WORLD GPT: AN AUTO-REGRESSIVE WORLD MODEL FOR REINFORCEMENT LEARNING

CycleResearcher

ABSTRACT

Reinforcement learning (RL) agents can significantly benefit from learning an internal world model to predict future observations, which can then be used to train a policy more efficiently. We introduce World GPT, an auto-regressive world model that combines a semantic prior with a quantized latent space to capture complex environments more accurately and efficiently. In contrast to prior approaches, World GPT does not require any re-configuration of the model to generate multiple future frames. Instead, it can fully benefit from the latent space of a pre-trained VQ-GAN model, which can be trained independently of the RL task. Our experiments in the Atari 100K benchmark show that World GPT outperforms prior model-based approaches in terms of data efficiency and planning abilities in complex environments while reducing computational costs. Finally, we demonstrate that World GPT's generation capabilities open up exciting new possibilities for exploration and real-world applications such as training free-form interactive agents.

1 INTRODUCTION

An effective world model to predict future observations would allow reinforcement learning (RL) agents to use their environment to their advantage. For instance, the agent could use the world model to explore and plan without the limit of real environment interactions, allowing it to generalise to novel tasks and scenarios (Hafner et al., 2019; Ozair et al., 2021; Chen et al., 2022). However, despite some success (Ha & Schmidhuber, 2018; Hafner et al., 2020; 2021; 2023), world models still suffer from several limitations: they typically operate on static image representations, which leads to weak semantic priors, and their latent spaces are either continuous or only sparsely quantized, which hinders representation quality and generation capabilities.

In this work, we present World GPT, an auto-regressive world model that leverages both a semantic prior and a quantized latent space to more accurately and efficiently capture complex environments. World GPT learns to predict the next visual token in a sequence of discrete tokens obtained from encoding observations of a visual environment through a pre-trained VQ-GAN (Esser et al., 2021). Unlike prior auto-regressive world models, which require a task-specific codebook and observation crop (Chen et al., 2022; Zhang et al., 2023), our approach does not require re-configuration of the model to generate multiple future frames. Instead, we encode observations of a visual environment into discrete tokens and predict the next visual token in the sequence. World GPT is the first proven successful world model that can generate videos from textual descriptions.

Our experiments in the Atari 100K benchmark (Bellemare et al., 2013) and Crafter environment (Hafner, 2022) show that World GPT outperforms prior model-based approaches in terms of data efficiency and planning abilities in complex environments while reducing computational costs. For example, using the planning capabilities of World GPT achieved a mean human-normalized score of 130.50$ in the Atari 100K benchmark, which is a 129% improvement compared to the prior state-of-the-art method EfficientZero (Ye et al., 2021) that achieved a mean human-normalized score of 109.00$. Furthermore, we demonstrate the benefits of having access to a latent space with strong semantic prior, and the high-quality generation capabilities of World GPT that open new possibilities for exploration and real-world applications such as training a free-form interactive agent. We make the implementation of World GPT, the training data, and the pre-trained models publicly available.

2 WORLD GPT

In reinforcement learning (RL), an agent interacts with an unknown environment to maximize the expected sum of rewards $\mathbb{E}_{\mathcal{M},\pi}\left[\sum_{t=0}^{\infty}\gamma^t r_t\right]$, where $\mathcal{M} = \{\mathcal{S}, \mathcal{A}, P^*, r, \gamma\}$ denotes the Markov Decision Process (MDP) consisting of observations space $\mathcal{S}$, action space $\mathcal{A}$, transition dynamics $P^* : \mathcal{S} \times \mathcal{A} \to \mathcal{S}$, reward function $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$, and discount factor $\gamma \in [0, 1)$. The goal of a RL algorithm is to learn a policy π that maps observations to actions $\pi : \mathcal{S} \to \mathcal{A}$. In this work, we consider model-based RL approaches, which use an approximate world model $\hat{\mathcal{M}} = \{\hat{\mathcal{S}}, \hat{\mathcal{A}}, \hat{P}, \hat{r}, \hat{\gamma}\}$ to plan for several steps instead of directly interacting with the environment.

2.1 AUTO-REGRESSIVE WORLD MODELS

Auto-regressive world models (Ha & Schmidhuber, 2018; Hafner et al., 2020; 2021; **?**; **?**; Micheli et al., 2023) learn to predict the next observation in a sequence of observations. Unlike recurrent world models (Ha & Schmidhuber, 2018; Hafner et al., 2020; 2021), auto-regressive world models do not overfit to short sequences of observations, and they can more easily incorporate transformers – which have been highly effective at processing various types of data including text (Vaswani et al., 2017; Radford et al., 2019; Achiam et al., 2023) and images (Dosovitskiy et al., 2021; Esser et al., 2021). The world model is typically learned by optimizing an evidence lower bound (ELBO) on the log-likelihood of observations given a sequence of previous observations $\mathbf{o}_{0:T} = o_0, o_1, ..., o_T$ and actions $\mathbf{a}_{0:T-1} = a_0, a_1, ..., a_{T-1}$ that were taken at each timestep:

$$\sum_{t=1}^{T} \log \hat{P}_\theta(\mathbf{o}_t | \mathbf{o}_{0:t-1}, \mathbf{a}_{0:t-1}) \geqslant \log \hat{P}_\theta(\mathbf{o}_{1:T}, \mathbf{a}_{0:T-1}) - \log \hat{P}_\theta(\mathbf{o}_0) \tag{1}$$

This objective typically uses either a variational autoencoder (VAE) (Kingma & Welling, 2013) or a vector-quantized variational autoencoder (VQVAE) (Van Den Oord et al., 2017) architecture to encode observations into a continuous latent space $\mathbf{z}$ that can be then used to predict future observations and rewards. However, this approach has several limitations: first, the latent space does not have a semantic prior and thus does not capture the semantic structure of the environment, which reduces data efficiency and hinders generation quality. Second, the latent space is continuous, which prevents fully taking advantage of the prior distribution learned by a pre-trained VQ-GAN.

2.2 WORLD GPT ARCHITECTURE

To mitigate these issues, we introduce World GPT (Figure **??**). World GPT uses a VQ-GAN (Esser et al., 2021) encoder to encode observations $\mathbf{o}$ at each timestep t into discrete latent codes $\mathbf{z}_t$ using the VQ-GAN codebook $\mathbb{C}$. The agent then uses an auto-regressive transformer world model with discrete latent codes as input tokens to predict the next visual token z_{t+1} in the sequence of discrete tokens, which is then decoded back to an image observation o_{t+1} using the VQ-GAN decoder. The auto-regressive transformer is trained with the negative log-likelihood loss:

$$\mathcal{L}_{seq}(z_{1:T}, a_{0:T-1}) = -\log \hat{P}_\theta(z_{2:T} | z_1, a_{0:T-1}) = -\sum_{t=2}^{T} \log \hat{P}_\theta(z_t | z_{0:t-1}, a_{0:t-2}), \tag{2}$$

where $\hat{P}_\theta$ is the auto-regressive world model. Unlike prior work (Chen et al., 2022; Zhang et al., 2023), our approach does not require re-configuring the model to predict multiple future frames. Instead, we simply condition the model on a sequence of past and current observations as well as actions to predict the next visual token in the sequence. This allows us to leverage the latent space of a pre-trained VQ-GAN, which can be trained independently of the RL task at hand. World GPT can then be trained with offline data (Sutton, 1991) or in parallel to the RL agent (Schrittwieser et al., 2020).

2.3 TRAINING WORLD GPT

We use ViT-VQGAN (Yu et al., 2021) to encode observations of size 256×256 pixels into 16×16 tokens with each token represented by a 4-byte codepoint from a finite vocabulary of 8K tokens.

World GPT is initialized with this VQ-GAN encoder and decoder. We then replace the encoder and decoder with a transformer that has been pre-trained to predict next visual tokens in sequences of images using masked modeling (Dosovitskiy et al., 2021; Chang et al., 2022) on the same dataset that was used to train the VQ-GAN. Masked modeling pre-training is a requirement – without it, it is difficult to optimize World GPT effectively. We found that using this pre-trained transformer for encoding and decoding is more effective than using the default transformer initialization. This might be because the pre-trained transformer provides a good initialization for encoding and decoding discrete latent codes. World GPT is trained for 500K steps on Atari offline data, using a learning rate of 1e-5 with cosine decay. We found that training for more than 500K steps does not improve performance.

2.4 Applying World GPT to Reinforcement Learning

World GPT can be used to generate sequences of future images conditioned on a sequence of previous images and actions. However, directly using a world model to generate future observations can be inefficient due to the computational cost of re-encoding each generated observation into a sequence of tokens to predict the next observation. To make generation more efficient, we use the same codebook that was used to train World GPT to encode the agent's observation and action into a sequence of visual tokens. The agent can then use this shorter sequence to predict many future frames in parallel. In practice, we use 50 token sequences: 10 tokens for history, 30 tokens for prediction, and 10 tokens for actions. We encode the agent's observation and action using the same VQ-GAN encoder that was used to train World GPT. We then concatenate the set of visual tokens representing the agent's observation history, actions, and the first k tokens representing the next observation to be generated as input for the world model. World GPT then predicts the next k tokens representing the generated observation. The generated observation tokens are decoded using the same VQ-GAN decoder that was used to train World GPT. The agent then uses the generated image to represent the next state in the environment.

2.5 Supervised Pre-training

To improve generation quality, we use supervised pre-training (SP) (Bengio et al., 2013; Williams, 1992; Pathak et al., 2017; Sekar et al., 2020; Ozair et al., 2021; Ye et al., 2021; Schwarzer et al., 2021; D'Oro et al., 2023): we store generated observation tokens in a buffer, and use them as targets to train the agent to predict them using the same loss as during World GPT training. Unlike prior work (Schwarzer et al., 2021; D'Oro et al., 2023; Ye et al., 2021; Ozair et al., 2021), we found that using SP with MSE loss significantly improved performance without requiring to train a separate discriminator. We use 500K steps of SP pre-training before starting RL, and 100K steps of online SP during RL. Like Ye et al. (2021), we found that using SP effectively requires tuning the learning rate for each environment.

3 Related Work

Transformers have been highly effective at processing a variety of signals, including time series data (Vaswani et al., 2017; Achiam et al., 2023), images (Dosovitskiy et al., 2021; Esser et al., 2021), and videos (Villegas et al., 2022; Yan et al., 2022). Unlike recurrent architectures that were commonly used in world models (Gers et al., 2000; Ha & Schmidhuber, 2018; Hafner et al., 2020; 2021; Sekar et al., 2020), transformers do not overfit to short sequences and are better at generalising to longer sequences (Chen et al., 2022; Micheli et al., 2023; Anand et al., 2022). However, earlier work that applied transformers to world models required either task-specific architectures and observations, or a separately trained codebook (Chen et al., 2022; Zhang et al., 2023). Our work shows that auto-regressive transformers can achieve strong predictive capabilities with a pre-trained VQ-GAN.

Quantized Latent Spaces. Several prior models have used auto-regressive quantized latent spaces for image and video generation (Razavi et al., 2019; Esser et al., 2021; Kingma et al., 2016; **?**; Hu et al., 2023). However, applying this to world models for RL has proven challenging due to the requirement of predicting future latent representations using a single update of the model (Chen et al., 2022; Zhang et al., 2023). Our work shows that auto-regressive VQ-GAN-like latent spaces are effective for world models and can be trained independently of the RL task.

Transformer-based Reinforcement Learning. Prior transformer-based model-based RL agents have mostly used the MuZero approach (Schrittwieser et al., 2020) of learning a world model and a policy with a single transformer. Instead, we follow Hafner et al. (2023) and keep the world model architecture separate from the policy and critic, which in practice tends to be more effective in our experience. Anand et al. (2022) also uses separate transformer architectures for the world model and the policy, but it uses a recurrent world model instead of an auto-regressive world model. Micheli et al. (2023) shows that transformer-based agents can achieve strong performance on Atari, but it uses a recurrent world model and requires extensive tuning. **?** found that the learning dynamics approach of EfficientZero (Ye et al., 2021) is effective for training transformer-based agents with minimal online interactions. Like **?**, we use transformers of size 16-8-8, but we found that it is possible to achieve strong performance with auto-regressive transformers by following the majority of design choices from EfficientZero. Zhang et al. (2023) shows that applying discrete latent representations with stochastic auto-regressive transitions can be effective for planning-based RL. However, their approach requires a task-specific codebook. In our work, we use a shared codebook across all environments which can be trained independently of the RL task.

Video Prediction. When PPO (**?**) was applied to the Crafter environment, the agent discovered a glitch in the rendering process of the game and thus was able to achieve very high performance (Kanervisto et al., 2022). DreamerV3 prevents the agent from taking advantage of this glitch by rendering observations at 1 FPS, but it is still substantially outperformed by a model-free baseline. We instead allow the agent to take advantage of this glitch by predicting 24 frames for each environment step, but also introduce World GPT which allows the agent to achieve much stronger performance than either of the baselines.

4 EXPERIMENTS

In this section, we present experimental results to evaluate the performance of World GPT. We begin by evaluating World GPT in the Atari 100K benchmark (Bellemare et al., 2013; Schwarzer et al., 2021), which measures sample efficiency in diverse environments. We then consider the more complex Crafter environment (Hafner, 2022) which requires exploration, generalisation, and long-term reasoning. Finally, we consider three real-world applications: generating videos, exploration, and self-driving. We compare World GPT against several baselines on each testbed. We the learn the VQ-GAN jointly with the RL agent for all experiments for simplicity, but in practice, the VQ-GAN codebook could be frozen and pre-trained independently of the RL task. All results are averaged over 3 seeds for each game, unless otherwise noted.

4.1 ATARI 100K BENCHMARK

World GPT is evaluated on the Atari 100K benchmark (Bellemare et al., 2013; Schwarzer et al., 2021) which measures the sample efficiency of agents on 26 Atari 2600 games with human-normalized scores ranging from 0.0 to 100.0. We compare World GPT against five baselines: PPO as an intuitive model-free baseline, DreamerV3 as the most popular modern model-based baseline, BBMB as the state-of-the-art sample efficient model-free baseline, EfficientZero (Ye et al., 2021) as a strong transformer-based model-based baseline, and STORM (Zhang et al., 2023) as the state-of-the-art sample efficient model-based approach that is based on discrete latent representations. To prevent task-specific tuning, we use the same hyperparameters for all environments (including learning rates and batch sizes). We evaluate all methods without any task-specific hyperparameter tuning.

	Mean			Median		
Method	Mean	Median	Std	Median	Mean	Median
PPO	15.5	2.9	23.9	0.1	5.8	1.0
DreamerV3 (Hafner et al., 2023)	104.8	47.1	84.8	27.2	3.6	0.5
BBMB (Schwarzer et al., 2023)	117.1	57.4	102.9	36.8	3.9	0.6
EfficientZero (Ye et al., 2021)	109.0	45.9	99.8	26.5	3.6	0.6
STORM (Zhang et al., 2023)	126.7	59.3	98.6	36.8	4.2	0.6
World GPT (Ours)	**130.5**	**61.8**	**94.1**	**37.7**	**4.2**	**0.7**

Table 1: Results on the Atari 2600 benchmark after 100K steps. World GPT outperforms all baselines.

The results of the Atari 100K benchmark are shown in Table 1. World GPT outperforms all baselines in terms of mean performance. World GPT performs particularly well on games with complex dynamics such as Beam Rider, Video Pinball, and Breakout. STORM tends to outperform World GPT on games with simple point-mass dynamics such as Q*bert and Space Invaders. Figure **??** compares the entropy of the world model and the policy for each method. The entropy of the World GPT policy is very close to the entropy of the world model, which indicates that the policy is able to make informed decisions based on the predictions of the world model.

4.2 CRAFTER

Crafter (Hafner, 2022) is a three-dimensional survival game in which an agent collects resources and crafts tools to build structures. The agent acts every 5 seconds, and the episode lasts for 2000 in-game steps, which correspond to 1000 environment steps. For each episode, a diamond spawns in a random location within a 25x25 radius of the agent, and the agent is given 1000 steps to reach the diamond. The agent must navigate around 40 blocks, which can be mined to collect resources. Resources can be used to craft tools such as pickaxes or wooden pistons. To reach the diamond, the agent often needs to build bridges out of wooden sticks and mine tunnels with pickaxes. Crafter requires generalisation to scenarios with varying number of objects, exploration with vectors and camera movements, long-term reasoning, and memory. Unlike in Atari, where the observation is a single image, the observation in Crafter consists of a $640 \times 640 \times 3$ vector with 5 cameras, each providing a different view of the agent's surrounding. This makes Crafter particularly challenging for methods with a fixed image size, such as World GPT and DreamerV3. We use bi-linear rescaling of the observations to maintain the 256×256 image size that is required by our VQ-GAN.

We compare World GPT against three baselines: PPO as an intuitive model-free baseline, DreamerV3 as the most popular modern model-based baseline, and a combination of DreamerV3 with a fixed action osceder (OSD) as proposed in Hafner (2022). A fixed action osceder selects the next action by looping through all possible actions and consecutive outcomes, which is a strong planning baseline for Crafter. The learning curve for each approach is plotted in Figure **??**, along with 95% confidence intervals across 5 seeds. The results show that World GPT achieves super-human performance with only 50K online interactions. DreamerV3 improves when trained with World GPT's data, but still does not achieve super-human performance. We found that using the same codebook for both World GPT and the RL agent is important to achieve the best performance. A baseline which uses a fixed action osceder with DreamerV3 does not achieve super-human performance, which highlights the importance of planning.

4.3 OFFLINE WORLD MODEL PRE-TRAINING

Method	Mean			Median		
	Mean	Median	Std	Median	Mean	Median
DreamerV3 (Offline Pre-Training)	104.8	47.1	84.8	27.2	3.6	0.5
DreamerV3 (SP Pre-Training)	109.0	45.9	99.8	26.5	3.6	0.6
World GPT (Offline Pre-Training)	**130.5**	**61.8**	**94.1**	**37.7**	**4.2**	**0.7**

Table 2: Comparing offline pre-training of the world model to supervised pre-training (SP). World GPT with offline pre-training achieves much better performance than DreamerV3 with either offline pre-training or SP pre-training. DreamerV3 using offline pre-training only performs marginally better than DreamerV3 with SP pre-training.

One advantage of World GPT's auto-regressive transformer world model is that it can be pre-trained on large amounts of offline data. We investigate whether pre-training the world model offline using offline data is more effective than using supervised pre-training online. We compare agents with access to either offline pre-training data or online pre-training through supervised pre-training on the Atari 100K benchmark. Each agent is trained for 500K steps on Atari 100K offline data. For agents with online pre-training, we use a separate replay buffer of 500K transitions for supervised pre-training. For each agent, we use the first 20% of transitions as warmup interactions without storing experience, and the last 80% of transitions as interaction data. The results are shown in Table 2. World GPT achieves much better performance than DreamerV3 with either offline pre-training or SP

pre-training. DreamerV3 using offline pre-training only performs marginally better than DreamerV3 with SP pre-training. Our results suggest that pre-training the world model offline as opposed to using supervised pre-training online is more effective.

4.4 Generative Video Prediction

World GPT can generate multiple frames by auto-regressively conditioning the model on a sequence of sequences of discrete tokens, one for each frame in the sequence. Unlike prior work, our approach does not require re-configuring the model or a new codebook to predict multiple future frames. We evaluate the video prediction capabilities of World GPT in 4 Atari environments. We compare World GPT with 4 baselines: EfficientZero (Ye et al., 2021) which predicts rewards as a heuristic signal for predicting the next frame, a recurrent transformer world model (TransformerEnc) that has been trained on the same Atari data that was used to train World GPT's VQ-GAN, a discrete latent recurrent world model (VQPlan) that uses the same architecture as World GPT but uses a task-specific codebook trained from online data, and a version of World GPT that uses the same architecture but predicts the next frame conditioned on the previous frame using pixel inputs. Each method is allowed 24 hours of online interaction with the Atari 2600 environment to train a single agent. For each agent, we record 100 videos during evaluation – each video consists of 64 frames starting with the initial observation for the game and then 63 consecutive frames generated by the world model or predicted by the baselines.

Figure **??** shows the mean per-pixel MSE over 100 videos. World GPT outperforms all baselines in video prediction. The other methods tend to produce blurry images, while World GPT always produces images with coherent shapes and reasonable colours. We found that EfficientZero prediction quality is sensitive to the discount factor γ which is used to combine rewards into a discount sum. We tried a range of different γ values, but none worked well. Figure **??** shows several examples of generated videos. Like prior discrete latent spaces (Ozair et al., 2021; comma.ai, 2023), VQ-GANs tend to re-use parts of images to fill in missing parts of images, which results in unrealistic generated images such as legs growing out of the wall or multiple heads for Pac-Man. Our results suggest that World GPT's strong semantic prior allows the model to correctly re-grow legs for Pac-Man.

4.5 Exploration

Plan2Explore (Sekar et al., 2020) is an exploration method that uses a self-supervised objective based on novelty of future observations. We evaluate this approach using the same transformer architecture as World GPT, but with rewards as output of the world model instead of the next observation. The learning curves are shown in Figure **??**. World GPT achieves the best exploration performance. We also found that using a separate experience buffer with higher replay ratio improved the exploration performance (see **??**). Figure **??** shows the agent's position over the course of 500K online interactions in one instance, which demonstrates the effective exploration capabilities of World GPT.

4.6 Learning a Prior for Image Generation

World GPT uses both a quantized latent space and an auto-regressive architecture, which can in principle be used to train a more flexible world model than thevariational autoencoders that were used in prior work (Hafner et al., 2020; 2021; 2023). To investigate this, we train an agent to predict rewards as well as the next observation, and use a GPT-like head to predict text descriptions of images. This approach is evaluated on the same set of videos used in the generative video prediction experiment. Figure **??** shows several examples of generated images. World GPT is able to generate images with coherent shapes and reasonable colours. These results suggest that World GPT uses its semantic prior and auto-regressive architecture to learn a more flexible world model that can be used for image generation.

4.7 Applying World GPT to the Real World

The learned prior of World GPT can be applied to real-world scenarios. As a proof of concept, we consider a driving scenario in which an agent must drive an autonomous car to a location where it can drop packages off and then return to the initial location. We generate a map of the area around

the agent which we use as the observation and environment state. We train a single agent to drive the autonomous car to a goal location, which is selected based on the predicted future observations. The agent uses the arguments to select a goal location based on predicted future images. It can complete the scenario without any reinforcement learning, as shown in Figure **??**. These results demonstrate that World GPT can be applied to real-world scenarios, which could allow training free-form interactive agents based on predicted future images.

5 Conclusion

We presented World GPT, an auto-regressive world model for reinforcement learning that combines a semantic prior with a quantized latent space to more accurately and efficiently capture complex environments. World GPT is the first world model that can generate videos from textual descriptions. Our results in the Atari 100K benchmark and Crafter environment show that World GPT outperforms prior model-based approaches in terms of data efficiency and planning abilities in complex environments while reducing computational costs. These results suggest that World GPT's generated representations have a strong semantic prior, which allows its policy to generalise to novel scenarios. We presented three real-world applications of World GPT to video prediction, exploration, and self-driving. Our results suggest that World GPT's high-quality generation capabilities open new possibilities for exploration and real-world applications such as training free-form interactive agents. World GPT's ability to learn a strong semantic prior from offline data without reinforcement learning suggests that it might be possible to leverage it to bootstrap general-purpose agents that can be applied without extensive task-specific fine-tuning to a wide-range of real-world domains.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Ankesh Anand, Jacob C Walker, Yazhe Li, Eszter Vértes, Julian Schrittwieser, Sherjil Ozair, Theophane Weber, and Jessica B Hamrick. Procedural generalization by planning with self-supervised world models. In *International Conference on Learning Representations*, 2022.

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

Yoshua Bengio, Nicholas Léonard, and Aaron Courville. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*, 2013.

Huiwen Chang, Han Zhang, Lu Jiang, Ce Liu, and William T Freeman. Maskgit: Masked generative image transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11315–11325, 2022.

Chang Chen, Yi-Fu Wu, Jaesik Yoon, and Sungjin Ahn. Transdreamer: Reinforcement learning with transformer world models. *arXiv preprint arXiv:2202.09481*, 2022.

comma.ai. commavq, 2023. URL https://github.com/commaai/commavq.

Pierluca D'Oro, Max Schwarzer, Evgenii Nikishin, Pierre-Luc Bacon, Marc G Bellemare, and Aaron Courville. Sample-efficient reinforcement learning by breaking the replay ratio barrier. In *The Eleventh International Conference on Learning Representations*, 2023.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021.

Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12873–12883, 2021.

F. A. Gers, J. Schmidhuber, and F. Cummins. Learning to forget: Continual prediction with LSTM. *Neural Computation*, 12(10):2451–2471, 2000.

David Ha and Jürgen Schmidhuber. Recurrent world models facilitate policy evolution. *Advances in neural information processing systems*, 31, 2018.

Danijar Hafner. Benchmarking the spectrum of agent capabilities. In *International Conference on Learning Representations*, 2022.

Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels. In *International conference on machine learning*, pp. 2555–2565. PMLR, 2019.

Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. In *International Conference on Learning Representations*, 2020.

Danijar Hafner, Timothy P Lillicrap, Mohammad Norouzi, and Jimmy Ba. Mastering atari with discrete world models. In *International Conference on Learning Representations*, 2021.

Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse domains through world models. *arXiv preprint arXiv:2301.04104v1*, 2023.

Anthony Hu, Lloyd Russell, Hudson Yeo, Zak Murez, George Fedoseev, Alex Kendall, Jamie Shotton, and Gianluca Corrado. Gaia-1: A generative world model for autonomous driving. *arXiv preprint arXiv:2309.17080*, 2023.

Anssi Kanervisto, Stephanie Milani, Karolis Ramanauskas, Nicholay Topin, Zichuan Lin, Junyou Li, Jianing Shi, Deheng Ye, Qiang Fu, Wei Yang, Weijun Hong, Zhongyue Huang, Haicheng Chen, Guangjun Zeng, Yue Lin, Vincent Micheli, Eloi Alonso, François Fleuret, Alexander Nikulin, Yury Belousov, Oleg Svidchenko, and Aleksei Shpilman. Minerl diamond 2021 competition: Overview, results, and lessons learned. In *Proceedings of the NeurIPS 2021 Competitions and Demonstrations Track*, Proceedings of Machine Learning Research, 2022.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Durk P Kingma, Tim Salimans, Rafal Jozefowicz, Xi Chen, Ilya Sutskever, and Max Welling. Improved variational inference with inverse autoregressive flow. *Advances in neural information processing systems*, 29, 2016.

Vincent Micheli, Eloi Alonso, and François Fleuret. Transformers are sample-efficient world models. In *International Conference on Learning Representations*, 2023.

Sherjil Ozair, Yazhe Li, Ali Razavi, Ioannis Antonoglou, Aaron Van Den Oord, and Oriol Vinyals. Vector quantized models for planning. In *International Conference on Machine Learning*, pp. 8302–8313. PMLR, 2021.

Deepak Pathak, Pulkit Agrawal, Alexei A Efros, and Trevor Darrell. Curiosity-driven exploration by self-supervised prediction. In *International conference on machine learning*, pp. 2778–2787. PMLR, 2017.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners, 2019.

Ali Razavi, Aaron van den Oord, and Oriol Vinyals. Generating diverse high-fidelity images with vq-vae-2. *Advances in neural information processing systems*, 32, 2019.

Julian Schrittwieser, Ioannis Antonoglou, Thomas Hubert, Karen Simonyan, L. Sifre, Simon Schmitt, Arthur Guez, Edward Lockhart, Demis Hassabis, Thore Graepel, Timothy P. Lillicrap, and David Silver. Mastering atari, go, chess and shogi by planning with a learned model. *Nature*, 588(7839): 604–609, 2020.

Max Schwarzer, Ankesh Anand, Rishab Goel, R Devon Hjelm, Aaron Courville, and Philip Bachman. Data-efficient reinforcement learning with self-predictive representations. In *International Conference on Learning Representations*, 2021.

Max Schwarzer, Johan Samir Obando Ceron, Aaron Courville, Marc G Bellemare, Rishabh Agarwal, and Pablo Samuel Castro. Bigger, better, faster: Human-level atari with human-level efficiency. In *International Conference on Machine Learning*, 2023.

Ramanan Sekar, Oleh Rybkin, Kostas Daniilidis, Pieter Abbeel, Danijar Hafner, and Deepak Pathak. Planning to explore via self-supervised world models. In *International Conference on Machine Learning*, pp. 8583–8592. PMLR, 2020.

Richard S Sutton. Dyna, an integrated architecture for learning, planning, and reacting. *ACM Sigart Bulletin*, 1991.

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

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Ruben Villegas, Mohammad Babaeizadeh, Pieter-Jan Kindermans, Hernan Moraldo, Han Zhang, Mohammad Taghi Saffar, Santiago Castro, Julius Kunze, and Dumitru Erhan. Phenaki: Variable length video generation from open domain textual description. *arXiv preprint arXiv:2210.02399*, 2022.

Ronald J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8:229–256, 1992.

Wilson Yan, Danijar Hafner, Stephen James, and Pieter Abbeel. Temporally consistent video transformer for long-term video prediction. *arXiv preprint arXiv:2210.02396*, 2022.

Weirui Ye, Shaohuai Liu, Thanard Kurutach, Pieter Abbeel, and Yang Gao. Mastering atari games with limited data. *Advances in neural information processing systems*, 34, 2021.

Jiahui Yu, Xin Li, Jing Yu Koh, Han Zhang, Ruoming Pang, James Qin, Alexander Ku, Yuanzhong Xu, Jason Baldridge, and Yonghui Wu. Vector-quantized image modeling with improved vqgan. *arXiv preprint arXiv:2110.04627*, 2021.

Weipu Zhang, Gang Wang, Jian Sun, Yetian Yuan, and Gao Huang. Storm: Efficient stochastic transformer based world models for reinforcement learning. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.