoder network
...
Original Tokens
$z_{01}z_{02}...z_{0K}\hat{a}_0$
t = 0
Predicted Tokens
$\hat{z}_{11}\hat{z}_{12}...\hat{z}_{1K}\hat{a}_1$
t = 1
Predicted Tokens
$\hat{z}_{21}\hat{z}_{22}...\hat{z}_{2K}\hat{a}_2$
t = 2
...

Figure 3.1 Discrete abstract representation for transformer-based learning (DART): In this approach, the original observation x_t is encoded into discrete tokens z_t using VQ-VAE. These tokenized observations and predicted actions serve as inputs for the world model. A Transformer decoder network is used for modelling the world. The predicted tokens, along with a `CLS` and a `MEM` token, are used as input by the policy. This policy is modelled using a transformer-encoder network. The `CLS` token aggregates information from the observation tokens and the `MEM` token to learn a common representation, which is then used for action and value predictions. This common representation also plays a role in modelling memory, acting as the `MEM` token at the subsequent time step. **Abbreviations:** R_t = reward, D_t = done, $\hat{v}_t$ = value, $\hat{a}_t$ = action

natural language processing (NLP) tasks (Wolf *et al.*, 2020) and addressing complex visual reasoning challenges in computer vision (CV) tasks (Khan *et al.*, 2022b). Considering these advantages, recent works have adapted transformers for modelling the dynamics in MBRL. Transdreamer (Chen, Yoon, Wu & Ahn, 2021a) first used a transformer-based world model by replacing Dreamer's RNN-based stochastic world model with a transformer-based state space model. It outperformed DreamerV2 in Hidden Order Discovery Tasks, which require long-term dependency and complex reasoning. To stabilize the training, it utilizes gated transformer-XL (GTrXL) (Parisotto *et al.*, 2020) architecture.

Masked world model (MWM) (Seo *et al.*, 2023b) utilizes a Convolutional-autoencoder and vision transformer (ViT) (Dosovitskiy *et al.*, 2021) for learning a representation that models dynamics following the RSSM objective. Their decoupling approach outperforms DreamerV2 on different robotic manipulation tasks from Meta-world (Yu *et al.*, 2020) and RLBench (James, Ma, Arrojo & Davison, 2020). Similarly, transformer-based world model (TWM) (Robine *et al.*, 2023a) uses transformer-XL (TrXL) (Dai *et al.*, 2019) for modeling the world and uses the predicted latent states for policy learning. Their work demonstrates sample-efficient performance on the Atari 100k benchmark.

Contrary to these approaches, imagination with auto-regression over an inner speech (IRIS) (Micheli *et al.*, 2023) models dynamics learning as a sequence modelling problem, utilizing discrete image tokens for modelling the world. It then uses reconstructed images using the predicted tokens for understanding the policy using CNNs and long short-term memorys (LSTMs), achieving improved sample efficiency on the Atari 100k compared to past models. However, it still faces difficulties in policy learning due to the use of reconstructed images as input, resulting in reduced performance.

In this work, we introduce discrete abstract representation for transformer-based learning (DART), a novel approach that leverages transformers for learning both the world model and policy. Unlike the previous work by Yoon, Wu, Bae & Ahn (2023), which solely utilized a transformer for extracting object-centric representation, our approach employs a transformer encoder to learn behaviour through discrete representation (Mao *et al.*, 2022), as predicted by the transformer-decoder that models the world. This choice allows the model to focus on fine-grained details, facilitating precise decision-making. Specifically, we utilize a transformer-decoder architecture, akin to the generative pre-trained transformer (GPT) framework (Radford *et al.*, 2019), to model the world, while adopting a transformer encoder, similar to the ViT architecture (Dosovitskiy *et al.*, 2021), to learn the policy (as illustrated in Figure 3.1).

Additionally, challenges related to partial observability necessitate memory modelling. Previous work Didolkar *et al.* (2022) modelled memory in transformers using a computationally intensive two-stream network. Inspired by (Bulatov, Kuratov & Burtsev, 2022), we model memory as a distinct token, aggregating task-relevant information over time using a self-attention mechanism.

The main contribution of our work includes a novel approach that utilizes transformers for both world and policy modelling. Specifically, we use a transformer-decoder (GPT) for world modelling and a transformer-encoder (ViT) for policy learning. This represents an improvement compared to IRIS, which relies on CNNs and LSTMs for policy learning, potentially limiting its performance. We use discrete representations for policy and world modelling. These discrete representations capture abstract features, enabling our transformer-based model to focus on task-specific fine-grained details. Attending to these details improves decision-making, as demonstrated by our results. To address the problem of partial observability, we introduce a novel mechanism for modelling the memory that aggregates task-relevant information from the previous time step to the next using a self-attention mechanism. Our model showcases enhanced interpretability and sample efficiency. It achieves state-of-the-art results (no-look-ahead search methods) on the Atari 100k benchmark with a median score of 0.790 and superhuman performance in 9 out of 26 games.

3.2 Research Problem and Motivations

Sample Efficiency in RL. Enhancing sample efficiency (i.e., the amount of data required to reach a specific performance level) constitutes a fundamental challenge in the field of RL. This efficiency directly impacts the time and resources needed for training an RL agent. Numerous approaches aimed at accelerating the learning process of RL agents have been proposed (Buckman, Hafner, Tucker, Brevdo & Lee, 2018; Mai, Mani & Paull, 2022; Yu, 2018). Model-based RL is one such approach that helps improve the sample efficiency. It reduces the number of interactions an agent needs to have with the environment to learn the policy (Ayoub, Jia, Szepesvari, Wang & Yang, 2020; Polydoros & Nalpantidis, 2017; Atkeson & Santamaria, 1997). This is done by allowing the policy to understand the task in the imagined world (Wang *et al.*, 2021b; Mu *et al.*, 2021; Okada & Taniguchi, 2021; Zhu, Zhang, Lee & Zhang, 2020). This motivates the need to have an accurate world model while providing the agent with concise and meaningful task-relevant information for faster learning. Considering this challenge Kurutach, Clavera, Duan, Tamar & Abbeel (2018) learns an ensemble of models to reduce the impact of model bias and variance. Uncertainty estimation is another approach as shown in Plaat, Kosters & Preuss (2023) to improve model accuracy. It involves estimating the uncertainty in the model's prediction so that the agent focuses its exploration on those areas. The other most common approach for an accurate world model is using a complex or higher-capacity model architecture that is better suited to the task at hand (Ji *et al.*, 2022). For example, using a transformer-based world model, as in TransDreamer (Chen *et al.*, 2021a), TWM (Robine *et al.*, 2023a), and IRIS (Micheli *et al.*, 2023).

Learning a low-dimensional representation of the environment can also help improve the sample efficiency of RL agents. By reducing the dimensionality of the state, the agent can learn an accurate policy with fewer interactions with the environment (Schwarzer *et al.*, 2020; Du, Kakade, Wang & Yang, 2020). Variational Autoencoders (VAEs) (Kingma & Welling, 2019b) are commonly used for learning low-dimensional representations in MBRL (Andersen, Goodwin & Granmo, 2018). The VAEs capture a compact and informative representation of the input data. This allows the agent to learn the policy faster (Ke *et al.*, 2018; Corneil, Gerstner & Brea, 2018). However, VAEs learn a continuous representation of the input data by forcing the latent variable to be normally distributed. This continuous representation poses a challenge for RL agents, where agents need to focus on precise details for decision-making (Dunion, McInroe, Luck, Hanna & Albrecht, 2023). Lee, Eysenbach, Salakhutdinov, Gu & Finn (2020a) shows that disentangling representations helps in modeling interpretable policy and improves the learning speed of RL agents on various manipulation tasks. Recent works (Robine, Uelwer & Harmeling, 2023b; Kujanpää, Pajarinen & Ilin, 2023) have used vector quantized-variational autoencoder (VQ-VAE) for learning independent latent representations of different low-level features present in the original observation. Their clustering properties have enabled robust, interpretable, and generalizable policies across a wide range of tasks.

3.3 Method

Our model, DART, is designed for mastering Atari games, within the framework of a partially observable Markov decision process (POMDP) (Kaelbling, Littman & Cassandra, 1998) which is defined as a tuple $(\mathcal{O}, \mathcal{A}, p, r, \gamma, d)$. Here, $\mathcal{O}$ is the observation space with image observations $x_t \subseteq \mathbb{R}^{h\times w\times 3}$, $\mathcal{A}$ represents the action space, and a_t is a discrete action taken at time step t from the action space $\mathcal{A}$, $p\left(x_t \mid x_{<t}, a_{<t}\right)$ is the transition dynamics, r is the reward function $r_t = r\left(x_{\leq t}, a_{<t}\right)$, $\gamma \in [0, 1)$ is the discount factor and $d \in \{0, 1\}$ indicates episode termination. The goal is to find a policy π that maximizes the expected sum of discounted rewards $\mathbb{E}_\pi\left[\sum_{t=1}^{\infty} \gamma^{t-1} r_t\right]$. Adopting the training methodology employed by IRIS, DART likewise consists of three main steps: (1) *Representation Learning*, where VQ-VAEs (Van Den Oord *et al.*, 2017; Esser, Rombach & Ommer, 2021) are used for tokenizing the original observations; (2) *World-Model Learning*, which involves auto-regressive modelling of the dynamics of the environment using a GPT architecture; and (3) *Policy Learning*, which is modelled using ViT for decision-making by attending to task-relevant cues. We now describe our overall approach in detail.

3.3.1 Representation Learning

Discrete symbols are essential in human communication, as seen in natural languages (Cartuyvels, Spinks & Moens, 2021). Likewise, in the context of RL, discrete representation is useful for abstraction and reasoning, leveraging the inherent structure of human communication (Islam *et al.*, 2022). This motivates our approach to model the observation space as a discrete set. In this work, we use VQ-VAE for discretizing the observation space. It learns a discrete latent representation of the input data by quantizing the continuous latent space into a finite number of discrete codes,

$$\hat{z}_q = q(\hat{z}_t^k; \phi_q, Z). \tag{3.1}$$

At time step t, the observation from the environment $x_t \in \mathbb{R}^{H\times W\times 3}$ is encoded by the image encoder f_θ to a continuous latent space $\hat{z}_t^k$. This encoder is modelled using CNNs. The quantization process q maps the predicted continuous latent space $\hat{z}_t^k$ to a discrete latent space $\hat{z}_q$. This is done by finding the closest embedding vector in the codebook Z from a set of N codes (see Equation 3.1). The discrete latent codes are passed to the decoder g_ϕ, which maps them back to the input data $\hat{x}_t$.

The training of this VQ-VAE comprises minimizing the *reconstruction loss* to ensure alignment between input and reconstructed images. Simultaneously, the codebook is learned by minimizing the *codebook loss*, encouraging the embedding vector in the codebook to be close to the encoder output. The *commitment loss* encourages the encoder output to be close to the nearest codebook vector. Additionally, *perceptual loss* is computed to enable the encoder to capture high-level features. The total loss in VQ-VAE is a weighted sum of these loss functions.

This approach enables the modelling of fine-grained, low-level information within the input image as a set of discrete latent codes.

3.3.2 World-model learning

The discrete latent representation forms the core of our approach, enabling the learning of dynamics (Madden, Fox & Thrampoulidis, 2025) through an auto-regressive next-token prediction approach. A transformer decoder based on the GPT architecture is used for modelling this sequence prediction framework. First, an aggregate sequence $\hat{z}_{ct} = f_\phi(\hat{z}_{<t}, \hat{a}_{<t})$ is modelled by encoding past latent tokens and actions at each time step. The aggregated sequence is used for estimating the distribution of the next token, contributing to the modelling of future states given as $\hat{z}_{q_t}^k \sim p_d(\hat{z}_{q_t}^k \mid \hat{z}_{ct})$. Simultaneously, it is also used for estimating the reward $\hat{r}_t \sim p_d(\hat{r}_t \mid \hat{z}_{ct})$ and the episode termination $\hat{d}_t \sim p_d(\hat{d}_t \mid \hat{z}_{ct})$. This training occurs in a self-supervised manner, with the next state predictor and termination modules trained using cross-entropy loss, while reward prediction uses mean squared error.

3.3.3 Policy-learning

The policy π is trained within the world model (also referred to as imagination) using a transformer encoder architecture based on ViT. At each time step t, the policy processes the current observation as K discrete tokens received from the world model. These observation tokens are extended with additional learnable embeddings, including a `CLS` token placed at the beginning and a `MEM` token appended to the end,

$$\mathbf{out} = [\mathtt{CLS}, \hat{z}^1_{q_t}, \ldots, \hat{z}^K_{q_t}, \mathtt{MEM}_{t-1}] + \mathbf{E}_{\text{pos}}. \tag{3.2}$$

The `CLS` token helps in aggregating information from the K observation tokens and the `MEM` token. Meanwhile, the `MEM` token acts as a memory unit, accumulating information from the previous time steps. Thus, at time step t the input to the policy can be represented as $(\mathtt{CLS}, \hat{z}^1_{q_t}, \ldots, \hat{z}^K_{q_t}, \mathtt{MEM}_{t-1})$, where $\hat{z}^K_{q_t}$ corresponds to the embedding of K^{th} index token from the codebook.

While these discrete tokens excel at capturing fine-grained low-level details (Li & Qiu, 2021), they lack spatial information about various features or objects within the image (Darcet, Oquab, Mairal & Bojanowski, 2024). Transformers, known for their permutational-equivariant nature, efficiently model global representation (Xu, Yang, Liu & He, 2023d; Yun, Bhojanapalli, Rawat, Reddi & Kumar, 2020). To incorporate local spatial information, we add learnable positional encoding $\mathbf{E}_{\text{pos}}$ to the original input (see Equation 3.2). During training, these embeddings converge into vector spaces that represent the spatial location of different tokens.

Following this spatial encoding step, the output is first processed with layer normalization (LN) within the residual block. This helps in enhancing gradient flow and eliminates the need for an additional warm-up strategy as recommended in Xiong *et al.* (2020). Subsequently, the output undergoes processing via multi-head self-attention (MSA) and a multi-layer perceptron (MLP) (see Equation 3.3). This series of operations is repeated for a total of L blocks,

$$\left.\begin{aligned}\mathbf{out} &= \mathbf{out} + \mathrm{MSA}(\mathrm{LN}(\mathbf{out})),\\ \mathbf{out} &= \mathbf{out} + \mathrm{MLP}(\mathrm{LN}(\mathbf{out})),\end{aligned}\right\} \times L \\ h_t = \mathbf{out}[0], \quad \texttt{MEM}_t = \mathbf{out}[0]\,. \tag{3.3}$$

Following L blocks of operations, the feature vector associated with the CLS token serves as the representation, modelling both the current state and memory. This representation h_t is used by the policy to sample action $\hat{a}_t \sim p_\theta\left(\hat{a}_t \mid \hat{h}_t\right)$ and by the critic to estimate the expected return, $v_\xi\left(\hat{h}_t\right) \approx \mathbb{E}_{p_\theta}\left[\sum_{\tau \geq t} \hat{\gamma}^{\tau-t}\hat{r}_\tau\right]$. The reward prediction, episode end prediction, and the token predictions of the next observation by the world model follow this.

The feature vector h_t now becomes the memory unit. This is possible because the self-attention mechanism acts like a gate, passing on information to the next time step as required by the task. This simple approach enables effective memory modelling without relying on recurrent networks, which can be challenging to train and struggle with long context (Pascanu, Mikolov & Bengio, 2013).

The imagination process unfolds for a duration of H steps, stopping on episode-end prediction. To optimize the policy, we follow a similar objective function as IRIS and DreamerV2 approaches.

3.4 Experiments

We evaluated our model alongside existing baselines using the Atari 100k benchmark (Kaiser *et al.*, 2020), a commonly used testbed for assessing the sample-efficiency of RL algorithms. It consists of 26 games from the Arcade Learning Environment (Bellemare, Naddaf, Veness & Bowling, 2013), each with distinct settings requiring perception, planning, and control skills.

We evaluated our model's performance based on several metrics, including the mean and median of the human-normalized score, which measures how well the agent performs compared to human and random players given as $\frac{\text{score}_{\text{agent}} - \text{score}_{\text{random}}}{\text{score}_{\text{human}} - \text{score}_{\text{random}}}$. We also used the super-human

Table 3.1 DART achieves a new state-of-art median score among no-look-ahead search methods. It attains the highest median score, inter-quartile mean (IQM), and optimality gap score. Moreover, DART outperforms humans in 9 out of 26 games and achieves a higher score than IRIS in 18 out of 26 games (underlined)

			No look-ahead search				
					Transformer based		
Game	Random	Human	SPR	DreamerV3	TWM	IRIS	DART
Alien	227.8	7127.7	841.9	959	674.6	420.0	**962.0**
Amidar	5.8	1719.5	**179.7**	139	121.8	143.0	125.7
Assault	222.4	742.0	565.6	706	682.6	**1524.4**	1316.0
Asterix	210.0	8503.3	962.5	932	**1116.6**	853.6	956.2
BankHeist	14.2	753.1	345.4	**649**	466.7	53.1	629.7
BattleZone	2360.0	37187.5	14834.1	12250	5068.0	13074.0	**15325.0**
Boxing	0.1	12.1	35.7	78	77.5	70.1	**83.0**
Breakout	1.7	30.5	19.6	31	20.0	**83.7**	41.9
ChopperCommand	811.0	7387.8	946.3	420	**1697.4**	1565.0	1263.8
CrazyClimber	10780.5	35829.4	36700.5	**97190**	71820.4	59324.2	34070.6
DemonAttack	152.1	1971.0	517.6	303	350.2	2034.4	**2452.3**
Freeway	0.0	29.6	19.3	0	24.3	31.1	**32.2**
Frostbite	65.2	4334.7	1170.7	909	**1475.6**	259.1	346.8
Gopher	257.6	2412.5	660.6	**3730**	1674.8	2236.1	1980.5
Hero	1027.0	30826.4	5858.6	**11161**	7254.0	7037.4	4927.0
Jamesbond	29.0	302.8	366.5	445	362.4	**462.7**	353.1
Kangaroo	52.0	3035.0	3617.4	**4098**	1240.0	838.2	2380.0
Krull	1598.0	2665.5	3681.6	**7782**	6349.2	6616.4	7658.3
KungFuMaster	258.5	22736.3	14783.2	21420	**24554.6**	21759.8	23744.3
MsPacman	307.3	6951.6	1318.4	1327	**1588.4**	999.1	1132.7
Pong	-20.7	14.6	-5.4	18	**18.8**	14.6	17.2
PrivateEye	24.9	69571.3	86.0	**882**	86.6	100.0	765.7
Qbert	163.9	13455.0	866.3	**3405**	3330.8	745.7	750.9
RoadRunner	11.5	7845.0	12213.1	**15565**	9109.0	4046.2	7772.5
Seaquest	68.4	42054.7	558.1	618	774.4	661.3	**895.8**
UpNDown	533.4	11693.2	10859.2	7667	**15981.7**	3546.2	3954.5
#Superhuman(↑)	0	N/A	6	9	7	9	9
Mean(↑)	0.000	1.000	0.616	1.120	0.956	1.046	1.022
Median(↑)	0.000	1.000	0.396	0.466	0.505	0.289	**0.790**
IQM(↑)	0.000	1.000	0.337	0.490	-	0.501	**0.575**
Optimality Gap(↓)	1.000	0.000	0.577	0.508	-	0.512	**0.458**

score to quantify the number of games in which our model outperformed human players. We further evaluated our model's performance using the Interquartile Mean (IQM) score and the

Optimality Gap, following the evaluation guidelines outlined in Agarwal, Schwarzer, Castro, Courville & Bellemare (2021).

We rely on the median score to evaluate overall model performance, as it is less affected by outliers. A few games with exceptional or poor performance can strongly influence the mean score. Additionally, the IQM score helps in assessing both consistency and average performance across all games.

Atari environments offer the model an RGB observation of 64×64 dimensions, featuring a discrete action space, and the model is allowed to be trained using only 100k environment steps (equivalent to 400k frames due to a frame-skip of 4), which translates to approximately 2 hours of real-time gameplay.

The world model is trained with a GPT-style causal (decoder) transformer, while the policy is trained using a ViT-style (encoder) transformer. This allows for parallel computation of multiple steps during world model training, making it computationally much faster than previous methods like the recurrent network-based DreamerV3. During policy training, actions for each time step are computed using the modelled CLS token, which then serves as a memory token for the next step. Although this process is calculated step by step, it remains efficient compared to methods like IRIS, as it doesn't require additional networks like LSTM to retain memory.

3.4.1 Results

In Figure 3.2, we present the IQM and optimality gap scores, as well as the mean and median scores. These scores pertain to various models assessed on Atari 100k. Figure 3.4a visualizes the performance profile, while Figure 3.4b illustrates the probability of improvement, which quantifies the likelihood of DART surpassing baseline models in any Atari game. To perform these comparisons, we use results from Micheli *et al.* (2023), which include scores of 100 runs of CURL (Laskin, Srinivas & Abbeel, 2020), DrQ (Yarats, Kostrikov & Fergus, 2021a), SPR (Schwarzer *et al.*, 2021), as well as data from 5 runs of SimPLe (Kaiser *et al.*, 2020) and IRIS.

DART exhibits a similar mean performance as IRIS. However, the median and IQM scores show that DART outperforms other models consistently.

Table 3.1 presents DART's score across all 26 games featured in the Atari 100k benchmark. We compare its performance against other strong world models, including DreamerV3 (Hafner *et al.*, 2025), as well as other transformer-based world models, such as TWM (Robine *et al.*, 2023a) and IRIS (Micheli *et al.*, 2023).

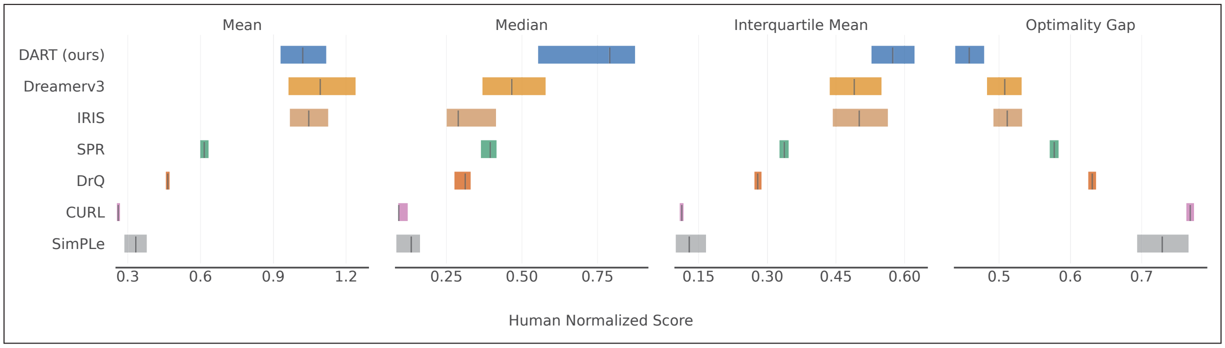
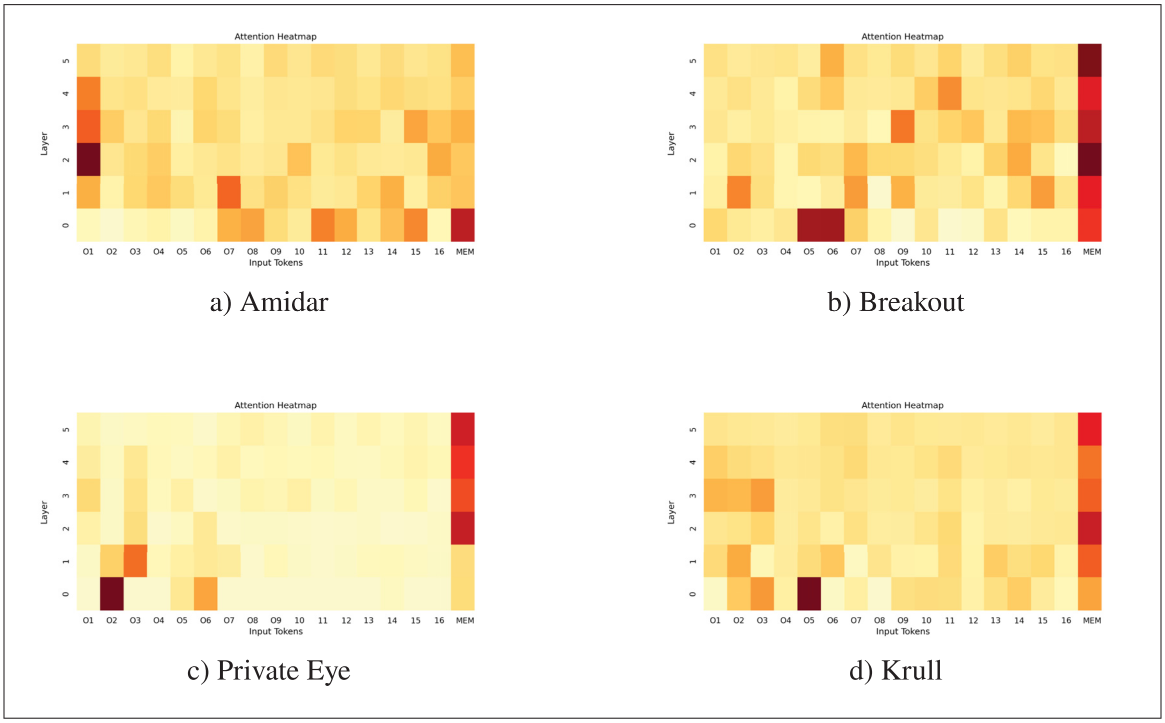


Figure 3.2 Comparison of Mean, Median, and Inter-quartile Mean Human-Normalized Scores

To assess DART's overall performance, we calculate the average score over 100 episodes post-training, utilizing five different seeds to ensure robustness. DART outperforms the previous best model, IRIS, in 18 out of 26 games. It achieves a median score of 0.790 (an improvement of 61% when compared to DreamerV3). Additionally, it reaches an IQM of 0.575, reflecting a 15% advancement, and significantly improves the OG score to 0.458, indicating a 10% improvement when compared to IRIS. DART also achieves a superhuman score of 9, outperforming humans in 9 out of 26 games.

3.4.2 Policy Analysis

In Figure 3.3, we present the attention maps for the six layers of our transformer policy using a heat-map visualization. These maps are generated by averaging the attention scores from each multi-head attention mechanism across all layers. The final visualization is obtained by further averaging these attention maps over 20 randomly selected observation states during an episode. This analysis provides insights into our approach to information aggregation through self-attention.



a) Amidar

b) Breakout

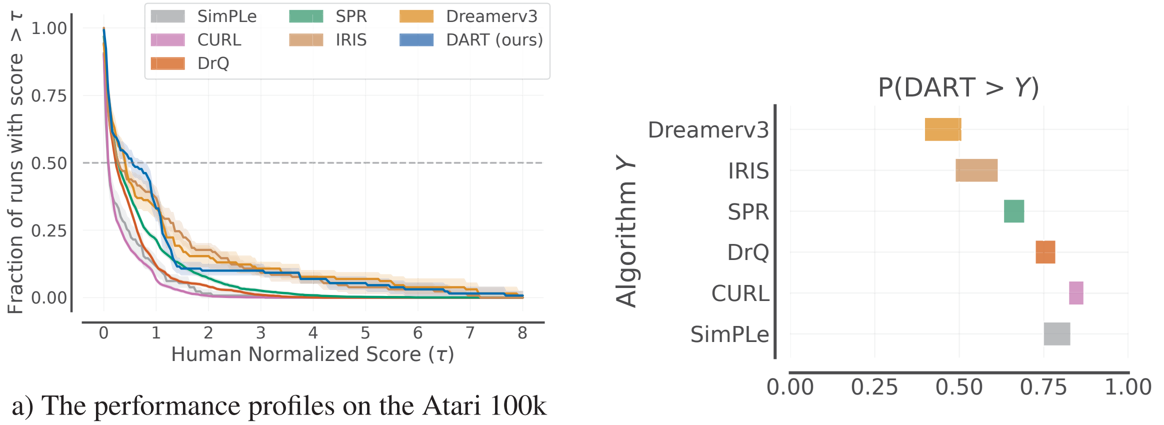
c) Private Eye

d) Krull

Figure 3.3 Comparison of Memory Requirements Across Atari Games: Atari games exhibit varying memory requirements, depending on their specific dynamics. Games with relatively static or slow-moving objects, like *Amidar*, maintain complete information at each time step and thus aggregate less information from the memory token. Conversely, games characterized by rapidly changing environments, such as *Breakout*, *Krull*, and *PrivateEye*, require modelling the past trajectories of objects. As a result, the policy for these games heavily relies on the memory token to aggregate information from past states into future states

The visualization in Figure 3.3 shows that the extent to which information is aggregated from the past and the current state to the next state depends on the specific task at hand. In games featuring slowly moving objects where the current observation provides complete information to the agent, the memory token receives less attention (see Figure 3.3a). Conversely, in environments with fast-moving objects like balls and paddles, where the agent needs to model the past trajectory of objects (e.g., Breakout and Private Eye), the memory token is given more attention (see

Figure 3.3b- 3.3d). This observation highlights the adaptability of our approach to varying task requirements.



a) The performance profiles on the Atari 100k benchmark illustrate the proportion of runs across all games (y-axis) that achieve a score normalized against human performance (x-axis)

b) The probabilities of improvement visualized here refer to the likelihood of DART surpassing the performance of baseline models in any game

Figure 3.4 Comparison of different models using performance profiles and probabilities of improvement

On further analysis, we observe that DART performs better in environments with many approaching enemies and infraction, such as Alien (three enemies need to be tracked) and Seaquest (keep track of divers and dodge enemy subs and killer sharks). However, DreamerV3 does better in games where global information is enough for planning actions. This is because DART uses discrete tokens to focus on critical task-related details with its attention mechanism, while DreamerV3's approach suits games with fewer components. We saw a similar pattern in long, complex tasks with multiple infractions in the game of Crafter (Section 3.4.4), where DART showed improved performance over DreamerV3 in modeling long-horizon tasks with multiple components.

3.4.3 Ablation Studies

We further analyzed DARTs performance across various experimental settings, as detailed in Table 3.2 for five distinct games. The original score of DART is presented in the second column. The different scenarios include:

Without Positional Encoding (PE): The third column demonstrates the performance of DART when learned positional encoding is excluded. We can observe that in environments where agents need to closely interact with their surroundings, such as in Boxing and KungFuMaster, the omission of positional encoding significantly impacts performance. However, in games where the enemy may not be near the agent, such as Amidar, there is a slight drop in performance without positional encoding. This is because transformers inherently model global context, allowing the agent to plan its actions based on knowledge of the overall environment state. However, precise decision-making requires positional information about the local context. In our case, adding learnable positional encoding provides this, resulting in a significant performance boost.

No Exploration (ϵ): The fourth column illustrates DARTs performance when trained without random exploration, relying solely on agent-predicted actions for collecting trajectories for world modelling. However, like IRIS, our model also faces the double-exploration challenge. This means that the agent's performance declines when new environment states aren't introduced through random exploration, which is crucial for effectively modelling the dynamics of the world. It's worth noting that for environments with simpler dynamics (e.g., Seaquest), the performance impact isn't as substantial.

Masking Memory Tokens: In the fifth column, we explore the impact of masking the memory token, thereby removing past information. Proper modelling of memory is crucial in RL to address the challenge of partial observability and provide information about various states (e.g., the approaching trajectory of a ball, and the velocity of the surrounding objects) that are important for decision-making. Our method of aggregating memory over time enhances

DARTs overall performance. However, since Atari games exhibit diverse dynamics, the effect of masking the memory tokens varies accordingly.

In some games, decisions are solely based on information from the current time step, making memory tokens unnecessary. However, in other games, such as Breakout, Private Eye, and Krull, tracking memory is crucial for optimal planning, like predicting the ball's trajectory or following clues. This is evident in the heat map visualization in Figure 3.3, where significant attention is given to memory tokens for the games as mentioned earlier. On the contrary, in games like Boxing and Amidar, where long-term trajectory information or extensive planning isn't needed, relying solely on recent state information is often sufficient for optimal decision-making. Thus, there is only a small impact on the final performance with the masking of memory tokens.

It is interesting to observe improvement in the agent's performance with masked memory tokens in the case of RoadRunner. This could be because the original state already contains complete information, rendering the memory token redundant, thereby impacting the final performance.

Random Observation Token Masking: The last set of columns explores the consequences of randomly masking observation tokens, which selectively removes low-level information. Given that each token among the K tokens models distinct low-level features of the observation, random masking has a noticeable impact on the agent's final performance. When observation tokens are masked 100%, the agent attends solely to the memory token, resulting in a significant drop in overall performance.

Table 3.2 Evaluating DART's performance through various techniques such as memory token masking, random observation masking, and the removal of positional encoding and random exploration

					Masked			
		w/o		Masked	Observation Token			
Game	Original	PE	ϵ	Memory	25%	50%	75%	100%
Boxing	83.0	3.86	58.67	81.45	77.79	51.14	15.64	-11.91
Amidar	125.7	77.1	92.75	113.69	102.47	56.37	52.22	30.43
Road Runner	7772.5	1030.0	3597.1	8021.0	7354.0	2730.0	988.0	961.0
Seaquest	895.8	64.2	753.93	704.8	491.4	207.8	104.0	142.0
KungFuMaster	23744.3	1028.0	15464.7	20378.0	16436.0	9760.0	4676.2	1571.8

3.4.4 Experiment on Crafter

Crafter (Hafner, 2022), inspired by Minecraft (Guss *et al.*, 2019), allows assessing an agent's general abilities within a single environment. This distinguishes it from Atari 100k, where the agent must be evaluated across 26 different games that test for various skills. In Crafter, 2D worlds are randomly generated, featuring diverse landscapes like forests, lakes, mountains, and caves on a 64×64 grid. Players aim to survive by searching for essentials like food, water, and shelter while defending against monsters, collecting materials, and crafting tools. This setup allows for evaluating a wide range of skills within a single environment, spanning multiple domains, and increasing assessment comprehensiveness. The environment is partially observable with observations covering a small 9×9 region centred around the agent.

Table 3.3 Comparing the sample efficiency of DreameV3, IRIS, and DART on the challenging Crafter environment, which involves long-horizon tasks. Reported returns are specified as average and standard deviation over five seeds

Model	DreamerV3	IRIS	DART
Steps	1M	1M	1M
Return	11.7 ± 1.9	9.23 ± 0.56	**12.2 ± 1.67**

In the results shown in Table 3.3, we compare DART with IRIS and Dreamer V3 in a low data regime of 1M environment steps, which is particularly challenging in Crafter since tasks require long-horizon credit assignment and typically demand 10M+ steps for stable performance. DART achieves a higher average return, further showcasing the efficiency of DART over previous models.

3.4.5 Experiment on Atari with More Environment Steps

By training it beyond 100k training steps, we see improved performance of DART as shown in Table 3.4, showcasing the scalability of DART with more data.

Table 3.4 Performance of DART with 100k and 150k environment steps (k). All results are shown as average and standard deviation over five seeds

Environment	**Steps** (k)	**Score**
Freeway	100k	32.2 ± 0.57
	150k	33.1 ± 0.37
KungFuMaster	100k	23744.3 ± 3271.53
	150k	24756.5 ± 2635.21
Pong	100k	17.2 ± 1.74
	150k	17.6 ± 2.79

3.4.6 Model Configuration

Recent works have used transformer-based architectures for MBRL. In Table 3.5 we compare the configurations used by different approaches for representation learning, world modelling, and behaviour learning.

Table 3.5 Comparing the model configuration of recent MBRL approaches. n/a- Not Available; Cat.-VAE - Categorical VAE.; MAE - Masked Auto Encoder

	MWM	TWM	IRIS	DreamerV3	STORM	DART
Parameters	n/a	n/a	3.04M	18M	n/a	3.07M
State model	MLP	MLP	CNN	MLP	MLP	ViT
Agent memory	ViT	Tr-XL	LSTM	GRU	GPT	ViT (Self-attention)
Representation	MAE	Cat.-VAE	VQ-VAE	Cat.-VAE	Cat.-VAE	VQ-VAE

3.4.7 Hyperparameters

A detailed list of hyperparameters is provided for each module: Table 3.6 for Image Tokenizer, Table 3.7 for World Modeling, and Table 3.8 for behaviour learning.

3.4.8 Comparing the performance with STORM

Recently released, another transformer-based model, STORM (Zhang, Wang, Sun, Yuan & Huang, 2023c) showcases close to similar performance when compared with DART as shown in Table 3.9.

Table 3.6 Hyperparameters for image tokenization using VQ-VAE

Hyperparameter	**Symbol**	**Value**
Encoder Convolutional layers	–	4
Decoder Convolutional layers	–	4
Per layer residual blocks	–	2
Self-attention layers	–	8 / 16
Codebook size	N	512
Embedding dimension	d	512
Input image resolution	–	64×64
Image channels	–	3
Activation	–	Swish
Tokens per image	K	16
Batch size	–	64
Learning rate	–	0.0001

Table 3.7 Hyperparameters used for modelling the dynamics using the transformer decoder

Hyperparameter	**Symbol**	**Value**
Embedding dimension	–	256
Transformer layers	–	10
Attention heads	–	4
Imagination steps	H	20
Embedding dropout	–	0.1
Weight decay	–	0.01
Attention dropout	–	0.1
Residual dropout	–	0.1
Attention type	–	Causal
Activation	–	GeLU
Batch size	–	64
Learning rate	–	0.0001

STORM relies on utilizing VAEs for modelling stochastic world models, while we use VQ-VAE for modelling discrete world models. While STORM performs similarly to DART overall, it struggles in games like Pong and Breakout with single, small moving objects. Comparing DART's performance with STORM, there's a notable improvement in DART's performance due to its discrete representation. DART's use of discrete tokens for each entity, coupled with an attention mechanism, enables the agent to focus precisely on relevant tasks, leading to significant

Table 3.8 Hyperparameters used for modelling behaviour using the transformer encoder

Hyperparameter	**Symbol**	**Value**
Input tokens	–	18
Embedding dimension	–	512
Attention heads	–	8
Transformer layers	L	6
Dropout	–	0.2
Activation	–	GeLU
Transformer layers	–	6
Attention type	–	Self-attention
Positional embedding	–	Learnable
Gamma	γ	0.995
Lambda	λ	0.95
Batch size	–	64
Epsilon	ϵ	0.01
Temperature (train)	–	1.0
Temperature (test)	–	0.5
Learning rate	–	0.0001

performance boosts, especially in such games. Additionally, this approach makes DART more interpretable compared to STORM, as illustrated in Figure 3.3's heat-map visualization.

3.5 Conclusion

In this work, we introduced DART, a model-based reinforcement learning agent that learns both the model and the policy using discrete tokens. Through our experiments, we demonstrated that our approach helps in improving performance and achieves a new state-of-the-art score on the Atari 100k benchmarks for methods with no look-ahead search during inference. Moreover, our approach for memory modelling and the use of a transformer for policy modelling provide additional benefits in terms of interpretability.

Limitations: As of now, our method is primarily designed for environments with discrete action spaces. This limitation poses a significant challenge, considering that many real-world robotic control tasks necessitate continuous action spaces. For future work, it would be interesting to adapt our approach to continuous action spaces and modelling better-disentangled tokens for faster learning.

Table 3.9 Comparing the performance of DART with STORM

Game	STORM	DART
Alien	984	962.0
Amidar	205	125.7
Assault	801	1316.0
Asterix	1028	956.2
BankHeist	641	629.7
BattleZone	13540	15325.0
Boxing	80	83.0
Breakout	16	41.9
ChopperCommand	1888	1263.8
CrazyClimber	66776	34070.6
DemonAttack	165	2452.3
Freeway	0	32.2
Frostbite	1316	346.8
Gopher	8240	1980.5
Hero	11044	4927.0
Jamesbond	509	353.1
Kangaroo	4208	2380.0
Krull	8413	7658.3
KungFuMaster	26182	23744.3
MsPacman	2673	1132.7
Pong	11	17.2
PrivateEye	7781	765.7
Qbert	4522	750.9
RoadRunner	17564	7772.5
Seaquest	525	895.8
UpNDown	7985	3954.5
#Superhuman(↑)	9	9
Mean(↑)	1.267	1.022
Median(↑)	0.584	0.790

CHAPTER 4

STRIDE: CONTINUAL LEARNING USING SKILL TRANSFER AND REUSE WITH INCREMENTAL DISCRETE ENCODING

Pranav Agarwal [1] , Michele Rocca [2] , Victor Zordan [3] , Sheldon Andrews[1]

[1] Department of Software and IT Engineering, École de Technologie Supérieure,
1100 Notre-Dame Ouest, Montréal, Québec, Canada H3C 1K3
[2] Department of Computer Science, University of Copenhagen,
Nørregade 10, 1172 København, Denmark
[3] Department of Visual Computing, Clemson University,
105 Sikes Hall, Clemson, SC 29634, United States

Article submitted to « Siggraph Asia » in May 2025

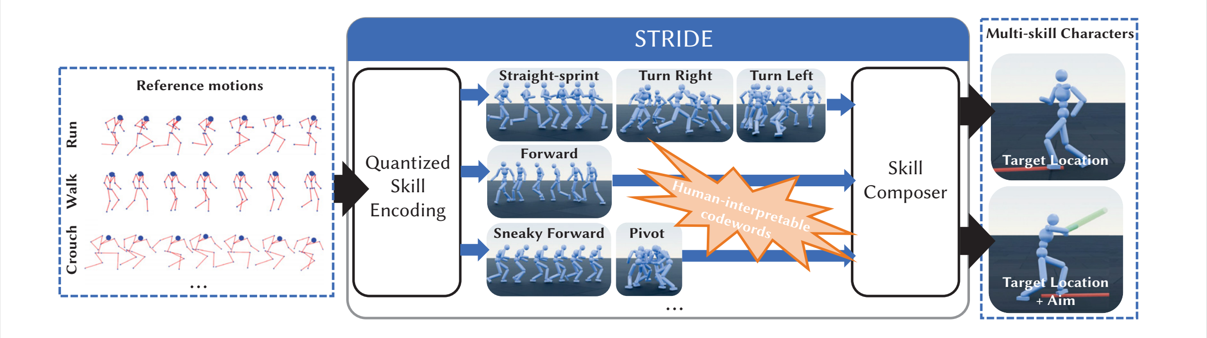


Figure 4.1 STRIDE models task learning as a continual RL problem: it continually learn new tasks while modelling skills into clear, human-understandable codewords (e.g., Run, Turn-Right, Pivot) stored in a codebook that models a discrete latent space. This reusable skill vocabulary enables a lightweight composer to rapidly mix past skills to master new tasks, such as navigating to a target while aiming, thereby enabling fast transfer, scalability, and interpretability

Résumé

Les personnages basés sur la physique offrent un grand potentiel pour créer des agents polyvalents dans des environnements immersifs. Des progrès significatifs ont été réalisés dans le développement de contrôleurs individuels pour des comportements spécifiques en utilisant l'apprentissage par renforcement. Cependant, l'adaptation de ces contrôleurs à de nouvelles conditions reste difficile en raison de l'interférence des politiques dans l'espace continu. Une solution courante consiste à construire indépendamment des contrôleurs pour chaque comportement, mais cela entraîne une surcharge d'entraînement supplémentaire et des transitions non fluides entre les comportements. Pour relever ces défis, nous introduisons STRIDE (Skill Transfer and Reuse with Incremental Discrete Encoding), un nouveau cadre conçu pour l'apprentissage continu des comportements de locomotion. Contrairement aux méthodes traditionnelles, STRIDE ne nécessite pas l'entraînement de contrôleurs séparés pour chaque comportement. Au lieu de cela, il exploite la quantification de politique à travers un livre de codes, représenté comme un espace latent discret, pour générer des mots de code réutilisables (compétences) qui permettent une composition facile de nouveaux comportements. Nous employons des autoencodeurs variationnels à quantification vectorielle (VQ-VAE) pour apprendre dynamiquement ces compétences discrètes pendant l'entraînement. Ce mécanisme permet à STRIDE de découvrir et d'encoder des composants de comportement réutilisables de manière non supervisée, facilitant ainsi un transfert efficace à travers diverses tâches. STRIDE réduit considérablement le temps d'entraînement (1,5x à 2x plus rapide par rapport à l'état de l'art) et minimise l'interférence lors de l'acquisition de nouvelles politiques. De plus, l'utilisation de représentations latentes discrètes améliore l'interprétabilité de la politique RL, car il devient possible de tracer quelles compétences ont été activées et comment elles ont contribué à l'accomplissement des tâches. Cette capacité répond à deux défis clés de l'apprentissage par renforcement : comprendre le raisonnement derrière les décisions de politique et rendre les comportements appris plus transparents. Nos résultats démontrent que STRIDE accélère l'apprentissage continu de nouveaux comportements sans compromettre la qualité du mouvement, offrant une solution évolutive et interprétable pour les applications d'apprentissage par renforcement dans l'animation de personnages.

Abstract

Physically based characters offer great potential for creating general-purpose agents in immersive environments. Significant progress has been made in developing individual controllers for specific behaviors using reinforcement learning. However, adapting these controllers to new conditions remains challenging due to policy interference in continuous space. One standard solution is to build controllers for each behavior independently, but this leads to additional training overhead and non-smooth transitions between behaviors. To address these challenges, we introduce Skill Transfer and Reuse with Incremental Discrete Encoding (STRIDE), a novel framework designed for continual learning of locomotion behaviors. Unlike traditional methods, STRIDE does not require training separate controllers for each behavior. Instead, it leverages policy quantization through a codebook, represented as a discrete latent space, to generate reusable codewords (skills) that enable easy composition of new behaviors. We employ Vector Quantized Variational Autoencoders (VQ-VAEs) to learn these discrete skills dynamically during training. This mechanism allows STRIDE to discover and encode reusable behavior components in an unsupervised manner, thereby facilitating efficient transfer across diverse tasks. STRIDE significantly reduces training time (1.5x to 2x faster compared to state-of-the-art) and minimizes interference when acquiring new policies. Furthermore, the use of discrete latent representations improves the interpretability of the RL policy, as it becomes possible to trace which skills were activated and how they contributed to completing tasks. This capability addresses two key challenges in reinforcement learning: understanding the reasoning behind policy decisions and making learned behaviors more transparent. Our results demonstrate that STRIDE accelerates the continual learning of new behaviors without compromising motion quality, offering a scalable and interpretable solution for reinforcement learning applications in character animation.

4.1 Introduction

Humans effortlessly adapt and expand their motor skills, acquiring new behaviors without forgetting past ones (Dayan & Cohen, 2011). This ability enables rapid learning and mastery of new movements. While Reinforcement Learning (RL) can model controllers that mimic human motion (Peng *et al.*, 2018), it struggles with continual learning (Khetarpal, Riemer, Rish & Precup, 2022; Kirkpatrick *et al.*, 2017; Gai, Lyu, Zhang & Wang, 2024). It often results in independently learned controllers that fail to leverage existing skills (Peng *et al.*, 2018; Merel *et al.*, 2019), even for closely related behaviors. This is problematic because the policy training time scales as the number of behaviors grows (Xu *et al.*, 2023a).

This work introduces Skill Transfer and Reuse with Incremental Discrete Encoding (STRIDE), a framework for continual reinforcement learning (CRL) of controlled behaviors. Our method adapts existing skills, aggregates new ones, and composes them to learn new behaviors.

In CRL, agents adapt to new tasks or environments sequentially while retaining past skills (Khetarpal *et al.*, 2022), mimicking how humans and animals learn through experience. Continual learning is a critical aspect of intelligent agents, enabling them to acquire new skills over time without forgetting previous ones. In dynamic settings like 3D gaming environments, agents are often required to know multiple skills, ranging from basic locomotion control, such as walking and running, to fine-grained dexterous activities, such as juggling balls or aiming at a target. A key challenge in designing controls for such agents is to leverage knowledge from earlier tasks to enhance performance and accelerate learning on new, related tasks—a concept known as *positive forward transfer* (Schwarz *et al.*, 2018)—while simultaneously avoiding catastrophic forgetting, which occurs when learning new tasks disrupts or overwrites prior knowledge (Kirkpatrick *et al.*, 2016).

At its core, STRIDE employs policy quantization, which maps encoded representations to vectors in a shared discrete latent space and subsequently translates them into final actions. This latent space serves as a dynamic knowledge base, facilitating skill reuse and positive forward transfer for learning new behaviors. We propose the use of Vector Quantized Variational Autoencoders (VQ-VAEs) for this quantization, enabling continuous learning of new tasks while leveraging previously acquired skills. The central ideas behind our framework are conceptualized in the teaser (Fig. 4.1).

Unlike hierarchical pipelines that hard-code a motion prior for a fixed catalogue of skills Peng, Ma, Abbeel, Levine & Kanazawa (2021); Kwiatkowski *et al.* (2022); Zhu, Zhang, Lan & Han (2023a), and must be rebuilt whenever the catalogue changes, our method formulates policy quantization as a dynamic VQ-VAE. The codebook expands on demand: when the agent encounters a movement it cannot express with its current entries, the VQ-VAE creates a new discrete code while still re-using existing ones for shared sub-skills. This continual reuse removes the need to train an entire controller each time a new animation behavior is added, as well as improving compute time and sample efficiency.

Our results show that STRIDE enables faster learning of new behaviors while preserving prior knowledge, with more structured representations than a Variational Autoencoder and quicker learning compared to previous methods like Adversarial Motion Priors (AMP). An essential aspect of our approach is that it gives the controller a built-in "explainability layer." As we highlight in our results, every codeword corresponds to a specific, human-interpretable skill, allowing users to i) see which skills are invoked when learning a new behavior and ii) inspect the sequence of codewords (or strategy) a policy employs to solve any given task, making the construction more intuitive and easier to understand.

The key contributions of our work can be summarized as follows:

- **On-demand skill library via dynamic policy quantization**: The policy's latent space is modelled as a growing VQ-VAE codebook that automatically adds a new code only when the current repertoire cannot express an incoming behavior. This turns the skill library from a static, hand-crafted prior into an adaptive, end-to-end learned component, eliminating complete re-training when a new style or task is introduced.
- **Positive forward transfer with catastrophic-forgetting immunity**: New behaviors are encoded as sparse combinations of existing codes (or, when needed, a small set of new ones), STRIDE reuses earlier skills by design. Across locomotion benchmarks, we show >1.5× faster convergence and near-zero performance drop on previously mastered tasks compared with AMP and Composite Motion baselines.
- **Built-in interpretability and skill attribution**: Each discrete codeword corresponds to a human-interpretable motor primitive. The learned policy therefore exposes (i) which skills are used while learning a new behavior and (ii) the exact code sequence it executes at run-time, offering a transparent window into the agent's strategy without external probes.

Our proposed framework, STRIDE, enables a novel continual reinforcement learning approach for character animation beyond fixed skill sets, modelling a controller that grows, reuses, and explains its abilities as it encounters new challenges.

4.2 Research Problem and Motivations

Continual learning in reinforcement learning (RL) enables agents to learn multiple tasks sequentially without catastrophic forgetting. Traditional RL methods, such as policy gradient and value-based approaches (Wang *et al.*, 2020d; Mnih *et al.*, 2016), excel in single-task scenarios but fail when required to generalize across new tasks (Yu *et al.*, 2019). This limitation stems from their inability to manage interference between tasks or preserve learned knowledge during transitions (Riemer *et al.*, 2019; Moeed *et al.*, 2020).

To address these challenges, several approaches have been proposed to retain knowledge while mitigating interference. Techniques like Elastic Weight Consolidation (EWC) (Kirkpatrick *et al.*, 2017) and Synaptic Intelligence (Zenke, Poole & Ganguli, 2017) employ regularization to prevent forgetting by anchoring network weights to previously learned tasks. Memory-based methods, such as Episodic Memory Replay (Lopez-Paz & Ranzato, 2017), explicitly store and replay past experiences, supporting sequential learning.

Modular architectures offer a powerful route for skill transfer in RL. Early work, such as Progressive Neural Networks (Moeed *et al.*, 2020), policy distillation (Czarnecki *et al.*, 2019), and hierarchical RL (Sutton, Precup & Singh, 1999), reuse knowledge through task-specific modules or motion priors, but newer systems push this idea further. AdaptNet (Xu *et al.*, 2023b) injects small adapter layers into the latent space of a pre-trained controller, allowing a single policy to be re-targeted rapidly to new tasks, environments, or morphologies without full retraining.

Similarly, Composite Motion Learning (Xu *et al.*, 2023a) decouples motion into independently trained body-part modules via multiple discriminators and stabilizes training with adaptive multi-objective weighting, so a character can, for example, gesture with the upper body while walking. Although these advances reduce training overhead and improve reuse, they still rely on clearly defined task boundaries or fixed hierarchies; when consecutive tasks differ substantially in dynamics or reward structure, interference and negative transfer remain problematic (Frans *et al.*, 2017).

Another limitation of prior methods, such as AMP (Peng *et al.*, 2021), ASE (Peng, Guo, Halper, Levine & Fidler, 2022), and DIAYN (Eysenbach, Gupta, Ibarz & Levine, 2019), lies in their dependence on continuous latent spaces for skill modelling. While these methods can discover diverse skills unsupervised, the continuous latent space makes it difficult to interpret and challenging to prevent policy interference, thereby hindering continual learning across diverse tasks (Parisi *et al.*, 2019).

Recent works have shifted towards using discrete latent spaces to overcome these challenges. Vector-Quantized Variational Autoencoders (VQ-VAEs) (Van Den Oord *et al.*, 2017) have emerged as a promising approach, offering structured and interpretable representations by mapping inputs to a codebook of discrete embeddings. This discrete paradigm has proven particularly useful in tasks requiring fine-grained control, diversity, or explicit modelling of high-dimensional data distributions, especially in graphics applications (Zhu *et al.*, 2023a). MoConVQ (Yao *et al.*, 2024) integrates model-based RL with VQ-VAE–learned motion embeddings harvested from tens of hours of unstructured data, yielding a discrete motion vocabulary that supports versatile, robust character control. Unlike MoConVQ's fixed, offline vocabulary, STRIDE grows its codebook incrementally and reuses frozen slices, enabling continual learning and much faster task adaptation.

While Neural Categorical Priors (NCP) (Zhu *et al.*, 2023a) takes a similar direction by using a Vector-Quantized Variational Autoencoder. However, they follow a hierarchical approach that requires pretraining on a corpus of motion data and then training the policy on top of it. However, STRIDE does not require pretraining and instead can be continually learned end-to-end as the new reference motion or task comes in.

In another example, Starke, Starke, He, Komura & Ye (2024) introduced an embodied character controller that combines motion matching with neural network flexibility. By encoding a large and diverse motion dataset into a discrete latent space using VQ-VAEs, their approach avoided error accumulation during auto-regressive inference and enabled high-quality motion synthesis. The use of discrete latent representations facilitated applications across AR/VR gaming scenarios by avoiding blending artifacts and enhancing motion quality.

These advancements highlight the broader utility of VQ-VAEs in graphics tasks requiring explicit output control, robustness to mode collapse, and efficient modelling of sparse or discontinuous phenomena. Building on these insights, we propose STRIDE. This framework extends VQ-VAEs to dynamically aggregate unique skills from diverse behaviors into a structured knowledge base modelled as a discrete latent space (codebook). STRIDE enables positive skill transfer by allowing agents to learn new behaviors rapidly without interfering with previously acquired knowledge, addressing key limitations of existing reinforcement learning methods for character animation.

4.3 Background

This section outlines key concepts behind our approach, focusing on Reinforcement Learning (RL) and the Vector Quantized Variational Autoencoder (VQ-VAE), which form the theoretical foundation of our methodology.

4.3.1 Reinforcement Learning

RL involves training an agent to make decisions by maximizing cumulative rewards in a given environment. The agent's interaction is typically modelled as a Markov Decision Process (MDP), defined by the tuple (S, A, P, R, γ), where S represents the state space, A the action space, $P(s'|s, a)$ the transition probability, $R(s, a)$ the reward function, and $\gamma \in [0, 1]$ the discount factor. The objective is to learn a policy $\pi(a|s)$ that maximizes the expected return

$$\mathbb{E}_{\pi}\left[\sum_{t=0}^{T}\gamma^{t}r_{t}\right]. \tag{4.1}$$

We use Proximal Policy Optimization (PPO) (Wang *et al.*, 2020d), a widely used RL algorithm that stabilizes policy updates by optimizing a clipped surrogate objective. To model the reward function for optimizing the policy, we use Generative Adversarial Imitation Learning (GAIL) (Ho & Ermon, 2016), which enables imitation learning by aligning the learned policy with expert demonstrations. The GAIL framework uses a discriminator to distinguish between expert and agent trajectories, providing feedback that guides the policy to mimic expert behavior effectively.

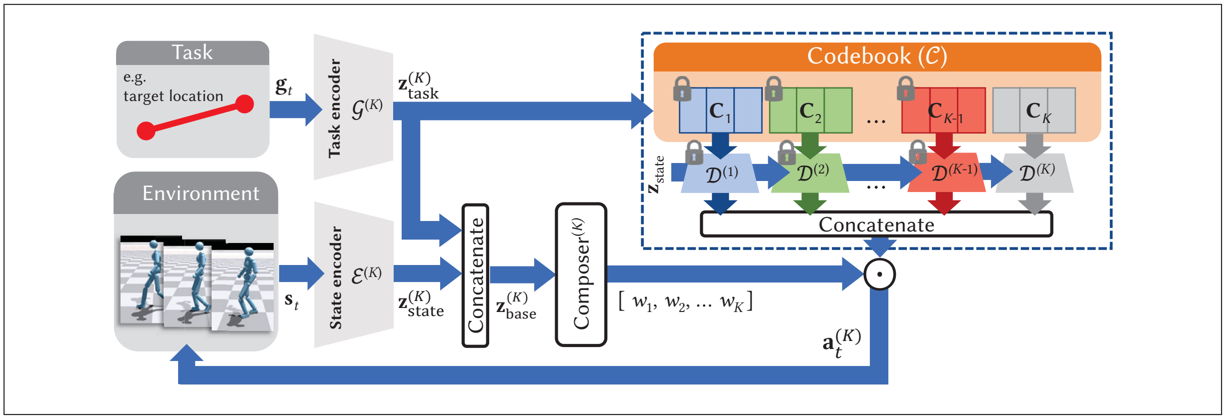


Figure 4.2 STRIDE training pipeline for the k^{th} behavior. The agent receives the current proprioceptive state s_t and a high-level goal g_t (e. g. target location). A *state encoder* $\mathcal{E}_{\text{state}}^{(k)}$ converts the recent kinematic window into a motion latent $z_{\text{state}}^{(k)}$, while a *task encoder* $\mathcal{E}_{\text{task}}^{(k)}$ embeds the goal as $z_{\text{task}}^{(k)}$. The task latent is projected into the codebook space and quantized to the nearest codeword in the union of all slices $C_1 \cup \cdots \cup C_k$. Every slice C_j has a decoder $D^{(j)}$ that outputs a Gaussian action head; slices learned in earlier behaviors are frozen (padlock icons). A lightweight composer mixes the K heads with soft weights $[w_1, \dots, w_k]$, producing the final action $a_t^{(k)}$

4.3.2 Vector Quantized Variational Autoencoder

In this work, we model the policy $\pi(a|s)$ using a VQ-VAE by introducing the concept of policy quantization. Unlike standard VAEs (Kingma & Welling, 2019b), the VQ-VAE encoder outputs a set of discrete codes and learns a prior distribution during training, avoiding posterior collapse caused by static priors in traditional VAEs (Van Den Oord *et al.*, 2017).

Given input data x, the encoder maps it to a continuous latent representation z. Instead of passing z directly to the decoder, it is quantized into a discrete codeword z_q from a predefined codebook C of embeddings $e_1, e_2, \ldots, e_K$. This enables the model to learn discrete representations. The training process jointly optimizes the overall objective, the quantization error, and the commitment loss.

The closest codeword z_q is sampled from the codebook using a distance metric, thus giving the quantized latent. Euclidean distance is commonly used for this purpose, although Cao *et al.* (2023) proposed that L2-normalizing both the codes and the encoded vector using a cosine similarity leads to more stable performance, such that

$$z_\mathrm{q} = \arg\max_{e_i \in C} \frac{z \cdot e_i}{|z||e_i|} \,. \tag{4.2}$$

Once the closest codeword is identified, the decoder maps it to the required output based on this quantized representation.

To ensure the encoder's output z remains aligned with the selected codeword z_q, a commitment loss is introduced:

$$\mathcal{L}_\mathrm{commit} = \|z - \mathrm{sg}(z_\mathrm{q})\|_2^2 \,. \tag{4.3}$$

Here, $\mathrm{sg}(\cdot)$ denotes the stop-gradient operation, ensuring that gradients are not propagated through the codebook during this step.

To update the codebook embeddings stably and gradually, Exponential Moving Average (EMA) is used. This method ensures smooth updates of the codebook vectors by taking a weighted average of the current and past embeddings. Instead of directly updating the codebook based on each batch, EMA accumulates the changes over time, which prevents abrupt shifts that could destabilize training.

The usage count N_i of each codeword e_i is first updated by

$$N_i^{t+1} = \lambda N_i^t + (1 - \lambda) m_i \, ,$$

where m_i is one if codeword e_i was selected (as shown in Eqn. 4.2) for the current input and zero otherwise, and λ is the EMA decay factor– a value between 0 and 1 that controls the rate of updates. The codebook vector e_i is then updated using the encoder's output z, and the codebook embedding e_i^t at time step t, such that

$$e_i^{t+1} = \frac{N_i^t e_i^t + (1 - \lambda) z}{N_i^{t+1}} \, .$$

This ensures that more frequently used codewords are updated more substantially than those used less often.

In our work, we propose an online version of VQ-VAE where codewords representing skills are dynamically modelled as required. This enables the codebook to expand as needed, facilitating continual learning.

4.4 STRIDE

STRIDE turns continuous controller learning into a problem of discovering and reusing a discrete library of motion skills. The key idea is to quantize the policy using a shared vector-quantized (VQ) codebook so that every newly-mastered behavior, or skill, contributes a fixed set of latent codewords that can be re-combined by later tasks, as conceptualized in Fig. 4.2.

We begin this section with an explanation of how the discrete codebook is built for a single behavior (Sec. 4.4.1), and then explain how collections of behaviors, or skills, can be composited by a lightweight gating network (Sec. 4.4.2).

4.4.1 Skill acquisition by policy quantization

STRIDE models skill embeddings from a reference motion. Our approach incrementally learns unique skills based on behavioral requirements. Unlike prior methods that directly map states to actions, we quantize the policy by integrating a codebook into the agent's pipeline.

4.4.1.1 State encoder

At each time step t, the state encoder $\mathcal{E}_{\text{state}}$ processes the character kinematics

$$\mathbf{s}_t = \{\mathbf{x}_{t-n}, \mathbf{x}_{t-n+1}, \ldots, \mathbf{x}_t\} \tag{4.4}$$

which encompass the positions, orientations, linear velocities, and angular velocities of the simulated character's body links over the previous $n + 1$ frames. This input is transformed into a latent representation $\mathbf{z}_{\text{state}} \in \mathbb{R}^d$, where d denotes the dimensionality of the latent space. The transformation is achieved by first modelling the temporal dynamics using a Gated Recurrent Unit (GRU) (Cho *et al.*, 2014b), followed by a feed-forward neural network.

4.4.1.2 Task encoder and the codebook

A smaller task description encoder $\mathcal{E}_{\text{task}}$ models the task representation $\mathbf{z}_{\text{task}}$ which is then quantized into a discrete codeword $\mathbf{z}_{\text{q}}$ using a codebook $C = \{e_1, e_2, \ldots, e_K\}$. Initially, the codebook is allotted a set of codewords. Quantization is performed by selecting the codeword in C that maximizes the cosine similarity with $\mathbf{z}_{\text{task}}$, as detailed in Eqn. 4.2. The quantized representation $\mathbf{z}_{\text{q}}$ is then concatenated with $\mathbf{z}_{\text{state}}$ and passed to the decoder $\mathcal{D}$, which models the action $\mathbf{a}_t$ at time step t as a distribution in the form of a Gaussian with mean $\boldsymbol{\mu}_t$ and standard deviation $\boldsymbol{\sigma}_t$:

$$\mathbf{a}_t \sim \mathcal{N}(\boldsymbol{\mu}_t, \boldsymbol{\sigma}_t^2) \,. \tag{4.5}$$

4.4.1.3 Augmenting the codebook

We follow the process outlined in Xu *et al.* (2023a) to train the discriminator and compute the reward. A commitment loss (Eqn. 4.3) ensures that the encoder's output, $\mathbf{z}_{\text{task}}$, aligns with the selected codeword, $\mathbf{z}_{\text{q}}$, despite the non-differentiable quantization step ($\mathbf{z}_{\text{task}} \rightarrow \mathbf{z}_{\text{q}}$). Gradients from the decoder's loss flow back to the encoder through $\mathbf{z}_{\text{task}}$, bypassing the quantization step, as the stop-gradient operation $\text{sg}(\cdot)$ prevents gradients from directly affecting $\mathbf{z}_{\text{q}}$. This loss provides the necessary signal for the encoder to adjust its output, keeping it close to the quantized codeword and enabling effective end-to-end training.

This approach enables simultaneous training of the policy and modelling of skills as codewords. For every new behavior, skills (or codewords) are incrementally added to the codebook C as needed.

4.4.2 Skill Composition

After the root behavior has been learned, STRIDE introduces a composer that utilizes previously learned skills with newly acquired ones to learn new behaviors. For each new behavior, a lightweight encoder and a composer, implemented as simple feed-forward neural networks—are progressively added, while freezing the parameters of the previous behaviors.

4.4.2.1 Multi-behavior codebook

The character state and task representation are encoded following the same strategy as the first behavior, while additionally utilizing the skills from previous behaviors. The $\mathbf{z}_{\text{task}}$ is quantized by selecting the closest codeword from each category of previously learned behaviors.

Suppose on training the policy for the Kth behaviors, with $K - 1$ behaviors already learned, each with its own set of codewords. Let $\mathbf{C}_k$ represent the set of codewords for the k-th behavior, where $k \in \{1, 2, \ldots, K\}$. The quantization process involves finding the nearest codeword in each category $\mathbf{C}_k$ to $\mathbf{z}_{\text{task}}$.

The final quantized representation of $\mathbf{z}_{\text{task}}$ is then a combination of the closest codewords from all categories:

$$\{\mathbf{z}_{\text{q}}^{(1)}, \mathbf{z}_{\text{q}}^{(2)}, \ldots, \mathbf{z}_{\text{q}}^{(K)}\}\,. \tag{4.6}$$

4.4.2.2 Composing actions

Each behavior has a decoder $D^{(j)}$ (simple Multi-Layer Perceptron (MLP)); the parameters of the decoders for previously learned behaviors remain fixed. This ensures that the representations of earlier skills remain stable, while only the new behavior's codewords and the skill corresponding to it are updated. The sampled codewords are then mapped to their corresponding Gaussian distribution parameters, obtaining K pairs of Gaussian parameters: $\{\mu_1, \sigma_1\}, \{\mu_2, \sigma_2\}, \ldots, \{\mu_K, \sigma_K\}$

To sample the final action, the K Gaussian distributions corresponding to the learned behaviors are composed into a single distribution. This composition is achieved using a composer, implemented as a simple feed-forward neural network. The composer takes the task representation $\mathbf{z}_{\text{task}}$ and $\mathbf{z}_{\text{state}}$ as input and outputs a set of softmax scores $\mathbf{w} = [w_1, w_2, \ldots, w_K]$, where each w_k represents the contribution of the k-th behavior to the final action. These scores dynamically weigh the influence of each behavior based on the context.

For each behavior k, the weighted mean $\bar{\mu}_k$ is computed as:

$$\bar{\mu}_k = w_k \cdot \mu_k \quad , \tag{4.7}$$

where μ_k is the mean of the k-th Gaussian distribution, respectively. The final mean μ of the composed distribution is obtained by aggregating the weighted means across all K behaviors:

$$\mu = \sum_{k=1}^{K} \bar{\mu}_k \quad \text{and} \quad \sigma^2 = \bar{\sigma}_k^2 \,. \tag{4.8}$$

The final standard deviation σ is then computed as $\sigma = \sqrt{\sigma^2}$.

This process ensures that the final action is a weighted combination of all learned behaviors, with the gating weights **w** dynamically determining the relative contribution of each behavior. By composing the distributions in this manner, the system effectively balances the influence of multiple skills while maintaining stability and adaptability.

4.4.2.3 Multi-skill knowledge distillation

We incorporate a commitment loss for each category into the final loss function to ensure effective knowledge distillation across all learned skills. The total commitment loss is the sum of the individual losses across all categories:

$$\mathcal{L}_{\text{commit}} = \sum_{k=1}^{K} \mathcal{L}_{\text{commit},k} \,. \tag{4.9}$$

To prevent forgetting and minimize interference between different skill behaviors, only the codewords corresponding to new behaviors are updated using Exponential Moving Average (EMA).

4.5 Experiments

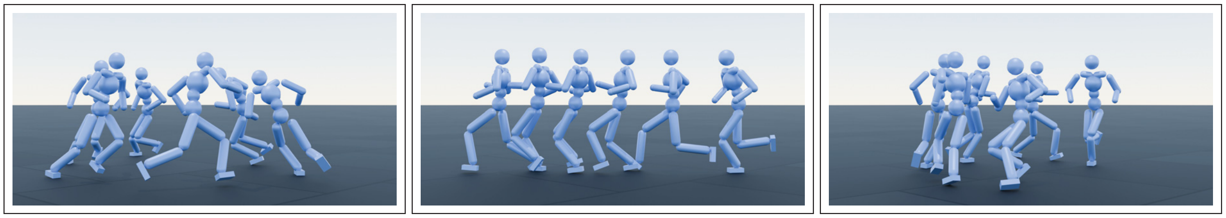
We benchmark STRIDE on a suite of physics-based character–control tasks designed to highlight (i) continual skill acquisition, (ii) goal-directed adaptation, and (iii) sample efficiency. The character state and action description, and the environment and reward setup are elaborated in the supplementary material.

Algorithm 4.1 STRIDE inference step (behavior index k)

```
Input: proprioceptive state window $\mathbf{s}_{t-n:t} \in \mathbb{R}^{(n+1)\times d_x}$, task goal $\mathbf{g}_t \in \mathbb{R}^{d_g}$, frozen
       codebook slices $\{C_j\}_{j=1}^{k}$, current hidden state $\mathbf{h}_{t-1} \in \mathbb{R}^{256}$
Output: action $\mathbf{a}_t \in \mathbb{R}^{d_a}$, updated hidden state $\mathbf{h}_t$
// Dual encoders
$\mathbf{z}_t^{\text{state}}, \mathbf{h}_t \leftarrow$ GRU-MLP_state$(\mathbf{s}_{t-n:t}, \mathbf{h}_{t-1})$        // $\mathbf{z}_t^{\text{state}} \in \mathbb{R}^{128}$
$\mathbf{z}_t^{\text{task}} \leftarrow$ MLP_task$(\mathbf{g}_t)$        // $\mathbf{z}_t^{\text{task}} \in \mathbb{R}^{128}$
$\mathbf{z}_t^{\text{base}} \leftarrow [\mathbf{z}_t^{\text{state}}, \mathbf{z}_t^{\text{task}}] \in \mathbb{R}^{256}$
// Composer
$\boldsymbol{\ell}_t \leftarrow$ MLP_composer$(\mathbf{z}_t^{\text{base}}) \in \mathbb{R}^{k}$
$\mathbf{w}_t \leftarrow \text{softmax}(\boldsymbol{\ell}_t)$        // $\sum_{j=1}^{k} w_{t,j} = 1$
// Codebook query
$\mathbf{e}^{\star} \leftarrow \arg\max_{\mathbf{e}\in C_{\le k}} \frac{\langle \mathbf{z}_t^{\text{task}}, \mathbf{e}\rangle}{\|\mathbf{z}_t^{\text{task}}\|_2 \|\mathbf{e}\|_2}$
// Per-behavior heads
// log (ensures $\sigma > 0$ after exponentiation)
for $j \leftarrow 1$ to $k$ do
    $(\boldsymbol{\mu}_t^{(j)}, \log \boldsymbol{\sigma}_t^{(j)}) \leftarrow D_j(\mathbf{e}^{\star}, \mathbf{h}_t)$        // $D_j$ frozen if $j < k$
end for
// Mixture of experts
$\boldsymbol{\mu}_t \leftarrow \sum_{j=1}^{k} w_{t,j} \boldsymbol{\mu}_t^{(j)}$
$\boldsymbol{\sigma}_t \leftarrow \boldsymbol{\sigma}_t^{(k)}$
// Sample and execute
$\mathbf{a}_t \sim \mathcal{N}(\boldsymbol{\mu}_t, \text{diag}(\boldsymbol{\sigma}_t^2))$
return $(\mathbf{a}_t, \mathbf{h}_t)$
```

4.5.1 Setup

We use a 15-link humanoid with 28 actuated DoFs (Xu *et al.*, 2023a) and train the eight behaviors from locomotion styles introduced in Section 4.4: {Run, Jaunty-Walk, Stomp-Walk, Crouch-Walk, Limp-Walk, Stoop-Walk, Joyful-Walk, Walk}. For style discriminators, we use 1–3 s clips re-targeted from the LAFAN1 mocap set (Harvey, Yurick, Nowrouzezahrai & Pal, 2020).



a) Codeword 1: Turn Right b) Codeword 2: Straight Sprint c) Codeword 3: Turn Left

Figure 4.3 Codeword-level primitives for the running slice. In inference analysis, each codeword shows distinct motion primitives (in this case, 3) that together span the behavior (running): (a) a left-turn with inside-foot shortening, (b) a straight sprint, and (c) a symmetric right turn. A single slice of the codebook decomposes the high-speed gait into human-interpretable sub-skills, matching the analysis in Section 4.6.3

We use PPO (Wang *et al.*, 2020d) with hyperparameters as shown in Table 4.1. Training roll-outs employ 512 parallel IsaacGym environments; each PPO epoch processes $H \times N_{\text{env}} = 4096$ transitions. A single base skill trains in ~ 1.4h on an NVIDIA 4070. Adding a new behavior (frozen slices + new composer) gradually takes less time to optimize, and although the number of slices grows linearly, we find the computational overhead negligible compared to environment rollouts, with no measurable degradation in performance across curricula of up to 8–10 skills.

Table 4.1 Optimization hyper-parameters

Group	**Symbol / name**	**Value**
Rollout	Horizon	$H = 8$
	Parallel environments	$N_{\text{env}} = 512$
Mini-batch	Batch size	$B = 256$
	PPO epochs	$E = 5$
Optimizer	Actor/critic learning rate	$\alpha_{a,c} = 10^{-4}$
	Discriminator learning rate	$\alpha_{\mathcal{D}} = 10^{-5}$
Discounting	Discount factor	$\gamma = 0.95$
	GAE parameter	$\lambda = 0.95$
Runtime	Max training epochs	2 500
	Codeword dimension	$d_c = 128$
	Termination reward	$r_{\text{min}} = -1$

4.6 Discussion

STRIDE introduces a continual learning framework that enables speed-up over existing RL locomotion pipelines, while additionally contributing towards sample efficiency, interpretability, and explainability.

4.6.1 Sample Efficiency

Fig. 4.5 quantifies how the policy quantization using a discrete codebook of STRIDE accelerates the acquisition of *new* skills compared with two strong baselines: AMP (Peng *et al.*, 2021) and Composite-Motion (Xu *et al.*, 2023a). The results show the *speed-up* ratio—baseline training epochs divided by STRIDE epochs—so values above the dashed line indicate positive forward transfer.

4.6.1.1 STRIDE versus AMP (Fig. 4.5a- 4.5b.)

We evaluate two curricula that differ in the order of training the behavior.

- **Order–1 (root = Run).** STRIDE starts slower than AMP on the *very first* stylized (*Jaunty*), pauses at parity on *Stomp*, and then accelerates monotonically, reaching a ×2.6 advantage by the eighth behavior (*Walk*). The upward trend confirms that each learnt behavior makes the next behavior cheaper when the curriculum transforms from energetic to slow gaits.
- **Order–2 (root = Walk).** When the root is the slow gait, the first behavior (*Run*) still gains a modest ×1.1 benefit by re-using plantar-flexion primitives from *Walk*. Subsequent low-stance gaits (*Crouch*, *Stoop*) show the largest gains ($\times 1.3-1.6$), because the encoder can recycle the knee-flexion strategies already present in *Walk*. The final stylized skills plateau around ×1.3, giving an overall geometric mean speed-up of ×1.36 across the sequence.

Taken together, the two experiments demonstrate *order-robust positive transfer*: regardless of which skill seeds the codebook, every later behavior learns faster than AMP, and the advantage compounds as the curriculum lengthens.

4.6.1.2 Comparison with Xu *et al.* (2023a) (Fig. 4.5c).

Composite Motion (Xu *et al.*, 2023a) models policies for learning multiple tasks simultaneously by training a dedicated multi-policy controller. While effective for a small number of tasks, CM scales poorly as the number of behaviors increases: every new skill requires retraining the shared controller, which leads to longer adaptation times and potential interference between unrelated tasks.

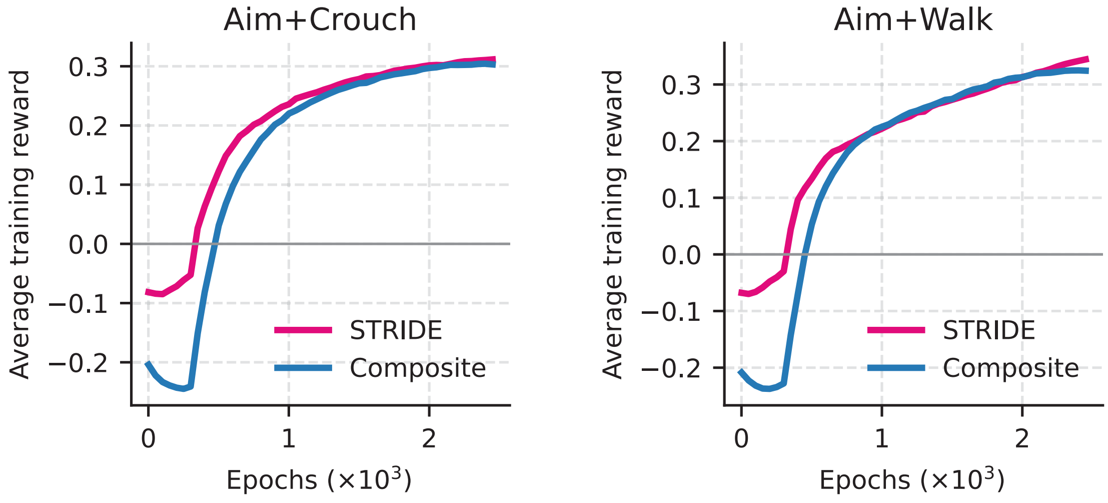
Table 4.2 Comparison of training efficiency between CM and STRIDE for composite motion tasks with base policy adaptation

Task	Base	CM epochs	STRIDE epochs
Aim + Run	—	800	940
Aim + Crouch	Run	520	370
Aim + Walk	Run	490	340

By contrast, STRIDE takes a modular approach: once a skill slice is added to the codebook, it is frozen and reused for subsequent tasks. This design means that the computational cost of adding a new skill grows only with the lightweight encoder/decoder for that skill, rather than retraining a large joint policy. As the curriculum expands, STRIDE avoids the combinatorial explosion of retraining all task mixtures, making it substantially more scalable for long-horizon or multi-skill settings.

Practically, this scalability manifests in faster adaptation (30–45% reduction in training epochs compared to CM) and smoother transitions, since the system composes skills from a stable and reusable dictionary instead of re-optimizing them jointly. In short, CM couples all behaviors into a single controller, whereas STRIDE accumulates and reuses them as modular building blocks — making it more suitable for lifelong continual learning.

For qualitative results, please refer to the supplementary video (link).



a) Aim + Crouch (2nd Behaviour)

b) Aim + Run (3rd Behaviour)

Figure 4.4 STRIDES start by training for the task of *Aim* + *Run* as the base controller; curves show the first 2.5×10^3 epochs required to adapt to (a) *Aim* + *Crouch* and (b) *Aim* + *Walk*. STRIDE (magenta) climbs out of the negative-reward region earlier. It reaches the success threshold roughly 40–50 % sooner than the Composite motion (Xu *et al.*, 2023a) baseline (blue), demonstrating stronger positive forward transfer from the base skill. Final rewards converge to similar values, indicating that the efficiency gain does not impact performance

4.6.2 Interpretability

Fig. 4.6 stacks two *skill-reuse matrices*, one for each curriculum order. Rows list the behavior that is currently being learned; columns list the *earlier* behaviors whose codewords are already frozen. An entry $w_{i\leftarrow j} = 0.71$ therefore reads “while training behavior i, the controller routes 71% of its actions through the skills learned for behavior j.” Scanning a row from left to right tells us which behavior’s skill dominates the learning of the new behavior.

Sparse mixtures. In both curricula, the average row-wise Gini coefficient is high (Order 1: 0.76, Order 2: 0.74), meaning that the composer relies on at most two behaviors at any point.

Consistent clusters. Hierarchical clustering of the rows yields the same two groups in both orders: (i) upright, stylized walks {*Jaunty*, *Stomp*, *Joyful*, *Limp*} and (ii) low-stance gaits {*Crouch*, *Stoop*}. Cluster boundaries are stable even when the curriculum order is reversed, indicating that the codebook captures genuine kinematic similarity, not ordering bias.

Direction of transfer. When *Run* is the first skill (Order 1), its slice feeds the entire curriculum: later stylized walks draw 64–81% of their skills from running, confirming that a fast gait is a kinematic *superset*. Stylized walks borrow heavily from *Walk* (35–61%) but little from each other (≤ 25%), suggesting that they add style-specific residuals on top of a shared foot-strike template.

In summary, freezing each 16-codeword slice does *not* hinder future learning; the earliest slice often becomes a backbone for later skills. The composer's sparse, directional weights produce a lineage that matches biomechanical intuition and makes the policy easy to explain in human terms.

4.6.3 Explainability

An exciting aspect of our approach is the emergence of derivative motion primitives. Namely, each codebook slice models a high-level, human-understandable basic motion action. For example, for *Run* the system generates exactly three code words which can be interpreted easily as a "straight-sprint" codeword and a pair of symmetric fast left and right turns as shown in Fig. 4.3. The system generates these primitives automatically, and they are viewable in the accompanying video. We can interpret this in the context of long-range, target-reaching as a symbolic sequence, such as run straight → left-turn → run straight. All gaits exhibit similar patterns with a handful of codewords, each an explainable motion controller on its own. In the video, we showcase two gaits, *Crouch* and *Walk*, with each allocating four codewords. We can interpret the codewords as the following: forward stepping, left/right turns, and a precision step, which is used mainly for fine alignment near a given goal.

Because every codeword is frozen once learned, these primitives are stable and can be inspected—— or even swapped—— without retraining. Thus, we see a strength in the STRIDE approach to not build a black-box policy, but a dictionary of basic interpretable skills —

something baselines such as AMP or Composite Motion cannot model because of the continuous representation for skill modelling. We believe that this development may open the door to human-in-the-loop editing, safety auditing, and task-specific re-targeting by codeword surgery in the future.

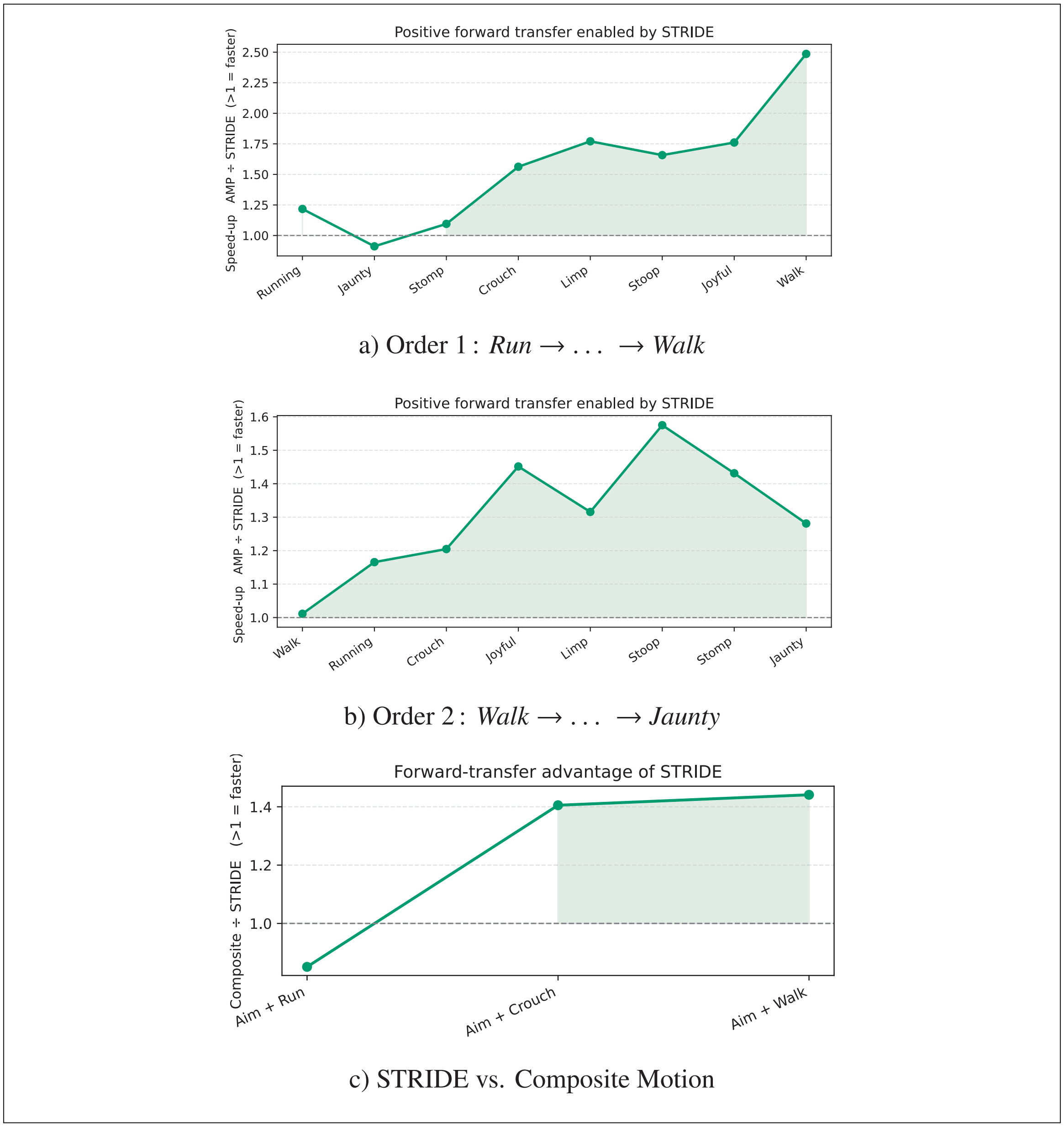
4.7 Conclusions

We introduced STRIDE, a continual-learning framework that quantizes a policy space into a growing dictionary of motion primitives. By freezing each codeword slice once learned and allowing subsequent behaviors to *compose* these slices through a lightweight gating network, STRIDE shows improved sample-efficiency, transparency, and scalability as a unified controller. Across eight locomotion behaviors, STRIDE leads to a $\mathbf{2.6}\times$ reduction in optimization epochs relative to AMP and a $30\% - 45\%$ reduction in adaptation time over the multi-policy composite motion baseline. Moreover, our approach allows the modelled policy to be interpretable and explainable. For any downstream behavior, the skills used from the previous behavior can be easily interpreted. Since the codewords model high-level skills, it is easy to interpret what strategy was used for completing the task.

4.8 Limitations

Our fixed quota of 16 codewords per behavior was chosen empirically; adaptive quotas informed by information-theoretic criteria could yield leaner dictionaries. Freezing codeword slices prevents forgetting but also forbids post-hoc refinement. If later tasks expose systematic errors in an early slice, STRIDE must either add corrective codewords or accept sub-optimal reuse. Incorporating a low-temperature, gated fine-tuning phase—akin to elastic weight consolidation that can mitigate this rigidity without sacrificing stability.

Moreover, our experiments are confined to proprioceptive states and low-dimensional goal signals. Extending to more complex goals such as multi-step objectives, semantic tasks, or goals specified in natural language that would likely require hierarchical or compositional



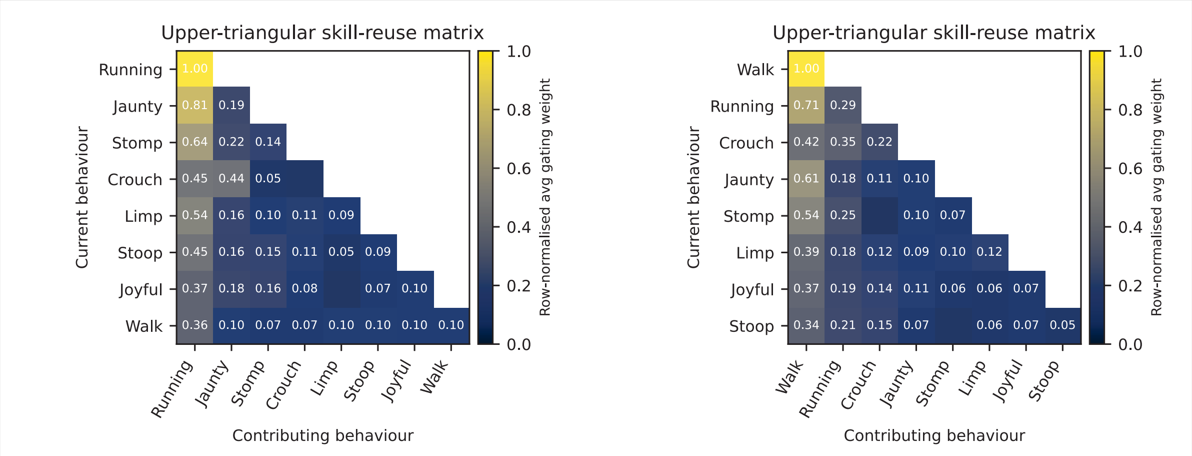
a) Order 1 : *Run* → . . . → *Walk*

b) Order 2 : *Walk* → . . . → *Jaunty*

c) STRIDE vs. Composite Motion

Figure 4.5 Relative sample-efficiency of STRIDE. Comparing *speed-up ratios* (baseline epochs ÷ STRIDE epochs); values above the dashed line indicate faster learning, hence positive forward transfer. (a,b) Against AMP, STRIDE accelerates convergence by up to **2.6**× on later skills, regardless of curriculum order. (c) Against the multi-policy Composite Motion baseline, STRIDE still achieves **1.45**× faster adaptation while adding <1 % parameters per new behavior

use of codewords. Because STRIDE ultimately operates on discrete tokens, it is in principle modality-agnostic; visual observations or language prompts could select or weight codewords



a) Curriculum **Order 1**: *Run* → . . . → *Walk*

b) Curriculum **Order 2**: *Walk* → . . . → *Stoop*

Figure 4.6 Skill reuse across the curriculum. Rows index the behavior currently being trained; columns index *earlier* behaviors whose codeword slices are frozen. Each entry is the row-normalized average gating weight $w_{i \leftarrow j}$ accumulated over the full 300-step segment. Hot colours, therefore, quantify how strongly behavior i relies on motion primitives inherited from behavior j. Comparing (a) and (b) reveals how swapping the root skill changes the lineage while preserving cluster structure

just as well. Scaling the framework to multimodal, structured, and long-horizon goals therefore remains an exciting direction for future work.

GENERAL DISCUSSION AND CONCLUSION

This dissertation has established that reinforcement-learning agents achieve markedly superior data-efficiency, transferability, and transparency when equipped with learnable, discrete, and adaptive priors—fundamentally challenging the field's prevailing tabula-rasa paradigm. Through three complementary works—automatic reward construction for heavy-equipment automation (Chapter 2), token-based world-model learning (Chapter 3), and dynamic codebook growth for continual skill acquisition (Chapter 4)—we have demonstrated that structured priors are not merely helpful but essential for practical RL deployment.

Our empirical results conclusively show that:

- Prior-guided learning reduces sample complexity compared to tabula-rasa baselines
- Discrete representations enable interpretable decision-making while maintaining competitive performance
- Adaptive codebooks successfully balance continual learning with catastrophic forgetting prevention

This chapter synthesizes these findings to articulate the broader implications: the future of efficient RL lies not in better algorithms for learning from scratch, but in better methods for incorporating, adapting, and growing structured knowledge representations.

Synthesis of Findings

A unifying pattern emerges when the three works are viewed side by side. Each begins with a standard RL problem that exhibits prohibitive sample complexity and opaque behaviour: sparse reward learning for excavator control, high-dimensional pixel-based Atari games, and long-horizon locomotion curricula. Each then injects prior structure at a different point in the learning loop. Yet, all converge on the same outcome: a substantial reduction in environment interaction, a measurable boost in generalization to novel tasks or settings, and an increased capacity for human inspection.

In Chapter 2, the prior takes the form of a *learned evaluation metric*. By training a score predictor to emulate expert judgements, we replaced post-hoc grading with dense, shaped feedback available at every time step. The practical effect was two-fold. First, policy search became less exploratory and more goal-directed, reducing the interaction budget relative to a sparse-reward baseline. Second, the same predictor doubled as a diagnostic tool: mismatches between predicted and actual scores pointed directly to failure modes in perception or control. Crucially, the score predictor was not a fixed hand-engineered rubric; it was itself learned from data, adapting as additional demonstrations accumulated.

Chapter 3 pushed the notion of discreteness into the world model. The DART architecture tokenised observations, actions, and returns into tokens and trained a transformer to model their joint distribution. This decision had three benefits. Tokens served as compressed, reusable fragments of world dynamics; they consistently reappeared across trajectories and games, enabling transformers to exploit their sequential structure for long-range dependencies, while planners could mix and match them in ways not possible with continuous vectors. Compared with latent-contiguous baselines, DART achieved higher performance with less than one-tenth of the data and produced roll-out sequences that could be inspected token by token, revealing which events or actions the agent deemed salient at each decision point.

Chapter 4 addressed the continual-learning setting, where the prior must not merely exist but also *grow*. STRIDE's vector-quantized variational auto-encoder begins with a modest codebook of motor primitives and expands it only when evidence suggests the current vocabulary is insufficient to explain new trajectories. Because codewords are immutable once created, previously solved tasks remain perfectly reconstructible, eliminating catastrophic forgetting without resorting to rehearsal buffers or regularization tricks. Moreover, the sequence of invoked codewords forms a time-indexed "skill trace" that human operators can read to understand what the agent believes it is doing at each instant. The approach delivered up to a two-fold speed-up on locomotion benchmarks while preserving zero-shot competence on earlier skills.

Taken together, these studies validate the central claim that priors are not optional decorations but essential infrastructure. A prior can manifest as a reward, a model, or a skill library; it can be learned, discretized, and adapted; and when these properties align, learning becomes faster, reuse becomes natural, and interpretation becomes feasible.

Implications for Efficient Reinforcement Learning

The empirical gains reported herein invite a fresh look at how progress in reinforcement learning should be quantified. While the community rightly continues to track headline numbers such as final episodic return, these experiments reveal three complementary dimensions—*interaction cost*, *transfer surface*, and *explanation bandwidth*—that are equally diagnostic of an algorithm's real-world utility.

Interaction cost: Sample efficiency is best understood not by the raw number of environment steps an agent consumes, but by how many *steps it avoids* when equipped with effective priors. Reporting the gap between a minimally informed baseline and a prior-rich variant makes explicit the degree to which domain knowledge has been encoded and leveraged. Such a differential view turns sample efficiency from an absolute statistic into a measure of knowledge reuse.

Transfer surface: Discrete latent spaces create a compositional interface through which skills, behaviours, or sub-tasks learned in one setting can be spliced into another with only lightweight fine-tuning. Future benchmarks should therefore organize tasks into systematic families that probe generalization across changes in dynamics, morphology, or reward structure. By rewarding agents that succeed with minimal additional data, these benchmarks would promote methods that treat prior experience as a first-class asset rather than starting over on every new problem.

Explanation bandwidth: Because discrete priors are expressed in tokens that resemble symbols more than activation patterns, an agent's internal reasoning becomes substantially more transparent. Engineers and domain experts can inspect the resulting codebooks, activation sequences, or learned program sketches to audit decisions and pinpoint failure modes. In safety-critical domains, this ability to perform targeted, post-hoc debugging is indispensable and remains largely out of reach for agents that rely purely on opaque continuous representations.

Taken together, these axes encourage the field to move away from ever larger networks trained with ever larger budgets and toward leaner, more reusable solutions—solutions whose efficiency is measured not only in returns but in how economically they interact, how broadly they generalize, and how clearly they explain themselves. Adopting such multidimensional reporting standards would help ensure that future advances translate into practical, trustworthy systems rather than merely bigger numbers on isolated leaderboards.

Limitations

Despite encouraging results, each works have limitations that must be acknowledged.

First, the reward predictor in Chapter 2 relies on the availability of high-quality demonstrations. In domains where expert data are scarce or expensive, bootstrapping the prior may itself be costly. Second, tokenization in Chapter 3 was learned in a relatively simple Atari environment; whether tokens modelled in the case of complex real-life environments can match the same sample efficiency remains an open question. Third, STRIDE assumes that fixed-length codewords well capture the granularity of skills; tasks requiring hierarchical or variable-duration primitives may demand more flexible encodings.

A broader limitation spans all studies: *discreteness does not guarantee optimality*. While tokens and codewords are tractable for planning and interpretation, they may under-approximate the actual solution space, particularly in high-precision control tasks. Balancing expressivity against interpretability is therefore an ongoing design challenge.

Future Work

Building on the present work's limitations, we outline four complementary avenues that could significantly broaden the scope and practicality of token-based reward shaping.

Adaptive Demonstration Gathering

Rather than collecting a fixed, potentially redundant set of expert roll-outs, **active-learning protocols** can estimate epistemic uncertainty in the reward prior and query annotators only for trajectories whose expected information gain is high. Concretely, Bayesian acquisition functions (e.g., BALD or variance reduction) could be combined with bandit-style budget allocation to decide *when* to request help, *which* states to present, and even *which expert* to consult when multiple levels of proficiency are available. Such targeted queries promise lower annotation cost without sacrificing—indeed, often improving—the fidelity of the shaped reward.

Online Token Discovery

Extending DART with a non-parametric component that spawns new tokens on-the-fly would blur the line between representation learning and control. The agent could start with a minimal vocabulary and, when faced with novel perceptual clusters or action motifs, propose additional tokens whose semantics are refined through continued interaction. This dynamic lexicon is particularly attractive for **long-horizon, open-world settings** such as autonomous driving, where unforeseen manoeuvres (e.g., rare traffic scenarios) emerge continually.

Hierarchical Codebooks

Flat codebooks force a trade-off between detailed low-level actions and abstract strategy. Introducing *nested or variable-length* codewords—akin to byte-pair encoding in NLP—would allow fine motor tweaks and coarse tactical decisions to coexist within a single prior. A hierarchical STRIDE-like framework could then transfer high-level behaviours across tasks while fine-tuning lower layers to environment-specific dynamics, yielding both data efficiency and interpretability.

A fourth theme concerns *human editability*. If the learned priors are truly discrete and intelligible, domain experts should be able to **insert, remove, or rewrite** tokens and codewords directly—much like editing a vocabulary list. Such manual intervention can inject commonsense constraints, correct failure modes observed in deployment, and accelerate alignment with evolving requirements, creating a tight feedback loop between human insight and machine optimization.

Concluding Remarks

This work has argued, and empirically demonstrated, that *improved prior modelling* is a powerful lever for efficient, general, and transparent reinforcement learning. By turning rewards, dynamics, and skills into discrete, adaptive vocabularies, we transformed three otherwise intractable problems into manageable ones, achieving substantial reductions in sample complexity, seamless transfer across tasks, and visibility into the agent's internal reasoning. While limitations remain, the future of RL lies not in ever larger unstructured networks but in systems whose knowledge is organized, expandable, and intelligible. As RL continues its march from game boards and simulators into real-world applications, such features will be indispensable.

RECOMMENDATIONS

The results presented in this work support the hypothesis: when reinforcement-learning agents are supplied with priors that are *learnable, discrete, and adaptive*, data efficiency improves, generalization widens, and decision processes become more transparent. The results span three domains—heavy-equipment automation, pixel-based Atari control, and continual locomotion learning—and show consistent gains in sample complexity, transfer performance, and interpretability. Translating these findings into broader impact, however, demands concrete recommendations for both the research community and practitioners who deploy RL systems in the wild. This closing chapter, therefore, distills a set of actionable guidelines that follow naturally from the results, and it highlights the open problems whose resolution is likely to yield the most significant returns.

Improved Prior Modelling

A central lesson is that priors deserve the same design attention as neural architectures and optimization algorithms. Researchers should document the structure and learning dynamics of their priors with the same precision currently reserved for network depth, layer normalization, or learning-rate schedules. Practitioners, in turn, should allocate engineering effort to maintaining and evolving prior libraries, treating them as reusable assets rather than experiment-specific artifacts. Whether the prior appears as a reward predictor, a tokenizer, or a codebook of motor primitives, version control and continuous integration will ensure that improvements propagate across projects rather than being rediscovered ad hoc.

Measure Efficiency, Transfer, and Transparency Jointly

Traditional benchmarks emphasize final return, yet Chapters 2–4 reveal that returns alone obscure critical trade-offs. Future empirical studies should therefore report three companion metrics: *interaction cost*, the number of environment steps required to reach a given performance threshold; *transfer surface*, the extent to which representations or policies accelerate learning on novel but related tasks; and *explanation bandwidth*, the fraction of decision steps whose intermediate representations can be mapped to human-interpretable concepts. Only by viewing these axes together can the field chart genuine progress toward agents that are not merely strong but also data-thrifty and intelligible.

Design Data Collection Around Prior Growth

The reward predictor of Chapter 2 and the codebook expansion mechanism of Chapter 4 both rely on targeted data to refine or extend their priors. Data-collection strategies should therefore be *prior-aware*: simulation scenarios, human demonstrations, or curriculum modules ought to be selected with the explicit aim of filling gaps in the current prior, rather than maximizing aggregate experience. Active-learning criteria—uncertainty, disagreement, or novelty—can flag regions where the prior is fragile, guiding budget-constrained data acquisition toward maximal leverage.

Integrate Human-Editable Representations

Discrete tokens and codewords provide natural hooks for human oversight. Engineering teams should expose simple interfaces that allow domain experts to inspect, rename, or disable individual components of a prior without retraining the entire agent. For example, safety auditors might mark a subset of skill codewords as forbidden in specific operational modes, or designers could merge synonymous tokens to simplify downstream analysis. Such editability fosters a virtuous cycle in which machine-generated structure invites human refinement, which in turn feeds more structured learning.

Plan for Continual Expansion and Maintenance

The dynamic nature of priors implies a maintenance burden analogous to that of any growing knowledge base. Regular "prior audits"—scheduled evaluations that probe coverage, redundancy, and drift—should be incorporated into development pipelines. Tooling for visualizing token co-occurrence, codeword usage over time, and reward-predictor calibration will help teams decide when to prune stale elements or to introduce higher-level abstractions. Neglecting such steps risks modelling priors that hamper both efficiency and interpretability.

Multi-Modal and Hierarchical Priors

While this work focused on visual, proprioceptive, and scalar reward signals, real-world tasks increasingly involve language, audio, and structured data. Extending the principles of discreteness and adaptivity to multi-modal settings is a ripe area for exploration. Hierarchical priors that couple low-level motor primitives with high-level symbolic goals could bridge the gap between reactive control and long-horizon planning, enabling agents to reason fluidly across timescales and modalities.

Ethical and Regulatory Considerations

Improved priors do not eliminate ethical challenges; instead, they surface new questions about provenance, bias, and control. Reward predictors trained on historical demonstrations inherit the values implicit in those demonstrations, which may embed undesirable biases. Tokenizers and codebooks reflect the distribution of training environments and may fail in corner cases. Developers must therefore adopt rigorous validation protocols—similar in spirit to fairness audits in supervised learning—before deploying prior-driven agents in settings where safety or equity is paramount. Clear documentation of data sources, prior structures, and known failure modes will facilitate regulatory approval and public trust.

Future Research Priorities

Despite encouraging progress, several scientific challenges remain unresolved. Online token discovery with theoretical guarantees, hierarchical codebook management that balances expressivity with tractability, and methods for fusing human-written symbolic knowledge with learned priors are promising directions. In addition, formal frameworks that quantify the capacity of a discrete prior to represent task families could guide principled prior sizing, avoiding both under-fitting and wasteful over-parametrization. Finally, integrating uncertainty estimation directly into prior-growth mechanisms may yield agents that not only learn quickly but also recognize when they are out of distribution and request human intervention.

Closing Remark

The recommendations above rest on a simple idea: as reinforcement learning matures from a laboratory curiosity into a backbone of autonomous technology, the community must shift its collective focus from raw representational capacity to structured, revisable, and accountable knowledge. Learnable, discrete, and adaptive priors offer a pragmatic path toward that goal. By treating priors as first-class entities—carefully measured, actively maintained, and transparently governed—we can build agents that not only excel in benchmarks but also earn their place in real-world systems where efficiency, reliability, and human understanding are non-negotiable.

BIBLIOGRAPHY

Adiban, M., Siniscalchi, M., Stefanov, K. & Salvi, G. (2022). Hierarchical Residual Learning Based Vector Quantized Variational Autoencorder for Image Reconstruction and Generation. *33rd British Machine Vision Conference*.

Adiban, M., Stefanov, K., Siniscalchi, S. M. & Salvi, G. (2023). S-HR-VQVAE: Sequential Hierarchical Residual Learning Vector Quantized Variational Autoencoder for Video Prediction. *IEEE Transactions on Multimedia*, 27, 4321-4332.

Agarwal, R., Schwarzer, M., Castro, P. S., Courville, A. C. & Bellemare, M. (2021). Deep reinforcement learning at the edge of the statistical precipice. *Advances in neural information processing systems*, 34, 29304–29320.

Ali, Y. A., Awwad, E. M., Al-Razgan, M. & Maarouf, A. (2023). Hyperparameter Search for Machine Learning Algorithms for Optimizing the Computational Complexity. *Processes*, 11(2), 349.

Allahverdi, A. (2016). A survey of scheduling problems with no-wait in process. *Eur. J. Oper. Res.*, 255(3), 665–686.

AlMahamid, F. & Grolinger, K. (2021). Reinforcement Learning Algorithms: An Overview and Classification. *34th IEEE Canadian Conference on Electrical and Computer Engineering, CCECE 2021*, pp. 1–7.

Almási, P., Moni, R. & Gyires-Tóth, B. (2020). Robust Reinforcement Learning-based Autonomous Driving Agent for Simulation and Real World. *2020 International Joint Conference on Neural Networks (IJCNN)*, pp. 1-8. doi: 10.1109/IJCNN48605.2020.9207497.

Amini, M., Zhang, Z., Penmetsa, S., Zhang, Y., Hao, J. & Liu, W. (2022). Generalizable Floorplanner through Corner Block List Representation and Hypergraph Embedding. *KDD '22: The 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, August 14 - 18, 2022*, pp. 2692–2702.

Andersen, P.-A., Goodwin, M. & Granmo, O.-C. (2018). The dreaming variational autoencoder for reinforcement learning environments. *Artificial Intelligence XXXV: 38th SGAI International Conference on Artificial Intelligence, AI 2018, Cambridge, UK, December 11–13, 2018, Proceedings 38*, pp. 143–155.

Andersson, J., Bodin, K., Lindmark, D., Servin, M. & Wallin, E. (2021). Reinforcement Learning Control of a Forestry Crane Manipulator. *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2121-2126. doi: 10.1109/IROS51168.2021.9636219.

Andrychowicz, M., Raichuk, A., Stanczyk, P., Orsini, M., Girgin, S., Marinier, R., Hussenot, L., Geist, M., Pietquin, O., Michalski, M., Gelly, S. & Bachem, O. (2021). What Matters for On-Policy Deep Actor-Critic Methods? A Large-Scale Study. *9th International Conference on Learning Representations, ICLR*.

Andrychowicz, O. M., Baker, B., Chociej, M., Józefowicz, R., McGrew, B., Pachocki, J., Petron, A., Plappert, M., Powell, G., Ray, A., Schneider, J., Sidor, S., Tobin, J., Welinder, P., Weng, L. & Zaremba, W. (2020). Learning dexterous in-hand manipulation. *The International Journal of Robotics Research*, 39(1), 3-20. doi: 10.1177/0278364919887447.

Anuchitanukul, A. & Ive, J. (2022). SURF: Semantic-level Unsupervised Reward Function for Machine Translation. *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL*, pp. 4508–4522.

Anzalone, L., Barra, P., Barra, S., Castiglione, A. & Nappi, M. (2022). An End-to-End Curriculum Learning Approach for Autonomous Driving Scenarios. *IEEE Transactions on Intelligent Transportation Systems*, 23, 19817-19826.

Arthur, P., Cohn, T. & Haffari, G. (2021). Learning Coupled Policies for Simultaneous Machine Translation using Imitation Learning. *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, EACL 2021, Online, April 19 - 23, 2021*, pp. 2709–2719.

Arulkumaran, K., Deisenroth, M. P., Brundage, M. & Bharath, A. A. (2017). Deep Reinforcement Learning: A Brief Survey. *IEEE Signal Process. Mag.*, 34(6), 26–38.

Atkeson, C. G. & Santamaria, J. C. (1997). A comparison of direct and model-based reinforcement learning. *Proceedings of international conference on robotics and automation*, 4, 3557–3564.

Ausin, M. S., Maniktala, M., Barnes, T. & Chi, M. (2021). Tackling the Credit Assignment Problem in Reinforcement Learning-Induced Pedagogical Policies with Neural Networks. *Artificial Intelligence in Education - 22nd International Conference, AIED, Utrecht, Netherlands*, 12748, 356–368.

Ayoub, A., Jia, Z., Szepesvari, C., Wang, M. & Yang, L. F. (2020). Model-Based Reinforcement Learning with Value-Targeted Regression. *Conference on Learning for Dynamics & Control*.

Bahdanau, D., Cho, K. & Bengio, Y. (2015). Neural machine translation by jointly learning to align and translate. *International Conference on Learning Representations*.

Bai, B., Liang, J., Zhang, G., Li, H., Bai, K. & Wang, F. (2021a). Why Attentions May Not Be Interpretable? *KDD '21: The 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining,*, pp. 25–34.

Bai, Y., Mei, J., Yuille, A. L. & Xie, C. (2021b). Are Transformers more robust than CNNs? *NeurIPS*, pp. 26831–26843.

Bakker, M., Chadwick, M., Sheahan, H., Tessler, M., Campbell-Gillingham, L., Balaguer, J., McAleese, N., Glaese, A., Aslanides, J., Botvinick, M. et al. (2022). Fine-tuning language models to find agreement among humans with diverse preferences. *Advances in neural information processing systems*, 35, 38176–38189.

Banino, A., Badia, A. P., Walker, J. C., Scholtes, T., Mitrovic, J. & Blundell, C. (2022a). CoBERL: Contrastive BERT for Reinforcement Learning. *10th International Conference on Learning Representations, ICLR*.

Banino, A., Badia, A. P., Walker, J. C., Scholtes, T., Mitrovic, J. & Blundell, C. (2022b). CoBERL: Contrastive BERT for Reinforcement Learning. *The Tenth International Conference on Learning Representations, ICLR*.

Bao, H., Dong, L., Piao, S. & Wei, F. (2022). BEiT: BERT Pre-Training of Image Transformers. *10th International Conference on Learning Representations, ICLR*.

Baroglio, C., Giordana, A., Piola, R., Kaiser, M. & Nuttin, M. (1996). Learning controllers for industrial robots. *Machine learning*, 23, 221–249.

Barreto, A., Dabney, W., Munos, R., Hunt, J. J., Schaul, T., van Hasselt, H. P. & Silver, D. (2017). Successor features for transfer in reinforcement learning. *Advances in Neural Information Processing Systems*, pp. 4055–4065.

Barto, A. G. & Mahadevan, S. (2003). Recent Advances in Hierarchical Reinforcement Learning. *Discret. Event Dyn. Syst.*, 13(1-2), 41–77.

Bastani, O., Inala, J. P. & Solar-Lezama, A. (2020). Interpretable, Verifiable, and Robust Reinforcement Learning via Program Synthesis. *xxAI - Beyond Explainable AI - International Workshop, Held in Conjunction with ICML*, 13200, 207–228.

Baxter, J. (2000). A model of inductive bias learning. *Journal of artificial intelligence research*, 12, 149–198.

Bellemare, M. G., Naddaf, Y., Veness, J. & Bowling, M. (2013). The arcade learning environment: An evaluation platform for general agents. *Journal of Artificial Intelligence Research*, 47, 253–279.

Bellemare, M. G., Naddaf, Y., Veness, J. & Bowling, M. (2015). The Arcade Learning Environment: An Evaluation Platform for General Agents (Extended Abstract). *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina, July 25-31, 2015*, pp. 4148–4152.

Bellman, R. (1954). Some Applications of the Theory of Dynamic Programming - A Review. *Oper. Res.*, 2(3), 275–288.

Bengio, Y., Louradour, J., Collobert, R. & Weston, J. (2009). Curriculum Learning. *Proceedings of the 26th Annual International Conference on Machine Learning*, (ICML '09), 41–48. doi: 10.1145/1553374.1553380.

Bergstra, J. & Bengio, Y. (2012a). Random Search for Hyper-Parameter Optimization. *J. Mach. Learn. Res.*, 13, 281–305.

Bergstra, J. & Bengio, Y. (2012b). Random Search for Hyper-Parameter Optimization. *J. Mach. Learn. Res.*, 13, 281-305.

Bertsekas, D. P. (2010). Pathologies of temporal difference methods in approximate dynamic programming. *49th IEEE Conference on Decision and Control (CDC)*, pp. 3034–3039.

Bhattamishra, S., Patel, A. & Goyal, N. (2020). On the Computational Power of Transformers and Its Implications in Sequence Modeling. *Proceedings of the 24th Conference on Computational Natural Language Learning, CoNLL 2020, Online, November 19-20, 2020*, pp. 455–475.

Bhutto, A. B., Vu, X., Elmroth, E., Tay, W. P. & Bhuyan, M. H. (2022). Reinforced Transformer Learning for VSI-DDoS Detection in Edge Clouds. *IEEE Access*, 10, 94677–94690.

Boysen, G. A. & Vogel, D. L. (2009). Bias in the classroom: Types, frequencies, and responses. *Teaching of Psychology*, 36(1), 12–17.

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I. & Amodei, D. (2020). Language Models are Few-Shot Learners. *NeurIPS*.

Buckman, J., Hafner, D., Tucker, G., Brevdo, E. & Lee, H. (2018). Sample-efficient reinforcement learning with stochastic ensemble value expansion. *Advances in neural information processing systems*, 31.

Bulatov, A., Kuratov, Y. & Burtsev, M. (2022). Recurrent memory transformer. *Advances in Neural Information Processing Systems*, 35, 11079–11091.

Burgess, A., van Diggele, C., Roberts, C. & Mellis, C. (2020). Feedback in the clinical setting. *BMC Medical Education*, 20. doi: 10.1186/s12909-020-02280-5.

Cao, S., Yin, Y., Huang, L., Liu, Y., Zhao, X., Zhao, D. & Huang, K. (2023). Efficient-vqgan: Towards high-resolution image generation with efficient vision transformers. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 7368–7377.

Cartuyvels, R., Spinks, G. & Moens, M.-F. (2021). Discrete and continuous representations and processing in deep learning: Looking forward. *AI Open*, 2, 143–159.

Chaplot, D. S., Gandhi, D., Gupta, S., Gupta, A. & Salakhutdinov, R. (2020). Learning To Explore Using Active Neural SLAM. *8th International Conference on Learning Representations, ICLR*.

Charlesworth, H. & Montana, G. (2020). Plangan: Model-based planning with sparse rewards and multiple goals. *Advances in Neural Information Processing Systems*, 33, 8532–8542.

Chefer, H., Gur, S. & Wolf, L. (2021). Transformer Interpretability Beyond Attention Visualization. *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pp. 782–791.

Chen, C., Yoon, J., Wu, Y.-F. & Ahn, S. (2021a). TransDreamer: Reinforcement Learning with Transformer World Models. *Deep RL Workshop NeurIPS 2021*.

Chen, L., Lu, K., Rajeswaran, A., Lee, K., Grover, A., Laskin, M., Abbeel, P., Srinivas, A. & Mordatch, I. (2021b). Decision transformer: Reinforcement learning via sequence modeling. *Advances in Neural Information Processing Systems*, 34, 15084–15097.

Chen, R. & Tan, Y. (2023). Credit assignment with predictive contribution measurement in multi-agent reinforcement learning. *Neural Networks*, 164, 681-690. doi: https://doi.org/10.1016/j.neunet.2023.05.021.

Chen, R., Li, W. & Yang, H. (2023). A Deep Reinforcement Learning Framework Based on an Attention Mechanism and Disjunctive Graph Embedding for the Job-Shop Scheduling Problem. *IEEE Trans. Ind. Informatics*, 19(2), 1322–1331.

Chen, T., Xu, J. & Agrawal, P. (2022a). A System for General In-Hand Object Re-Orientation. *Proceedings of the 5th Conference on Robot Learning*, 164(Proceedings of Machine Learning Research), 297–307.

Chen, W., Qiu, X., Cai, T., Dai, H., Zheng, Z. & Zhang, Y. (2021c). Deep Reinforcement Learning for Internet of Things: A Comprehensive Survey. *IEEE Commun. Surv. Tutorials*, 23(3), 1659–1692.

Chen, X., Toyer, S., Wild, C., Emmons, S., Fischer, I., Lee, K.-H., Alex, N., Wang, S. H., Luo, P., Russell, S., Abbeel, P. & Shah, R. (2021d). An Empirical Investigation of Representation Learning for Imitation. *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.

Chen, Y., der Merwe, M. V., Sipos, A. & Fazeli, N. (2022b). Visuo-Tactile Transformers for Manipulation. *Conference on Robot Learning, CoRL*, 205, 2026–2040.

Cho, K., van Merrienboer, B., Bahdanau, D. & Bengio, Y. (2014a). On the Properties of Neural Machine Translation: Encoder-Decoder Approaches. *Proceedings of SSST@EMNLP 2014, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation, Doha, Qatar, 25 October 2014*, pp. 103–111.

Cho, K., van Merrienboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H. & Bengio, Y. (2014b). Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1724–1734.

Choi, K., Hawthorne, C., Simon, I., Dinculescu, M. & Engel, J. H. (2020). Encoding Musical Style with Transformer Autoencoders. *Proceedings of the 37th International Conference on Machine Learning, ICML*, 119, 1899–1908.

Christiano, P. F., Leike, J., Brown, T. B., Martic, M., Legg, S. & Amodei, D. (2017). Deep Reinforcement Learning from Human Preferences. *NeurIPS*, pp. 4299–4307.

Clegg, A., Tan, J., Turk, G. & Liu, C. K. (2018). Animating human dressing. *ACM Transactions on Graphics (TOG)*, 37(4), 179:1–179:14.

Clevert, D.-A., Unterthiner, T. & Hochreiter, S. (2016). Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs). *International Conference on Learning Representations (ICLR)*.

CM-Labs Simulations Inc. (2024). Vortex Studio (Version 2024.1). Montreal, Quebec, Canada: CM-Labs Simulations Inc. Retrieved from: https://www.cm-labs.com/vortex-studio/.

Co-Reyes, J., Liu, Y., Gupta, A., Eysenbach, B., Abbeel, P. & Levine, S. (2021). Recursive classification of compositional structure in neural networks. *International Conference on Machine Learning*, pp. 2060–2070.

Corneil, D., Gerstner, W. & Brea, J. (2018). Efficient model-based deep reinforcement learning with variational state tabulation. *International Conference on Machine Learning*, pp. 1049–1058.

Cornia, M., Stefanini, M., Baraldi, L. & Cucchiara, R. (2020). Meshed-Memory Transformer for Image Captioning. *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR*, pp. 10575–10584.

Czarnecki, W. M., Pascanu, R., Osindero, S., Jayakumar, S., Swirszcz, G. & Jaderberg, M. (2019, 16–18 Apr). Distilling Policy Distillation. *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*, 89(Proceedings of Machine Learning Research), 1331–1340.

Dai, Z., Cai, B., Lin, Y. & Chen, J. (2021). UP-DETR: Unsupervised Pre-Training for Object Detection With Transformers. *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pp. 1601–1610.

Dai, Z., Yang, Z., Yang, Y., Carbonell, J. G., Le, Q. V. & Salakhutdinov, R. (2019). Transformer-XL: Attentive Language Models beyond a Fixed-Length Context. *Annual Meeting of the Association for Computational Linguistics*.

Darcet, T., Oquab, M., Mairal, J. & Bojanowski, P. (2024). Vision Transformers Need Registers. *The Twelfth International Conference on Learning Representations*.

Dasari, S. & Gupta, A. (2021, 16–18 Nov). Transformers for One-Shot Visual Imitation. *Proceedings of the 2020 Conference on Robot Learning*, 155(Proceedings of Machine Learning Research), 2071–2084.

d'Ascoli, S., Touvron, H., Leavitt, M. L., Morcos, A. S., Biroli, G. & Sagun, L. (2021). ConViT: Improving Vision Transformers with Soft Convolutional Inductive Biases. *Proceedings of the 38th ICML*, 139, 2286–2296.

Dayal, A., Cenkeramaddi, L. R. & Jha, A. (2022). Reward criteria impact on the performance of reinforcement learning agent for autonomous navigation. *Applied Soft Computing*, 126, 109241.

Dayan, E. & Cohen, L. G. (2011). Neuroplasticity subserving motor skill learning. *Neuron*, 72(3), 443–454.

de Bruin, T., Kober, J., Tuyls, K. & Babuška, R. (2018). Integrating State Representation Learning Into Deep Reinforcement Learning. *IEEE Robotics and Automation Letters*, 3(3), 1394-1401. doi: 10.1109/LRA.2018.2800101.

De Oliveira Fonseca, A. H., Zappala, E., Ortega Caro, J. & Dijk, D. V. (2023, 23–29 Jul). Continuous Spatiotemporal Transformer. *Proceedings of the 40th International Conference on Machine Learning*, 202(Proceedings of Machine Learning Research), 7343–7365.

Deng, F., Park, J. & Ahn, S. (2023). Facing Off World Model Backbones: RNNs, Transformers, and S4. *Thirty-seventh Conference on Neural Information Processing Systems*.

Devlin, J., Chang, M., Lee, K. & Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pp. 4171–4186.

Didolkar, A., Gupta, K., Goyal, A., Gundavarapu, N. B., Lamb, A. M., Ke, N. R. & Bengio, Y. (2022). Temporal latent bottleneck: Synthesis of fast and slow processing mechanisms in sequence learning. *Advances in Neural Information Processing Systems*, 35, 10505–10520.

Doerr, A., Daniel, C., Schiegg, M., Duy, N.-T., Schaal, S., Toussaint, M. & Sebastian, T. (2018). Probabilistic recurrent state-space models. *International conference on machine learning*, pp. 1280–1289.

Dolgov, D., Thrun, S., Montemerlo, M. & Diebel, J. (2010). Path Planning for Autonomous Vehicles in Unknown Semi-structured Environments. *The International Journal of Robotics Research*, 29(5), 485-501. doi: 10.1177/0278364909359210.

Dong, L., Xu, S. & Xu, B. (2018). Speech-Transformer: A No-Recurrence Sequence-to-Sequence Model for Speech Recognition. *2018 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2018, Calgary, AB, Canada, April 15-20, 2018*, pp. 5884–5888.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J. & Houlsby, N. (2021). An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. *9th International Conference on Learning Representations, ICLR*.

Dou, Z., Xu, Y., Gan, Z., Wang, J., Wang, S., Wang, L., Zhu, C., Zhang, P., Yuan, L., Peng, N., Liu, Z. & Zeng, M. (2022). An Empirical Study of Training End-to-End Vision-and-Language Transformers. *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR*, pp. 18145–18155.

Drugowitsch, J., DeAngelis, G. C., Klier, E. M., Angelaki, D. E. & Pouget, A. (2014). Optimal multisensory decision-making in a reaction-time task. *Elife*, 3.

Du, S. S., Kakade, S. M., Wang, R. & Yang, L. F. (2020). Is a Good Representation Sufficient for Sample Efficient Reinforcement Learning? *International Conference on Learning Representations*.

Duell, S., Udluft, S. & Sterzing, V. (2012). Solving Partially Observable Reinforcement Learning Problems with Recurrent Neural Networks. In *Neural Networks: Tricks of the Trade - Second Edition* (vol. 7700, pp. 709–733). Springer.

Dulac-Arnold, G., Levine, N., Mankowitz, D. J., Li, J., Paduraru, C., Gowal, S. & Hester, T. (2021). Challenges of real-world reinforcement learning: definitions, benchmarks and analysis. *Machine Learning*, 110(9), 2419–2468.

Dunion, M., McInroe, T., Luck, K. S., Hanna, J. P. & Albrecht, S. V. (2023). Temporal Disentanglement of Representations for Improved Generalisation in Reinforcement Learning. *The Eleventh International Conference on Learning Representations*.

Egli, P. & Hutter, M. (2020). Towards RL-Based Hydraulic Excavator Automation. *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2692-2697. doi: 10.1109/IROS45743.2020.9341598.

Egli, P. & Hutter, M. (2022). A General Approach for the Automation of Hydraulic Excavator Arms Using Reinforcement Learning. *IEEE Robotics and Automation Letters*, 7(2), 5679-5686. doi: 10.1109/LRA.2022.3152865.

Emami, N., Maio, A. D. & Braun, T. (2022). INTRAFORCE: Intra-Cluster Reinforced Social Transformer for Trajectory Prediction. *18th International Conference on Wireless and Mobile Computing, Networking and Communications, WiMob 2022, Thessaloniki, Greece, October 10-12, 2022*, pp. 333–338.

Epstein, D., Wu, J., Schmid, C. & Sun, C. (2021). Learning Temporal Dynamics from Cycles in Narrated Video. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 1460-1469. doi: 10.1109/ICCV48922.2021.00151.

Esser, P., Rombach, R. & Ommer, B. (2021). Taming transformers for high-resolution image synthesis. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 12873–12883.

Everitt, T., Krakovna, V., Orseau, L., Lefrancq, A., Legg, S. & Hutter, M. (2021). Reward Tampering Problems and Solutions in Reinforcement Learning: A Causal Influence Diagram Perspective. *Advances in Neural Information Processing Systems (NeurIPS)*, 34, 26812–26824.

Eysenbach, B., Gupta, A., Ibarz, J. & Levine, S. (2019). Diversity is All You Need: Learning Skills without a Reward Function. *International Conference on Learning Representations (ICLR)*.

Faal, F., Schmitt, K. A. & Yu, J. Y. (2023). Reward modeling for mitigating toxicity in transformer-based language models. *Appl. Intell.*, 53(7), 8421–8435.

Faloutsos, P., van de Panne, M. & Terzopoulos, D. (2001). Composable controllers for physics-based character animation. *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pp. 251–260.

Fan, Y., Deng, F., Shi, X. & Yang, J. (2022). Learning to Solve Pod Retrieval as Sequential Decision Making Problem. *17th IEEE International Conference on Control & Automation, ICCA 2022, Naples, Italy, June 27-30, 2022*, pp. 220–224.

Farquhar, G., Gustafson, L., Lin, Z., Whiteson, S., Usunier, N. & Synnaeve, G. (2020). Growing action spaces. *International Conference on Machine Learning*, pp. 3040–3051.

Featherstone, R. (2014). *Rigid body dynamics algorithms*. Springer.

Finn, C., Abbeel, P. & Levine, S. (2017). Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks. *Proceedings of the 34th ICML*, 70, 1126–1135.

Firdaus, M., Shandilya, A., Ekbal, A. & Bhattacharyya, P. (2022). Being Polite: Modeling Politeness Variation in a Personalized Dialog Agent. *IEEE Transactions on Computational Social Systems*.

Floridi, L. & Chiriatti, M. (2020). GPT-3: Its Nature, Scope, Limits, and Consequences. *Minds Mach.*, 30(4), 681–694.

Foerster, J. N., Assael, Y. M., de Freitas, N. & Whiteson, S. (2016). Learning to Communicate with Deep Multi-Agent Reinforcement Learning. *NeurIPS*, pp. 2137–2145.

Franceschelli, G. & Musolesi, M. (2023). Reinforcement Learning for Generative AI: State of the Art, Opportunities and Open Research Challenges. *J. Artif. Intell. Res.*, 79, 417-446.

Frans, K., Ho, Y. D., Chen, X., Abbeel, P., Schulman, J. & Daumé III, H. (2017). Meta-Learning Shared Hierarchies. *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 5059–5070.

Fu, J., Singh, A., Ghosh, D., Yang, L. & Levine, S. (2018). Variational inverse control with events: A general framework for data-driven reward definition. *Advances in Neural Information Processing Systems*, pp. 8538–8547.

Fu, W., Li, Y., Ye, Z. & Liu, Q. (2022). Decision Making for Autonomous Driving Via Multimodal Transformer and Deep Reinforcement Learning*. *IEEE International Conference on Real-time Computing and Robotics, RCAR*, pp. 481–486.

Fujimoto, S., Chang, W.-D., Smith, E., Gu, S. S., Precup, D. & Meger, D. (2024). For sale: State-action representation learning for deep reinforcement learning. *Advances in Neural Information Processing Systems*, 36.

Gai, S., Lyu, S., Zhang, H. & Wang, D. (2024). Continual reinforcement learning for quadruped robot locomotion. *Entropy*, 26(1).

Ghosh, D. & Bellemare, M. G. (2020). Representations for Stable Off-Policy Reinforcement Learning. *Proceedings of the 37th ICML*, 119, 3556–3565.

Glanois, C., Weng, P., Zimmer, M., Li, D., Yang, T., Hao, J. & Liu, W. (2024). A survey on interpretable reinforcement learning. *Machine Learning*, 1–44.

Glorot, X. & Bengio, Y. (2010). Understanding the difficulty of training deep feedforward neural networks. *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, AISTATS 2010, Chia Laguna Resort, Sardinia, Italy, May 13-15, 2010*, 9(JMLR Proceedings), 249–256.

Gondhi, N. K. & Gupta, A. (2017). Survey on machine learning based scheduling in cloud computing. *Proceedings of the 2017 International Conference on Intelligent Systems, Metaheuristics & Swarm Intelligence*, pp. 57–61.

González, A. V. (2019). Reinforcement Learning for Improved Low Resource Dialogue Generation. *AAAI Conference on Artificial Intelligence*.

Gopalakrishnan, A., Irie, K., Schmidhuber, J. & van Steenkiste, S. (2023). Unsupervised Learning of Temporal Abstractions With Slot-Based Transformers. *Neural Comput.*, 35(4), 593–626.

Goulão, M. & Oliveira, A. L. (2022). Pretraining the Vision Transformer using self-supervised methods for vision based Deep Reinforcement Learning. *European Conference on Artificial Intelligence*.

Graves, A., Bellemare, M. G., Menick, J., Munos, R. & Kavukcuoglu, K. (2017). Automated Curriculum Learning for Neural Networks. *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, pp. 1311–1320.

Gronauer, S. & Diepold, K. (2022). Multi-agent deep reinforcement learning: a survey. *Artif. Intell. Rev.*, 55(2), 895–943.

Guss, W. H., Houghton, B., Topin, N., Wang, P., Codel, C. R., Veloso, M. M. & Salakhutdinov, R. (2019). MineRL: A Large-Scale Dataset of Minecraft Demonstrations. *International Joint Conference on Artificial Intelligence*.

Ha, D. & Schmidhuber, J. (2018). World models. *Advances in Neural Information Processing Systems*, pp. 2454–2462.

Hafner, D. (2022). Benchmarking the Spectrum of Agent Capabilities. *International Conference on Learning Representations*.

Hafner, D., Lillicrap, T., Ba, J. & Norouzi, M. (2020a). Dream to Control: Learning Behaviors by Latent Imagination. *International Conference on Learning Representations (ICLR)*.

Hafner, D., Lillicrap, T. P., Ba, J. & Norouzi, M. (2020b). Dream to Control: Learning Behaviors by Latent Imagination. *8th International Conference on Learning Representations, ICLR*.

Hafner, D., Lillicrap, T. P., Norouzi, M. & Ba, J. (2021). Mastering Atari with Discrete World Models. *9th International Conference on Learning Representations, ICLR*.

Hafner, D., Paukonis, J., Ba, J. & Lillicrap, T. (2025). Mastering diverse control tasks through world models. *Nature*, 640, 647 - 653.

Hahn, M. (2022). *Language Guided Localization and Navigation*. (Ph.D. thesis, Georgia Institute of Technology, Atlanta, GA, USA).

Han, H., Wu, X., Liao, H., Xu, Z., Hu, Z., Li, R., Zhang, Y. & Li, X. (2025). Atom: Aligning text-to-motion model at event-level with gpt-4vision reward. *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 22746–22755.

Hansen, N. & Wang, X. (2021). Generalization in Reinforcement Learning by Soft Data Augmentation. *IEEE International Conference on Robotics and Automation, ICRA 2021, Xi'an, China, May 30 - June 5, 2021*, pp. 13611–13617.

Harvey, F. G., Yurick, M., Nowrouzezahrai, D. & Pal, C. (2020). Robust motion in-betweening. *ACM Transactions on Graphics (TOG)*, 39(4), 60–1.

Haugh, M. B. & Lo, A. W. (2001). Computational challenges in portfolio management. *Comput. Sci. Eng.*, 3(3), 54–59.

Hausknecht, M. J. & Stone, P. (2015). Deep Recurrent Q-Learning for Partially Observable MDPs. *2015 AAAI Fall Symposia, Arlington, Virginia, USA, November 12-14, 2015*, pp. 29–37.

Heuillet, A., Couthouis, F. & Rodríguez, N. D. (2021). Explainability in deep reinforcement learning. *Knowl. Based Syst.*, 214, 106685.

Hewing, L., Wabersich, K. P., Menner, M. & Zeilinger, M. N. (2020). Learning-based model predictive control: Toward safe learning in control. *Annual Review of Control, Robotics, and Autonomous Systems*, 3(1), 269–296.

Ho, J. & Ermon, S. (2016). Generative adversarial imitation learning. *Advances in neural information processing systems*, 29.

Hochreiter, S. (1998). The Vanishing Gradient Problem During Learning Recurrent Neural Nets and Problem Solutions. *Int. J. Uncertain. Fuzziness Knowl. Based Syst.*, 6(2), 107–116.

Hochreiter, S. & Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Comput.*, 9, 1735–1780. doi: 10.1162/neco.1997.9.8.1735.

Hopf, K., Nahr, N., Staake, T. & Lehner, F. (2024). The group mind of hybrid teams with humans and intelligent agents in knowledge-intense work. *Journal of Information Technology*, 40, 9 - 34.

Hossain, S., Chakrabarty, A., Gadekallu, T. R., Alazab, M. & Piran, M. J. (2023). Vision Transformers, Ensemble Model, and Transfer Learning Leveraging Explainable AI for Brain Tumor Detection and Classification. *IEEE Journal of Biomedical and Health Informatics*, 28, 1261-1272.

Hu, A., Cotter, F., Mohan, N., Gurau, C. & Kendall, A. (2020a). Probabilistic Future Prediction for Video Scene Understanding. *Computer Vision – ECCV 2020*, pp. 767–785.

Hu, S., Zhu, F., Chang, X. & Liang, X. (2021). Transformer Based Multi-Agent Framework. *2021 IEEE International Conference on Multimedia & Expo Workshops, ICME Workshops*, pp. 1–2.

Hu, Y., Wang, W., Jia, H., Wang, Y., Chen, Y., Hao, J., Wu, F. & Fan, C. (2020b). Learning to Utilize Shaping Rewards: A New Approach of Reward Shaping. *Proceedings of the 34th International Conference on Neural Information Processing Systems*, (NIPS'20).

Hu, Y., Wang, W., Jia, H., Wang, Y., Chen, Y., Hao, J., Wu, F. & Fan, C. (2020c). Learning to utilize shaping rewards: A new approach of reward shaping. *Advances in Neural Information Processing Systems*, 33, 15931–15941.

Huang, J., Xu, Y., Wang, Q., Wang, Q. C., Liang, X., Wang, F., Zhang, Z., Wei, W., Zhang, B., Huang, L. et al. (2025). Foundation models and intelligent decision-making: Progress, challenges, and perspectives. *The Innovation*.

Huang, L., Mao, F., Zhang, K. & Li, Z. (2022). Spatial-Temporal Convolutional Transformer Network for Multivariate Time Series Forecasting. *Sensors*, 22(3), 841.

Huang, W., Mordatch, I. & Pathak, D. (2020a). One Policy to Control Them All: Shared Modular Policies for Agent-Agnostic Control. *Proceedings of the 37th International Conference on Machine Learning, ICML*, 119, 4455–4464.

Huang, X. S., Pérez, F., Ba, J. & Volkovs, M. (2020b). Improving Transformer Optimization Through Better Initialization. *Proceedings of the 37th International Conference on Machine Learning, ICML*, 119, 4475–4483.

Huang, Y., Gu, Y., Yuan, K., Yang, S., Liu, T. & Chen, H. (2024). Human Knowledge Enhanced Reinforcement Learning for Mandatory Lane-Change of Autonomous Vehicles in Congested Traffic. *IEEE Transactions on Intelligent Vehicles*, 9, 3509-3519.

Huang, Y., Xie, K., Bharadhwaj, H. & Shkurti, F. (2021). Continual model-based reinforcement learning with hypernetworks. *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 799–805.

Huang, Y. & Kanai, T. (2024). Brittle Fracture Animation with VQ-VAE-Based Generative Method. *Eurographics/ ACM SIGGRAPH Symposium on Computer Animation - Posters*. doi: 10.2312/sca.20241163.

Ikegami, T. & Ganesh, G. (2014). Watching novice action degrades expert motor performance: causation between action production and outcome prediction of observed actions by humans. *Scientific reports*, 4(1), 6989.

Inala, J. P. (2022). *Neurosymbolic Learning for Robust and Reliable Intelligent Systems*. (Ph.D. thesis, Massachusetts Institute of Technology, USA).

Inala, J. P., Yang, Y., Paulos, J., Pu, Y., Bastani, O., Kumar, V., Rinard, M. C. & Solar-Lezama, A. (2020). Neurosymbolic Transformers for Multi-Agent Communication. *NeurIPS*.

Islam, R., Zang, H., Goyal, A., Lamb, A. M., Kawaguchi, K., Li, X., Laroche, R., Bengio, Y. & Tachet des Combes, R. (2022). Discrete Compositional Representations as an Abstraction for Goal Conditioned Reinforcement Learning. *Advances in Neural Information Processing Systems*, 35, 3885–3899.

Jaakkola, T. S., Singh, S. & Jordan, M. I. (1994). Reinforcement Learning Algorithm for Partially Observable Markov Decision Problems. *NeurIPS*, pp. 345–352.

Jaegle, A., Gimeno, F., Brock, A., Vinyals, O., Zisserman, A. & Carreira, J. (2021). Perceiver: General Perception with Iterative Attention. *Proceedings of the 38th International Conference on Machine Learning, ICML*, 139, 4651–4664.

James, S., Ma, Z., Arrojo, D. R. & Davison, A. J. (2020). Rlbench: The robot learning benchmark & learning environment. *IEEE Robotics and Automation Letters*, 5(2), 3019–3026.

Jangir, R., Hansen, N., Ghosal, S., Jain, M. & Wang, X. (2022). Look Closer: Bridging Egocentric and Third-Person Views With Transformers for Robotic Manipulation. *IEEE Robotics Autom. Lett.*, 7(2), 3046–3053.

Janner, M., Li, Q. & Levine, S. (2021a). Offline Reinforcement Learning as One Big Sequence Modeling Problem. *NeurIPS*, pp. 1273–1286.

Janner, M., Li, Q. & Levine, S. (2021b). Offline Reinforcement Learning as One Big Sequence Modeling Problem. *34th Annual Conference on Neural Information Processing Systems*, pp. 1273–1286.

Janner, M., Li, Q. & Levine, S. (2021c). Offline Reinforcement Learning as One Big Sequence Modeling Problem. *NeurIPS*, pp. 1273–1286.

Ji, T., Luo, Y., Sun, F., Jing, M., He, F. & Huang, W. (2022). When to Update Your Model: Constrained Model-based Reinforcement Learning. *Advances in Neural Information Processing Systems*, 35, 23150–23163.

Jiang, Z., Hou, Q., Yuan, L., Zhou, D., Shi, Y., Jin, X., Wang, A. & Feng, J. (2021). All Tokens Matter: Token Labeling for Training Better Vision Transformers. *NeurIPS*, pp. 18590–18602.

Jo, D., Kwon, T., Kim, E. & Kim, S. (2022). Selective Token Generation for Few-shot Natural Language Generation. *Proceedings of the 29th International Conference on Computational Linguistics, COLING 2022, Gyeongju, Republic of Korea, October 12-17, 2022*, pp. 5837–5856.

Jud, D., Leemann, P., Kerscher, S. & Hutter, M. (2019). Autonomous Free-Form Trenching Using a Walking Excavator. *IEEE Robotics and Automation Letters*, 4(4), 3208-3215. doi: 10.1109/LRA.2019.2925758.

Juravsky, J., Guo, Y., Fidler, S. & Peng, X. B. (2022). Padl: Language-directed physics-based character control. *SIGGRAPH Asia 2022 Conference Papers*, pp. 1–9.

Kaelbling, L. P., Littman, M. L. & Cassandra, A. R. (1998). Planning and acting in partially observable stochastic domains. *Artificial intelligence*, 101(1-2), 99–134.

Kaiser, L., Babaeizadeh, M., Milos, P., Osinski, B., Campbell, R. H., Czechowski, K., Erhan, D., Finn, C., Kozakowski, P., Levine, S., Mohiuddin, A., Sepassi, R., Tucker, G. & Michalewski, H. (2020). Model Based Reinforcement Learning for Atari. *8th International Conference on Learning Representations, ICLR*.

Kalashnikov, D., Irpan, A., Pastor, P., Ibarz, J., Herzog, A., Jang, E., Quillen, D., Holly, E., Kalakrishnan, M., Vanhoucke, V. & Levine, S. (2018). Scalable Deep Reinforcement Learning for Vision-Based Robotic Manipulation. *2nd Annual Conference on Robot Learning, CoRL 2018, Zürich*, 87, 651–673.

Kargar, E. & Kyrki, V. (2022). Vision Transformer for Learning Driving Policies in Complex and Dynamic Environments. *IEEE Intelligent Vehicles Symposium*, pp. 1558–1564.

Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R. & Fei-Fei, L. (2014). Large-Scale Video Classification with Convolutional Neural Networks. *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pp. 1725–1732.

Katharopoulos, A., Vyas, A., Pappas, N. & Fleuret, F. (2020). Transformers are RNNs: Fast Autoregressive Transformers with Linear Attention. *Proceedings of the 37th ICML*, 119, 5156–5165.

Ke, N. R., Singh, A., Touati, A., Goyal, A., Bengio, Y., Parikh, D. & Batra, D. (2018). Modeling the long term future in model-based reinforcement learning. *International Conference on Learning Representations*.

Keles, F. D., Wijewardena, P. M. & Hegde, C. (2023). On The Computational Complexity of Self-Attention. *International Conference on Algorithmic Learning Theory*, 201, 597–619.

Kempka, M., Wydmuch, M., Runc, G., Toczek, J. & Jaskowski, W. (2016). ViZDoom: A Doom-based AI research platform for visual reinforcement learning. *IEEE Conference on Computational Intelligence and Games, CIG 2016, Santorini, Greece, September 20-23, 2016*, pp. 1–8.

Khan, M. J., Ahmed, S. H. & Sukthankar, G. (2022a). Transformer-Based Value Function Decomposition for Cooperative Multi-Agent Reinforcement Learning in StarCraft. *Proceedings of the Eighteenth AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment, AIIDE 2022, Pomona, CA, USA, October 24-28, 2022*, pp. 113–119.

Khan, S., Naseer, M., Hayat, M., Zamir, S. W., Khan, F. S. & Shah, M. (2022b). Transformers in Vision: A Survey. *ACM Computing Surveys*, 54(10), 1–41.

Khetarpal, K., Riemer, M., Rish, I. & Precup, D. (2022). Towards continual reinforcement learning: A review and perspectives. *Journal of Artificial Intelligence Research*, 75, 1401–1476.

Killian, T. W., Parbhoo, S. & Ghassemi, M. (2023). Risk Sensitive Dead-end Identification in Safety-Critical Offline Reinforcement Learning. *Trans. Mach. Learn. Res.*, 2023.

Kim, C., Park, J., Shin, J., Lee, H., Abbeel, P. & Lee, K. (2023). Preference Transformer: Modeling Human Preferences using Transformers for RL. *International Conference on Learning Representations*. Retrieved from: https://openreview.net/forum?id=Peot1SFDX0.

Kim, H., Ohmura, Y. & Kuniyoshi, Y. (2021). Transformer-based deep imitation learning for dual-arm robot manipulation. *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2021, Prague, Czech Republic, September 27 - Oct. 1, 2021*, pp. 8965–8972.

Kim, H., Ohmura, Y. & Kuniyoshi, Y. (2022). Memory-based gaze prediction in deep imitation learning for robot manipulation. *2022 International Conference on Robotics and Automation, ICRA 2022, Philadelphia, PA, USA, May 23-27, 2022*, pp. 2427–2433.

Kingma, D. P. & Welling, M. (2019a). An Introduction to Variational Autoencoders. *Found. Trends Mach. Learn.*, 12(4), 307–392.

Kingma, D. P. & Welling, M. (2019b). An Introduction to Variational Autoencoders. *Foundations and Trends in Machine Learning*, 12(4), 307-392. doi: 10.1561/2200000056.

Kiran, B. R., Sobh, I., Talpaert, V., Mannion, P., Al Sallab, A. A., Yogamani, S. & Pérez, P. (2021). Deep reinforcement learning for autonomous driving: A survey. *IEEE transactions on intelligent transportation systems*, 23(6), 4909–4926.

Kiran, B. R., Sobh, I., Talpaert, V., Mannion, P., Sallab, A. A. A., Yogamani, S. K. & Pérez, P. (2022). Deep Reinforcement Learning for Autonomous Driving: A Survey. *IEEE Trans. Intell. Transp. Syst.*, 23(6), 4909–4926.

Kirkpatrick, J., Pascanu, R., Rabinowitz, N. C., Veness, J., Desjardins, G., Rusu, A. A., Milan, K., Quan, J., Ramalho, T., Grabska-Barwinska, A., Hassabis, D., Clopath, C., Kumaran, D. & Hadsell, R. (2016). Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114, 3521 - 3526.

Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A. A., Milan, K., Quan, J., Ramalho, T., Grabska-Barwinska, A. et al. (2017). Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13), 3521–3526.

Koivo, A. J., Thoma, M., Kocaoglan, E. & Andrade-Cetto, J. (1996). Modeling and Control of Excavator Dynamics during Digging Operation. *Journal of Aerospace Engineering*, 9(1), 10-18.

Koivumäki, J. & Mattila, J. (2015). High performance nonlinear motion/force controller design for redundant hydraulic construction crane automation. *Automation in construction*, 51, 59–77.

Krebs, F., Meixner, A., Patzer, I. & Asfour, T. (2021). The KIT Bimanual Manipulation Dataset. *IEEE/RAS International Conference on Humanoid Robots (Humanoids)*, pp. 499–506.

Krishnan, S., Garg, A., Patil, S., Lea, C., Hager, G., Abbeel, P. & Goldberg, K. (2017). DDCO: Discovery of deep continuous options for robot learning from demonstrations. *Conference on Robot Learning*, pp. 418–437.

Kuba, J. G., Chen, R., Wen, M., Wen, Y., Sun, F., Wang, J. & Yang, Y. (2022). Trust Region Policy Optimisation in Multi-Agent Reinforcement Learning. *10th International Conference on Learning Representations, ICLR*.

Kujanpää, K., Pajarinen, J. & Ilin, A. (2023). Hierarchical imitation learning with vector quantized models. *International Conference on Machine Learning*, pp. 17896–17919.

Kumar, A., Zhou, A., Tucker, G. & Levine, S. (2020). Conservative Q-Learning for Offline Reinforcement Learning. *NeurIPS*.

Kurin, V., Igl, M., Rocktäschel, T., Boehmer, W. & Whiteson, S. (2021). My Body is a Cage: the Role of Morphology in Graph-Based Incompatible Control. *9th International Conference on Learning Representations, ICLR*.

Kurutach, T., Clavera, I., Duan, Y., Tamar, A. & Abbeel, P. (2018). Model-Ensemble Trust-Region Policy Optimization. *International Conference on Learning Representations*.

Kwiatkowski, A., Alvarado, E., Kalogeiton, V., Liu, C. K., Pettré, J., van de Panne, M. & Cani, M.-P. (2022). A survey on reinforcement learning methods in character animation. *Computer Graphics Forum*, 41(2), 613–639.

La Hera, P. & Morales, D. O. (2019). What do we observe when we equip a forestry crane with motion sensors? *Croatian Journal of Forest Engineering: Journal for Theory and Application of Forestry Engineering*, 40(2), 259–280.

Lake, B. M., Ullman, T. D., Tenenbaum, J. B. & Gershman, S. J. (2016). Building machines that learn and think like people. *Behavioral and Brain Sciences*, 40.

Lake, B. M., Ullman, T. D., Tenenbaum, J. B. & Gershman, S. J. (2017). Building machines that learn and think like people. *Behavioral and Brain Sciences*, 40, e253.

Larsen, T. N., Teigen, H. Ø., Laache, T., Varagnolo, D. & Rasheed, A. (2021). Comparing Deep Reinforcement Learning Algorithms' Ability to Safely Navigate Challenging Waters. *Frontiers Robotics AI*, 8, 738113.

Laskin, M., Srinivas, A. & Abbeel, P. (2020). CURL: Contrastive Unsupervised Representations for Reinforcement Learning. *Proceedings of the 37th ICML*, 119, 5639–5650.

Lathuilière, S., Massé, B., Mesejo, P. & Horaud, R. (2019). Neural network based reinforcement learning for audio-visual gaze control in human-robot interaction. *Pattern Recognit. Lett.*, 118, 61–71.

Latif, S., Cuayáhuitl, H., Pervez, F., Shamshad, F., Ali, H. S. & Cambria, E. (2023). A survey on deep reinforcement learning for audio-based applications. *Artif. Intell. Rev.*, 56(3), 2193–2240.

Lee, J., Lee, Y., Kim, J., Kosiorek, A. R., Choi, S. & Teh, Y. W. (2019). Set Transformer: A Framework for Attention-based Permutation-Invariant Neural Networks. *Proceedings of the 36th International Conference on Machine Learning, ICML*, 97, 3744–3753.

Lee, K., Nachum, O., Yang, M., Lee, L., Freeman, D., Guadarrama, S., Fischer, I., Xu, W., Jang, E., Michalewski, H. & Mordatch, I. (2022a). Multi-Game Decision Transformers. *NeurIPS*.

Lee, K.-H., Nachum, O., Yang, M. S., Lee, L., Freeman, D., Guadarrama, S., Fischer, I., Xu, W., Jang, E., Michalewski, H. et al. (2022b). Multi-game decision transformers. *Advances in Neural Information Processing Systems*, 35, 27921–27936.

Lee, L., Eysenbach, B., Salakhutdinov, R. R., Gu, S. S. & Finn, C. (2020a). Weakly-supervised reinforcement learning for controllable behavior. *Advances in Neural Information Processing Systems*, 33, 2661–2673.

Lee, M. A., Zhu, Y., Zachares, P., Tan, M., Srinivasan, K., Savarese, S., Fei-Fei, L., Garg, A. & Bohg, J. (2020b). Making Sense of Vision and Touch: Learning Multimodal Representations for Contact-Rich Tasks. *IEEE Transactions on Robotics*, 36(3), 582-596. doi: 10.1109/TRO.2019.2959445.

Lesort, T., Díaz-Rodríguez, N., Goudou, J.-F. & Filliat, D. (2018). State representation learning for control: An overview. *Neural Networks*, 108, 379-392. doi: https://doi.org/10.1016/j.neunet.2018.07.006.

Li, C., Yamanaka, C., Kaitoh, K. & Yamanishi, Y. (2022a). Transformer-based Objective-reinforced Generative Adversarial Network to Generate Desired Molecules. *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI*, pp. 3884–3890.

Li, G., Gomez, R., Nakamura, K. & He, B. (2019a). Human-centered reinforcement learning: A survey. *IEEE Transactions on Human-Machine Systems*, 49(4), 337–349.

Li, L. & Qiu, X. (2021). Token-aware virtual adversarial training in natural language understanding. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(9), 8410–8418.

Li, S., Jin, X., Xuan, Y., Zhou, X., Chen, W., Wang, Y. & Yan, X. (2019b). Enhancing the Locality and Breaking the Memory Bottleneck of Transformer on Time Series Forecasting. *NeurIPS*, pp. 5244–5254.

Li, S., Hu, W., Cao, D., Zhang, Z., Huang, Q., Chen, Z. & Blaabjerg, F. (2022b). A Multiagent Deep Reinforcement Learning Based Approach for the Optimization of Transformer Life Using Coordinated Electric Vehicles. *IEEE Trans. Ind. Informatics*, 18(11), 7639–7652.

Li, T., Xi, W., Fang, M., Xu, J. & Meng, M. Q. (2019c). Learning to Solve a Rubik's Cube with a Dexterous Hand. *2019 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, 1387-1393.

Li, X., Cai, H., Jiang, T., Liu, C. & Ji, Y. (2022c). Emotion Aware Reinforcement Network for Visual Storytelling. *31st International Conference on Artificial Neural Networks*, 13530, 26–37.

Li, Z., Wallace, E., Shen, S., Lin, K., Keutzer, K., Klein, D. & Gonzalez, J. (2020). Train Big, Then Compress: Rethinking Model Size for Efficient Training and Inference of Transformers. *Proceedings of the 37th International Conference on Machine Learning, ICML*, 119, 5958–5968.

Liang, P. P., Zadeh, A. & philippe Morency, L. (2022). Foundations & Trends in Multimodal Machine Learning: Principles, Challenges, and Open Questions. *ACM Computing Surveys*, 56, 1 - 42.

Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D. & Wierstra, D. (2016). Continuous control with deep reinforcement learning. *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*.

Lim, J., Ha, S. & Choi, J. (2020). Prediction of Reward Functions for Deep Reinforcement Learning via Gaussian Process Regression. *IEEE/ASME Transactions on Mechatronics*, 25(4), 1739-1746. doi: 10.1109/TMECH.2020.2993564.

Lin, A., Wohlwend, J., Chen, H. & Lei, T. (2020). Autoregressive Knowledge Distillation through Imitation Learning. *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pp. 6121–6133.

Lin, H., Geishauser, C., Feng, S., Lubis, N., van Niekerk, C., Heck, M. & Gasic, M. (2022a). GenTUS: Simulating User Behaviour and Language in Task-oriented Dialogues with Generative Transformers. *Proceedings of the 23rd Annual Meeting of the Special Interest Group on Discourse and Dialogue, SIGDIAL 2022, Edinburgh, UK, 07-09 September 2022*, pp. 270–282.

Lin, J., Zeng, A., Lu, S., Cai, Y., Zhang, R., Wang, H. & Zhang, L. (2023a). Motion-x: A large-scale 3d expressive whole-body human motion dataset. *Advances in Neural Information Processing Systems*, 36, 25268–25280.

Lin, T., Wang, Y., Liu, X. & Qiu, X. (2022b). A survey of transformers. *AI Open*, 3, 111–132.

Lin, X., Yu, L., Cheng, K.-T. & Yan, Z. (2023b). BATFormer: Towards Boundary-Aware Lightweight Transformer for Efficient Medical Image Segmentation. *IEEE Journal of Biomedical and Health Informatics*.

Liu, C. K., Hertzmann, A. & Popović, Z. (2005). Learning physics-based motion style with nonlinear inverse optimization. *ACM Transactions on Graphics (TOG)*, 24(3), 1071–1081.

Liu, H., Socher, R. & Xiong, C. (2019). Taming MAML: Efficient unbiased meta-reinforcement learning. *Proceedings of the 36th ICML*, 97, 4061–4071.

Liu, H., Huang, Z., Mo, X. & Lv, C. (2022a). Augmenting Reinforcement Learning With Transformer-Based Scene Representation Learning for Decision-Making of Autonomous Driving. *IEEE Transactions on Intelligent Vehicles*, 9, 4405-4421.

Liu, L., Yin, K., van de Panne, M., Shao, T. & Xu, W. (2010). Sampling-based contact-rich motion control. *ACM Transactions on Graphics (TOG)*, 29(4), 128:1–128:10.

Liu, L., Liu, X., Gao, J., Chen, W. & Han, J. (2020). Understanding the Difficulty of Training Transformers. *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pp. 5747–5763.

Liu, Q., Chung, A., Szepesvári, C. & Jin, C. (2022b). When Is Partially Observable Reinforcement Learning Not Scary? *Conference on Learning Theory*, 178, 5175–5220.

Liu, R., Nageotte, F., Zanne, P., de Mathelin, M. & Dresp, B. (2021a). Deep reinforcement learning for the control of robotic manipulation: a focussed mini-review. *Robotics*, 10. doi: 10.3390/robotics10010022.

Liu, X., Lai, H., Yu, H., Xu, Y., Zeng, A., Du, Z., Zhang, P., Dong, Y. & Tang, J. (2023). WebGLM: Towards An Efficient Web-Enhanced Question Answering System with Human Preferences. *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*.

Liu, Y., Sangineto, E., Bi, W., Sebe, N., Lepri, B. & Nadai, M. D. (2021b). Efficient Training of Visual Transformers with Small Datasets. *NeurIPS*, pp. 23818–23830.

Liu, Y., Li, H., Guo, Y., Kong, C., Li, J. & Wang, S. (2022c). Rethinking Attention-Model Explainability through Faithfulness Violation Test. *ICML*, 162, 13807–13824.

Liu, Z., Mao, H., Wu, C., Feichtenhofer, C., Darrell, T. & Xie, S. (2022d). A ConvNet for the 2020s. *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR*, pp. 11966–11976.

Locatello, F., Bauer, S., Lucic, M., Raetsch, G., Gelly, S., Schölkopf, B. & Bachem, O. (2019). Challenging common assumptions in the unsupervised learning of disentangled representations. *International Conference on Machine Learning*, pp. 4114–4124.

Locatello, F., Weissenborn, D., Unterthiner, T., Mahendran, A., Heigold, G., Uszkoreit, J., Dosovitskiy, A. & Kipf, T. (2020a). Object-Centric Learning with Slot Attention. *NeurIPS*.

Locatello, F., Weissenborn, D., Unterthiner, T., Mahendran, A., Heigold, G., Uszkoreit, J., Dosovitskiy, A. & Kipf, T. (2020b). Object-Centric Learning with Slot Attention. *NeurIPS*.

Lohse, O., Pütz, N. & Hörmann, K. (2021). Implementing an Online Scheduling Approach for Production with Multi Agent Proximal Policy Optimization (MAPPO). *Advances in Production Management Systems. Artificial Intelligence for Sustainable and Resilient Production Systems: IFIP WG 5.7 International Conference, APMS 2021, Nantes, France, September 5–9, 2021, Proceedings, Part V*, pp. 586–595.

Lopez-Paz, D. & Ranzato, M. (2017). Gradient episodic memory for continual learning. *Advances in neural information processing systems*, 30.

Lu, C., Bao, Q., Xia, S. & Qu, C. (2022a). Centralized reinforcement learning for multi-agent cooperative environments. *Evolutionary Intelligence*, 1–7.

Lu, C., Ball, P. J., Rudner, T. G. J., Parker-Holder, J., Osborne, M. A. & Teh, Y. W. (2022b). Challenges and Opportunities in Offline Reinforcement Learning from Visual Observations. *Trans. Mach. Learn. Res.*, 2023.

Lu, H., Zhang, X. & Yang, S. (2020). A Learning-based Iterative Method for Solving Vehicle Routing Problems. *8th International Conference on Learning Representations, ICLR*.

Lu, J., Yao, J., Zhang, J., Zhu, X., Xu, H., Gao, W., Xu, C., Xiang, T. & Zhang, L. (2021). SOFT: Softmax-free Transformer with Linear Complexity. *NeurIPS*, pp. 21297–21309.

Lu, J., Batra, D., Parikh, D. & Lee, S. (2019). ViLBERT: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. *Advances in Neural Information Processing Systems*, pp. 13–23.

Lu, Z., Rallapalli, S., Chan, K. S. & Porta, T. L. (2017). Modeling the Resource Requirements of Convolutional Neural Networks on Mobile Devices. *Proceedings of the 2017 ACM on Multimedia Conference, MM 2017, Mountain View, CA, USA, October 23-27, 2017*, pp. 1663–1671.

Luo, J., Dong, P., Wu, J., Kumar, A., Geng, X. & Levine, S. (2023). Action-quantized offline reinforcement learning for robotic skill learning. *Conference on Robot Learning*, pp. 1348–1361.

Luo, Y., Dong, K., Zhao, L., Sun, Z., Zhou, C. & Song, B. (2020). Balance Between Efficient and Effective Learning: Dense2Sparse Reward Shaping for Robot Manipulation with Environment Uncertainty. *2022 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, 1192-1198.

Lygerakis, F., Dave, V. & Rueckert, E. (2024). M2CURL: Sample-Efficient Multimodal Reinforcement Learning via Self-Supervised Representation Learning for Robotic Manipulation. *2024 21st International Conference on Ubiquitous Robots (UR)*, 490-497.

Ma, J., Chen, Y., Wu, F., Ji, X. & Ding, Y. (2022). Multimodal Reinforcement Learning with Effective State Representation Learning. *21st International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2022, Auckland, New Zealand, May 9-13, 2022*, pp. 1684–1686.

Ma, M., D'Oro, P., Bengio, Y. & Bacon, P.-L. (2021a). Long-Term Credit Assignment via Model-based Temporal Shortcuts. *Deep RL Workshop NeurIPS 2021*.

Ma, Z., Zhuang, Y., Weng, P., Li, D., Shao, K., Liu, W., Zhuo, H. H. & Jianye, H. (2021b). Interpretable reinforcement learning with neural symbolic logic.

MacGlashan, J., Ho, M. K., Loftin, R. T., Peng, B., Wang, G., Roberts, D. L., Taylor, M. E. & Littman, M. L. (2017). Interactive Learning from Policy-Dependent Human Feedback. *Proceedings of the 34th ICML*, 70, 2285–2294.

Madden, L., Fox, C. & Thrampoulidis, C. (2025). Next-token prediction capacity: general upper bounds and a lower bound for transformers. *IEEE Transactions on Information Theory*.

Mai, V., Mani, K. & Paull, L. (2022). Sample Efficient Deep Reinforcement Learning via Uncertainty Estimation. *International Conference on Learning Representations*.

Makoviychuk, V., Wawrzyniak, L., Guo, Y., Lu, M., Storey, K., Macklin, M., Hoeller, D., Rudin, N., Allshire, A., Handa, A. & State, G. (2021). Isaac Gym: High Performance GPU Based Physics Simulation For Robot Learning. *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.

Manchin, A., Abbasnejad, E. & van den Hengel, A. (2019). Reinforcement Learning with Attention that Works: A Self-Supervised Approach. *Neural Information Processing - 26th International Conference, ICONIP*, 1143, 223–230.

Mandery, C., Terlemez, O., Do, M., Vahrenkamp, N. & Asfour, T. (2016). Unifying Representations and Large-Scale Whole-Body Motion Databases for Studying Human Motion. *IEEE Transactions on Robotics*, 32(4), 796–809.

Mao, C., Jiang, L., Dehghani, M., Vondrick, C., Sukthankar, R. & Essa, I. (2022). Discrete Representations Strengthen Vision Transformer Robustness. *International Conference on Learning Representations*.

Mao, Y., Zhang, T., Cao, X., Chen, Z., Liang, X., Xu, B. & Fang, H. (2024). NL2STL: Transformation from Logic Natural Language to Signal Temporal Logics using Llama2. *2024 IEEE International Conference on Cybernetics and Intelligent Systems (CIS) and IEEE International Conference on Robotics, Automation and Mechatronics (RAM)*, 469-474.

Mathews, R. P., Panicker, M. R., Hareendranathan, A. R., Chen, Y. T., Jaremko, J. L., Buchanan, B., Narayan, K. V., Chandrasekharan, K. & Mathews, G. (2022). RL Based Unsupervised Video Summarization Framework for Ultrasound Imaging. *Simplifying Medical Ultrasound - Third International Workshop, ASMUS, MICCAI*, 13565, 23–33.

Mazyavkina, N., Sviridov, S., Ivanov, S. & Burnaev, E. (2021). Reinforcement learning for combinatorial optimization: A survey. *Comput. Oper. Res.*, 134, 105400.

Mees, O., Hermann, L. & Burgard, W. (2022). What Matters in Language Conditioned Robotic Imitation Learning Over Unstructured Data. *IEEE Robotics Autom. Lett.*, 7(4), 11205–11212.

Melnychuk, V., Frauen, D. & Feuerriegel, S. (2022). Causal transformer for estimating counterfactual outcomes. *International conference on machine learning*, pp. 15293–15329.

Melo, L. C. (2022). Transformers are Meta-Reinforcement Learners. *International Conference on Machine Learning, ICML*, 162, 15340–15359.

Meng, L., Wen, M., Le, C., Li, X., Xing, D., Zhang, W., Wen, Y., Zhang, H., Wang, J., Yang, Y. & Xu, B. (2023). Offline Pre-trained Multi-agent Decision Transformer. *Mach. Intell. Res.*, 20(2), 233–248.

Merel, J., Hasenclever, L., Galashov, A., Ahuja, A., Pham, V., Wayne, G., Teh, Y. W. & Heess, N. (2019). Neural Probabilistic Motor Primitives for Humanoid Control. *International Conference on Learning Representations*.

Mesnard, T., Weber, T., Viola, F., Thakoor, S., Saade, A., Harutyunyan, A., Dabney, W., Stepleton, T. S., Heess, N., Guez, A., Moulines, E., Hutter, M., Buesing, L. & Munos, R. (2021). Counterfactual Credit Assignment in Model-Free Reinforcement Learning. *Proceedings of the 38th ICML*, 139, 7654–7664.

Meulemans, A., Schug, S., Kobayashi, S., Daw, N. & Wayne, G. (2023). Would I have gotten that reward? Long-term credit assignment by counterfactual contribution analysis. *Thirty-seventh Conference on Neural Information Processing Systems*.

Micheli, V., Alonso, E. & Fleuret, F. (2023). Transformers are Sample-Efficient World Models. *The Eleventh International Conference on Learning Representations*.

Mikhaylov, A., Mazyavkina, N., Salnikov, M., Trofimov, I., Qiang, F. & Burnaev, E. (2022). Learned Query Optimizers: Evaluation and Improvement. *IEEE Access*, 10, 75205–75218.

Milani, S., Topin, N., Veloso, M. & Fang, F. (2024). Explainable Reinforcement Learning: A Survey and Comparative Review. *ACM Comput. Surv.*, 56(7).

Mitchell, E., Rafailov, R., Peng, X. B., Levine, S. & Finn, C. (2021). Offline Meta-Reinforcement Learning with Advantage Weighting. *Proceedings of the 38th International Conference on Machine Learning, ICML*, 139, 7780–7791.

Miura, Y., Zhang, Y., Tsai, E. B., Langlotz, C. P. & Jurafsky, D. (2021). Improving Factual Completeness and Consistency of Image-to-Text Radiology Report Generation. *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, pp. 5288–5304.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S. & Hassabis, D. (2015). Human-level control through deep reinforcement learning. *Nature*, 518(7540), 529–533.

Mnih, V., Badia, A. P., Mirza, M., Graves, A., Lillicrap, T. P., Harley, T., Silver, D. & Kavukcuoglu, K. (2016). Asynchronous Methods for Deep Reinforcement Learning. *Proceedings of the 33nd International Conference on Machine Learning, ICML*, 48, 1928–1937.

Moeed, A., Hagerer, G., Dugar, S., Gupta, S., Ghosh, M., Danner, H., Mitevski, O., Nawroth, A. & Groh, G. (2020). An Evaluation of Progressive Neural Networksfor Transfer Learning in Natural Language Processing. *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pp. 1376–1381.

Moerland, T. M., Broekens, J., Plaat, A., Jonker, C. M. et al. (2023). Model-based reinforcement learning: A survey. *Foundations and Trends® in Machine Learning*, 16(1), 1–118.

Mohanty, S. P., Poonganam, J., Gaidon, A., Kolobov, A., Wulfe, B., Chakraborty, D., Semetulskis, G., Schapke, J., Kubilius, J., Pasukonis, J., Klimas, L., Hausknecht, M. J., MacAlpine, P., Tran, Q. N., Tumiel, T., Tang, X., Chen, X., Hesse, C., Hilton, J., Guss, W. H., Genc, S., Schulman, J. & Cobbe, K. (2020). Measuring Sample Efficiency and Generalization in Reinforcement Learning Benchmarks: NeurIPS 2020 Procgen Benchmark. *NeurIPS Competition and Demonstration Track*, 133, 361–395.

Moor, M., Banerjee, O., Abad, Z. S. H., Krumholz, H. M., Leskovec, J., Topol, E. J. & Rajpurkar, P. (2023). Foundation models for generalist medical artificial intelligence. *Nature*, 616(7956), 259–265.

Mor, A. & Speranza, M. G. (2022). Vehicle routing problems over time: a survey. *Ann. Oper. Res.*, 314(1), 255–275.

Motokawa, Y. & Sugawara, T. (2021). MAT-DQN: Toward Interpretable Multi-agent Deep Reinforcement Learning for Coordinated Activities. *30th International Conference on Artificial Neural Networks*, 12894, 556–567.

Motokawa, Y. & Sugawara, T. (2023). Interpretability for Conditional Coordinated Behavior in Multi-Agent Reinforcement Learning. *2023 International Joint Conference on Neural Networks (IJCNN)*, 1-8.

Mu, Y., Zhuang, Y., Wang, B., Zhu, G., Liu, W., Chen, J., Luo, P., Li, S., Zhang, C. & Hao, J. (2021). Model-based reinforcement learning via imagination with derived memory. *Advances in Neural Information Processing Systems*, 34, 9493–9505.

Mu, Y. M., Chen, S., Ding, M., Chen, J., Chen, R. & Luo, P. (2022). CtrlFormer: Learning Transferable State Representation for Visual Control via Transformer. *International Conference on Machine Learning, ICML*, 162, 16043–16061.

Nakatani, Y., Kajiwara, T. & Ninomiya, T. (2022). Comparing BERT-based Reward Functions for Deep Reinforcement Learning in Machine Translation. *Proceedings of the 9th Workshop on Asian Translation, WAT@COLING 2022, Gyeongju, Republic of Korea, October 17, 2022*, pp. 37–43.

Nakhaei, M. & Moghadam, R. A. (2025). Improving Sample Efficiency and Exploration in Upside-Down Reinforcement Learning. *Journal of Information and Intelligence*.

Nasir, Y. & Durlofsky, L. J. (2023). Deep reinforcement learning for optimal well control in subsurface systems with uncertain geology. *J. Comput. Phys.*, 477, 111945.

Nasiriany, S., Pong, V. H., Lin, S. & Levine, S. (2019). Planning with Goal-Conditioned Policies. *Proceedings of the 33rd International Conference on Neural Information Processing Systems*.

Negrinho, R., Gormley, M. R. & Gordon, G. J. (2018). Learning Beam Search Policies via Imitation Learning. *NeurIPS*, pp. 10675–10684.

Ng, A. Y., Harada, D. & Russell, S. (1999). Policy invariance under reward transformations: Theory and application to reward shaping. *Icml*, 99, 278–287.

Ng, A. Y., Russell, S. et al. (2000). Algorithms for inverse reinforcement learning. *Icml*, 1, 2.

Nguyen, S. M., Zadem, M. & Ji, Z. (2024). Compositionality in Continual Learning through Dynamic Symbolic Representation and LLM. *NeurIPS 2024 Workshop on Behavioral Machine Learning*.

Nguyen, T. T., Nguyen, N. D. & Nahavandi, S. (2020). Deep Reinforcement Learning for Multiagent Systems: A Review of Challenges, Solutions, and Applications. *IEEE Trans. Cybern.*, 50(9), 3826–3839.

Ni, J., Pandelea, V., Young, T., Zhou, H. & Cambria, E. (2022). HiTKG: Towards Goal-Oriented Conversations via Multi-Hierarchy Learning. *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI*, pp. 11112–11120.

Ni, T., Ma, M., Eysenbach, B. & Bacon, P.-L. (2024). When do transformers shine in RL? Decoupling memory from credit assignment. *Advances in Neural Information Processing Systems*, 36.

Nikishin, E., Izmailov, P., Athiwaratkun, B., Podoprikhin, D., Garipov, T., Shvechikov, P., Vetrov, D. & Wilson, A. G. (2018). Improving stability in deep reinforcement learning with weight averaging. *Uncertainty in artificial intelligence workshop on uncertainty in Deep learning*.

Ning, Z. & Xie, L. (2024). A survey on multi-agent reinforcement learning and its application. *Journal of Automation and Intelligence*.

Niu, R., Wei, Z., Wang, Y. & Wang, Q. (2022). AttExplainer: Explain Transformer via Attention by Reinforcement Learning. *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI*, pp. 724–731.

Obando-Ceron, J. S. & Castro, P. S. (2021). Revisiting Rainbow: Promoting more insightful and inclusive deep reinforcement learning research. *Proceedings of the 38th International Conference on Machine Learning, ICML*, 139, 1373–1383.

Okada, M. & Taniguchi, T. (2021). Dreaming: Model-based reinforcement learning by latent imagination without reconstruction. *2021 ieee international conference on robotics and automation (icra)*, pp. 4209–4215.

Oroojlooy, A. & Hajinezhad, D. (2023). A review of cooperative multi-agent deep reinforcement learning. *Appl. Intell.*, 53(11), 13677–13722.

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C. L., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., Schulman, J., Hilton, J., Kelton, F., Miller, L., Simens, M., Askell, A., Welinder, P., Christiano, P. F., Leike, J. & Lowe, R. (2022). Training language models to follow instructions with human feedback. *NeurIPS*.

Ozair, S., Li, Y., Razavi, A., Antonoglou, I., Van Den Oord, A. & Vinyals, O. (2021). Vector quantized models for planning. *International Conference on Machine Learning*, pp. 8302–8313.

Papineni, K., Roukos, S., Ward, T. & Zhu, W. (2002). Bleu: a Method for Automatic Evaluation of Machine Translation. *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pp. 311–318.

Parisi, G. I., Kemker, R., Part, J. L., Kanan, C. & Wermter, S. (2019). Continual lifelong learning with neural networks: A review. *Neural networks*, 113, 54–71.

Parisotto, E. & Salakhutdinov, R. (2021). Efficient Transformers in Reinforcement Learning using Actor-Learner Distillation. *9th International Conference on Learning Representations, ICLR*.

Parisotto, E., Song, H. F., Rae, J. W., Pascanu, R., Gülçehre, Ç., Jayakumar, S. M., Jaderberg, M., Kaufman, R. L., Clark, A., Noury, S., Botvinick, M. M., Heess, N. & Hadsell, R. (2020). Stabilizing Transformers for Reinforcement Learning. *Proceedings of the 37th International Conference on Machine Learning, ICML*, 119, 7487–7498.

Parmentier, A. & T'kindt, V. (2023). Structured learning based heuristics to solve the single machine scheduling problem with release times and sum of completion times. *Eur. J. Oper. Res.*, 305(3), 1032–1041.

Pascanu, R., Mikolov, T. & Bengio, Y. (2013). On the difficulty of training recurrent neural networks. *Proceedings of the 30th International Conference on Machine Learning, ICML*, 28, 1310–1318.

Paster, K., McIlraith, S. A. & Ba, J. (2022). You Can't Count on Luck: Why Decision Transformers and RvS Fail in Stochastic Environments. *Advances in Neural Information Processing Systems*.

Pathak, D., Agrawal, P., Efros, A. A. & Darrell, T. (2017). Curiosity-driven Exploration by Self-supervised Prediction. *Proceedings of the 34th International Conference on Machine Learning*, 70(Proceedings of Machine Learning Research), 2778–2787.

Paul, A. & Mitra, S. (2022). Deep reinforcement learning based cooperative control of traffic signal for multi-intersection network in intelligent transportation system using edge computing. *Trans. Emerg. Telecommun. Technol.*, 33(11).

Peng, X. B., Berseth, G., Yin, K. & Van De Panne, M. (2017). DeepLoco: Dynamic locomotion skills using hierarchical deep reinforcement learning. *ACM Transactions on Graphics (TOG)*, 36(4), 41:1–41:13.

Peng, X. B., Abbeel, P., Levine, S. & van de Panne, M. (2018). DeepMimic: Example-guided deep reinforcement learning of physics-based character skills. *ACM Transactions on Graphics (TOG)*, 37(4), 143:1–143:14.

Peng, X. B., Chang, M., Zhang, G., Abbeel, P. & Levine, S. (2019). MCP: Learning composable hierarchical control with multiplicative compositional policies. *Advances in Neural Information Processing Systems*, pp. 3681–3692.

Peng, X. B., Ma, Z., Abbeel, P., Levine, S. & Kanazawa, A. (2021). Amp: Adversarial motion priors for stylized physics-based character control. *ACM Transactions on Graphics (ToG)*, 40(4), 1–20.

Peng, X. B., Guo, Y., Halper, L., Levine, S. & Fidler, S. (2022). Ase: Large-scale reusable adversarial skill embeddings for physically simulated characters. *ACM Transactions On Graphics (TOG)*, 41(4), 1–17.

Pertsch, K., Lee, Y. & Levine, S. (2021). Accelerating reinforcement learning with learned skill priors. *Conference on Robot Learning*, pp. 188–204.

Petit, O., Thome, N., Rambour, C., Themyr, L., Collins, T. & Soler, L. (2021). U-Net Transformer: Self and Cross Attention for Medical Image Segmentation. *Machine Learning in Medical Imaging - 12th International Workshop, MLMI, MICCAI*, 12966, 267–276.

Plaat, A., Kosters, W. & Preuss, M. (2023). High-accuracy model-based reinforcement learning, a survey. *Artificial Intelligence Review*, 1–33.

Polydoros, A. S. & Nalpantidis, L. (2017). Survey of model-based reinforcement learning: Applications on robotics. *Journal of Intelligent & Robotic Systems*, 86(2), 153–173.

Prakash, A., Chitta, K. & Geiger, A. (2021). Multi-Modal Fusion Transformer for End-to-End Autonomous Driving. *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pp. 7077–7087.

Qu, J., Miwa, S. & Domae, Y. (2022). Interpretable Navigation Agents Using Attention-Augmented Memory. *IEEE International Conference on Systems, Man, and Cybernetics, SMC 2022, Prague, Czech Republic, October 9-12, 2022*, pp. 2575–2582.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I. et al. (2019). Language models are unsupervised multitask learners. *OpenAI blog*, 1(8), 9.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J. et al. (2021). Learning transferable visual models from natural language supervision. *International Conference on Machine Learning*, pp. 8748–8763.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W. & Liu, P. J. (2020). Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *J. Mach. Learn. Res.*, 21, 140:1–140:67.

Rakelly, K., Zhou, A., Finn, C., Levine, S. & Quillen, D. (2019). Efficient Off-Policy Meta-Reinforcement Learning via Probabilistic Context Variables. *Proceedings of the 36th International Conference on Machine Learning, ICML*, 97, 5331–5340.

Ramachandram, D. & Taylor, G. W. (2017). Deep Multimodal Learning: A Survey on Recent Advances and Trends. *IEEE Signal Process. Mag.*, 34(6), 96–108.

Ren, H., Dai, H., Dai, Z., Yang, M., Leskovec, J., Schuurmans, D. & Dai, B. (2021). Combiner: Full Attention Transformer with Sparse Computation Cost. *NeurIPS*, pp. 22470–22482.

Rengarajan, D., Vaidya, G., Sarvesh, A., Kalathil, D. & Shakkottai, S. (2022). Reinforcement Learning with Sparse Rewards using Guidance from Offline Demonstration. *International Conference on Learning Representations*.

Ribeiro, A. H., Tiels, K., Aguirre, L. A. & Schön, T. B. (2020). Beyond exploding and vanishing gradients: analysing RNN training using attractors and smoothness. *The 23rd International Conference on Artificial Intelligence and Statistics, AISTATS*, 108, 2370–2380.

Riedmiller, M., Hafner, R., Lampe, T., Neunert, M., Degrave, J., van de Wiele, T., Mnih, V., Heess, N. & Springenberg, J. T. (2018). Learning by Playing Solving Sparse Reward Tasks from Scratch. *Proceedings of the 35th International Conference on Machine Learning*, 80, 4344–4353.

Riemer, M., Cases, I., Ajemian, R., Liu, M., Rish, I., Tu, Y., & Tesauro, G. (2019). Learning to Learn without Forgetting By Maximizing Transfer and Minimizing Interference. *International Conference on Learning Representations*.

Robine, J., Höftmann, M., Uelwer, T. & Harmeling, S. (2023a). Transformer-based World Models Are Happy With 100k Interactions. *The Eleventh International Conference on Learning Representations*.

Robine, J., Uelwer, T. & Harmeling, S. (2023b). Smaller world models for reinforcement learning. *Neural Processing Letters*, 1–31.

Rogers, A., Kovaleva, O. & Rumshisky, A. (2020). A Primer in BERTology: What We Know About How BERT Works. *Trans. Assoc. Comput. Linguistics*, 8, 842–866.

Rosbach, S., James, V., Großjohann, S., Homoceanu, S. & Roth, S. (2019). Driving with Style: Inverse Reinforcement Learning in General-Purpose Planning for Automated Driving. *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2658-2665.

Rummery, G. A. & Niranjan, M. (1994). *On-line Q-learning using connectionist systems*. University of Cambridge, Department of Engineering Cambridge, UK.

Rusu, A. A., Colmenarejo, S. G., Gulcehre, C., Desjardins, G., Kirkpatrick, J., Pascanu, R., Mnih, V., Kavukcuoglu, K. & Hadsell, R. (2016). Policy distillation. *International Conference on Learning Representations*.

Saha, T., Gakhreja, V., Das, A. S., Chakraborty, S. & Saha, S. (2022). Towards Motivational and Empathetic Response Generation in Online Mental Health Support. *SIGIR '22: The 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 2650–2656.

Samad, A. M., Mishra, K., Firdaus, M. & Ekbal, A. (2022). Empathetic Persuasion: Reinforcing Empathy and Persuasiveness in Dialogue Systems. *Findings of the Association for Computational Linguistics: NAACL*, pp. 844–856.

Savva, M., Malik, J., Parikh, D., Batra, D., Kadian, A., Maksymets, O., Zhao, Y., Wijmans, E., Jain, B., Straub, J., Liu, J. & Koltun, V. (2019). Habitat: A Platform for Embodied AI Research. *IEEE/CVF International Conference on Computer Vision, ICCV*, pp. 9338–9346.

Schrittwieser, J., Antonoglou, I., Hubert, T., Simonyan, K., Sifre, L., Schmitt, S., Guez, A., Lockhart, E., Hassabis, D., Graepel, T. et al. (2020). Mastering atari, go, chess and shogi by planning with a learned model. *Nature*, 588(7839), 604–609.

Schwarz, J., Czarnecki, W., Luketina, J., Grabska-Barwinska, A., Teh, Y. W., Pascanu, R. & Hadsell, R. (2018). Progress & compress: A scalable framework for continual learning. *International conference on machine learning*, pp. 4528–4537.

Schwarzer, M., Anand, A., Goel, R., Hjelm, R. D., Courville, A. C. & Bachman, P. (2020). Data-Efficient Reinforcement Learning with Self-Predictive Representations. *International Conference on Learning Representations*.

Schwarzer, M., Anand, A., Goel, R., Hjelm, R. D., Courville, A. & Bachman, P. (2021). Data-Efficient Reinforcement Learning with Self-Predictive Representations. *International Conference on Learning Representations*.

Seo, M., Vecchietti, L. F., Lee, S. & Har, D. (2019). Rewards Prediction-Based Credit Assignment for Reinforcement Learning With Sparse Binary Rewards. *IEEE Access*, 7, 118776–118791.

Seo, Y., Hafner, D., Liu, H., Liu, F., James, S., Lee, K. & Abbeel, P. (2023a, Dec). Masked World Models for Visual Control. *Proceedings of The 6th Conference on Robot Learning*, 205(Proceedings of Machine Learning Research), 1332–1344.

Seo, Y., Hafner, D., Liu, H., Liu, F., James, S., Lee, K. & Abbeel, P. (2023b). Masked world models for visual control. *Conference on Robot Learning*, pp. 1332–1344.

Serrano, S. & Smith, N. A. (2019). Is Attention Interpretable? *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL*, pp. 2931–2951.

Shamshad, F., Khan, S. H., Zamir, S. W., Khan, M. H., Hayat, M., Khan, F. S. & Fu, H. (2022). Transformers in Medical Imaging: A Survey. *Medical image analysis*, 88, 102802.

Shang, J., Kahatapitiya, K., Li, X. & Ryoo, M. S. (2022). StARformer: Transformer with State-Action-Reward Representations for Visual Reinforcement Learning. *Computer Vision - ECCV - 17th European Conference*, 13699, 462–479.

Shang, J., Ma, T., Xiao, C. & Sun, J. (2019). Pre-training of Graph Augmented Transformers for Medication Recommendation. *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, pp. 5953–5959.

Shao, K., Zhu, Y. & Zhao, D. (2019). StarCraft Micromanagement With Reinforcement Learning and Curriculum Transfer Learning. *IEEE Trans. Emerg. Top. Comput. Intell.*, 3(1), 73–84.

Sharma, A., Gu, S., Levine, S., Kumar, V. & Hausman, K. (2020). Dynamics-aware unsupervised discovery of skills. *International Conference on Learning Representations*.

Shawki, N., Nunez, R. R., Obeid, I. & Picone, J. (2021). On automating hyperparameter optimization for deep learning applications. *2021 IEEE Signal Processing in Medicine and Biology Symposium (SPMB)*, pp. 1–7.

Shazeer, N., Mirhoseini, A., Maziarz, K., Davis, A., Le, Q., Hinton, G. & Dean, J. (2017a). The sparsely-gated mixture-of-experts layer. *Outrageously large neural networks*, 2.

Shazeer, N., Mirhoseini, A., Maziarz, K., Davis, A., Le, Q., Hinton, G. & Dean, J. (2017b). Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *International Conference on Learning Representations*.

She, J., Gupta, J. K. & Kochenderfer, M. J. (2022). Agent-Time Attention for Sparse Rewards Multi-Agent Reinforcement Learning. *21st International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2022, Auckland, New Zealand, May 9-13, 2022*, pp. 1723–1725.

Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., van den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M. et al. (2016). Mastering the game of Go with deep neural networks and tree search. *Nature*, 529(7587), 484–489.

Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L., Lai, M., Bolton, A., Chen, Y., Lillicrap, T. P., Hui, F., Sifre, L., van den Driessche, G., Graepel, T. & Hassabis, D. (2017). Mastering the game of Go without human knowledge. *Nat.*, 550(7676), 354–359.

Smith, C. (2022). *Attention-Based Learning for Combinatorial Optimization*. (Ph.D. thesis, Massachusetts Institute of Technology).

Sniedovich, M. (2010). *Dynamic programming: foundations and principles*. CRC press.

Snoek, J., Rippel, O., Swersky, K., Kiros, R., Satish, N., Sundaram, N., Patwary, M. M. A., Prabhat & Adams, R. P. (2015). Scalable Bayesian Optimization Using Deep Neural Networks. *Proceedings of the 32nd ICML*, 37, 2171–2180.

Song, H., Li, A., Wang, T. & Wang, M. (2021). Multimodal Deep Reinforcement Learning with Auxiliary Task for Obstacle Avoidance of Indoor Mobile Robot. *Sensors*, 21(4), 1363.

Sopov, V. & Makarov, I. (2021). Transformer-Based Deep Reinforcement Learning in VizDoom. *Recent Trends in Analysis of Images, Social Networks and Texts - 10th International Conference*, 1573, 96–110.

Starke, S., Starke, P., He, N., Komura, T. & Ye, Y. (2024). Categorical Codebook Matching for Embodied Character Controllers. *ACM Transactions on Graphics (TOG)*, 43(4), 1–14.

Strnad, F. M., Barfuss, W., Donges, J. F. & Heitzig, J. (2019). Deep reinforcement learning in World-Earth system models to discover sustainable management strategies. *Chaos*, 29 12, 123122.

Sutton, R. S. (1991). Dyna, an integrated architecture for learning, planning, and reacting. *ACM Sigart Bulletin*, 2(4), 160–163.

Sutton, R. S. & Barto, A. G. (1998). *Reinforcement learning - an introduction*. MIT Press.

Sutton, R. S., Precup, D. & Singh, S. (1999). Between MDPs and Semi-MDPs: A Framework for Temporal Abstraction in Reinforcement Learning. *Proceedings of the 12th International Conference on Machine Learning (ICML)*, pp. 282–290.

Svidchenko, O. & Shpilman, A. (2021). Maximum Entropy Model-based Reinforcement Learning. *Deep RL Workshop NeurIPS 2021*.

Ta, V.-D., Liu, C.-M. & Tadesse, D. A. (2020). Portfolio optimization-based stock prediction using long-short term memory network in quantitative trading. *Applied Sciences*, 10(2), 437.

Talvitie, E. (2017). Self-correcting models for model-based reinforcement learning. *Proceedings of the AAAI conference on artificial intelligence*, 31(1).

Tang, C.-Y., Liu, C.-H., Chen, W.-K. & You, S. D. (2020). Implementing action mask in proximal policy optimization (PPO) algorithm. *ICT Express*, 6(3), 200–203.

Tang, J., Liang, Y. & Li, K. (2024). Dynamic Scene Path Planning of UAVs Based on Deep Reinforcement Learning. *Drones*.

Tang, Y., Yang, D., Li, W., Roth, H. R., Landman, B. A., Xu, D., Nath, V. & Hatamizadeh, A. (2022). Self-Supervised Pre-Training of Swin Transformers for 3D Medical Image Analysis. *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR*, pp. 20698–20708.

Taylor, M. E. & Stone, P. (2009). Transfer learning for reinforcement learning domains: A survey. *Journal of Machine Learning Research*, 10, 1633–1685.

Tessler, C., Guo, Y., Nabati, O., Chechik, G. & Peng, X. B. (2024). Maskedmimic: Unified physics-based character control through masked motion inpainting. *ACM Transactions on Graphics (TOG)*, 43(6), 1–21.

Thakkar, A. & Chaudhari, K. (2021). A comprehensive survey on portfolio optimization, stock price and trend prediction using particle swarm optimization. *Archives of Computational Methods in Engineering*, 28, 2133–2164.

Todorov, E., Erez, T. & Tassa, Y. (2012). MuJoCo: A physics engine for model-based control. *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2012, Vilamoura, Algarve, Portugal, October 7-12, 2012*, pp. 5026–5033.

Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A. & Jégou, H. (2021). Training data-efficient image transformers & distillation through attention. *Proceedings of the 38th International Conference on Machine Learning, ICML*, 139, 10347–10357.

Tunstall, L., von Werra, L. & Wolf, T. (2022). *Natural language processing with transformers*. " O'Reilly Media, Inc.".

Van Den Oord, A., Vinyals, O. & Kavukcuoglu, K. (2017). Neural discrete representation learning. *Advances in neural information processing systems*, 30.

van Hasselt, H. (2010). Double Q-learning. *NeurIPS*, pp. 2613–2621.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. & Polosukhin, I. (2017). Attention is All you Need. *30th Annual Conference on Neural Information Processing Systems*, pp. 5998–6008.

Vedantam, R., Zitnick, C. L. & Parikh, D. (2015). CIDEr: Consensus-based image description evaluation. *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, pp. 4566–4575.

Versaw, R., Schultz, S., Lu, K. & Zhao, R. (2021). Modular Reinforcement Learning Framework for Learners and Educators. *Proceedings of the 16th International Conference on the Foundations of Digital Games*, pp. 1–5.

Villaflor, A. R., Huang, Z., Pande, S., Dolan, J. M. & Schneider, J. (2022). Addressing Optimism Bias in Sequence Modeling for Reinforcement Learning. *International Conference on Machine Learning, ICML*, 162, 22270–22283.

Villarrubia-Martin, E. A., Rodriguez-Benitez, L., Jimenez-Linares, L., Munoz-Valero, D. & Liu, J. (2023). A hybrid online off-policy reinforcement learning agent framework supported by transformers. *International Journal of Neural Systems*, 33(12), 2350065.

Vinyals, O., Babuschkin, I., Czarnecki, W. M., Mathieu, M., Dudzik, A., Chung, J., Choi, D. H., Powell, R., Ewalds, T., Georgiev, P., Oh, J., Horgan, D., Kroiss, M., Danihelka, I., Huang, A., Sifre, L., Cai, T., Agapiou, J. P., Jaderberg, M., Vezhnevets, A. S., Leblond, R., Pohlen, T., Dalibard, V., Budden, D., Sulsky, Y., Molloy, J., Paine, T. L., Gülçehre, Ç., Wang, Z., Pfaff, T., Wu, Y., Ring, R., Yogatama, D., Wünsch, D., McKinney, K., Smith, O., Schaul, T., Lillicrap, T. P., Kavukcuoglu, K., Hassabis, D., Apps, C. & Silver, D. (2019). Grandmaster level in StarCraft II using multi-agent reinforcement learning. *Nat.*, 575(7782), 350–354.

Vo, Q., Nguyen, H., Le, B. & Nguyen, M. (2017). Multi-channel LSTM-CNN model for Vietnamese sentiment analysis. *9th International Conference on Knowledge and Systems Engineering, KSE 2017, Hue, Vietnam, October 19-21, 2017*, pp. 24–29.

Vollmer, A.-L., Wrede, B., Rohlfing, K. J. & Oudeyer, P.-Y. (2016). Pragmatic frames for teaching and learning in human–robot interaction: Review and challenges. *Frontiers in neurorobotics*, 10, 10.

Wang, C., Wang, J., Wang, J. & Zhang, X. (2020a). Deep-reinforcement-learning-based autonomous UAV navigation with sparse rewards. *IEEE Internet of Things Journal*, 7(7), 6180–6190.

Wang, J., Wei, L., Wang, L., Zhou, Q., Zhu, L. & Qin, J. (2021a). Boundary-Aware Transformers for Skin Lesion Segmentation. *Medical Image Computing and Computer Assisted Intervention - MICCAI 24th International Conference*, 12901, 206–216.

Wang, J., Li, W., Jiang, H., Zhu, G., Li, S. & Zhang, C. (2021b). Offline reinforcement learning with reverse model-based imagination. *Advances in Neural Information Processing Systems*, 34, 29420–29432.

Wang, J., Zhao, H., Liu, H., Geng, L. & Sun, Z. (2022a). A Distributed Vehicle-assisted Computation Offloading Scheme based on DRL in Vehicular Networks. *22nd IEEE International Symposium on Cluster, Cloud and Internet Computing, CCGrid 2022, Taormina, Italy, May 16-19, 2022*, pp. 200–209.

Wang, J., Hsieh, C., Wang, M., Wang, X., Wu, Z., Jiang, D., Liao, B., Zhang, X., Yang, B., He, Q., Cao, D., Chen, X. & Hou, T. (2021c). Multi-constraint molecular generation based on conditional transformer, knowledge distillation and reinforcement learning. *Nat. Mach. Intell.*, 3(10), 914–922.

Wang, K., Zhao, H., Luo, X., Ren, K., Zhang, W. & Li, D. (2022b). Bootstrapped Transformer for Offline Reinforcement Learning. *NeurIPS*.

Wang, L., Yang, Z. & Wang, Z. (2020b). Breaking the Curse of Many Agents: Provable Mean Embedding Q-Iteration for Mean-Field Reinforcement Learning. *Proceedings of the 37th International Conference on Machine Learning, ICML*, 119, 10092–10103.

Wang, M., Feng, M., Zhou, W. & Li, H. (2022c). Stabilizing Voltage in Power Distribution Networks via Multi-Agent Reinforcement Learning with Transformer. *KDD '22: The 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, August 14 - 18, 2022*, pp. 1899–1909.

Wang, P., Zhu, M. & Shen, S. (2024). Environment transformer and policy optimization for model-based offline reinforcement learning. *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 12625–12631.

Wang, Q., Lai, K. H. & Tang, C. (2023a). Solving combinatorial optimization problems over graphs with BERT-Based Deep Reinforcement Learning. *Inf. Sci.*, 619, 930–946.

Wang, T., Liao, R., Ba, J. & Fidler, S. (2018). NerveNet: Learning Structured Policy with Graph Neural Networks. *6th International Conference on Learning Representations, ICLR*.

Wang, W., Yang, T., Liu, Y., Hao, J., Hao, X., Hu, Y., Chen, Y., Fan, C. & Gao, Y. (2020c). From Few to More: Large-Scale Dynamic Multiagent Curriculum Learning. *The Thirty-Fourth Conference on Artificial Intelligence, AAAI*, pp. 7293–7300.

Wang, Y. & Chen, Z. (2022). A Deep Reinforcement Learning Algorithm Using A New Graph Transformer Model for Routing Problems. *Intelligent Systems and Applications - Proceedings of the Intelligent Systems Conference, IntelliSys*, 544, 365–379.

Wang, Y., Min, Z. & Jia, S. (2021d). Local-Global-Aware Convolutional Transformer for Hyperspectral Image Classification. *23rd Int Conf on High Performance Computing*, pp. 1188–1194.

Wang, Y., Xu, M., Shi, L. & Chi, Y. (2023b, 31 Jul–04 Aug). A trajectory is worth three sentences: multimodal transformer for offline reinforcement learning. *Proceedings of the Thirty-Ninth Conference on Uncertainty in Artificial Intelligence*, 216(Proceedings of Machine Learning Research), 2226–2236.

Wang, Y., He, H. & Tan, X. (2020d, 22–25 Jul). Truly Proximal Policy Optimization. *Proceedings of The 35th Uncertainty in Artificial Intelligence Conference*, 115(Proceedings of Machine Learning Research), 113–122.

Wang, Z., Cun, X., Bao, J., Zhou, W., Liu, J. & Li, H. (2022d). Uformer: A General U-Shaped Transformer for Image Restoration. *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR*, pp. 17662–17672.

Wang, Z., Wang, J., Zhou, Q., Li, B. & Li, H. (2022e). Sample-Efficient Reinforcement Learning via Conservative Model-Based Actor-Critic. *Thirty-Sixth AAAI Conference on Artificial Intelligence*, pp. 8612–8620.

Watkins, C. J. C. H. & Dayan, P. (1992). Technical Note Q-Learning. *Mach. Learn.*, 8, 279–292.

Wei, B., Wang, M., Zhou, H., Lin, J. & Sun, X. (2019). Imitation Learning for Non-Autoregressive Neural Machine Translation. *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL*, pp. 1304–1312.

Wei, J., Tay, Y., Bommasani, R., Raffel, C., Zoph, B., Borgeaud, S., Yogatama, D., Bosma, M., Zhou, D., Metzler, D., Chi, E. H., Hashimoto, T., Vinyals, O., Liang, P., Dean, J. & Fedus, W. (2022a). Emergent Abilities of Large Language Models. *Trans. Mach. Learn. Res.*, 2022.

Wei, X., Huang, X., Yang, L., Cao, G., Tao, Z., Wang, B. & An, J. (2022b). Hierarchical RNNs-Based transformers MADDPG for mixed cooperative-competitive environments. *J. Intell. Fuzzy Syst.*, 43(1), 1011–1022.

Wen, G., Fu, J., Dai, P. & Zhou, J. (2021). DTDE: A new cooperative multi-agent reinforcement learning framework. *The Innovation*, 2(4).

Wen, M., Kuba, J. G., Lin, R., Zhang, W., Wen, Y., Wang, J. & Yang, Y. (2022). Multi-Agent Reinforcement Learning is a Sequence Modeling Problem. *NeurIPS*.

Williams, R. J. (1992). Simple Statistical Gradient-Following Algorithms for Connectionist Reinforcement Learning. *Mach. Learn.*, 8, 229–256.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., von Platen, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Scao, T. L., Gugger, S., Drame, M., Lhoest, Q. & Rush, A. M. (2020). Transformers: State-of-the-Art Natural Language Processing. *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, EMNLP 2020 - Demos, Online, November 16-20, 2020*, pp. 38–45.

Won, J. & Lee, J. (2019). Learning body shape variation in physics-based characters. *ACM Transactions on Graphics (TOG)*, 38(6), 207:1–207:12.

Wong, A., Bäck, T., Kononova, A. V. & Plaat, A. (2023). Deep multiagent reinforcement learning: challenges and directions. *Artif. Intell. Rev.*, 56(6), 5023–5056.

Woo, S., Debnath, S., Hu, R., Chen, X., Liu, Z., Kweon, I. S. & Xie, S. (2023). Convnext v2: Co-designing and scaling convnets with masked autoencoders. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 16133–16142.

Wu, N., Xie, Y. & Hao, C. (2023). IronMan-Pro: Multiobjective Design Space Exploration in HLS via Reinforcement Learning and Graph Neural Network-Based Modeling. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 42, 900-913.

Wu, S., Haque, K. I. & Yumak, Z. (2024). ProbTalk3D: Non-Deterministic Emotion Controllable Speech-Driven 3D Facial Animation Synthesis Using VQ-VAE. *Proceedings of the 17th ACM SIGGRAPH Conference on Motion, Interaction, and Games*, pp. 1–12.

Wu, Y., Song, W., Cao, Z., Zhang, J. & Lim, A. (2022). Learning Improvement Heuristics for Solving Routing Problems. *IEEE Trans. Neural Networks Learn. Syst.*, 33(9), 5057–5069.

Wulfmeier, M., Ondruska, P. & Posner, I. (2016). Deep maximum entropy inverse reinforcement learning. *NIPS Deep Reinforcement Learning Workshop*.

Xiao, T., Singh, M., Mintun, E., Darrell, T., Dollár, P. & Girshick, R. B. (2021). Early Convolutions Help Transformers See Better. *NeurIPS*, pp. 30392–30400.

Xiao, X., Sun, Z., Li, T. & Yu, Y. (2022). Relational Graph Reasoning Transformer for Image Captioning. *IEEE International Conference on Multimedia and Expo, ICME*, pp. 1–6.

Xie, Q., Ma, X., Dai, Z. & Hovy, E. H. (2017). An Interpretable Knowledge Transfer Model for Knowledge Base Completion. *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada*, pp. 950–962.

Xiong, R., Yang, Y., He, D., Zheng, K., Zheng, S., Xing, C., Zhang, H., Lan, Y., Wang, L. & Liu, T. (2020). On layer normalization in the transformer architecture. *International Conference on Machine Learning*, pp. 10524–10533.

Xiong, Y., Du, B. & Yan, P. (2019). Reinforced Transformer for Medical Image Captioning. *Machine Learning in Medical Imaging - 10th International Workshop, MLMI, MICCAI*, 11861, 673–680.

Xu, K., Zhang, Y., Ye, D., Zhao, P. & Tan, M. (2020a). Relation-Aware Transformer for Portfolio Policy Learning. *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020*, pp. 4647–4653.

Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A. C., Salakhutdinov, R., Zemel, R. S. & Bengio, Y. (2015). Show, Attend and Tell: Neural Image Caption Generation with Visual Attention. *Proceedings of the 32nd ICML*, 37, 2048–2057.

Xu, M., Shen, Y., Zhang, S., Lu, Y., Zhao, D., Tenenbaum, J. B. & Gan, C. (2022a). Prompting Decision Transformer for Few-Shot Policy Generalization. *International Conference on Machine Learning, ICML*, 162, 24631–24645.

Xu, N., Chang, J., Nie, X., Huo, C., Xiang, S. & Pan, C. (2022b). AME: Attention and Memory Enhancement in Hyper-Parameter Optimization. *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR*, pp. 480–489.

Xu, P., Yin, Q., Zhang, J. & Huang, K. (2022c). Deep Reinforcement Learning With Part-Aware Exploration Bonus in Video Games. *IEEE Transactions on Games*, 14, 644-653.

Xu, P., Shang, X., Zordan, V. & Karamouzas, I. (2023a). Composite motion learning with task control. *ACM Transactions on Graphics (TOG)*, 42(4), 1–16.

Xu, P., Xie, K., Andrews, S., Kry, P. G., Neff, M., Mcguire, M., Karamouzas, I. & Zordan, V. (2023b). AdaptNet: Policy Adaptation for Physics-Based Character Control. *ACM Trans. Graph.*, 42(6). doi: 10.1145/3618375.

Xu, P., Kumar, D., Yang, W., Zi, W., Tang, K., Huang, C., Cheung, J. C. K., Prince, S. J. D. & Cao, Y. (2021). Optimizing Deeper Transformers on Small Datasets. *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP*, pp. 2089–2102.

Xu, P., Zhu, X. & Clifton, D. A. (2023c). Multimodal learning with transformers: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(10), 12113–12132.

Xu, R., Yang, K., Liu, K. & He, F. (2023d). $E(2)$-Equivariant Vision Transformer. *Uncertainty in Artificial Intelligence*, pp. 2356–2366.

Xu, Y., Chen, L., Fang, M., Wang, Y. & Zhang, C. (2020b). Deep Reinforcement Learning with Transformers for Text Adventure Games. *IEEE Conference on Games, CoG 2020, Osaka, Japan, August 24-27, 2020*, pp. 65–72.

Yamagata, T., Khalil, A. & Santos-Rodriguez, R. (2023). Q-learning decision transformer: Leveraging dynamic programming for conditional sequence modelling in offline rl. *International Conference on Machine Learning*, pp. 38989–39007.

Yan, D., Yu, W., Zhang, Z. & Gong, J. (2021a). Transformer with Prior Language Knowledge for Image Captioning. *Neural Information Processing - 28th International Conference, ICONIP*, 13109, 40–51.

Yan, S., Xie, J. & He, X. (2021b). Der: Dynamically expandable representation for class incremental learning. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 3014–3023.

Yang, B., Yang, J., Ni, R., Yang, C. & Liu, X. (2022a). Multi-granularity scenarios understanding network for trajectory prediction. *Complex & Intelligent Systems*, 1–14.

Yang, F., Jin, T., Liu, T., Sun, X. & Zhang, J. (2018). Boosting Dynamic Programming with Neural Networks for Solving NP-hard Problems. *Proceedings of The 10th Asian Conference on Machine Learning, ACML*, 95, 726–739.

Yang, L., Yang, G., Bing, Z., Tian, Y., Niu, Y., Huang, L. & Yang, L. (2021). Transformer-Based Generative Model Accelerating the Development of Novel BRAF Inhibitors. *ACS Omega*, 6, 33864 - 33873.

Yang, Y., Xing, D. & Xu, B. (2022b). Efficient Spatiotemporal Transformer for Robotic Reinforcement Learning. *IEEE Robotics Autom. Lett.*, 7(3), 7982–7989.

Yao, H., Song, Z., Zhou, Y., Ao, T., Chen, B. & Liu, L. (2024). MoConVQ: Unified Physics-Based Motion Control via Scalable Discrete Representations. *ACM Trans. Graph.*, 43(4).

Yao, T., Pan, Y., Li, Y., Ngo, C.-W. & Mei, T. (2022). Wave-vit: Unifying wavelet and transformers for visual representation learning. *European conference on computer vision*, pp. 328–345.

Yarats, D., Kostrikov, I. & Fergus, R. (2021a). Image Augmentation Is All You Need: Regularizing Deep Reinforcement Learning from Pixels. *International Conference on Learning Representations*.

Yarats, D., Zhang, A., Kostrikov, I., Amos, B., Pineau, J. & Fergus, R. (2021b). Improving Sample Efficiency in Model-Free Reinforcement Learning from Images. *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI*, pp. 10674–10681.

Ye, W., Liu, S., Kurutach, T., Abbeel, P. & Gao, Y. (2021). Mastering atari games with limited data. *Advances in Neural Information Processing Systems*, 34, 25476–25488.

Yin, Z.-H., Ye, W., Chen, Q. & Gao, Y. (2022). Planning for sample efficient imitation learning. *Advances in Neural Information Processing Systems*, 35, 2577–2589.

Ying, C., Cai, T., Luo, S., Zheng, S., Ke, G., He, D., Shen, Y. & Liu, T. (2021). Do Transformers Really Perform Badly for Graph Representation? *NeurIPS*, pp. 28877–28888.

Yoon, J., Wu, Y.-F., Bae, H. & Ahn, S. (2023). An Investigation into Pre-Training Object-Centric Representations for Reinforcement Learning. *International Conference on Machine Learning*.

Yu, C., Yang, X., Gao, J., Yang, H., Wang, Y. & Wu, Y. (2022). Learning Efficient Multi-agent Cooperative Visual Exploration. *Computer Vision - ECCV - 17th European Conference*, 13699, 497–515.

Yu, C., Liu, J., Nemati, S. & Yin, G. (2023). Reinforcement Learning in Healthcare: A Survey. *ACM Comput. Surv.*, 55(2), 5:1–5:36.

Yu, T., Kumar, S., Gupta, A., Levine, S., Hausman, K. & Finn, C. (2019). Multi-task reinforcement learning without interference. *Proc. Optim. Found. Reinforcement Learn. Workshop NeurIPS*.

Yu, T., Quillen, D., He, Z., Julian, R., Hausman, K., Finn, C. & Levine, S. (2020). Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. *Conference on robot learning*, pp. 1094–1100.

Yu, Y. (2018). Towards Sample Efficient Reinforcement Learning. *IJCAI*, pp. 5739–5743.

Yuan, L., Chen, Y., Wang, T., Yu, W., Shi, Y., Jiang, Z., Tay, F. E. H., Feng, J. & Yan, S. (2021). Tokens-to-Token ViT: Training Vision Transformers from Scratch on ImageNet. *IEEE/CVF International Conference on Computer Vision, ICCV*, pp. 538–547.

Yun, C., Bhojanapalli, S., Rawat, A. S., Reddi, S. & Kumar, S. (2020). Are Transformers universal approximators of sequence-to-sequence functions? *International Conference on Learning Representations*.

Zambaldi, V. F., Raposo, D., Santoro, A., Bapst, V., Li, Y., Babuschkin, I., Tuyls, K., Reichert, D. P., Lillicrap, T. P., Lockhart, E., Shanahan, M., Langston, V., Pascanu, R., Botvinick, M. M., Vinyals, O. & Battaglia, P. W. (2018a). Deep reinforcement learning with relational inductive biases. *International Conference on Learning Representations*.

Zambaldi, V. F., Raposo, D., Santoro, A., Bapst, V., Li, Y., Babuschkin, I., Tuyls, K., Reichert, D. P., Lillicrap, T. P., Lockhart, E., Shanahan, M., Langston, V., Pascanu, R., Botvinick, M. M., Vinyals, O. & Battaglia, P. W. (2018b). Deep reinforcement learning with relational inductive biases. *International Conference on Learning Representations*.

Zavodni, O., Nojpert, J. & Arnold, E. (2009). Actual trends in crane automation: Directions for the future. *FME Transactions*, 37(4), 167–174.

Zenke, F., Poole, B. & Ganguli, S. (2017). Continual learning through synaptic intelligence. *International conference on machine learning*, pp. 3987–3995.

Zhai, X., Kolesnikov, A., Houlsby, N. & Beyer, L. (2022). Scaling Vision Transformers. *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR*, pp. 1204–1213.

Zhan, Y., Bou-Ammar, H. & Taylor, M. E. (2017). Scalable lifelong reinforcement learning. *Pattern Recognit.*, 72, 407–418.

Zhang, F., Liu, B., Wang, K., Tan, V. Y. F., Yang, Z. & Wang, Z. (2022a). Relational Reasoning via Set Transformers: Provable Efficiency and Applications to MARL. *NeurIPS*.

Zhang, H., Wang, H. & Kan, Z. (2023a). Exploiting Transformer in Sparse Reward Reinforcement Learning for Interpretable Temporal Logic Motion Planning. *IEEE Robotics and Automation Letters*, 8(8), 4831-4838.

Zhang, J., Zhao, T. & Yu, Z. (2018a). Multimodal Hierarchical Reinforcement Learning Policy for Task-Oriented Visual Dialog. *Proceedings of the 19th Annual SIGdial Meeting on Discourse and Dialogue, Melbourne, Australia*, pp. 140–150.

Zhang, J., Nie, Y., Chang, J. & Zhang, J. (2022b). SIG-Former: monocular surgical instruction generation with transformers. *Int. J. Comput. Assist. Radiol. Surg.*, 17(12), 2203–2210.

Zhang, K., Yang, Z., Liu, H., Zhang, T. & Basar, T. (2018b). Fully Decentralized Multi-Agent Reinforcement Learning with Networked Agents. *Proceedings of the 35th ICML*, 80, 5867–5876.

Zhang, K., Lin, X. & Li, M. (2022c). Transformer-Based Reinforcement Learning for Pickup and Delivery Problems With Late Penalties. *IEEE Trans. Intell. Transp. Syst.*, 23(12), 24649–24661.

Zhang, L., Zhao, J., Long, P., Wang, L., Qian, L., Lu, F., Song, X. & Manocha, D. (2021). An autonomous excavator system for material loading tasks. *Science Robotics*, 6(55).

Zhang, M., Zhou, G., Yu, W., Huang, N. & Liu, W. (2023b). GA-SCS: Graph-Augmented Source Code Summarization. *ACM Trans. Asian Low Resour. Lang. Inf. Process.*, 22(2), 53:1–53:19.

Zhang, T., Hu, X., Xiao, J. & Zhang, G. (2022d). TVENet: Transformer-Based Visual Exploration Network for Mobile Robot in Unseen Environment. *IEEE Access*, 10, 62056–62072.

Zhang, W., Wang, G., Sun, J., Yuan, Y. & Huang, G. (2023c). STORM: Efficient Stochastic Transformer based World Models for Reinforcement Learning. *Thirty-seventh Conference on Neural Information Processing Systems*.

Zhang, Y. & Wang, T. (2022). Applying Value-Based Deep Reinforcement Learning on KPI Time Series Anomaly Detection. *IEEE 15th International Conference on Cloud Computing, CLOUD*, pp. 197–202.

Zhang, Z., Wei, Z., Huang, Z., Niu, R. & Wang, P. (2023d). One for all: One-stage referring expression comprehension with dynamic reasoning. *Neurocomputing*, 518, 523–532.

Zhao, W., Queralta, J. P. & Westerlund, T. (2020). Sim-to-real transfer in deep reinforcement learning for robotics: a survey. *2020 IEEE symposium series on computational intelligence (SSCI)*, pp. 737–744.

Zheng, B., Zheng, R., Ma, M. & Huang, L. (2019). Simultaneous Translation with Flexible Policy via Restricted Imitation Learning. *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL*, pp. 5816–5822.

Zheng, B., Verma, S., Zhou, J., Tsang, I. W. & Chen, F. (2022). Imitation Learning: Progress, Taxonomies and Challenges. *IEEE Transactions on Neural Networks and Learning Systems*, 1-16. doi: 10.1109/TNNLS.2022.3213246.

Zhong, D., Yang, Y. & Zhao, Q. (2024). No Prior Mask: Eliminate Redundant Action for Deep Reinforcement Learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(15), 17078–17086.

Zhong, H., Chen, J., Shen, C., Zhang, H., Huang, J. & Hua, X. (2021). Self-Adaptive Neural Module Transformer for Visual Question Answering. *IEEE Trans. Multim.*, 23, 1264–1273.

Zhou, C., Li, Q., Li, C., Yu, J., Liu, Y., Wang, G., Zhang, K., Ji, C., Yan, Q., He, L. et al. (2024). A comprehensive survey on pretrained foundation models: A history from bert to chatgpt. *International Journal of Machine Learning and Cybernetics*, 1–65.

Zhou, H., Lu, C., Yang, S. & Yu, Y. (2021). ConvNets vs. Transformers: Whose Visual Representations are More Transferable? *IEEE/CVF International Conference on Computer Vision Workshops, ICCVW*, pp. 2230–2238.

Zhou, K., Zhang, H. & Li, F. (2022). TransNav: spatial sequential transformer network for visual navigation. *J. Comput. Des. Eng.*, 9(5), 1866–1878.

Zhou, M., Liu, Z., Sui, P., Li, Y. & Chung, Y. Y. (2020). Learning implicit credit assignment for cooperative multi-agent reinforcement learning. *Advances in neural information processing systems*, 33, 11853–11864.

Zhu, B. (2022). *Energy-Efficient and Fresh Data Collection in IoT Networks by Machine Learning*. (Ph.D. thesis, University of Saskatchewan).

Zhu, G., Zhang, M., Lee, H. & Zhang, C. (2020). Bridging imagination and reality for model-based deep reinforcement learning. *Advances in Neural Information Processing Systems*, 33, 8993–9006.

Zhu, Q., Zhang, H., Lan, M. & Han, L. (2023a). Neural categorical priors for physics-based character control. *ACM Transactions on Graphics (TOG)*, 42(6), 1–16.

Zhu, Z., Lin, K., Dai, B. & Zhou, J. (2022). Self-adaptive imitation learning: Learning tasks with delayed rewards from sub-optimal demonstrations. *Proceedings of the AAAI conference on artificial intelligence*, 36(8), 9269–9277.

Zhu, Z., Lin, K., Jain, A. K. & Zhou, J. (2023b). Transfer learning in deep reinforcement learning: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(11), 13344–13362.

Zhu, Z., Lin, K., Jain, A. K. & Zhou, J. (2023c). Transfer learning in deep reinforcement learning: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(11), 13344–13362.

Ziebart, B. D., Maas, A. L., Bagnell, J. A., Dey, A. K. et al. (2008). Maximum entropy inverse reinforcement learning. *AAAI*, 8, 1433–1438.

Zou, Y., Wu, H., Yin, Y., Dhamotharan, L., Chen, D. & Tiwari, A. K. (2024). An improved transformer model with multi-head attention and attention to attention for low-carbon multi-depot vehicle routing problem. *Annals of Operations Research*, 339(1), 517–536.