RECURRENT WORLD MODEL WITH TOKENIZED LATENT STATES

Guangyao Zhai[1,2] * **Xingyuan Zhang**[1,3] * **Nassir Navab**[1,2]

[1]Technical University of Munich [2]Munich Center for Machine Learning [3]Volkswagen AG

{guangyao.zhai, xingyuan.zhang}@tum.de

ABSTRACT

World models are getting more and more popular in recent years. We introduce a new architecture – TokenWM, that maintains the recurrent nature of state-space models while incorporating tokenized latent states and a memory-augmented attention mechanism to improve modeling capacity in complex environments. The preliminary results on LIBERO benchmarks demonstrate that the new architecture is more favorable to complex tasks than the popular RSSM architecture. We believe TokenWM introduces a new design paradigm for recurrent world models, enabling more expressive and scalable decision-making in complex environments. We will open source the code in a short notice.

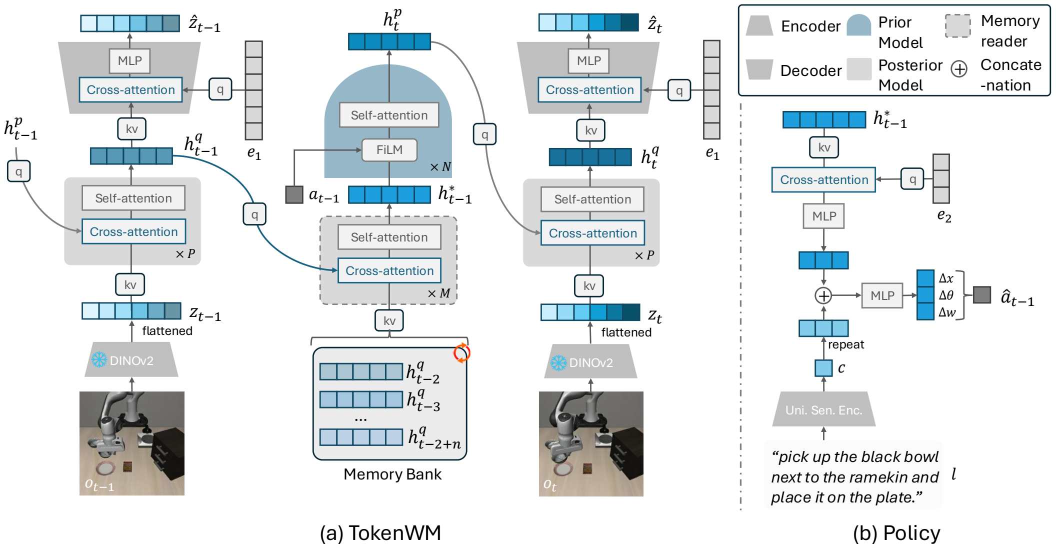
1 INTRODUCTION

The concept of a world model has become popular in recent years. Originally proposed in Ha & Schmidhuber (2018), a world model needs to learn both a meaningful representation of the state of the system (understanding) and a dynamic model that can predict the future states based on an action sequence (reasoning). Then, such a model can facilitate decision-making by doing planning within the world model instead of interacting with the real environments (Ha & Schmidhuber, 2018; Hafner et al., 2019a). This type of model is heavily studied in the model-based Reinforcement Learning (RL) field and also sometimes referring as State-Space Model (SSM) (Karl et al., 2017; Hafner et al., 2019b; Becker et al., 2019). The most well-known model from this family is the RSSM (Hafner et al., 2019b), which has been shown to be able to efficiently solve online RL on simulated locomotion tasks (Hafner et al., 2019b;a), games (Hafner et al., 2020; 2023) and real-world robots (Wu et al., 2023) and also Imitation Learning (IL) (Zhang et al., 2023b; Mazzaglia et al., 2024).

However, the RSSM structure cannot easily scaled to handle more complex environments such as Minecraft, as evidenced by the poor reconstruction from Hafner et al. (2023), and robot manipulations, as evidenced by the poor performances in Seo et al. (2022). On the other hand, due to the development of new techniques from other fields like transformers (Vaswani et al., 2017), linear attention models (Gu et al., 2021; Smith et al., 2022), and diffusion models (Alonso et al., 2024), recent works try to integrate these techniques into world models. One line of works use the new backbones to enhance the ability of SSM by replacing the recurrent models, e.g. GRU (Cho et al., 2014) in RSSM, with transformers (Chen et al., 2022; Zhang et al., 2023a) or linear attention models (Deng et al., 2023; Becker et al., 2024). Although the new backbones enhance the modeling capability of the dynamic model, the latent space is still a single vector, which limits the amount of information that can be modeled. Another line of work focuses only on prediction on the observation space, or a feature space of the observation, with the transformers and diffusion models (Micheli et al., 2023; Alonso et al., 2024) operating on a space of tokens, ignoring the need for representation learning.

In this paper, we study the possibility of an intermediate approach – we follow the recurrent nature of the SSM, designing a standard prior-to-posterior module framework so that it can still learn sequential state representation. The method is built on attention blocks accepting tokens as the information resource, which is step-by-step fused with tokenized states, enhancing the capacity of the model when facing complex environments. Due to the nature of limited memory preservation of recurrent

*The first two authors contribution equally.



(a) TokenWM

(b) Policy

Figure 1: Overview of the TokenWM and policy network. (a) A prior state h^p_{t-1} learns to query the information z_{t-1} from the new observation o_{t-1} to form the posterior state h^q_{t-1}, which is regularized by the decoder. h^q_{t-1} queries a memory bank filled with past states to form h^*_{t-1}, which is then sent to the prior module to predict h^p_t together with a_{t-1}. (b) h^*_{t-1} is queried by learnable tokens to have action features, which are then concatenated with the repeated embedding c out of a pre-trained language encoder. The fused features predict the action via several MLP layers.

attention blocks, we additionally design a memory bank and a reading mechanism to help consolidate the past information in the framework without losing the merits in previous SSMs. We coin this new architecture of world model TokenWM, which, to the best of our knowledge, is the first token-based SSM-style world model. We have some preliminary results showcasing the advantage of TokenWM over RSSM on the LIBERO benchmark (Liu et al., 2023).

2 Preliminary

We consider a language-conditioned POMDP problem defined by the tuple $\{S, A, T, L, R, O, \Omega\}$, where S is the state space, A is the action space, $T : S \times A \to S$ is the dynamic function, L is the space of language instruction, $R : S \times L \to \mathbb{R}$ is the reward function, O is the observation space, and $\Omega : S \to O$ is the emission function. The goal is to find a policy $\pi : S \times L \to A$ which maximizes the expected accumulated reward, or return, i.e. $\mathcal{R}(\pi) = \mathbb{E}_{a\sim\pi}[\sum_t r_t]$. The agent is given a dataset of demonstrations $\{(o_{1:T}, a_{1:T}, l)\}$ to learn the policy. Rewards are generally not available in the dataset and will only be used for evaluation.

3 TokenWM

In this section, we present the structure of TokenWM in detail. The framework largely follows the SSM-style world models, which typically have four components:

$$\begin{aligned} &\text{encoder } z_t = f_\phi(o_t), && \text{posterior } s_t \sim q_\phi(s_t|s_{t-1}, a_{t-1}, z_t), \\ &\text{decoder } o_t \sim p_\theta(o_t|s_t), && \text{prior } s_t \sim p_\theta(s_t|s_{t-1}, a_{t-1}). \end{aligned}$$

$f_\phi(o_t)$ is the encoder to extract the features from the observation; $q_\phi(s_t|s_{t-1}, a_{t-1}, z_t)$ and $p_\theta(s_t|s_{t-1}, a_{t-1})$ are the posterior and the prior of the latent state variable; while $p_\theta(o_t|s_t)$ is the decoder that decodes the observation distribution from the state. ϕ and θ represent the parameters of the inference model and the generative model, respectively.

As mentioned before, normally, the state s is a vector of $\mathbb{R}^D$ where D is the dimension of the state. But in TokenWM, we will implement it as a set of tokens $\mathbb{R}^{N_h \times D}$, where N_h is the number of tokens.

To not abuse the notation, we will use h to refer to the latent state in TokenWM. During this change, we need to redesign all four components in SSM to make them compatible with the new token-based format. We present an overview of the TokenWM structure in Figure 1 and will describe the designs on each component below.

3.1 Encoder

Normally, for image observation, the encoder is implemented as a CNN (Ha & Schmidhuber, 2018; Hafner et al., 2019b; 2020) to output a vector feature of the image to be compatible with the later modules. However, since we aim to work with tokens in TokenWM, we don't need to have such constraints. Thus, we generally consider adopting a ViT model (Dosovitskiy et al., 2020) as the encoder and take the patch feature as the output, which should allow richer information to be passed to the latent state. The adoption of ViT also opens the opportunity to use a lot of readily available visual foundation models to deal with high-resolution images, which is known to be one of the weaknesses of RSSM. To this end, the encoder in TokenWM, i.e. $z_t = f_\phi(o_t)$, outputs $z_t \in \mathbb{R}^{N_z \times D_z}$, where N_z is the number of the patch tokens and D_z is the feature dimension of patch tokens.

3.2 Prior model

In terms of functionality, the prior model $h_t^p = f_\theta^p(h_{t-1}, a_{t-1})$ needs to map one set of tokens to another set of tokens with the same size by integrating a vector. Inspired by RT-1 (Brohan et al., 2023), we use a combination of FiLM (Perez et al., 2018) and self-attention (Vaswani et al., 2017). For the prior model, first, a Positional Encoding (PE) is added to the tokens, and then it goes through a few blocks of FiLM and the post-norm self-attention layer. Note that we didn't use the more popular pre-norm structure for the self-attention layer because, in the experiment, we observed the pre-norm structure tends to diverge when rolling out the model for longer. We speculate that the direct residual connection in the pre-norm may not fit the recurrent scheme, while the post-norm structure renormalizes the output after every layer stabilizes the transition.

3.3 Posterior model

As a common practice, the posterior model is normally based on the output of the prior model so that some parameters are shared to make the prior model easier to train (Karl et al., 2017; Hafner et al., 2019b). Thus, we define the posterior model in TokenWM as $h_t^q = f_\phi^q(h_t^p, z_t)$. The structure of the posterior should be very similar to the prior, but instead of integrating a vector, a set of tokens z_t from the encoder needs to be integrated. We choose to use the cross-attention layer (Vaswani et al., 2017) to integrate the observation tokens. To be specific, the latent state provides a query, while the observation tokens provide the keys and values. In this way, the latent state can choose which part of the observation to focus on when updating the states. To this end, we summarize the structure of the posterior model: first PEs are added to both the prior state h_t^p and the observation tokens z_t, then they go through a few blocks of cross-attention and self-attention layers.

3.4 Decoder

The decoder needs to extract the relative information from the latent state h_t^q to output a prediction of the quantity we care about and provide guidance for training the world model. Without losing any generality, we also consider the outputs of the decoder to be a set of tokens, but it may not be the same number as the latent tokens. To handle this nature, we adopt a perceiver-decoder (Jaegle et al., 2021) structure, where a set of learnable tokens match the output size, serves as queries, and queries the latent tokens with a cross-attention layer. Then, the result tokens go through a shared MLP to the final outputs $\hat{o}_t$ or $\hat{z}_t$. Noted that, normally, we decode the original observation $\hat{o}_t$ for images. It can be directly tokenized by reshaping, but when we use a rich pre-trained encoder, we can output $\hat{z}_t$ instead. We will abuse the notation to use $\hat{z}_t$ in the rest of the paper.

3.5 Additional Memory Module

As we observed in complex real-robot tasks, there can be delays in the action execution, which may inhibit the model from learning a Markovian state. To combat that, we design an explicit memory

Table 1: Success rates on LIBERO benchmark.

	libero-spatial	libero-object	libero-goal	libero-long
RSSM-XL (Hafner et al., 2023)	26.4%	46.6%	18.6%	4.4%
TokenWM (Ours)	68.2%	87.2%	68.6%	22.6%

module for TokenWM. We implement a memory bank as a queue with a limited size. The memory bank maintains a small window of the past states. After the model infers the current state h_t, it queries the memory bank for the past information with a series of cross-attention and self-attention layers. Formally, we define the memory module as $h_t^* = f_\theta^M(h_t, [h_{t-n+2}, ..., h_{t-1}])$.

3.6 Policy

Although not a component of the world model, using the world model for a decision-making task normally also involves a policy module. In TokenWM, we define the policy as $\pi_\psi(a_t|h_t^*, c_t)$, where h_t^* is the augmented states after the memory module and c_t is an optional condition variable that offers context to the policy. For example, in the language-conditioned control, a representation of the language can be viewed as c_t. We adopt a similar structure with the decoder, where we first use the perceiver-decoder to extract the output tokens, then it optional concatenates with the condition and goes through MLPs for the final outputs. This structure also opens some new ways to design the policy, where we can also treat actions in a group form. For example, when doing robotics manipulation, the action space can be $\mathbb{R}^7$, where we have 3 for translation, 3 for rotation, and 1 for the gripper. We can semantically group the actions into 3 groups and hope each output token can capture more meaningful information for the group.

3.7 Loss function

The model is trained by optimizing three losses together, i.e. the information bottleneck loss, which replaces the KL loss, the reconstruction loss, and the policy loss, i.e.

$$L = \alpha||h_t^q - h_t^p||_2^2 + \beta||z_t - \hat{z}_t||_2^2 - \gamma \log \pi(a_t|h_t^*, c), \tag{1}$$

where α, β, and γ control the weight of each loss. Note that, in TokenWM, we discard the often-used probabilistic formulation for the latent state since, in the experiments, we find it makes the training unstable. We conjecture it could be caused by the independent sampling of multiple tokens. Thus, we use the deterministic formulation (Ghosh et al., 2019) to stabilize the training.

4 Experiment

We conduct preliminary experiments on the LIBERO benchmark (Liu et al., 2023) to showcase the performance gain of the TokenWM. The LIBERO benchmark was originally proposed as a testbench for lifelong learning, but in this paper, we are following Kim et al. (2024) to use it as a IL benchmark. We use four suites, namely spatial, object, goal, and long, for evaluation. Each suite consists of 10 languaged conditioned tasks and provides a dataset with 500 demonstrations. We use the filtered dataset from Kim et al. (2024), which removes the no-ops actions and re-renders the image observation to 224 x 224. Following previous works, the actions are discretized to 256 bins between the 1st and 99th percentile of the dataset distributions. We use a pre-trained DINOv2 (Oquab et al., 2023) as the encoder. For the language condition, we use a pre-trained Universal Sentence Encoder (Cer et al., 2018) from Reimers & Gurevych (2019). For the RSSM baseline, since it doesn't scale well to high-resolution images, we resize the image to 64 x 64 for it. We use the XL size model of RSSM from Hafner et al. (2023) to match the size of TokenWM for a fair comparison. Both models are trained with Adam optimizer (Kingma & Ba, 2017) with a learning rate of 1e-4, and the model is evaluated every 1000 gradient steps with 500 trajectories on the test suite (50 trajectories for each task). We report the best success rate during the training course. The results on the LIBERO benchmark are shown in Table 1. We can clearly see that TokenWM significantly outperforms RSSM on all four suites.

5 Discussion

In this paper, we propose TokenWM, a novel architecture for a recurrent world model. TokenWM can leverage the rich capacity of token-form representation to handle complex environments. Preliminary results on the LIBERO benchmark show that TokenWM significantly outperforms the popular RSSM model of similar size, demonstrating its great potential. In future work, we plan to test TokenWM in more settings, such as online RL, and conduct more ablation studies to understand the influence of different design choices. We believe our work opens new possibilities for designing more effective recurrent world models.

References

Eloi Alonso, Adam Jelley, Vincent Micheli, Anssi Kanervisto, Amos Storkey, Tim Pearce, and François Fleuret. Diffusion for World Modeling: Visual Details Matter in Atari. In *Thirty-Eighth Conference on Neural Information Processing Systems*, May 2024. URL `http://arxiv.org/abs/2405.12399`.

Philipp Becker, Harit Pandya, Gregor Gebhardt, Cheng Zhao, C. James Taylor, and Gerhard Neumann. Recurrent Kalman Networks: Factorized Inference in High-Dimensional Deep Feature Spaces. In *Proceedings of the 36th International Conference on Machine Learning*, pp. 544–552. PMLR, May 2019. URL `https://proceedings.mlr.press/v97/becker19a.html`.

Philipp Becker, Niklas Freymuth, and Gerhard Neumann. KalMamba: Towards Efficient Probabilistic State Space Models for RL under Uncertainty, June 2024. URL `http://arxiv.org/abs/2406.15131`.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alexander Herzog, Jasmine Hsu, Julian Ibarz, Brian Ichter, Alex Irpan, Tomas Jackson, Sally Jesmonth, Nikhil Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Isabel Leal, Kuang-Huei Lee, Sergey Levine, Yao Lu, Utsav Malla, Deeksha Manjunath, Igor Mordatch, Ofir Nachum, Carolina Parada, Jodilyn Peralta, Emily Perez, Karl Pertsch, Jornell Quiambao, Kanishka Rao, Michael Ryoo, Grecia Salazar, Pannag Sanketi, Kevin Sayed, Jaspiar Singh, Sumedh Sontakke, Austin Stone, Clayton Tan, Huong Tran, Vincent Vanhoucke, Steve Vega, Quan Vuong, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Tianhe Yu, and Brianna Zitkovich. RT-1: Robotics Transformer for Real-World Control at Scale. In *Robotics: Science and Systems XIX*. Robotics: Science and Systems Foundation, July 2023. ISBN 978-0-9923747-9-2. doi: 10.15607/RSS.2023.XIX.025. URL `http://www.roboticsproceedings.org/rss19/p025.pdf`.

Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Yun-Hsuan Sung, Brian Strope, and Ray Kurzweil. Universal Sentence Encoder, April 2018. URL `http://arxiv.org/abs/1803.11175`.

Chang Chen, Yi-Fu Wu, Jaesik Yoon, and Sungjin Ahn. TransDreamer: Reinforcement Learning with Transformer World Models, February 2022. URL `http://arxiv.org/abs/2202.09481`.

Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. In Alessandro Moschitti, Bo Pang, and Walter Daelemans (eds.), *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1724–1734, Doha, Qatar, October 2014. Association for Computational Linguistics. doi: 10.3115/v1/D14-1179. URL `https://aclanthology.org/D14-1179/`.

Fei Deng, Junyeong Park, and Sungjin Ahn. Facing Off World Model Backbones: RNNs, Transformers, and S4. In *Thirty-Seventh Conference on Neural Information Processing Systems*, November 2023. URL `https://openreview.net/forum?id=GDYuzX0rwj`.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An Image is Worth 16x16 Words: Transformers for Image Recognition at

Scale. In *International Conference on Learning Representations*, October 2020. URL https://openreview.net/forum?id=YicbFdNTTy.

Partha Ghosh, Mehdi S. M. Sajjadi, Antonio Vergari, Michael Black, and Bernhard Scholkopf. From Variational to Deterministic Autoencoders. In *International Conference on Learning Representations*, September 2019. URL https://openreview.net/forum?id=S1g7tpEYDS.

Albert Gu, Karan Goel, and Christopher Re. Efficiently Modeling Long Sequences with Structured State Spaces. In *International Conference on Learning Representations*, October 2021. URL https://openreview.net/forum?id=uYLFoz1vlAC.

David Ha and Jürgen Schmidhuber. World Models. March 2018. doi: 10.5281/zenodo.1207631. URL http://arxiv.org/abs/1803.10122.

Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to Control: Learning Behaviors by Latent Imagination. In *ICLR 2020*, September 2019a. URL https://openreview.net/forum?id=S1lOTC4tDS.

Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels. In *Proceedings of the 36th International Conference on Machine Learning*, pp. 2555–2565. PMLR, May 2019b. URL https://proceedings.mlr.press/v97/hafner19a.html.

Danijar Hafner, Timothy P. Lillicrap, Mohammad Norouzi, and Jimmy Ba. Mastering Atari with Discrete World Models. In *International Conference on Learning Representations*, September 2020. URL https://openreview.net/forum?id=0oabwyZbOu.

Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering Diverse Domains through World Models, January 2023. URL http://arxiv.org/abs/2301.04104.

Andrew Jaegle, Sebastian Borgeaud, Jean-Baptiste Alayrac, Carl Doersch, Catalin Ionescu, David Ding, Skanda Koppula, Daniel Zoran, Andrew Brock, Evan Shelhamer, Olivier J. Henaff, Matthew Botvinick, Andrew Zisserman, Oriol Vinyals, and Joao Carreira. Perceiver IO: A General Architecture for Structured Inputs & Outputs. In *International Conference on Learning Representations*, October 2021. URL https://openreview.net/forum?id=fILj7WpI-g.

Maximilian Karl, Maximilian Soelch, Justin Bayer, and Patrick van der Smagt. Deep Variational Bayes Filters: Unsupervised Learning of State Space Models from Raw Data. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=HyTqHL5xg.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan P. Foster, Pannag R. Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. OpenVLA: An Open-Source Vision-Language-Action Model. In *8th Annual Conference on Robot Learning*, September 2024. URL https://openreview.net/forum?id=ZMnD6QZAE6.

Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. In *3rd International Conference for Learning Representations*, San Diego, January 2017. doi: 10.48550/arXiv.1412.6980. URL http://arxiv.org/abs/1412.6980.

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. LIBERO: Benchmarking Knowledge Transfer for Lifelong Robot Learning. In *Thirty-Seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, November 2023. URL https://openreview.net/forum?id=xzEtNSuDJk.

Pietro Mazzaglia, Tim Verbelen, Bart Dhoedt, Aaron Courville, and Sai Rajeswar. GenRL: Multimodal-foundation world models for generalization in embodied agents. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, November 2024. URL https://openreview.net/forum?id=za9Jx8yqUA&referrer=%5Bthe%20profile%20of%20Sai%20Rajeswar%5D(%2Fprofile%3Fid%3D~Sai_Rajeswar2).

Vincent Micheli, Eloi Alonso, and François Fleuret. Transformers are Sample-Efficient World Models. In *The Eleventh International Conference on Learning Representations*, February 2023. URL https://openreview.net/forum?id=vhFu1Acb0xb.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mido Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning Robust Visual Features without Supervision. *Transactions on Machine Learning Research*, July 2023. ISSN 2835-8856. URL https://openreview.net/forum?id=a68SUt6zFt.

Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. FiLM: Visual Reasoning with a General Conditioning Layer. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1), April 2018. ISSN 2374-3468, 2159-5399. doi: 10.1609/aaai.v32i1.11671. URL https://ojs.aaai.org/index.php/AAAI/article/view/11671.

Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan (eds.), *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 3982–3992, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1410. URL https://aclanthology.org/D19-1410/.

Younggyo Seo, Danijar Hafner, Hao Liu, Fangchen Liu, Stephen James, Kimin Lee, and Pieter Abbeel. Masked World Models for Visual Control. In *6th Conference on Robot Learning*, June 2022. URL http://arxiv.org/abs/2206.14244.

Jimmy T. H. Smith, Andrew Warrington, and Scott Linderman. Simplified State Space Layers for Sequence Modeling. In *The Eleventh International Conference on Learning Representations*, September 2022. URL https://openreview.net/forum?id=Ai8Hw3AXqks.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is All you Need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

Philipp Wu, Alejandro Escontrela, Danijar Hafner, Pieter Abbeel, and Ken Goldberg. DayDreamer: World Models for Physical Robot Learning. In *Proceedings of The 6th Conference on Robot Learning*, pp. 2226–2240. PMLR, March 2023. URL https://proceedings.mlr.press/v205/wu23c.html.

Weipu Zhang, Gang Wang, Jian Sun, Yetian Yuan, and Gao Huang. STORM: Efficient Stochastic Transformer based World Models for Reinforcement Learning. *Advances in Neural Information Processing Systems*, 36:27147–27166, December 2023a. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/5647763d4245b23e6a1cb0a8947b38c9-Abstract-Conference.html.

Xingyuan Zhang, Philip Becker-Ehmck, Patrick van der Smagt, and Maximilian Karl. Action Inference by Maximising Evidence: Zero-Shot Imitation from Observation with World Models. In *Thirty-Seventh Conference on Neural Information Processing Systems*, November 2023b. URL https://openreview.net/forum?id=WjlCQxpuxU.

Appendix

A Hyper-parameters

The hyper-parameters for TokenWM have been shown in Table 2.

Table 2: TokenWM hyper-parameters.

Model	
World state size	64×512
Encoder structure	ViT
Encoder pre-trained weight	DINOv2-L
Prior layer	N=2
Posterior layer	P=4
Memory reader layer	M=2
All attention head	8
Memory bank size	5×64×512
Policy MLP layer	2
Model Size	113M
Training	
Batch Size	32
Horizon	64
Optimizer	AdamW
Training step	16,000
Gradient clip	100
Model & Policy initial learning rate	1e-4
Reconstruction loss weight	α=10.0
State loss weight	β=0.1
Policy loss weight	γ=1.0

B Training details and ablations

Policy cheating During training, we find a failure mode for the RSSM model on the LIBERO benchmark exists. LIBERO tasks normally involve a few steps to solve. For example, for the task "Pick up the black bowl between the plate and the ramekin and place it on the plate," one needs to first approach the black bowl, pick it up, move to the plate, and then drop the bowl. During each step, the actions can be very similar to each other, and only at the changing point between steps can the actions change dramatically. When we allow the gradient from the policy to update the model weights, the policy can cheat by forcing the state to store the previous action and decode the action again as the current action. So, for RSSM, we have to detach the policy's gradient from influencing the model learning.

On the other hand, TokenWM seems to suffer less from this problem as it can train stabling with the gradient from the policy. We conjecture that this robustness may be from the FiLM layer we use to integrate the action. As the FiLM layer uses the action to compute a scale and a shift to modify the latent state, it may be much harder for the policy to decode the previous action.

Remove the decoder to boost the policy performance Although the decoder provides essential guides to learn meaningful latent states, it may have a conflict of interest with the policy. Thus, we explore a two-phase training scheme, achieving the maximal performance on the policy side by removing the decoder from some point.

In the first phase, the model is trained with both the decoder and the policy, guaranteeing a meaningful state embedding flow. After a certain amount of gradient steps, we remove the decoder, i.e., by setting the decoder loss weight β to 0. In the meantime, we also find to scale up the policy loss weight γ to 100 so that the information bottleneck does not collapse after the decoder is removed. We show the learning curve of TokenWM on the libero-spatial suite in Figure 2. As we can see, after removing the decoder at gradient step 10000, the success rate immediately drops to 0 since the previous balance

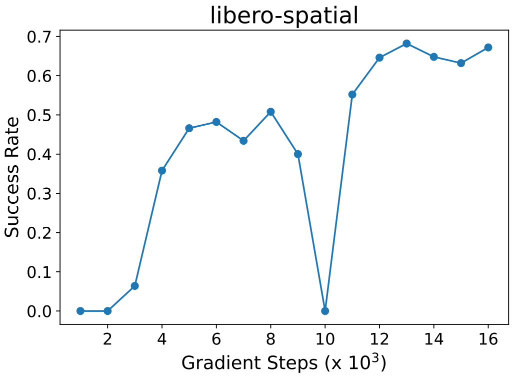


Figure 2: Learning curve of TokenWM on libero-spatial suite.

Table 3: Ablation of the memory module.

	libero-spatial
TokenWM w/o memory	60.4%
TokenWM w/ memory	68.2%

between the target and the information bottleneck has been broken. Then, it quickly recovers to a new balance point between policy loss and information bottlenecks and reaches a significantly higher success rate than before.

Effect of the memory module To showcase the effect of the memory module, we conduct an ablation study on the libero-spatial suite. The result is shown in Table 3. We can see that, although LIBERO is a simulated environment without much delay, having the memory module still gives a better success rate. We believe it is evident that the memory module can boost the model's capacity in general.