Learning to Play Atari in a World of Tokens

Pranav Agarwal [1 2] **Sheldon Andrews** [1 3] **Samira Ebrahimi Kahou** [4 2 5]

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 modeling 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 modeling both the world and learning behavior. We incorporate a transformer-decoder for auto-regressive world modeling and a transformer-encoder for learning behavior by attending to task-relevant cues 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/.

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 et al., 2023a). 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. (Ha & Schmidhuber, 2018). Additionally, the learned model allows the agent for safe and accurate decision-making by utilizing different look-ahead search algorithms for planning its action (Hamrick et al., 2020).

Most MBRL methods commonly follow a structured three-step approach: 1) Representation Learning $\phi : S \to \mathbb{R}^n$, the agents capture a simplified representation $\mathbb{R}^n$ of the high dimensional environment state S; 2) Dynamics and Reward Learning $f : S \times A \to S', \psi : S \times A \times S' \to R$, 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 $\pi : S \to \mathcal{P}(A)$, 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., 2020) employed a recurrent state space model (RSSM) (Doerr et al., 2018) to learn the world model. DreamerV2 (Hafner et al., 2021), an improved version of DreamerV1, offers better sample efficiency and scalability by incorporating a discrete latent space for modeling the dynamics. Building on the advancements of DreamerV2, DreamerV3 (Hafner et al., 2023) takes a similar approach with additions involving the use of symlog predictions and various regularisation 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 et al., 2022; Svidchenko & Shpilman, 2021). The training of Dreamer models can require an impractical amount of gameplay time, ranging from months to thousands of years, depending on the complexity of the game (Micheli et al., 2023). 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 (Xiao et al., 2019). This challenge is largely associated with the use of

[1]École de Technologie Supérieure, Canada [2]Mila [3]Roblox, USA [4]University of Calgary, Canada [5]Canada CIFAR AI Chair. Correspondence to: Pranav Agarwal <pranav.agarwal.1@ens.etsmtl.ca>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

convolutional neural networks (CNNs) and recurrent neural networks (RNNs) (Deng et al., 2023) that, while effective in many domains, face limitations in capturing complex and long-range dependencies, which are common in RL scenarios (Ni et al., 2024).

This motivates the need to use transformers (Vaswani et al., 2017; Lin et al., 2022), which have proven highly effective in capturing long-range dependencies in various natural language processing (NLP) tasks (Wolf et al., 2020) and addressing complex visual reasoning challenges in computer vision (CV) tasks (Khan et al., 2022). Considering these advantages, recent works have adapted transformers for modeling the dynamics in MBRL. Transdreamer (Chen et al., 2022) 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 requires long-term dependency and complex-reasoning. In order to stabilize the training, it utilizes gated transformer-XL (GTrXL) (Parisotto et al., 2020) architecture.

Masked world model (MWM) (Seo et al., 2023) utilizes a convolutional-autoencoder and vision transformer (ViT) (Dosovitskiy et al., 2020) 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 et al., 2020). Similarly, transformer-based world model (TWM) (Robine et al., 2023a) use transformer-XL (TrXL) (Dai et al., 2019) for modeling the world and use the predicted latent states for policy learning. Their work demonstrates sample-efficient performance on the Atari 100k benchmark.

Contrary to these approaches, imagination with autoregression over an inner speech (IRIS) (Micheli et al., 2023) models dynamics learning as a sequence modeling problem, utilizing discrete image tokens for modeling the world. It then uses reconstructed images using the predicted tokens for learning 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 et al. (2023), which solely utilized a transformer for extracting object-centric representation, our approach employs a transformer encoder to learn behavior 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., 2020), to learn the policy (as illustrated in Figure 1).

Additionally, challenges related to partial observability necessitate memory modeling. Previous work (Didolkar et al., 2022) modeled memory in transformers using a computationally intensive two-stream network. Inspired by (Bulatov et al., 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 modeling. Specifically, we utilize a transformer-decoder (GPT) for world modeling 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 modeling. 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 modeling 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.

2. Method

Our model, DART, is designed for mastering Atari games, within the framework of a partially observable Markov decision process (POMDP) (Kaelbling et al., 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 vector quantized-variational autoencoders (VQ-VAEs) (Van Den Oord et al., 2017; Esser et al., 2021) are used for tokenizing the original observations; (2) *World-Model Learning*, which involves

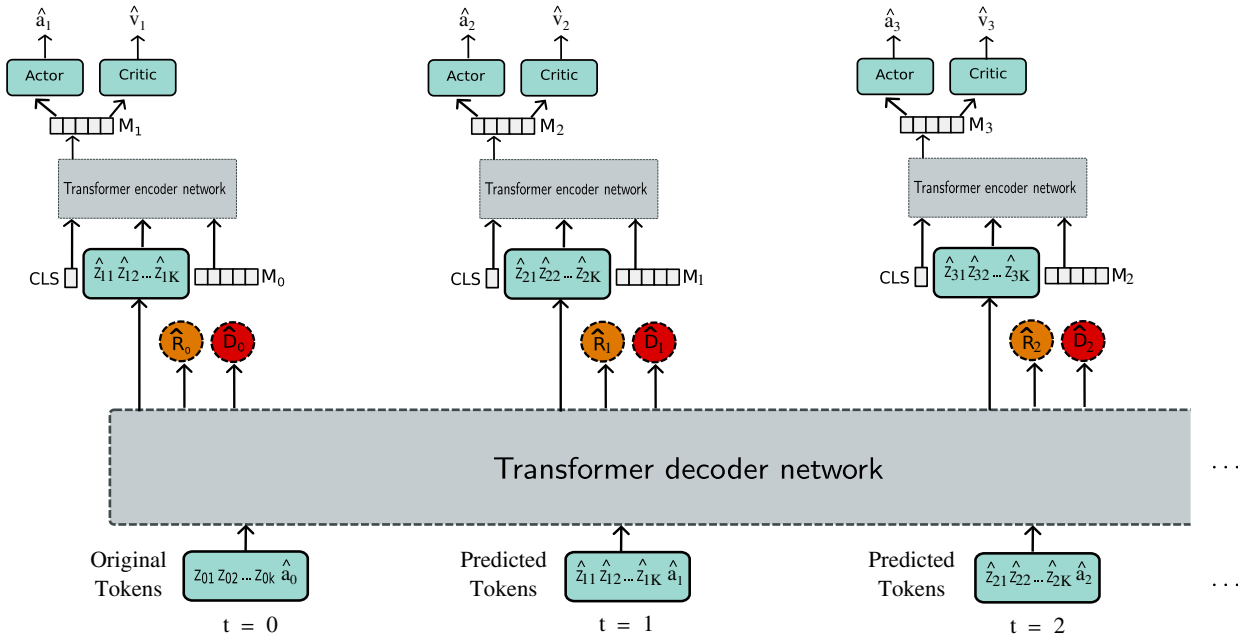


Figure 1: Discrete abstract representation for transformer-based learning (DART): In this approach, the original observation x_t is encoded into discrete tokens z_t using VQ-VAE. These tokenized observations, and predicted action, serve as inputs for the world model. A Transformer decoder network is used for modeling the world. The predicted tokens, along with a `CLS` and a `MEM` token are used as input by the policy. This policy is modeled 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 modeling memory, acting as the `MEM` token at the subsequent time step.

auto-regressive modeling of the dynamics of the environment using a GPT architecture; and (3) *Policy Learning*, which is modeled using ViT for decision-making by attending to task-relevant cues. We now describe our overall approach in detail.

2.1. Representation Learning

Discrete symbols are essential in human communication, as seen in natural languages (Cartuyvels et al., 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,

$$\hat{z}_q = q(\hat{z}_t^k; \phi_q, Z). \tag{1}$$

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 modeled 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 1). The discrete latent codes are passed to the decoder g_ϕ, which maps it 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 encourage 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 modeling of fine-grained, low-level information within the input image as a set of discrete latent codes.

2.2. World-model learning

The discrete latent representation forms the core of our approach, enabling the learning of dynamics through an autoregressive next-token prediction approach (Qi et al., 2024).

A transformer decoder based on the GPT architecture is used for modeling this sequence prediction framework. First, an aggregate sequence $\hat{z}_{ct} = f_\phi(\hat{z}_{<t}, \hat{a}_{<t})$ is modeled 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 modeling of future states given as $\hat{z}^k_{q_t} \sim p_d(\hat{z}^k_{q_t} \mid \hat{z}_{ct})$. 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.

2.3. Policy-learning

The policy π is trained within the world model (also referred as imagination) using a transformer encoder architecture based on vision transformer (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,

$$\mathbf{out} = [\texttt{CLS}, \hat{z}^1_{q_t}, \ldots, \hat{z}^K_{q_t}, \texttt{MEM}_{t-1}] + \mathbf{E}_{\text{pos}}. \tag{2}$$

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 $(\texttt{CLS}, \hat{z}^1_{q_t}, \ldots, \hat{z}^K_{q_t}, \texttt{MEM}_{t-1})$, where $\hat{z}^K_{q_t}$ corresponds to the embedding of Kth 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 et al., 2024). Transformers, known for their permutational-equivariant nature, efficiently model global representation (Xu et al., 2023; Yun et al., 2020). To incorporate local spatial information, we add learnable positional encoding $\mathbf{E}_{\text{pos}}$ to the original input (see Equation 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). This series of operations is repeated for a total of L blocks,

$$\begin{aligned} &\mathbf{out} = \mathbf{out} + \text{MSA}(\text{LN}(\mathbf{out})), \\ &\left.\begin{aligned} &\mathbf{out} = \mathbf{out} + \text{MLP}(\text{LN}(\mathbf{out})), \end{aligned}\right\} \times L \\ &h_t = \mathbf{out}[0], \quad \texttt{MEM}_t = \mathbf{out}[0]\,. \end{aligned} \tag{3}$$

Following L blocks of operations, the feature vector associated with the CLS token serves as the representation, modeling 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, $v_\xi\left(\hat{h}_t\right) \approx \mathbb{E}_{p_\theta}\left[\sum_{\tau \geq t} \hat{\gamma}^{\tau - t} \hat{r}_\tau\right]$. This is followed by the reward prediction, episode end prediction, and the token predictions of the next observation by the world model.

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 modeling without relying on recurrent networks, which can be challenging to train and struggle with long context (Pascanu et al., 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. Experiments

We evaluated our model alongside existing baselines using the Atari 100k benchmark (Łukasz 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 et al., 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 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 et al. (2021).

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

Table 1: DART achieves a new state-of-art median score among no-look-ahead search methods. It attains the highest median score, interquartile 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**

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 frameskip 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 modeled CLS token, which then serves as a memory token for the next step. Although this process is computed step by step, it remains efficient compared to methods like IRIS, as it doesn't require additional networks like LSTM to retain memory.

3.1. Results

In Figure 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 3a visualizes the performance profile, while Figure 3b 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 et al., 2020), DrQ (Yarats et al., 2021), SPR (Schwarzer et al., 2021), as well as data from 5 runs of SimPLe (Łukasz 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 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 Dream-

erV3 (Hafner et al., 2023), as well as other transformer-based world models, such as TWM (Robine et al., 2023a) and IRIS (Micheli et al., 2023).

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.

3.2. Policy Analysis

In Figure 4, we present the attention maps for the 6 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.

The visualization in Figure 4 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 4a). 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 4b- 4d). This observation highlights the adaptability of our approach to varying task requirements.

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 important 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 A.1), where DART showed improved performance over DreamerV3 in modeling long horizon tasks with multiple components.

3.3. Ablation Studies

We further analyzed DARTs performance across various experimental settings, as detailed in Table 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 in close proximity to 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 modeling. 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 modeling 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 modeling 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 heatmap visualization in Figure 4, where significant attention is given to memory tokens for the aforementioned games. On the contrary, in games like Boxing and Amidar, where long-term trajectory information

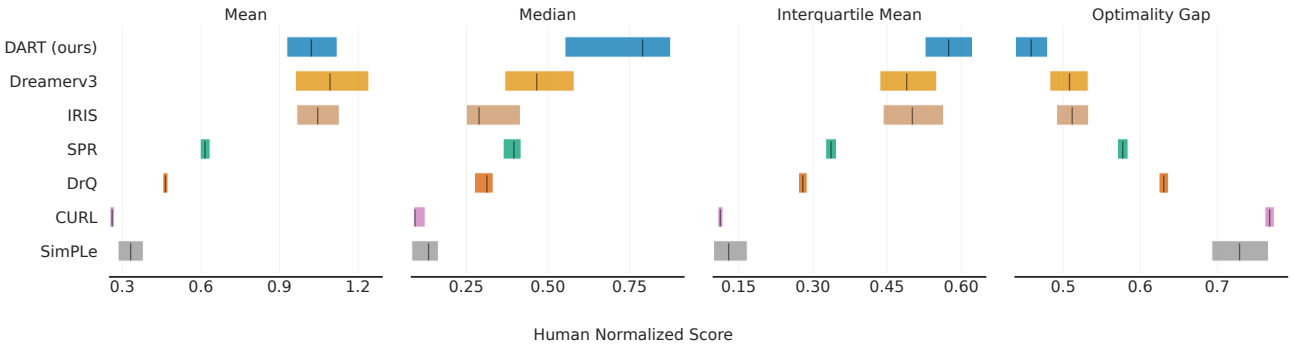
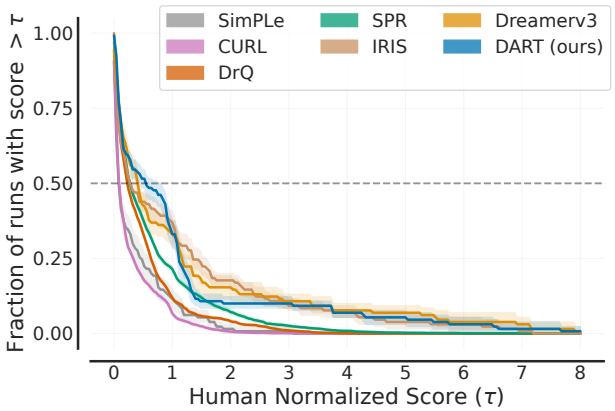
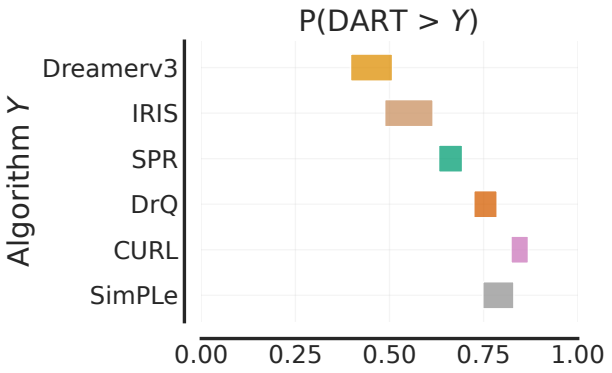


Figure 2: Comparison of Mean, Median, and Interquartile Mean Human-Normalized Scores



(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: Comparison of different models using performance profiles and probabilities of improvement.

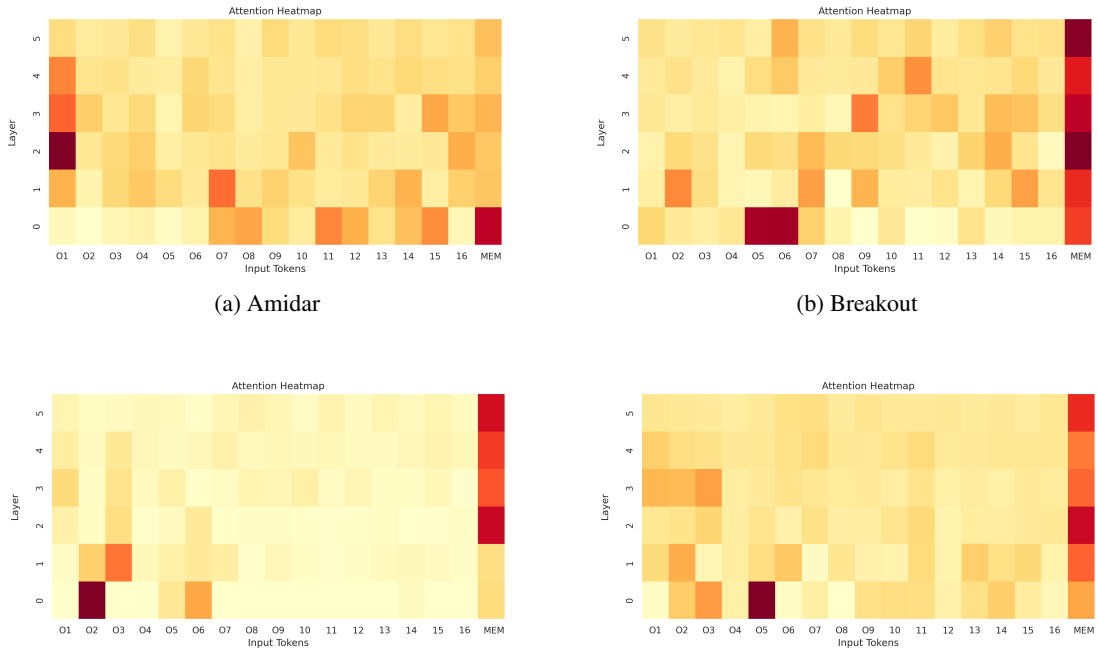
or extensive planning isn't needed, relying solely on recent state information is often sufficient for optimal decision-making, and thus there is only a small impact on the final performance with the masking of memory tokens.

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 model 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.

4. Related Work

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 et al., 2018; Mai et al., 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 (Moerland et al., 2023b; Polydoros & Nalpantidis, 2017; Atkeson & Santamaria, 1997). This is done by allowing the policy to learn the task in the imagined world (Wang et al., 2021b; Mu et al., 2021; Okada & Taniguchi, 2021; Zhu et al., 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 et al. (2018) learns an ensemble of models to reduce the



(a) Amidar (b) Breakout

(c) Private Eye (d) Krull

Figure 4: **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 modeling 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.

Table 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.

		w/o		Masked	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

impact of model bias and variance. Uncertainty estimation is another approach as shown in Plaat et al. (2023) to improve model accuracy. It involves estimating the uncertainty in the model's prediction so that the agent focuses its exploration in 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 (Wang et al., 2021a; Ji et al., 2022). For example, using a transformer-based world model, as in TransDreamer (Chen et al., 2022), 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 (McInroe et al., 2021; Du et al., 2020). Variational Autoencoders (VAEs) (Kingma et al., 2019) are commonly used for learning low-dimensional representations in MBRL (Andersen et al., 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 et al., 2018). However, VAEs learn a continuous representation of the input data by forcing the latent variable to be normally distributed. This poses a challenge for RL agents, where agents need to focus on precise details

for decision-making (Dunion et al., 2023). Lee et al. (2020) show disentangling representations helps in modeling interpretable policy and improves the learning speed of RL agents on various manipulation tasks. Recent works (Robine et al., 2023b; Zhang et al., 2022) have used 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 policy across a wide range of tasks.

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 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 modeling and the use of a transformer for policy modeling 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 modeling better-disentangled tokens for faster learning.

Acknowledgement

The authors would like to thank the reviewers for their valuable feedback. We are also thankful to the Digital Research Alliance of Canada for the computing resources and CIFAR for research funding.

Impact Statement

This paper presents works where the goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none of which we feel must be specifically highlighted here.

References

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

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

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

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

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

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

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

Chen, C., Wu, Y.-F., Yoon, J., and Ahn, S. Transdreamer: Reinforcement learning with transformer world models. *arXiv preprint arXiv:2202.09481*, 2022.

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

Dai, Z., Yang, Z., Yang, Y., Carbonell, J. G., Le, Q. V., and Salakhutdinov, R. Transformer-xl: Attentive language models beyond a fixed-length context. In *Annual Meeting of the Association for Computational Linguistics*, 2019.

Darcet, T., Oquab, M., Mairal, J., and Bojanowski, P. Vision transformers need registers. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=2dnO3LLiJ1.

Deng, F., Park, J., and Ahn, S. Facing off world model backbones: RNNs, transformers, and s4. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=GDYuzX0rwj.

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

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

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Du, S. S., Kakade, S. M., Wang, R., and Yang, L. F. Is a good representation sufficient for sample efficient reinforcement learning? In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=r1genAVKPB.

Dunion, M., McInroe, T., Luck, K. S., Hanna, J. P., and Albrecht, S. V. Temporal disentanglement of representations for improved generalisation in reinforcement learning. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=sPgP6aISLTD.

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

Guss, W. H., Houghton, B., Topin, N., Wang, P., Codel, C. R., Veloso, M. M., and Salakhutdinov, R. Minerl: A large-scale dataset of minecraft demonstrations. In *International Joint Conference on Artificial Intelligence*, 2019. URL https://api.semanticscholar.org/CorpusID:199000710.

Ha, D. and Schmidhuber, J. World models. *arXiv preprint arXiv:1803.10122*, 2018.

Hafner, D. Benchmarking the spectrum of agent capabilities. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=1W0z96MFEoH.

Hafner, D., Lillicrap, T., Ba, J., and Norouzi, M. Dream to control: Learning behaviors by latent imagination. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=S1lOTC4tDS.

Hafner, D., Lillicrap, T. P., Norouzi, M., and Ba, J. Mastering atari with discrete world models. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=0oabwyZbOu.

Hafner, D., Pasukonis, J., Ba, J., and Lillicrap, T. Mastering diverse domains through world models. *arXiv preprint arXiv:2301.04104*, 2023.

Hamrick, J. B., Friesen, A. L., Behbahani, F., Guez, A., Viola, F., Witherspoon, S., Anthony, T., Buesing, L., Veličković, P., and Weber, T. On the role of planning in model-based deep reinforcement learning. *arXiv preprint arXiv:2011.04021*, 2020.

Islam, R., Zang, H., Goyal, A., Lamb, A. M., Kawaguchi, K., Li, X., Laroche, R., Bengio, Y., and Tachet des Combes, R. Discrete compositional representations as an abstraction for goal conditioned reinforcement learning. *Advances in Neural Information Processing Systems*, 35: 3885–3899, 2022.

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

Ji, T., Luo, Y., Sun, F., Jing, M., He, F., and Huang, W. When to update your model: Constrained model-based reinforcement learning. *Advances in Neural Information Processing Systems*, 35:23150–23163, 2022.

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

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

Khan, S., Naseer, M., Hayat, M., Zamir, S. W., Khan, F. S., and Shah, M. Transformers in vision: A survey. *ACM computing surveys (CSUR)*, 54(10s):1–41, 2022.

Kingma, D. P., Welling, M., et al. An introduction to variational autoencoders. *Foundations and Trends® in Machine Learning*, 12(4):307–392, 2019.

Kurutach, T., Clavera, I., Duan, Y., Tamar, A., and Abbeel, P. Model-ensemble trust-region policy optimization. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=SJJinbWRZ.

Laskin, M., Srinivas, A., and Abbeel, P. Curl: Contrastive unsupervised representations for reinforcement learning. In *International Conference on Machine Learning*, pp. 5639–5650. PMLR, 2020.

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

Li, L. and Qiu, X. Token-aware virtual adversarial training in natural language understanding. *Proceedings of the*

AAAI Conference on Artificial Intelligence, 35(9):8410–8418, 2021.

Lin, T., Wang, Y., Liu, X., and Qiu, X. A survey of transformers. *AI Open*, 2022.

Mai, V., Mani, K., and Paull, L. Sample efficient deep reinforcement learning via uncertainty estimation. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=vrW3tvDfOJQ.

Mao, C., Jiang, L., Dehghani, M., Vondrick, C., Sukthankar, R., and Essa, I. Discrete representations strengthen vision transformer robustness. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=8hWs60AZcWk.

McInroe, T., Schäfer, L., and Albrecht, S. V. Learning temporally-consistent representations for data-efficient reinforcement learning. *arXiv preprint arXiv:2110.04935*, 2021.

Micheli, V., Alonso, E., and Fleuret, F. Transformers are sample-efficient world models. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=vhFu1Acb0xb.

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

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

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

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

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

Parisotto, E., Song, F., Rae, J., Pascanu, R., Gulcehre, C., Jayakumar, S., Jaderberg, M., Kaufman, R. L., Clark, A., Noury, S., et al. Stabilizing transformers for reinforcement learning. In *International conference on machine learning*, pp. 7487–7498. PMLR, 2020.

Pascanu, R., Mikolov, T., and Bengio, Y. On the difficulty of training recurrent neural networks. In *International conference on machine learning*, pp. 1310–1318. Pmlr, 2013.

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

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

Qi, M., Huang, Y., Yao, Y., Wang, M., Gu, B., and Sundaresan, N. Is next token prediction sufficient for gpt? exploration on code logic comprehension. *arXiv preprint arXiv:2404.08885*, 2024.

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

Robine, J., Höftmann, M., Uelwer, T., and Harmeling, S. Transformer-based world models are happy with 100k interactions. In *The Eleventh International Conference on Learning Representations*, 2023a. URL https://openreview.net/forum?id=TdBaDGCpjly.

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

Schwarzer, M., Anand, A., Goel, R., Hjelm, R. D., Courville, A., and Bachman, P. Data-efficient reinforcement learning with self-predictive representations. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=uCQfPZwRaUu.

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

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

Svidchenko, O. and Shpilman, A. Maximum entropy model-based reinforcement learning. In *Deep RL Workshop NeurIPS 2021*, 2021. URL https://openreview.net/forum?id=uBDG3yX-2sQ.

Van Den Oord, A., Vinyals, O., et al. Neural discrete representation learning. *Advances in neural information processing systems*, 30, 2017.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Wang, C., Yang, T., Hao, J., Zheng, Y., Tang, H., Barez, F., Liu, J., Peng, J., Piao, H., and Sun, Z. Ed2: An environment dynamics decomposition framework for world model construction. *arXiv preprint arXiv:2112.02817*, 2021a.

Wang, J., Li, W., Jiang, H., Zhu, G., Li, S., and Zhang, C. Offline reinforcement learning with reverse model-based imagination. *Advances in Neural Information Processing Systems*, 34:29420–29432, 2021b.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., et al. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations*, pp. 38–45, 2020.

Xiao, C., Wu, Y., Ma, C., Schuurmans, D., and Müller, M. Learning to combat compounding-error in model-based reinforcement learning. *arXiv preprint arXiv:1912.11206*, 2019.

Xiong, R., Yang, Y., He, D., Zheng, K., Zheng, S., Xing, C., Zhang, H., Lan, Y., Wang, L., and Liu, T. On layer normalization in the transformer architecture. In *International Conference on Machine Learning*, pp. 10524–10533. PMLR, 2020.

Xu, R., Yang, K., Liu, K., and He, F. $e(2)$-equivariant vision transformer. In *Uncertainty in Artificial Intelligence*, pp. 2356–2366. PMLR, 2023.

Yarats, D., Kostrikov, I., and Fergus, R. Image augmentation is all you need: Regularizing deep reinforcement learning from pixels. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=GY6-6sTvGaf.

Ye, W., Liu, S., Kurutach, T., Abbeel, P., and Gao, Y. Mastering atari games with limited data. *Advances in neural information processing systems*, 34:25476–25488, 2021.

Yin, Z.-H., Ye, W., Chen, Q., and Gao, Y. Planning for sample efficient imitation learning. *Advances in Neural Information Processing Systems*, 35:2577–2589, 2022.

Yoon, J., Wu, Y.-F., Bae, H., and Ahn, S. An investigation into pre-training object-centric representations for reinforcement learning. In *International Conference on Machine Learning*, 2023. URL https://api.semanticscholar.org/CorpusID:256697459.

Yu, T., Quillen, D., He, Z., Julian, R., Hausman, K., Finn, C., and Levine, S. Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. In *Conference on robot learning*, pp. 1094–1100. PMLR, 2020.

Yu, Y. Towards sample efficient reinforcement learning. In *IJCAI*, pp. 5739–5743, 2018.

Yun, C., Bhojanapalli, S., Rawat, A. S., Reddi, S., and Kumar, S. Are transformers universal approximators of sequence-to-sequence functions? In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=ByxRM0Ntvr.

Zhang, L., Lieffers, J., and Pyarelal, A. Deep reinforcement learning with vector quantized encoding. *arXiv preprint arXiv:2211.06733*, 2022.

Zhang, W., Wang, G., Sun, J., Yuan, Y., and Huang, G. STORM: Efficient stochastic transformer based world models for reinforcement learning. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=WxnrX42rnS.

Zhu, G., Zhang, M., Lee, H., and Zhang, C. Bridging imagination and reality for model-based deep reinforcement learning. *Advances in Neural Information Processing Systems*, 33:8993–9006, 2020.

Łukasz Kaiser, Babaeizadeh, M., Miłos, P., Osiński, B., Campbell, R. H., Czechowski, K., Erhan, D., Finn, C., Kozakowski, P., Levine, S., Mohiuddin, A., Sepassi, R., Tucker, G., and Michalewski, H. Model based reinforcement learning for atari. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=S1xCPJHtDB.

A. Appendix

A.1. 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 different 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 centered around the agent.

Table 3: Comparing the sample efficiency of DreameV3, IRIS, and DART on challenging Crafter environment which involves long-horizon tasks. Reported returns are specified as average and standard deviation over 5 seeds.

Model	DreamerV3	IRIS	DART
Steps	1M	1M	1M
Return	11.7 ± 1.9	9.23 ± 0.56	$\mathbf{12.2 \pm 1.67}$

In the results shown in Table 3, we compare DART with IRIS and Dreamer V3 in a low data regime and observed that DART achieves a higher average return, further showcasing the efficiency of DART over previous models.

A.2. Experiment on Atari with More Environment Steps

Table 4: Performance of DART with 100k and 150k environment steps (k). All results are shown as average and standard deviation over 5 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

By training it beyond 100k training steps, we see improved performance of DART as shown in Table 4, showcasing the scalability of DART with more data.

A.3. Model Configuration

Recent works have used transformer-based architectures for MBRL. In Table 5 we compare the configurations used by different approaches for representation learning, world modeling, and behavior learning.

Table 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

A.4. Hyperparameters

A detailed list of hyperparameters is provided for each module: Table 6 for Image Tokenizer, Table 7 for World Modeling, and Table 8 for behaviour learning.

Table 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 7: Hyperparameters used for modeling the dynamics using 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

Table 8: Hyperparameters used for modeling behavior using 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

A.5. Comparing the performance with STORM

Recently released another transformer-based model STORM (Zhang et al., 2023) showcases close to similar performance when compared with DART as shown in Table 9. STORM relies on utilizing VAEs for modeling stochastic world models, while we use VQ-VAE for modeling 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 performance boosts, especially in such games. Additionally, this approach makes DART more interpretable compared to

STORM, as illustrated in Figure 4's heatmap visualization.

Table 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

A.6. Integrating DART with lookahead search methods.

The proposed model DART doesn't utilize lookahead search, limiting its ability to leverage the learned dynamics for future trajectory planning. In contrast, Efficient Zero (Ye et al., 2021) the state-of-the-art model-based reinforcement learning uses the lookahead search method to plan future trajectories based on the learned model of the environment. Simulating future states and rewards, enables more effective decision-making. This approach predicts outcomes of different actions and selects the best course of action, utilizing Monte Carlo tree search (MCTS) for planning future trajectories.

Our method of modeling DART with discrete representation enables it to be easily integrated with lookahead search methods. In the Algorithm 1 below, we'll briefly outline how we integrate the lookahead search method into DART for efficient future trajectory planning, which we plan to explore further in future work. However, this approach is computationally expensive because exploring the entire state-action space is infeasible. Additionally, exploring the vast state-action space at each iteration requires significant computational resources, posing a challenge. Therefore, we currently limit DART's results without incorporating lookahead search methods.

A.7. Modeling DART for continuous action space.

As outlined in Algorithm 2, adapting DART for continuous action space involves training an additional action tokenizer, alongside an image tokenizer as explained in Section 2.1. The action tokenizer converts continuous action tokens into discrete codebook embeddings, while the world modeling process remains unchanged. In policy learning, the transformer encoder translates state tokens into action tokens, which are then decoded into continuous actions using the VQVAE decoder.

Algorithm 1 Integrating DART with lookahead search methods

Initialize pretrained dynamics model f (transformer decoder) and policy π (transformer encoder).
for each episode **do**
  Sample initial state s from the environment
  Map state s into a sequence of discrete tokens $s_0, s_1, ..., s_N$ using pre-trained VQ-VAE
  for $t = 0$ to $T - 1$ **do**
    Planning Action Sequence:
      Given current state tokens, use transformer decoder f to simulate possible future state tokens s' and rewards for each possible action
      Apply a planning algorithm (e.g., Monte Carlo Tree Search) to search for the optimal sequence of actions that maximize cumulative reward
      Select the first action a_t from the optimal sequence
    Execute a_t, observe next state s_{t+1} and reward r_t
  end for
end for

Algorithm 2 Modeling DART for continuous action space.

Representation Learning
$\rightarrow$ Sample trajectories τ (s, a, s', r).
$\rightarrow$ Use states in the form of an image to train an image tokenizer.
$\rightarrow$ Use the collected actions to train an action tokenizer.
World Model Learning
$\rightarrow$ Concatenate State tokens and action tokens for each time step.
$\rightarrow$ Process each state and action token using the transformer decoder.
$\rightarrow$ Optimize for the next state token, reward, and termination criterion.
Policy Learning
$\rightarrow$ Transformer encoder modeling the policy processes the tokenized state to predict discrete action tokens.
$\rightarrow$ VQVAE decoder maps the action token to continuous action.
$\rightarrow$ Train the transformer encoder policy for the next H steps using the imagined trajectories from the world model.