BEYOND SINGLE-STEP: MULTI-FRAME ACTION-CONDITIONED VIDEO GENERATION FOR REINFORCEMENT LEARNING ENVIRONMENTS

Zongyue Li[1,2,*], **Sikuan Yan**[1,*], **Yunpu Ma**[1,2,*,†], **Yusong Li**[1], **Xing Lyu**[1], **Matthias Schubert**[1,2]
[1]LMU Munich [2]Munich Center for Machine Learning (MCML)
{zongyue.li, matthias.schubert}@lmu.de, cognitive.yunpu@gmail.com

ABSTRACT

World models achieved great success in learning the dynamics from both low-dimensional and high-dimensional states. Yet, there is no existing work to address multi-step generation task with high dimensional data. In this paper, we propose the first action-conditioned multi-frame video generation model, advancing world model development by generating future states from actions. As opposed to recent single-step or autoregressive approaches, our model directly generates multiple future frames conditioned on past observations and action sequences. Our framework extends its capabilities to action-conditioned video generation by introducing an action encoder. This addition enables the spatiotemporal variational autoencoder and diffusion transformer in Open-Sora to effectively incorporate action information, ensuring precise and coherent video generation. We evaluated performance on Atari environments (Breakout, Pong, DemonAttack) using MSE, PSNR, and LPIPS. Results show that conditioning solely on future actions and embedding-based encoding improve generation accuracy and perceptual quality while capturing complex temporal dependencies like inertia. Our work paves the way for action-conditioned multi-step generative world models in dynamic environments.

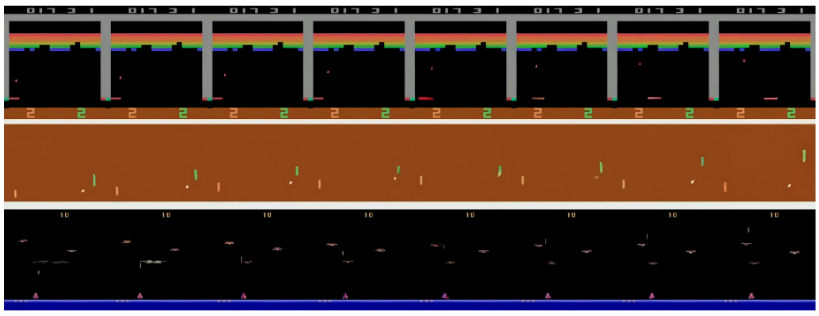

Figure 1: Examples of our generation results for three Atari games: Breakout (top row), Pong (middle row), and DemonAttack (bottom row). The four leftmost frames in each row represent the condition frames provided to the model, while the subsequent frames are generated based on the condition frames and corresponding action sequences. The generated frames demonstrate the model's ability to produce temporally consistent and action-aligned predictions.

1 INTRODUCTION

In the pursuit of Artificial General Intelligence, diffusion-based world models serve as fundamental building blocks for planning and reasoning. The introduction of the Sora video generation model

[*]Equal contribution.
[†]Corresponding author.

Table 1: A comparison of representative diffusion-model based world models

Method	Diffusion Model	State Modality	Multi-Step
DIAMOND Alonso et al. (2024)	$p(s_{n+T}\|s_{n:n+T-1}, a_{n:n+T-1})$	visual frame	✗
Diffuser Janner et al. (2022)	$p(a_{n:n+T-1}, s_{n+1:n+T}\|s_n)$	low-dimensional	✓
DD Ajay et al. (2023)	$p(s_{n+1:n+T-1}\|s_n, g_n)$	low-dimensional	✓
DWM Ding et al. (2024)	$p(r_{n:n+T-1}, s_{n+1:n+T-1}\|s_n, a_n, g_n)$	low-dimensional	✓
GameNgine Valevski et al. (2024)	$p(s_{n+1}\|s_{0:n}, a_{0:n})$	visual frame	✗
D-MPC Zhou et al. (2024)	$p(s_{n+1:n+T}\|s_{0:n}, a_{0:n+T-1})$	low-dimensional	✓
Our best model	$p(s_{n+1:n+T}\|s_{0:n}, a_{n:n+T-1})$	visual frame	✓

Brooks et al. (2024) has attained significant attention. However, most existing diffusion-based world models are constrained to either single-step (SS) next-frame generation, such as GameNgine Valevski et al. (2024), DIAMOND Alonso et al. (2024) or multi-step (MS) low-dimensional state representations generation Zhou et al. (2024); Ding et al. (2024). We provide a comprehensive summary of current related works in Table 1.

Building on the progress of action-conditioned diffusion-based generative models like DIAMOND and GameNgine, which have demonstrated strong capabilities in video generation, we explore the potential of MS video generation under action-conditioning scenarios. Following Zhou et al. (2024), we condition our model on action sequences to guide the generation process. While existing video generation models primarily focus on SS generation, MS generative models are limited to low-dimensional state representation generation. In contrast, our work bridges this gap by enabling the generation of multiple frames at once in high-dimensional state spaces. Our MS approach reduces the redundancy inherent in step-by-step generation, resulting in a more computationally efficient framework.

Furthermore, although DIAMOND claims to mitigate compounding errors effectively, Ding et al. (2024) emphasizes that reducing the frequency of calls to the world model is crucial for minimizing error accumulation. Moreover, Bar et al. (2024); Earle et al. (2024) report that both DIAMOND and GameNgine still produce trajectories with significant compounding errors. Our newly proposed MS approach mitigates these errors by predicting several future steps conditional to a given action sequence in parallel.

To make video-based world models practically effective and efficient, two primary challenges must be addressed. First, the accuracy of the dynamics model is crucial in mitigating compounding errors Venkatraman et al. (2015); Asadi et al. (2019), where small prediction errors accumulate over time and lead to significant deviations from the true trajectory. Second, while the Sora model generates videos based on language prompts, our objective is to shift from text-conditioning to action-conditioned generation. This requires adapting to a different data modality—specifically, action sequences while ensuring the generated future frames remain visually realistic and temporally consistent.

To address these challenges, we introduce the first **action-conditioned MS** video generation model that learns a joint trajectory-level representation of world dynamics. Specifically, given a condition length n, and a generation length T, we model the world dynamics $p(s_{(n+1):(n+T)}|s_{0:n}, a_{i:(n+T-1)})$ with different action conditions, where $i \in \{0, n\}$, re-formulating the state-of-the-art OpenSora framework Zheng et al. (2024). Furthermore, we investigate how historical action sequences influence video generation results. Our key contributions are as follows:

- We propose the first action-conditioned multi-step video generation model that directly generates multiple future frames without relying on the autoregressive framework, addressing a critical gap in the literature.
- The proposed action-conditioned world model is based on OpenSora. We shift the original text-conditioned generation approach into an action condition generation. Our model enhances the accuracy of the generation process by perfectly aligning the action conditions with the generated frames.

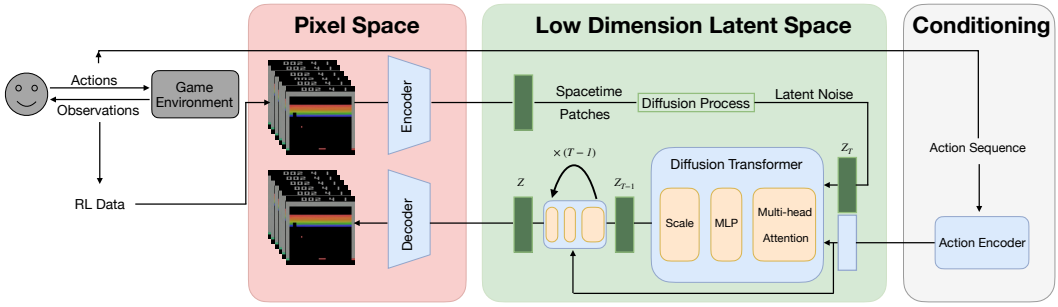

Figure 2: Overview of the world model pipeline

- Extensive experiments show that conditioning solely on future actions improves generation quality compared to incorporating both past and future actions. Additionally, applying an embedding layer for action representation significantly enhances accuracy over one-hot encodings.

We evaluate our approach on the Atari Mnih et al. (2013) benchmarks, our results demonstrate the model's effectiveness in generating high-quality video sequences with modality shifts while maintaining reasonable compounding errors. Our findings challenge conventional assumptions about the importance of past actions in world modeling, suggesting a promising direction for future research.

2 Related Work

World models Ha & Schmidhuber (2018) are generative models of environments, and we can categorize different world models mainly as single-step (SS) or multi-step (MS). We structure related work in Table 1. SS model the dynamics as $p(s_{n+1}|s_{n-H:t}, a_{n-H:n})$ (where H indicates the length of the sequence for dynamics modeling), so we predict the next state s_{n+1} conditioned on past observations of states and actions Alonso et al. (2024); Valevski et al. (2024). On the contrast, MS methods model the joint distribution at an episode level, which could be formulated as $(s_{n+1:n+T}|s_{n-H:n}, a_{n-H:n+T-1})$ Ding et al. (2024); Zhou et al. (2024). In contrast with previous MS methods, our method is capable of generating videos conditioned to action sequences instead of generating dense low dimensional representation Tassa et al. (2018).

Video Generation as a general world model The video generation models are considered as world models that create realistic videos, which requires the video generative model to understand and simulate the mechanism in the physical world. The emergence of the Sora model has attracted attention because of its generation and simulation capabilities, and OpenSora Zheng et al. (2024) was the original inspiration for our methods, is originally designed for text-prompted video generation. Liu et al. (2024) categorizes video generative models into three main classes: text-prompted, image-prompted, and video-prompted generation. However, none of these approaches focus on action sequences, making our work fundamentally different as we introduce action-conditioned video generation.

3 Method

We present a framework for generating action-conditioned trajectories using a diffusion-based world model. Our approach consists of two key phases: (1) collecting interaction data from an RL environment and (2) training a world model based on OpenSora, incorporating customized adaptations for action-conditioned video generation. To ensure stable training and effective trajectory modeling, we employ a teacher-forcing objective Valevski et al. (2024).

Figure 2 provides an overview of our method. In the first phase, an agent interacts with the environment by executing actions and receiving observations, generating a dataset of trajectories. In the second phase, we fine-tuned a pre-trained video generation model based on OpenSora Zheng et al. (2024), modifying its conditioning mechanism to incorporate action sequences. The model

Table 2: Results for video generation across different Atari environments, condition on future actions *only*. Lower MSE and LPIPS, and higher PSNR indicate better performance.

Game	Generation Length	MSE ↓	PSNR ↑	LPIPS ↓
Breakout	1 frame	13.1859	36.9485	0.0132
	2 frames	13.3697	36.8881	0.0356
	4 frames	14.3638	36.5727	0.0662
Pong	1 frame	12.2973	37.4956	0.0122
	2 frames	12.4529	37.3184	0.0217
	4 frames	12.9572	37.8253	0.0390
DemonAttack	1 frame	4.6977	41.4244	0.0558
	2 frames	5.0467	41.1289	0.0665
	4 frames	5.1662	41.0179	0.0670

learns to generate temporally coherent video by integrating action-conditioned cross-attention at every denoising step. Further details on data collection and model training are provided below. Implementation details are provided in Appendix B.

3.1 Data Collection via Agent Play

An *Interactive Environment* $\mathcal{E}$ is defined by a latent state space S, a visual space of partial projections of the latent state O, a partial projection function $f : S \to O$, an action space A, and a transition probability function that indicates the dynamics $p(s|a, s')$ such that $s, s' \in S$, and $a \in A$.

Our objective is to develop a world model capable of generating high-quality video data. In this context, the policy π for data collection is not a primary focus, as it is only responsible for sampling data from the true environment $\mathcal{E}$. In this case, to construct the dataset $\mathcal{T}_{PPO}$, we utilize a standard pre-trained PPO Schulman et al. (2017); Anand et al. (2019) RL agent. We collected 1M interactions for each of the environments used for training our world model.

3.2 Training the World Model

We repurpose a pre-trained action-to-video diffusion model based on OpenSora Zheng et al. (2024), fine-tuning it on collected trajectories $\mathcal{T}_{PPO}$ with teacher forcing objective. Specifically, we condition the model f_θ on sequences of actions $a_{i:n+T-1}$, where $i \in \{0, n\}$ and T represents the generation length. To condition on actions, we first encode them using an embedding layer network $\mathcal{E}_\mathcal{A}$ that learns an action embedding A_{emb}, which maps each action to a single token. This embedding is then used to replace the text embedding in the cross-attention module Chen et al. (2021) from OpenSora, as shown in the right side of Figure 2. For conditioning on observations, we followed the OpenSora to apply the Spatio-temporal Variational Autoencoder (VAE) Kingma & Welling (2022), denoted as ϕ to encode the sequence of observations $o_{0:n+T}$ into a latent space, facilitating their use in the diffusion generation process. During the diffusion process, a mask m is applied, ensuring that Gaussian noise is only added to $o_{n:n+T}$ at each diffusion step t, denoted as $\phi(o_{0:n+T}, m)$. In the denoising process, the diffusion model needs to denoise $\phi(o_{0:n+T})$.

We train the model to minimize the diffusion loss with velocity model Salimans & Ho (2022):

$$\mathcal{L} = \mathbb{E}_{t,\epsilon,\mathcal{T}}\left[\|v(\epsilon, x_0, t) - v'_\theta(x_t, t, \mathcal{E}_\mathcal{A}(a_{i:n+T-1}))\|_2^2\right] \tag{1}$$

where $\mathcal{T} = \{o_{0:n+T}, a_{i:n+T-1}\} \sim \mathcal{T}_{PPO}$, $i \in \{0, n\}$, $x_0 = \phi(o_{0:n+T}, m)$, $t \sim \mathcal{U}(0, 1)$, $\epsilon \sim \mathcal{N}(0, I)$. The diffusion process is defined as $x_t = \sqrt{\bar{\alpha}_t}x_0 + \sqrt{1 - \bar{\alpha}_t}$, with the corresponding reverse process given by $v(\epsilon, x_0, t) = \sqrt{\bar{\alpha}_t}\epsilon - \sqrt{1 - \bar{\alpha}_t}x_0$, and v'_θ represents the predicted velocity output by the model f_θ. During training, the parameters of the VAE $\phi(\cdot)$ are kept frozen. The noise schedule $\sqrt{\bar{\alpha}_t}$ is described in Liu et al. (2022).

Table 3: Results for video generation across different action conditions for Breakout Environment. Lower MSE and LPIPS, and higher PSNR indicate better performance.

Generation Length	Action Conditions	MSE ↓	PSNR ↑	LPIPS ↓
1 frames	1 Future Action	**13.1859**	**36.9485**	**0.0132**
	Past + 1 Future Action	13.2076	36.7749	0.0145
2 frames	2 Future Action	13.3697	**36.8881**	**0.0356**
	Past + 2 Future Action	**13.2152**	36.6384	0.0376
4 frames	4 Future Action	**14.3638**	**36.5727**	0.0662
	Past + 4 Future Action	14.5893	36.1825	**0.0554**

4 Experiments and Discussion

4.1 Experimental Setup

To thoroughly evaluate our approach, we utilize the RL benchmark Atari Mnih et al. (2013), which includes 26 games to assess the quality of generated videos. Our experiments focus on three representative environments: Breakout, Pong, and DemonAttack. This benchmark was chosen as it is also used by DIAMOND for evaluation. Additionally, our work paves the way for future research by exploring the potential and broader applications of our approach in the RL domain.

Our evaluation aims to address the following research questions: (1) Can the text-conditioned world model adapt to an action-conditioned world model with minimal data and capture environment dynamics? (2) How should action sequence conditioning and action embeddings be designed to enhance the model's understanding of the environment? (3) Is the world model capable of effectively learning and simulating inertia?

We investigated these questions through qualitative and quantitative experiments and analysis. For the quantitative analysis, we measure Mean Squared Error (MSE), peak signal-to-noise ratio (PSNR), and LPIPS Zhang et al. (2018) to compare the generated frames with ground-truth observations at both the pixel and perceptual levels. For the qualitative analysis, we visualized generated video frames under different action conditions.

4.2 Experimental Results

Table 2 shows the quantitative results of our approach on the three Atari environments with conditions $a_{n:n+T-1}$. As there are no current works that evaluate Atari with such metrics, we simply compare the performance differences on different condition designs to facilitate discussion. Table 3 shows the results on the environment Breakout with different action conditions, namely $a_{1:n+T-1}$ and $a_{n:n+T-1}$. As observed in both Tables, the MSE and PSNR metrics remain relatively stable across different frame generation lengths, indicating that the generated frames are pixel-wise similar to the ground truth. Across all games, as the generation length increases from 1 to 4 frames, MSE rises by 1.2–6.9%, PSNR decreases by only (-)1.3–0.8%, suggesting that while errors accumulate, the degradation remains within a controlled and reasonable range. These findings indicate that the compounding error is effectively managed and does not escalate drastically over time in pixel level. However, the LPIPS scores reveal that as more frames are generated, perceptual differences accumulate between the generated sequences and the ground truth.Moreover, Table 3 shows that the generation results improve when conditioning solely on future actions. To further examine the impact of action embedding design on performance, we compared two different action embedding strategies and report the results in Table 4. Our findings also suggest that the learned action embedding layer significantly outperforms the one-hot action vector, implying that a more refined action embedding design could enhance perceptual quality. This insight opens up potential avenues for future research.

Additionally, we conducted a qualitative evaluation to gain deeper insights for discussion. Figure 3 presents a set of examples from the Breakout environment, allowing us to assess whether the world model correctly interprets the given conditions and generates plausible outputs. Furthermore, it en-

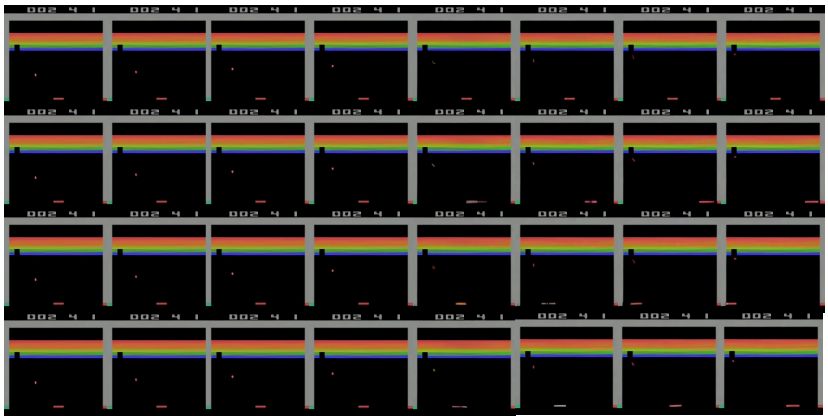

Figure 3: frames for inertia analysis

ables us to evaluate the model's ability to understand and simulate inertia. The first four columns depict the same observation conditions, where the paddle's position remains unchanged, while different future action sequences are applied. The generated frames illustrate that the paddle moves according to the given action conditions $a_{n:n+4}$. The last four frames in Figure 3, from top to bottom, correspond to the action sequences **0000, 2222, 3333, 3222**, where actions are defined as {**0: Noop, 1: Fire, 2: Right, 3: Left**}. By analyzing the generated frames, we observe that the model captures inertia effects. For instance, in the second row, the paddle's leftward motion intensifies as the action **2** is repeated. Similarly, in the third row, where the action sequence is **3333**, the paddle's movement follows the same pattern, but in the opposite direction, reinforcing the model's ability to simulate inertia in both directions. In the last row, the action **2** counteracts the previous leftward movement, gradually bringing the paddle to a stop. Additional results for 1 step and 2 steps generations can be found in Appendix C.

The results demonstrate that our model can effectively perform domain adaptation with minimal data, transitioning from text conditioning to action conditioning, while also generating reasonable trajectories under different action conditions. Moreover, the model successfully learns and simulates inertia.

4.3 Discussion

Can the text-conditioned world model adapt to an action-conditioned world model with minimal data and capture environment dynamics? The pre-trained OpenSora model was originally trained on text-video data. We show that it can be effectively adapted to a different data modality using an RL dataset with just 1M environment interactions. In contrast, OpenSora pretraining involved 10M samples from WebVid-10M Bain et al. (2022) and 20M samples from Panda-70M Chen et al. (2024), followed by a final training stage on a dataset of 2M video clips Zheng et al. (2024). Our quantitative and qualitative results show that the model successfully leverages action sequences instead of text to generate visually coherent frames, indicating a strong understanding of this new data modality. By conditioning only on future actions, the model preserves consistency between the generated frames and action inputs, further reinforcing its capability to align visual predictions with structured temporal dynamics.

How should action sequence conditioning and action embeddings be designed to enhance the model's understanding of the environment? Our analysis indicates that incorporating past action information introduces redundancy, as it is already implicitly encoded in the previously generated frames. Explicitly conditioning on past actions may degrade performance by introducing ambiguity between past and future inputs. Instead, relying solely on future actions leads to more precise and controllable frame generation, particularly for short-horizon predictions. Based on these findings, we show the "future actions only" conditioning approach to enhance both data efficiency and predictive accuracy.

Does the video diffusion World Model understand dynamics behind the video? Our qualitative experiments demonstrate that the model not only generates visually coherent frames but also faithfully adheres to the inherent physical dynamics of the game environments, such as inertia. The model accurately simulates the effects of different action sequences, ensuring that generated frames reflect not only the immediate action input but also the implicit continuation of prior motion. This capability highlights the model's ability to encode and predict dynamic transitions in Atari environments with high temporal consistency.

5 CONCLUSION

We introduced a novel action-conditioned multi-step video generation model that enables direct video generation while preserving temporal coherence and visual fidelity. Our approach integrates an action encoder that adapt OpenSora to efficiently capture complex temporal dependencies in action-conditioned video data. Experimental results demonstrate that (1) the text-video generative model can be efficiently adapted to an action-video generative model, (2) conditioning solely on future actions improves generation performance by eliminating redundant past information, and (3) learned action embeddings outperform one-hot encodings for action representation. Notably, the model successfully captures inertia effects in Atari environments, showcasing its ability to generate realistic, action-conditioned sequences. These findings highlight the model's effectiveness in producing high-quality, temporally consistent video predictions and provide a new framework for model-based reinforcement learning.

REFERENCES

Anurag Ajay, Yilun Du, Abhi Gupta, Joshua Tenenbaum, Tommi Jaakkola, and Pulkit Agrawal. Is conditional generative modeling all you need for decision-making?, 2023. URL https://arxiv.org/abs/2211.15657.

Eloi Alonso, Adam Jelley, Vincent Micheli, Anssi Kanervisto, Amos Storkey, Tim Pearce, and François Fleuret. Diffusion for world modeling: Visual details matter in atari, 2024.

Ankesh Anand, Evan Racah, Sherjil Ozair, Yoshua Bengio, Marc-Alexandre Côté, and R Devon Hjelm. Unsupervised state representation learning in atari. *arXiv preprint arXiv:1906.08226*, 2019.

Kavosh Asadi, Dipendra Misra, Seungchan Kim, and Michel L. Littman. Combating the compounding-error problem with a multi-step model, 2019. URL https://arxiv.org/abs/1905.13320.

Max Bain, Arsha Nagrani, Gül Varol, and Andrew Zisserman. Frozen in time: A joint video and image encoder for end-to-end retrieval, 2022. URL https://arxiv.org/abs/2104.00650.

Amir Bar, Gaoyue Zhou, Danny Tran, Trevor Darrell, and Yann LeCun. Navigation world models, 2024. URL https://arxiv.org/abs/2412.03572.

Tim Brooks, Bill Peebles, Connor Holmes, Will DePue, Yufei Guo, Li Jing, David Schnurr, Joe Taylor, Troy Luhman, Eric Luhman, Clarence Ng, Ricky Wang, and Aditya Ramesh. Video generation models as world simulators. 2024. URL https://openai.com/research/video-generation-models-as-world-simulators.

Chun-Fu Chen, Quanfu Fan, and Rameswar Panda. Crossvit: Cross-attention multi-scale vision transformer for image classification, 2021. URL https://arxiv.org/abs/2103.14899.

Tsai-Shien Chen, Aliaksandr Siarohin, Willi Menapace, Ekaterina Deyneka, Hsiang wei Chao, Byung Eun Jeon, Yuwei Fang, Hsin-Ying Lee, Jian Ren, Ming-Hsuan Yang, and Sergey Tulyakov. Panda-70m: Captioning 70m videos with multiple cross-modality teachers, 2024. URL https://arxiv.org/abs/2402.19479.

Zihan Ding, Amy Zhang, Yuandong Tian, and Qinqing Zheng. Diffusion world model: Future modeling beyond step-by-step rollout for offline reinforcement learning, 2024. URL https://arxiv.org/abs/2402.03570.

Sam Earle, Samyak Parajuli, and Andrzej Banburski-Fahey. Dreamgarden: A designer assistant for growing games from a single prompt, 2024. URL https://arxiv.org/abs/2410.01791.

David Ha and Jürgen Schmidhuber. World models. 2018. doi: 10.5281/ZENODO.1207631. URL https://zenodo.org/record/1207631.

Michael Janner, Yilun Du, Joshua B. Tenenbaum, and Sergey Levine. Planning with diffusion for flexible behavior synthesis, 2022. URL https://arxiv.org/abs/2205.09991.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes, 2022. URL https://arxiv.org/abs/1312.6114.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow, 2022. URL https://arxiv.org/abs/2209.03003.

Yixin Liu, Kai Zhang, Yuan Li, Zhiling Yan, Chujie Gao, Ruoxi Chen, Zhengqing Yuan, Yue Huang, Hanchi Sun, Jianfeng Gao, Lifang He, and Lichao Sun. Sora: A review on background, technology, limitations, and opportunities of large vision models, 2024. URL https://arxiv.org/abs/2402.17177.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning, 2013. URL https://arxiv.org/abs/1312.5602.

William Peebles and Saining Xie. Scalable diffusion models with transformers, 2023. URL https://arxiv.org/abs/2212.09748.

Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models, 2022. URL https://arxiv.org/abs/2202.00512.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms, 2017. URL https://arxiv.org/abs/1707.06347.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models, 2022. URL https://arxiv.org/abs/2010.02502.

Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding, 2023. URL https://arxiv.org/abs/2104.09864.

Yuval Tassa, Yotam Doron, Alistair Muldal, Tom Erez, Yazhe Li, Diego de Las Casas, David Budden, Abbas Abdolmaleki, Josh Merel, Andrew Lefrancq, Timothy Lillicrap, and Martin Riedmiller. Deepmind control suite, 2018. URL https://arxiv.org/abs/1801.00690.

Dani Valevski, Yaniv Leviathan, Moab Arar, and Shlomi Fruchter. Diffusion models are real-time game engines, 2024. URL https://arxiv.org/abs/2408.14837.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2023. URL https://arxiv.org/abs/1706.03762.

Arun Venkatraman, Martial Hebert, and J. Andrew Bagnell. Improving multi-step prediction of learned time series models. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, AAAI'15, pp. 3024–3030. AAAI Press, 2015. ISBN 0262511290.

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric, 2018. URL https://arxiv.org/abs/1801.03924.

Zangwei Zheng, Xiangyu Peng, Tianji Yang, Chenhui Shen, Shenggui Li, Hongxin Liu, Yukun Zhou, Tianyi Li, and Yang You. Open-sora: Democratizing efficient video production for all, March 2024. URL https://github.com/hpcaitech/Open-Sora.

Guangyao Zhou, Sivaramakrishnan Swaminathan, Rajkumar Vasudeva Raju, J. Swaroop Guntupalli, Wolfgang Lehrach, Joseph Ortiz, Antoine Dedieu, Miguel Lázaro-Gredilla, and Kevin Murphy. Diffusion model predictive control, 2024. URL https://arxiv.org/abs/2410.05364.

A Dataset Construction

A.1 Data Collection

To construct a high-quality dataset for training and evaluation, we collected expert-level trajectories using pre-trained Proximal Policy Optimization (PPO) agents in three Atari environments: Breakout, Pong, and Boxing. For each environment, we collected 1 million steps, resulting in a dataset comprising 3 million steps of state-action pairs. Learning from expert trajectories allows the model to internalize high-quality decision-making patterns and capture essential spatiotemporal relationships in the environment.

A.2 Sliding Window for Sequence Generation

To prepare sequential data for training, we applied a sliding window approach that segments raw state observations into structured video clips. Specifically, we used a sliding window of 32 frames with a stride of 4, ensuring that each resulting video sequence captures temporal dynamics over time while maintaining continuity between overlapping sequences. This overlap enhances data diversity and helps the model learn smooth transitions between frames. By structuring the data in this way, we provide sufficient temporal context for spatiotemporal models while keeping computational costs manageable.

B Model Structure

The Spatio-temporal Variational Autoencoder (VAE) Kingma & Welling (2022) serves as the initial processing stage, encoding observations $o_{o:n}$ into a compressed latent representation. This latent space is subsequently used by the diffusion model for generation. The VAE captures both spatial and temporal dependencies while ensuring computational efficiency.

The spatial and temporal encoders employ a two-stage compression strategy, where each video frame is first spatially downsampled by a factor of four, followed by temporal downsampling with the same factor. This hierarchical approach, implemented using causal 3D convolutional layers, ensures temporal causality while encoding compact, information-rich representations. The latent space is regularized via a Gaussian prior to enhance smoothness and sampling efficiency, and the decoder reconstructs video frames through upsampling layers, preserving both local and global structures for robust video generation.

Spatio-temporal Diffusion Transformer The Spatiotemporal Diffusion Transformer (ST-DiT) integrates transformer-based architectures with diffusion modeling Peebles & Xie (2023) to capture complex spatio-temporal dependencies in video data. ST-DiT comprises a hierarchical structure with dedicated spatial and temporal processing blocks, enabling effective modeling for generation.

The input video frames are first partitioned into spatio-temporal patches, which are then projected into a high-dimensional latent space using a PatchEmbed3D module. A PositionEmbedding2D layer further enhances these embeddings by encoding spatial and temporal positional information, ensuring the model retains sequence structure.

ST-DiT consists of alternating spatial and temporal transformer Vaswani et al. (2023) blocks. Spatial blocks use self-attention mechanisms Vaswani et al. (2023) to model intra-frame dependencies, while temporal blocks focus on inter-frame correlations. Each block incorporates rotary embeddings (RoPE) Su et al. (2023) for improved position encoding and flash attention for computational efficiency. Modulation parameters, generated by timestep and size embeddings, dynamically adjust attention and MLP operations, enhancing adaptability for sequential data tasks.

ST-DiT concludes with a diffusion-based generation layer that maps latent representations back to the video space. An unpatchification step reconstructs the original spatial and temporal dimensions, yielding temporally coherent video outputs.

Action Encoder for Cross-Attention Conditioning To integrate action sequences as conditioning signals for video generation, we introduce an Action Encoder $\mathcal{E}_{\mathcal{A}}$ that maps discrete action indices into a continuous latent representation. This representation is compatible with the cross-attention mechanism within ST-DiT, ensuring that generated frames align with provided action sequences.

The Action Encoder consists of an embedding layer that transforms discrete action indices into trainable dense vectors. These embeddings are processed through two fully connected layers with ReLU activations, progressively mapping them into a higher-dimensional latent space. The resulting representation is used as the key and value inputs in ST-DiT's cross-attention layers, where spatial-temporal embeddings serve as queries. This setup allows the model to dynamically attend to relevant actions during generation.

Conditioning with Video Frames and Noise Injection To balance temporal context from observed frames with stochastic variation, our framework employs a masking mechanism that conditions on video frames while injecting controlled noise. This approach ensures temporally consistent yet diverse video generation Song et al. (2022).

A binary mask determines which frames are retained from the input sequence, while the remaining frames are replaced with Gaussian noise. The masked input is passed to the diffusion model along with timestep and additional conditioning signals, such as encoded action sequences. This conditioning strategy allows the model to generate coherent and temporally consistent video sequences while incorporating stochastic variation, making it well-suited for action-conditioned video generation.

C ADDITIONAL EXPERIMENTS RESULTS

Figure 4 showcases a set of examples from the Breakout environment, allowing us to assess whether the world model understands the conditions and generates reasonable outputs. Each row in the figure 4 starts with 4 conditioning frames from left to right, with the last frame representing the generated result of our world model. The four different actions provided, listed from top to bottom, are **Noop, Fire, Right** and **Left**. Furthermore, to evaluate the world model's ability to understand and simulate inertia, we present examples where the final conditioning frame involves the "Right" action. The subsequent generated frames reflect inertia-driven behavior under varying actions. When no action or "Fire" is taken, the paddle continues moving right due to the previous inertia. Repeated "Right" actions increase the movement, while a "Left" action counteracts the motion, bringing the paddle to a stop. The results are shown in Figure 5.

To compare the performance of different designs of action encoders, we demonstrate also both quantitative and qualitative experimental results, using future action sequence conditions $a_{n,n+T}$. The results summarized in Table 4 demonstrate that utilizing the Embedding Layer for action encoding leads to superior performance across all metrics and generation lengths, except for the LPIPS score in the 2-frame generation scenario. The MSE remains consistently lower, while the PSNR and LPIPS values suggest improved visual and perceptual quality of the generated videos. Qualitative analysis reveals that the One-hot Vector encoding occasionally introduces noticeable artifacts in the generated frames, as shown in Figure 6. Meanwhile, in the 2-frame generation scenario, the frames generated using one-hot vector encoding exhibit an artifact where two paddles appear, which is perceptually incorrect. This observation also opens up a discussion on what could serve as a more precise evaluation metric for video generation.

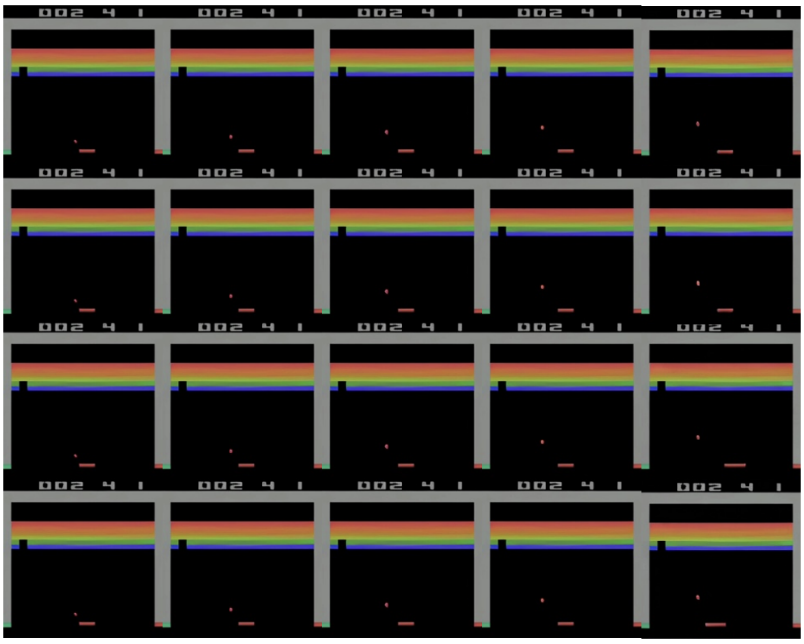

Figure 4: frames for consistency analysis

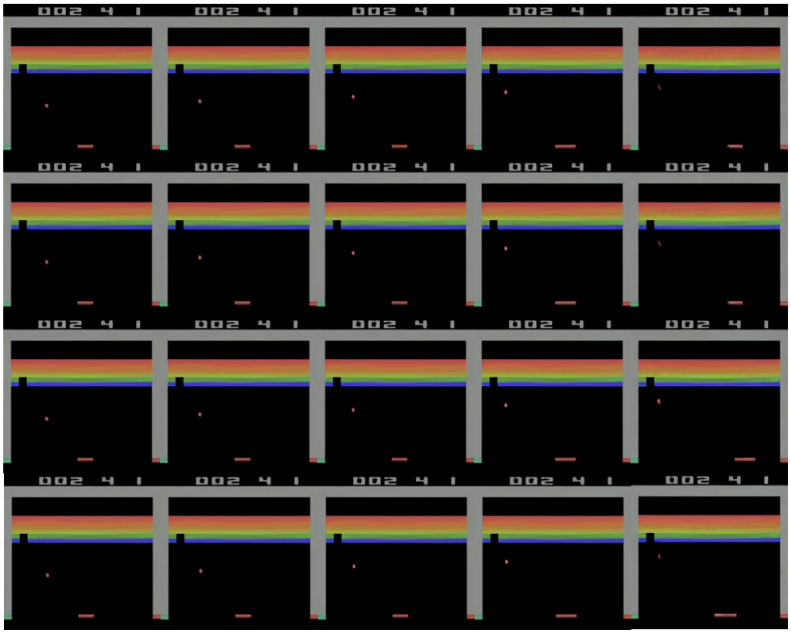

Figure 5: frames for inertia analysis

Table 4: Performance comparison between Embedding Layer and One-hot Vector representations for different generation lengths.

Gen. Length	Metrics	Embedding Layer	One-hot Vector
1 Frame	MSE ↓	**13.1859**	13.2162
	PSNR ↑	**36.9485**	36.7846
	LPIPS ↓	**0.0132**	0.0141
2 Frames	MSE ↓	**13.3697**	13.8342
	PSNR ↑	**36.8881**	36.7308
	LPIPS ↓	0.0356	**0.0190**
4 Frames	MSE ↓	**14.3638**	14.9298
	PSNR ↑	**36.5727**	36.2293
	LPIPS ↓	**0.0662**	0.0698

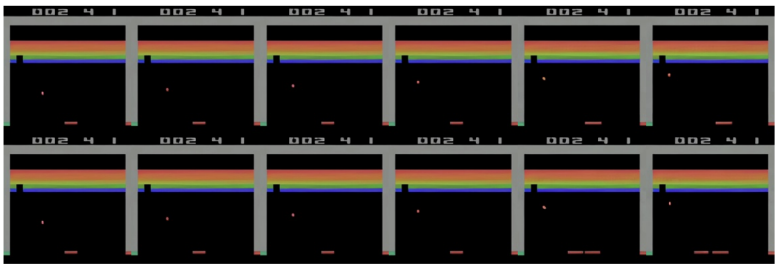

Figure 6: Comparison of generated video frames using Embedding Layer and One-hot Vector for action encoding in Breakout. The top row shows video frames generated using the Embedding Layer, while the bottom row shows frames generated using the One-hot Vector. In both cases, the first four frames are condition frames, and the last two frames are generated frames.