Dreaming with Transformers

Catherine Zeng,[1,2] Jordan Docter, [1]
Alexander Amini, [1] Igor Gilitschenski, [3] Ramin Hasani, [1] Daniela Rus [1]

[1] MIT CSAIL
[2] Harvard University
[3] University of Toronto
catherinezeng@college.harvard.edu

Abstract

Transformers have proven to be effective in various application domains, such as natural language processing and vision, as result of their ability to perform credit assignment in long time horizons and their scalability with large amounts of data. In this paper, we explore the effectiveness of the transformer architecture in world model-based deep reinforcement learning (RL). The performance of a world model-based deep RL agent depends on the quality of its state transition model and the imagination horizon, and we believe that transformers may enhance the memory capabilities and predictive performance of such agents. To this end, we extend the world model-based RL framework Dreamer using transformers in its dynamics model. Our experimental results on Deepmind Lab and the data-driven driving simulator VISTA suggest that Dreaming with Transformers can outperform RNN-based models, and we discuss the challenges and potential future directions working with this framework.

Introduction

Deep reinforcement learning (RL) tasks cover a wide range of possible applications with the potential to impact domains such as robotics, healthcare, smart grids, finance, and autonomous vehicles. A fundamental challenge in these domains is how to learn an optimal policy in high-dimensional and long time horizon tasks.

World models (Ha and Schmidhuber 2018) explicitly represent an agent's knowledge about its environment. Recent world model-based RL frameworks (Hafner et al. 2020a; Liu, Gu, and Liu 2020) leverage world models to facilitate generalization and can predict the outcomes of potential actions in an imagination space to improve decision making. One such framework, Dreamer, can achieve state-of-the-art performance across a series of standard RL benchmarks (Hafner et al. 2020b). Dreamer presents agents that can learn long-horizon behavior directly from high-dimensional inputs by using latent imagination.

Dreamer agents use an actor-critic algorithm to compute rewards and use recurrent neural networks (RNNs) to make predictions within a latent imagination state-space. The use of RNNs and their gated versions such as the long short-term memory (LSTMs) (Hochreiter and Schmidhuber 1997) and gated recurrent units (GRU) (Cho et al. 2014) is natural due to the spatiotemporal dependencies in various environments. However, the memory span of a recurrent network is limited (Lechner and Hasani 2020), as their information processing mode is sequential, and mutual information of RNNs decays exponentially in temporal distance of sequence inputs (Shen 2019). Recently, transformer architectures using the attention mechanism proved to enable parallel credit assignment in very long sequences, significantly outperforming recurrent models (Vaswani et al. 2017). Transformers are the primary choice in natural language processing (NLP) tasks (Devlin et al. 2019), and are becoming the dominant architecture in vision tasks as well (Dosovitskiy et al. 2020). Recent works provided additional evidence that transformers can be effective in the context of RL (Parisotto et al. 2019; Chen et al. 2021; Janner, Li, and Levine 2021). In the present study, we investigate whether transformers are better candidates for learning world models.

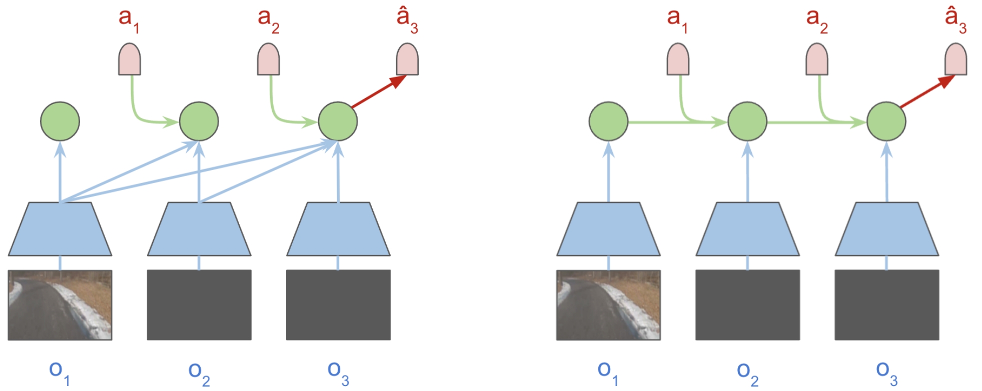
In particular, we improve upon the Dreamer framework by addressing an agent's capacity for world representations and credit assignment in long-horizon tasks. We propose an architecture that integrates the concept of self-attention seen in transformers into agents with representation of the world. Our main contribution lies in the architecture of the latent dynamics model which particularly allows for non-markovian transitions of the latent space. This freedom greatly increases the predictive power of imagined trajectories, which in turn can yield more optimal actions.

We first provide background on the architecture of world models as well as the concept of self-attention. Next, we provide an overview of our proposed architecture. We motivate our choices by providing analysis of the shortcomings of current models specifically within the context of environments requiring memory of the past. Then, we describe our experimental setup and present preliminary results. Finally, we discuss conclusions and future directions.

Background

Reinforcement Learning

The objective of reinforcement learning is to search for an optimal policy in a Partially Observable Markov Decision Process (POMDP) described by the tuple $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \mathcal{O})$.



(a) Transformer-based agent acting in the environment

(b) GRU-based agent acting in the environment

Figure 1: A comparison between transformer-based and GRU-based Dreamer agents. Transformers can directly access encoded information from previous observations, whereas recurrent networks can only access past information stored in a single hidden state. Therefore, transformers should perform better on tasks requiring long-term memory. Specifically, on a driving simulator task where sensors fail and visual input stops, transformers should be able to make use of its most recent view of the road.

Particularly in a *partially* observable MDP, the agent makes observations of the environment that may only contain partial information about the underlying state. Formally, we let $o_t \in \mathcal{O}, r_t \in \mathcal{R}, s_t \in \mathcal{S}, a_t \in \mathcal{A}$ be the observation, reward, state, and action at time step $t \in \{1, \ldots, T\}$. At each time step t, the agent will generate and execute an action $a_t \sim p(a_t|o_{\leq t}, a_{\leq t})$. The environment will change to a new state according to some transition probability function $s_t \sim P(s_t|s_{t-1}, a_t)$, but the agent will only receive observations and reward $o_t, r_t \sim p(o_t, r_t|o_{\leq t}, a_{\leq t})$ from the environment. The goal of the agents is to maximize the expected reward $\mathbb{E}(\sum_{t=0}^{T} r_t)$.

Model-Free vs Model-Based RL

Recent reinforcement learning models (Hafner et al. 2020a; Liu, Gu, and Liu 2020) have found success by learning world models that explicitly represent an agent's knowledge about its environment. World models stand in contrast to model-free frameworks which directly learn a correspondence between the state-space and action-space. It is shown (Haarnoja et al. 2018) that in large unknown environments, model-free frameworks suffer from low sample efficiency and high sample complexity, and in some cases are not optimal. World models attempt to address this issue by providing the means for agents to extrapolate in situations they have never encountered before. This is accomplished by learning a representation of the world in a latent space, and then forming policies on top of this latent space.

Dreaming with Transformers

We consider reinforcement learning tasks with highly complex observation and action spaces such as image inputs and continuous movement within the environment. Inspired by recent works in model-based RL and sequence-to-sequence machine learning models, we propose a deep reinforcement learning model with two key components: a world representation and dynamics modeling with transformers. Our main contribution lies in the integration of transformers into world models.

World Model

Our world model consists of several high-level components: (1) an encoder from observations (images) to a latent state space, (2) a latent dynamics model that imagines trajectories in the latent space, and (3) an actor-critic model that predict actions and reward of imagined trajectories. The agent makes decisions by imagining trajectories in the latent space of the world model based on past experience, and estimating trajectory rewards through learned action and value models. In this work, we focus on the latent dynamics component. We first define the following:

- o_t is the observation at time t
- $\hat{o}_t$ is the reconstructed observation at time t
- a_t is the action at time t
- s_t is a stochastic state at time t that incorporates information about o_t
- $\hat{s}_t$ is a stochastic state at time t that does not incorporate information about o_t
- h_t is the deterministic state from which the s_t and $\hat{s}_t$ are predicted off of
- $\mathcal{M}$ is the memory length of the sequential model.

The model can thus be formulated by the following distributions where we use p for distributions that generate samples in the real environment, q for their approximations that enable latent imagination and ϕ to describe their shared parameters:

$$\begin{aligned}
\text{Transformer model:} \quad & h_t \sim f_\phi(h_{[t-\mathcal{M},t-1]}, s_{[t-\mathcal{M},t-1]}, a_{t-1}) \\
\text{Representation model:} \quad & s_t \sim p_\phi(h_t, o_t) \\
\text{Transition model:} \quad & \hat{s_t} \sim q_\phi(h_t) \\
\text{Image model:} \quad & \hat{o}_t \sim q_\phi(h_{t-1}, s_{t-1}) \\
\text{Reward model:} \quad & r_t \sim q_\phi(h_t, s_t).
\end{aligned}$$

The representation model encodes observations and actions to create continuous states s_t with non-markovian transitions. The transition model predicts future states in the latent space without seeing the corresponding observations that will later cause them. The image model reconstructs observations from model states. The reward model predicts the rewards given the model states. The policy is formed by imagining hypothetical trajectories in the compact latent space of the world model using the transition model, and choosing actions that maximize expected value.

We refer the reader to Hafner et al.'s (Hafner et al. 2020b) work for a more detailed description of the remaining components of the architecture which we largely base ours off of.

Dynamics Modeling with Transformers

Our model imagines trajectories in the latent space via transformers. Transformers (Vaswani et al. 2017) are neural nets that *transform* a given sequence of elements, such as the sequence of words in a sentence, into another sequence. Similarly to other sequence-to-sequence architectures, they consist of encoders and decoders to produce an output sequence from an input sequence. Recent works have shown that transformers achieve staggering improvement over previous sequence-to-sequence models. The attention mechanism can take into account several different inputs at the same time and decides which ones are important by attributing higher weights to those inputs.

By analyzing the auto-mutual information (across time lags) of sequence-to-sequence models, (Shen 2019) shows that the mutual information decays exponentially in temporal distance in RNNs, whereas long-range dependence can be captured efficiently by Transformers. The sequential data within sophisticated reinforcement learning tasks, such as self-driving cars, are highly correlated across time. As such, we expect transformers to have potential to better represent the latent state space and make predictions of future states. See Figure 1 for a visualization of transformer memory capabilities.

The transformer takes the past $\mathcal{M}$ deterministic states $h_{[t-M,t-1]}$, stochastic states $s_{[t-M,t-1]}$, and action a_t to predict future states h_t. Observations are encoded via an encoder/decoder model. The transformer imagines future states $h_{\geq t}$ off of past $h_{[t-\mathcal{M},t-1]}$ and $s_{[t-\mathcal{M},t-1]}$, and the most recent action a_{t-1}. The imagined states are used to imagine the world (states, value, reward) in the future $\hat{s}_{\geq t}, \hat{v}_{\geq t}, \hat{r}_{\geq t}$, and find optimal policies $\hat{a}_{\geq t}$ within the imagined space. The hat operator indicates values that are predicted without their corresponding observations.

In our framework, we make use of recent literature studying architectural changes that may benefit transformers in reinforcement learning contexts. In particular, we use a 3-layer Gated Transformer-XL (Parisotto et al. 2019), which changes the position of the layer normalization and adds a gating layer.

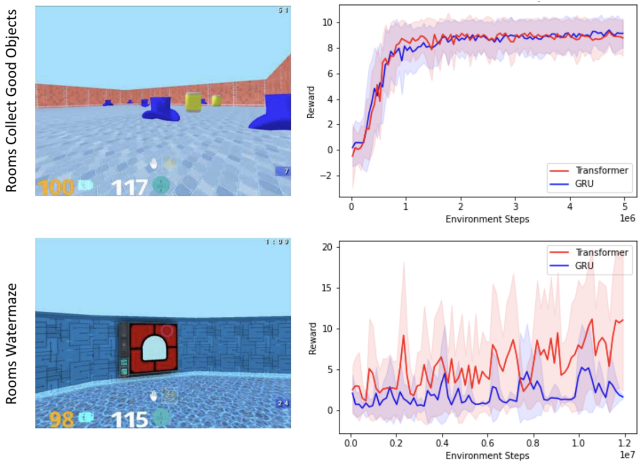


Figure 2: On Deepmind Lab tasks, Dreaming with Transformers achieves comparable or better performance than the original Dreamer agent.

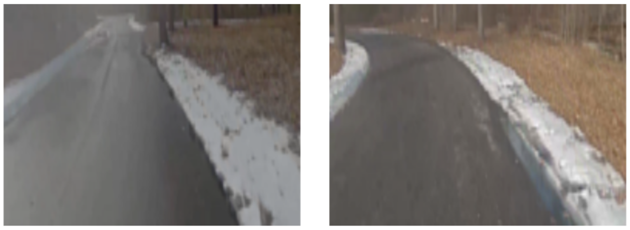


Figure 3: Outtakes from the VISTA driving simulator. When roads do not change in curvature (left), agents may succeed during sensor failures by repeating their last action. When failures occur at changes in road curvature (right), agents must predict curvature based on previous views of the road.

Preliminary Experiments

We experiment on short-term and long-term memory tasks. In the figures shown, the parameters used are mostly the same parameters originally used in Dreamer for Deepmind Lab, and each experiment was run five times. The agents using a GRU and using a transformer have exactly the same parameters, only different dynamics models. See appendix for experimental details.

Deepmind Lab

We first tested our framework on the Deepmind Lab (Beattie et al. 2016) tasks Collect Good Objects and Watermaze, as these two environments had been tested in the original Dreamer paper (results in appendix). Collect Good Objects is a short-term memory task that requires the agent to collect good objects and avoid bad objects, and Rooms

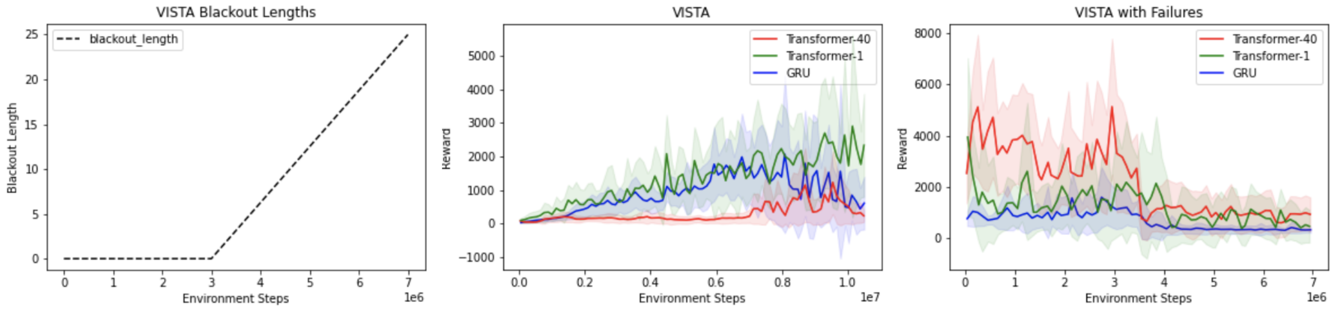


Figure 4: VISTA experiments. On the original VISTA task, the transformer with memory length 40 learns slowly, and the transformer with memory length 1 slightly outperforms the GRU (center). In the sensor failures task, blackout length increases steadily in frames as the agent takes more steps in the environment (left). The transformers are loaded with weights from the length-1 transformer trained on the original VISTA task, and the GRU is loaded with weights from the GRU trained on the original VISTA task. The agents are subject to the sensor failures task, where we see the length-40 transformer performing the best before a performance degradation at 3.5 million steps (right).

Watermaze is a slightly longer-term memory task that requires the agent to find a hidden platform and revisit it. For these experiments, we set the transformer memory length to 40. As shown in Figure 3, Dreaming with Transformers achieved very similar performance on Collect Good Objects as Dreamer and performed better on Watermaze. Note that the Dreamer results differ from those the ones reported in (Hafner et al. 2020a) because our framework builds on top of the changes introduced in DreamerV2, and we compare against DreamerV2.

VISTA

From the optimistic results in Deepmind Lab, we moved towards testing Dreaming with Transformers on a more real-world application, autonomous driving. We used the data-driven simulator VISTA (Amini et al. 2020), where our agent received visual input and output steering directions. The agent is rewarded +1 for every frame it successfully stays on the road. We were particularly interested in testing the potential long-term memory benefits brought by the transformer, so we looked into a navigation task that would require long-term memory: driving with sensor failures. In this task, the visual input periodically blacks out for a number of frames, and the agent must remember the last clear frames of the road to turn or drive straight as necessary to stay on the road. This task mimics scenarios where autonomous vehicles' sensors temporarily malfunction or are obstructed by obstacles or weather conditions. Driving through the beginning of a sensor failure may give a human more time to react and take control of the vehicle.

In our initial testing, we tried using a transformer with memory length 40 in our agent ('Transformer-40'). Figure 4 (center) shows that this agent learns very slowly on the original VISTA task without sensor failures. However, we found that an agent using a transformer with memory length 1 ('Transformer-1') can quickly learn to navigate well and slightly outperforms the agent using a GRU ('GRU'), whose performance consistently degrades after around 7 million steps. Though we expect transformers to help with tasks requiring memory, we see that transformer-powered RL agents can outperform the GRU agents even on the short-term memory original VISTA task. After this initial result, we tried loading the weights of the transformer with memory length 1 into the transformer with memory length 40.

Figure 4 (left) shows the setup of the sensor failure task. For the first 3 million steps, there are no sensor failures. Afterwards, the potential blackout length (in frames) experienced by the agent steadily increases as it takes more steps in the environment. At around 7 million steps, the agent is challenged to navigate when the sensor periodically receives black input for 25 frames. We verified that a human driver should be able to estimate the steering direction 25 frames in advance, so the maximum blackout length is not an impossible task. Note that the agent may not actually experience the full duration of the blackout if it drives off the road during the sensor failure.

Figure 4 (right) shows three models on the sensor failures task: the Transformer-40 agent, the Transformer-1 agent, and the GRU agent. Each model is loaded with weights from the best run on the original VISTA task after 10 million steps. The transformer-based agents are both loaded with weights from the Transformer-1 agent, and the GRU agent is loaded with weights from the GRU agent.

We see that after being loaded with the Transformer-1 agent's weights, the Transformer-40 agent performs better than the other two models. The Transformer-1 agent continues to slightly outperform the GRU agent. However, contrary to our expectations, all three models show performance degradation at around 3.5 million steps, when the maximum blackout length is only 4 frames. After 3.5 million steps, the Transformer-40 agent performs comparably with the Transformer-1 agent, despite the Transformer-40 agent's memory capacity theoretically allowing it to remember frames from the past. The investigation of these results would be an interesting direction for future work. We theorize one possibility is the blackout inputs interfere with the transformer's attention mechanism.

Discussion

In this work, we introduce Dreaming with Transformers, an extension of Dreamer which leverages the advantages of transformers in latent imagination-based reinforcement learning. We showed that Dreaming with Transformers can perform at least as well as Dreamer on two Deepmind Lab tasks and the VISTA driving simulator task. In the process, we showed that it is viable to quickly train a transformer with shorter memory length as an initialization point for a transformer with longer memory length in the context of reinforcement learning. Future directions include investigating whether inputs of zero can interfere with transformers' attention mechanism, studying what representations work well with transformers in reinforcement learning, and deeper exploration into the advantages transformers in world models can offer in long-term memory tasks.

References

Amini, A.; Gilitschenski, I.; Phillips, J.; Moseyko, J.; Banerjee, R.; Karaman, S.; and Rus, D. 2020. Learning Robust Control Policies for End-to-End Autonomous Driving From Data-Driven Simulation. *IEEE*, 5: 1143 – 1150.

Beattie, C.; Leibo, J. Z.; Teplyashin, D.; Ward, T.; et al. 2016. DeepMind Lab. arXiv:1612.03801.

Chen, L.; Lu, K.; Rajeswaran, A.; Lee, K.; Grover, A.; Laskin, M.; Abbeel, P.; Srinivas, A.; and Mordatch, I. 2021. Decision Transformer: Reinforcement Learning via Sequence Modeling. arXiv:2106.01345.

Cho, K.; Van Merriënboer, B.; Bahdanau, D.; and Bengio, Y. 2014. On the properties of neural machine translation: Encoder-decoder approaches. arXiv:1409.1259.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv:1810.04805.

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

Ha, D.; and Schmidhuber, J. 2018. Recurrent World Models Facilitate Policy Evolution. In Bengio, S.; Wallach, H.; Larochelle, H.; Grauman, K.; Cesa-Bianchi, N.; and Garnett, R., eds., *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc.

Haarnoja, T.; Zhou, A.; Abbeel, P.; and Levine, S. 2018. Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor. arXiv:1801.01290.

Hafner, D.; Lillicrap, T.; Ba, J.; and Norouzi, M. 2020a. Dream to Control: Learning Behaviors by Latent Imagination. arXiv:1912.01603.

Hafner, D.; Lillicrap, T.; Norouzi, M.; and Ba, J. 2020b. Mastering Atari with Discrete World Models. arXiv:2010.02193.

Hochreiter, S.; and Schmidhuber, J. 1997. Long short-term memory. *Neural computation*, 9(8): 1735–1780.

Janner, M.; Li, Q.; and Levine, S. 2021. Reinforcement Learning as One Big Sequence Modeling Problem. arXiv:2106.02039.

Lechner, M.; and Hasani, R. 2020. Learning Long-Term Dependencies in Irregularly-Sampled Time Series. arXiv:2006.04418.

Liu, J.; Gu, X.; and Liu, S. 2020. Reinforcement learning with world model. arXiv:1908.11494.

Parisotto, E.; Song, H. F.; Rae, J. W.; Pascanu, R.; Gulcehre, C.; Jayakumar, S. M.; Jaderberg, M.; Kaufman, R. L.; Clark, A.; Noury, S.; Botvinick, M. M.; Heess, N.; and Hadsell, R. 2019. Stabilizing Transformers for Reinforcement Learning. arXiv:1910.06764.

Shen, H. 2019. Mutual Information Scaling and Expressive Power of Sequence Models. arXiv:1905.04271.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; and Polosukhin, I. 2017. Attention Is All You Need. arXiv:1706.03762.

Experimental Details

The experiments run on Deepmind Lab use the same setup Dreamer originally used for Deepmind Lab (graciously provided by Danijar Hafner). The action space is discretized into the following possible actions: move backward, strafe left, strafe right, look left, look right, look left and forward, look right and forward, fire. The parameters are set the same way as they are set for Atari in the open source code[1], except that `kl_scale` is set to 0.3, `imag_gradient_mix` is set to 0.0, `actor_entropy` is set to 1e-4, `discount` is set to 0.99, and `precision` is set to 32. The only deviation from the original Deepmind Lab setup is that the batch size is reduced to 10 for memory purposes.

The experiments run on VISTA have the same setup, with the exceptions that `action_repeat` is set to 2, and `model_lr` is lowered to 5e-5, and `actor_lambda_n` is set to 1e-3. These small changes seemed to very slightly help both GRU-based and transformer-based agents on the VISTA environment. Additionally, since the action space is continuous, we set `actor_dist` as tanh_normal.

In both sets of experiments, the transformer has 3 layers, 4 attention heads of dimension 64, inner dimension of 1024, and dropout set to 0.1.

[1] https://github.com/danijar/dreamerv2