Flow Equivariant World Models: Structured Dynamics Outside the Field of View

Hansen Jin Lillemark^{*1,2}
hlillemark@ucsd.edu

Benhao Huang^{*3}
benhaoh@andrew.cmu.edu

Fangneng Zhan^{4}
fnzhan@seas.harvard.edu

Yilun Du^{1}
ydu@seas.harvard.edu

T. Anderson Keller^{1}
t.anderson.keller@gmail.com

[1]Kempner Institute, Harvard University [2]CSE, UC San Diego
[3]ML, Carnegie Mellon University [4]SEAS, Harvard University

Abstract

Embodied systems experience the world as 'a symphony of flows': a combination of many continuous streams of sensory input entrained to self-motion and intertwined with the motion of external objects. These streams obey smooth, time-parameterized symmetries (e.g. translating or expanding optic flow), yet most neural network sequence models ignore this structure, and instead laboriously re-learn the same transformations from data. In this work, we introduce 'Flow Equivariant World Models', a framework in which both self-motion and the motion of external objects are unified as one-parameter Lie group 'flows' thereby enabling group equivariance with respect to these ubiquitous transformations. On a 2D partially observed world modeling benchmark, Flow Equivariant World Models learn with an order of magnitude fewer training iterations and consequently outperform a comparable state-of-the-art diffusion-based world-modeling architecture – particularly when there are predictable world dynamics outside the agent's current field of view. The flow equivariant update rule also remains stable over hundreds of future rolled-out timesteps, generating a latent map robust to internal and external motion. Project page: Link.

1 Introduction

The sensory experience of embodied agents can be understood as a composition of transformations originating from both internal and external sources. As an agent navigates in the world, its experience transforms in a highly structured manner with respect to its own actions: as it turns to the right, its visual input inversely flows left. Simultaneously, the natural dynamics of external objects combine with this self-motion to yield a complex entangled flow of stimuli. This flow is challenging to model accurately even when fully observed; when combined with the inherently limited field of view of embodied agents, the task becomes nearly insurmountable, even for today's state-of-the-art world models.

In this work, we study this task of *partially observed dynamical world modeling* (visualized in Figure 1), combined with the inherent self-motion of embodied agents, and investigate if we might be able to account for both external and internal sources of visual variation in a geometrically structured manner.

[*] Equal contribution

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: SPACE in Vision, Language, and Embodied AI.

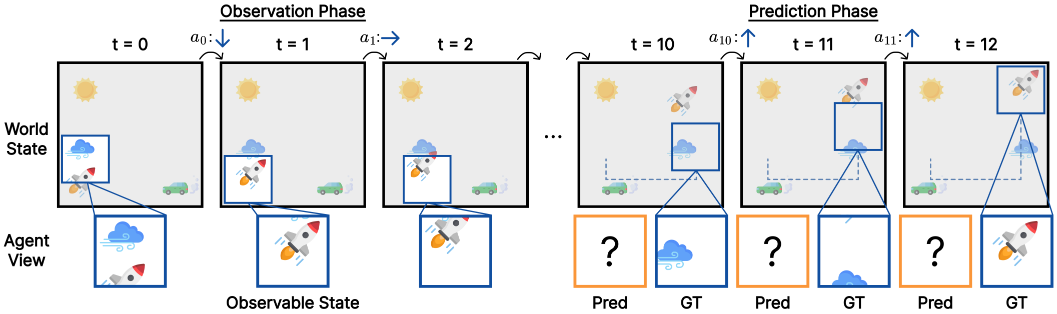

Figure 1: Partially observable dynamic world modeling. Grayed out areas are not visible to the agent at time t. The agent moves its view each timestep via action a_t.

Specifically, we find that both internal and external motion can be understood as mathematical 'flows', enabling both sources of variation to be handled exactly as time-parameterized symmetries through the framework of 'flow equivariance' [Keller, 2025]. We demonstrate that we can construct flow equivariant world models that handle self-generated motion in a precisely structured manner, while simultaneously capturing the motion of external objects, even outside the agent's field of view. We show that this yields substantially improved world modeling performance and generalization to significantly longer sequences than those seen during training, highlighting the benefits of precise spatial and dynamical structure in world models.

2 Flow Equivariant World Models

A neural network ϕ is said to be equivariant if its output, $\phi(f)$, changes in a structured, predictable manner when the input f is transformed by an element g of the group G, i.e. $\phi(g \cdot f) = g \cdot \phi(f)$. This constraint, often imposed analytically through weight-tying [Cohen and Welling, 2016, Ravanbakhsh et al., 2017], induces a type of representational structure which enables both improved data efficiency and generalization [Worrall et al., 2017, Batzner et al., 2022]. Recently, Keller [2025] introduced the concept of flow equivariance, extending existing 'static' group equivariance to time-parameterized sequence transformations ('flows'), such as visual motion. These flows are generated by vector fields ν (elements of a Lie algebra $\mathfrak{g}$ of G), and written as $\psi_t(\nu) \in G$. A sequence-to-sequence model Φ, mapping from $(f_0, \ldots, f_T) \mapsto (y_0, \ldots, y_T)$ is then said to be flow equivariant if, when the input sequence undergoes a flow, the output sequence transforms according to a proper flow representation, i.e. $\Phi\big(\psi_0(\nu)\cdot f_0,\ \ldots,\ \psi_T(\nu)\cdot f_T\big) \ = \ \big(\psi_0(\nu)\cdot y_0,\ \ldots,\ \psi_T(\nu)\cdot y_T\big)$, To achieve this, Keller [2025] demonstrated that it is sufficient to perform computation in the co-moving reference frame of the input. In other words, for a simple Recurrent Neural Network (RNN), the hidden state must flow in unison with the input, i.e. $h_{t+\Delta t} = \sigma\left(\psi_{\Delta t}(\nu) \cdot h_t + f_t\right)$. To achieve equivariance with respect to a set of multiple flows ($\nu \in V$), Flow Equivariant RNNs possess multiple hidden state 'velocity channels', each flowing according to their own vector fields ν (denoted as $h_t(\nu)$), illustrated as stacked rows in Fig. 2 a).

Self-Motion Equivariance In this work, we leverage the fact that motion is relative (i.e. self-motion is equivalent to the motion of the input) to additionally achieve equivariance to self-motion in a unified manner – with the core difference being that self-motion is accompanied by a known action (a_t) between the intervening observations. This additional information (knowledge of a_t) allows us to build a world model which operates in the co-moving reference frame of the agent, thereby achieving self-motion equivariance, without any additional 'velocity channels' – we call this model **FloWM**.

Specifically, given the action a_t, we transform the hidden state of the network to flow according to the corresponding induced visual flow. In this work, we assert the action is an element of the Lie algebra, i.e. $a_t \in \mathfrak{g}$ (otherwise a mapping can be performed). The visual flow induced by the action in the agent's reference frame, the 'Action Flow' ($\psi_1(-a_t)$), then combines with the 'Internal Flows' ($\psi_1(\nu)$) of the 'velocity channels', resulting in the following *Self-Motion Flow Equivariant* Recurrence Relation:

$$h_{t+1}(\nu) = \sigma\big(\psi_1(\nu - a_t) \cdot \mathcal{W} \star h_t(\nu) + \mathrm{pad}(\mathcal{U} \star f_t)\big), \tag{1}$$

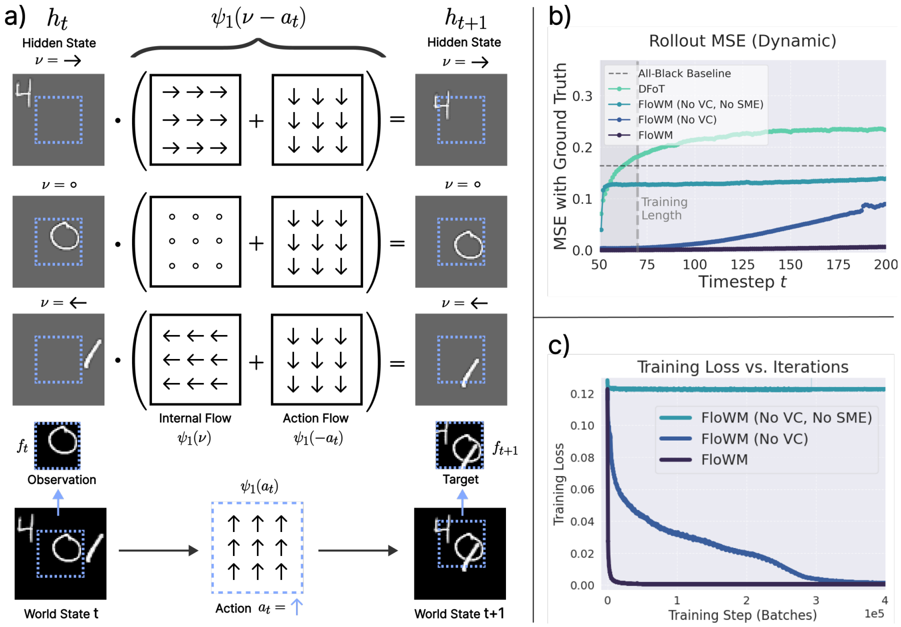

Figure 2: **a)** FloWM Recurrence relation. Velocity channels are plotted as rows, with the 'read-in' and 'read-out' portion of the hidden state highlighted in blue. **b)** Rollout MSE shows length generalization of FloWM. **c)** Training curves show efficient learning of FloWM.

where $\mathcal{W} \star h_t$, and $\mathcal{U} \star f_t$ denote convolutions over the hidden state and input spatial dimensions (or group dimensions for groups beyond translation). We see that this unified framework effectively 'factors out' both self-motion and the motion of external objects from agents' observations in a group-theoretic manner, thereby supporting a broader set of actions than prior work [Parisotto and Salakhutdinov, 2017]. To model partial observability, we simply write-in-to (denoted '$\text{pad}(\cdot)$'), and read-out-from, a fixed $\text{window\_size} < \text{world\_size}$ portion of the hidden state (blue dashed square in Fig. 2(a)), letting the rest of the hidden state flow around the agent's field of view according to $\psi_1(\nu - a_t)$. In particular, the hidden state is windowed at each timestep, pixel-wise max-pooled over 'velocity channels' and passed through a decoder g_θ to predict the next observation, explicitly: $\hat{f}_{t+1} = g_\theta\big(\max_\nu \text{window}(h_{t+1})\big)$

We provide more model details in Appendix D, and in A we review related models that have similarly structured representations with respect to self-motion, but may be seen as special cases of this framework without input-flow equivariance.

3 Experiments

To test our architecture on partially observable dynamic world modeling, we propose a simple MNIST World dataset. The world is a 2D black canvas with multiple MNIST digits moving with random constant velocities. The agent is provided a view of the world, smaller than the world size, yielding partial observability. At each discrete timestep, the world evolves according to the velocity of each object, and the agent takes a random action (relative (x, y) offset) to move its viewpoint. Given 50 observation frames, the task is to predict the dynamics played out for 20 future frames, integrating future self-motion (given) and world dynamics. To test length generalization, each validation video has 50 observation frames and 150 prediction frames. We include ablations on data subsets with different combinations of partial observability, object dynamics, and self-motion in Appendix B.

On the MNIST World dataset, we train and evaluate our proposed Flow Equivariant World Model (FloWM), which includes velocity channels (VC) and self-motion equivariance (SME). We also include ablations FloWM (no VC), FloWM (no VC, no SME), and a state of the art diffusion forcing transformer video world model, termed here DFoT. We note here that FloWM (no VC, no SME) is just a simple convolutional RNN. More training and model details are available in Appendices D and

Table 1: MSE for rollouts of 20 and 150 frames. We see the FloWM without velocity channels is able to model the sequences well for the training length (20), but generalizes far beyond to 150 frames with full flow equivariance.

Model	20	150
All-Black Baseline	0.16410	0.16410
FloWM (Ours)	**0.00058**	**0.00294**
(no VC)	**0.00410**	0.03380
(no VC, no SME)	0.12242	0.13072
DFoT	0.14874	0.21281

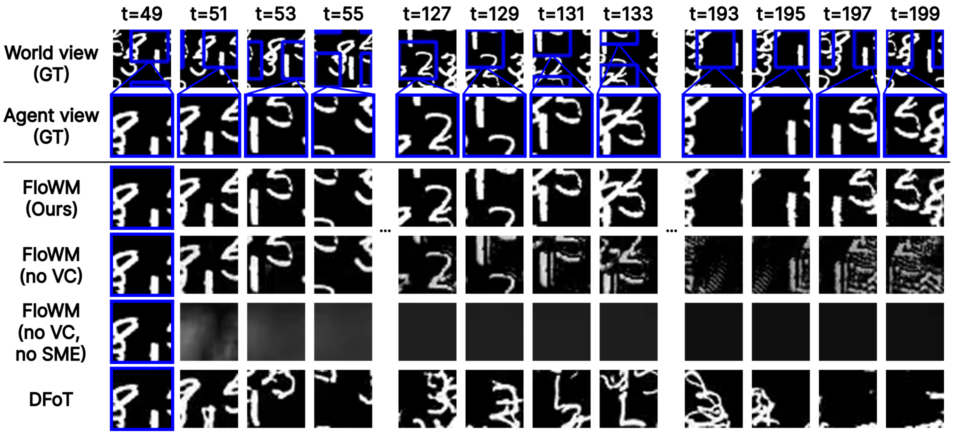

Figure 3: Prediction rollouts over time for FloWM and baselines. Timesteps 0 to 49 are given as observations. Note that FloWM does not diverge even at timestep 199.

E. At each timestep, we calculate the MSE of the predicted frame for each model, reported in Figure 2(b) and Table 1. Example rollouts and full world view visualizations are available in Figure 3.

Predictions from the FloWM remain consistent with the motion of objects out of its view for 150 timesteps past the observation window, well beyond its training horizon of 20 prediction timesteps, while FloWM with (no VC, no SME) fails. We find that the FloWM with (no VC) can still somewhat learn to model unobserved dynamics, especially within its training window, but drifts over time. We find models combining SME and VC require orders of magnitude less training steps to converge, shown in Figure 2(c). The DFoT model's predictions quickly diverge from the ground truth, even within its training window, generating plausible digit-like artifacts. Through additional results in Appendix C, we explore how the DFoT model can sometimes handle partial observability, object dynamics, and self-motion individually, but not in any combination.

4 Conclusion

In this work, we have introduced Flow Equivariant World Models, a new framework unifying both internally and externally generated motion for more accurate and efficient world modeling in partially observable settings with dynamic objects out of the agent's view. Our results on a simple dataset with these properties demonstrate the limitations with current state-of-the-art diffusion-based video world models and highlight the importance of unified equivariance with respect to motion for handling such settings.

Acknowledgements This work has been made possible in part by a gift from the Chan Zuckerberg Initiative Foundation to establish the Kempner Institute for the Study of Natural and Artificial Intelligence at Harvard University.

References

Niket Agarwal, Arslan Ali, Maciej Bala, Yogesh Balaji, Erik Barker, Tiffany Cai, Prithvijit Chattopadhyay, Yongxin Chen, Yin Cui, Yifan Ding, et al. Cosmos world foundation model platform for physical ai. *arXiv preprint arXiv:2501.03575*, 2025.

Muhammad Qasim Ali, Aditya Sridhar, Shahbuland Matiana, Alex Wong, and Mohammad Al-Sharman. Humanoid world models: Open world foundation models for humanoid robotics. *arXiv preprint arXiv:2506.01182*, 2025.

Philip J. Ball, Jakob Bauer, Frank Belletti, Bethanie Brownfield, Ariel Ephrat, Shlomi Fruchter, Agrim Gupta, Kristian Holsheimer, Aleksander Holynski, Jiri Hron, Christos Kaplanis, Marjorie Limont, Matt McGill, Yanko Oliveira, Jack Parker-Holder, Frank Perbet, Guy Scully, Jeremy Shar, Stephen Spencer, Omer Tov, Ruben Villegas, Emma Wang, Jessica Yung, Cip Baetu, Jordi Berbel, David Bridson, Jake Bruce, Gavin Buttimore, Sarah Chakera, Bilva Chandra, Paul Collins, Alex Cullum, Bogdan Damoc, Vibha Dasagi, Maxime Gazeau, Charles Gbadamosi, Woohyun Han, Ed Hirst, Ashyana Kachra, Lucie Kerley, Kristian Kjems, Eva Knoepfel, Vika Koriakin, Jessica Lo, Cong Lu, Zeb Mehring, Alex Moufarek, Henna Nandwani, Valeria Oliveira, Fabio Pardo, Jane Park, Andrew Pierson, Ben Poole, Helen Ran, Tim Salimans, Manuel Sanchez, Igor Saprykin, Amy Shen, Sailesh Sidhwani, Duncan Smith, Joe Stanton, Hamish Tomlinson, Dimple Vijaykumar, Luyu Wang, Piers Wingfield, Nat Wong, Keyang Xu, Christopher Yew, Nick Young, Vadim Zubov, Douglas Eck, Dumitru Erhan, Koray Kavukcuoglu, Demis Hassabis, Zoubin Gharamani, Raia Hadsell, Aäron van den Oord, Inbar Mosseri, Adrian Bolton, Satinder Singh, and Tim Rocktäschel. Genie 3: A new frontier for world models. 2025.

Amir Bar, Gaoyue Zhou, Danny Tran, Trevor Darrell, and Yann LeCun. Navigation world models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 15791–15801, 2025.

Simon Batzner, Albert Musaelian, Lixin Sun, Mario Geiger, Jonathan P. Mailoa, Mordechai Kornbluth, Nicola Molinari, Tess E. Smidt, and Boris Kozinsky. E(3)-equivariant graph neural networks for data-efficient and accurate interatomic potentials. *Nature Communications*, 13(1):2453, 2022. doi: 10.1038/s41467-022-29939-5. URL `https://doi.org/10.1038/s41467-022-29939-5`.

Charles Beattie, Joel Z. Leibo, Denis Teplyashin, Tom Ward, Marcus Wainwright, Heinrich Küttler, Andrew Lefrancq, Simon Green, Víctor Valdés, Amir Sadik, Julian Schrittwieser, Keith Anderson, Sarah York, Max Cant, Adam Cain, Adrian Bolton, Stephen Gaffney, Helen King, Demis Hassabis, Shane Legg, and Stig Petersen. Deepmind lab, 2016. URL `https://arxiv.org/abs/1612.03801`.

Edward Beeching, Christian Wolf, Jilles Dibangoye, and Olivier Simonin. Egomap: Projective mapping and structured egocentric memory for deep rl, 2020. URL `https://arxiv.org/abs/2002.02286`.

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`.

Jake Bruce, Michael Dennis, Ashley Edwards, Jack Parker-Holder, Yuge Shi, Edward Hughes, Matthew Lai, Aditi Mavalankar, Richie Steigerwald, Chris Apps, Yusuf Aytar, Sarah Bechtle, Feryal Behbahani, Stephanie Chan, Nicolas Heess, Lucy Gonzalez, Simon Osindero, Sherjil Ozair, Scott Reed, Jingwei Zhang, Konrad Zolna, Jeff Clune, Nando de Freitas, Satinder Singh, and Tim Rocktäschel. Genie: Generative interactive environments, 2024. URL `https://arxiv.org/abs/2402.15391`.

Stanley H. Chan. Tutorial on diffusion models for imaging and vision, 2025. URL `https://arxiv.org/abs/2403.18103`.

Boyuan Chen, Diego Marti Monso, Yilun Du, Max Simchowitz, Russ Tedrake, and Vincent Sitzmann. Diffusion forcing: Next-token prediction meets full-sequence diffusion, 2024. URL `https://arxiv.org/abs/2407.01392`.

Delong Chen, Willy Chung, Yejin Bang, Ziwei Ji, and Pascale Fung. Worldprediction: A benchmark for high-level world modeling and long-horizon procedural planning, 2025. URL https://arxiv.org/abs/2506.04363.

Maxime Chevalier-Boisvert, Bolun Dai, Mark Towers, Rodrigo de Lazcano, Lucas Willems, Salem Lahlou, Suman Pal, Pablo Samuel Castro, and Jordan Terry. Minigrid & miniworld: Modular & customizable reinforcement learning environments for goal-oriented tasks, 2023. URL https://arxiv.org/abs/2306.13831.

Nathan Cloos, Meagan Jens, Michelangelo Naim, Yen-Ling Kuo, Ignacio Cases, Andrei Barbu, and Christopher J. Cueva. Baba is ai: Break the rules to beat the benchmark, 2024. URL https://arxiv.org/abs/2407.13729.

Taco Cohen and Max Welling. Group equivariant convolutional networks. In Maria Florina Balcan and Kilian Q. Weinberger, editors, *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 2990–2999, New York, New York, USA, 20–22 Jun 2016. PMLR. URL https://proceedings.mlr.press/v48/cohenc16.html.

Abhishek Das, Samyak Datta, Georgia Gkioxari, Stefan Lee, Devi Parikh, and Dhruv Batra. Embodied question answering. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1–10, 2018. doi: 10.1109/CVPR.2018.00008.

Etched Decart, Quinn McIntyre, Spruce Campbell, Xinlei Chen, and Robert Wachen. Oasis: A universe in a transformer. *URL: https://oasis-model. github. io*, 2024.

Dynamics Lab. Research preview: The world's first ai-native ugc game engine powered by real-time world model. https://blog.dynamicslab.ai/, August 2025. Accessed: 2025-08-29.

Qiyue Gao, Xinyu Pi, Kevin Liu, Junrong Chen, Ruolan Yang, Xinqi Huang, Xinyu Fang, Lu Sun, Gautham Kishore, Bo Ai, Stone Tao, Mengyang Liu, Jiaxi Yang, Chao-Jung Lai, Chuanyang Jin, Jiannan Xiang, Benhao Huang, Zeming Chen, David Danks, Hao Su, Tianmin Shu, Ziqiao Ma, Lianhui Qin, and Zhiting Hu. Do vision-language models have internal world models? towards an atomic evaluation, 2025. URL https://arxiv.org/abs/2506.21876.

Hafez Ghaemi, Eilif Muller, and Shahab Bakhtiari. seq-jepa: Autoregressive predictive learning of invariant-equivariant world models, 2025. URL https://arxiv.org/abs/2505.03176.

Rohit Girdhar and Deva Ramanan. Cater: A diagnostic dataset for compositional actions and temporal reasoning, 2020. URL https://arxiv.org/abs/1910.04744.

Yuchao Gu, Weijia Mao, and Mike Zheng Shou. Long-context autoregressive video modeling with next-frame prediction. *arXiv preprint arXiv:2503.19325*, 2025.

Junliang Guo, Yang Ye, Tianyu He, Haoyu Wu, Yushu Jiang, Tim Pearce, and Jiang Bian. Mineworld: a real-time and open-source interactive world model on minecraft, 2025. URL https://arxiv.org/abs/2504.08388.

Agrim Gupta, Lijun Yu, Kihyuk Sohn, Xiuye Gu, Meera Hahn, Li Fei-Fei, Irfan Essa, Lu Jiang, and José Lezama. Photorealistic video generation with diffusion models, 2023. URL https://arxiv.org/abs/2312.06662.

David Ha and Jürgen Schmidhuber. World models. *arXiv preprint arXiv:1803.10122*, 2(3), 2018.

Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse domains through world models, 2024. URL https://arxiv.org/abs/2301.04104.

Nicklas Hansen, Hao Su, and Xiaolong Wang. Td-mpc2: Scalable, robust world models for continuous control, 2024. URL https://arxiv.org/abs/2310.16828.

Shibo Hao, Yi Gu, Haodi Ma, Joshua Jiahua Hong, Zhen Wang, Daisy Zhe Wang, and Zhiting Hu. Reasoning with language model is planning with world model, 2023. URL https://arxiv.org/abs/2305.14992.

Matthew Hausknecht and Peter Stone. Deep recurrent q-learning for partially observable mdps, 2017. URL https://arxiv.org/abs/1507.06527.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners, 2021. URL https://arxiv.org/abs/2111.06377.

Xianglong He, Chunli Peng, Zexiang Liu, Boyang Wang, Yifan Zhang, Qi Cui, Fei Kang, Biao Jiang, Mengyin An, Yangyang Ren, et al. Matrix-game 2.0: An open-source, real-time, and streaming interactive world model. *arXiv preprint arXiv:2508.13009*, 2025.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance, 2022. URL https://arxiv.org/abs/2207.12598.

Emiel Hoogeboom, Thomas Mensink, Jonathan Heek, Kay Lamerigts, Ruiqi Gao, and Tim Salimans. Simpler diffusion (sid2): 1.5 fid on imagenet512 with pixel-space diffusion. *arXiv preprint arXiv:2410.19324*, 2024.

Fan Jia, Weixin Mao, Yingfei Liu, Yucheng Zhao, Yuqing Wen, Chi Zhang, Xiangyu Zhang, and Tiancai Wang. Adriver-i: A general world model for autonomous driving. *arXiv preprint arXiv:2311.13549*, 2023.

T. Anderson Keller. Flow equivariant recurrent neural networks, 2025. URL https://arxiv.org/abs/2507.14793.

Michał Kempka, Marek Wydmuch, Grzegorz Runc, Jakub Toczek, and Wojciech Jaśkowski. Vizdoom: A doom-based ai research platform for visual reinforcement learning, 2016. URL https://arxiv.org/abs/1605.02097.

Bosung Kim and Prithviraj Ammanabrolu. Beyond needle(s) in the embodied haystack: Environment, architecture, and training considerations for long context reasoning, 2025. URL https://arxiv.org/abs/2505.16928.

Diederik P. Kingma and Ruiqi Gao. Understanding diffusion objectives as the elbo with simple data augmentation, 2023. URL https://arxiv.org/abs/2303.00848.

Andrey Kurenkov, Michael Lingelbach, Tanmay Agarwal, Emily Jin, Chengshu Li, Ruohan Zhang, Li Fei-Fei, Jiajun Wu, Silvio Savarese, and Roberto Martín-Martín. Modeling dynamic environments with scene graph memory, 2023. URL https://arxiv.org/abs/2305.17537.

Emilio Parisotto and Ruslan Salakhutdinov. Neural map: Structured memory for deep reinforcement learning, 2017. URL https://arxiv.org/abs/1702.08360.

Jung Yeon Park, Ondrej Biza, Linfeng Zhao, Jan Willem van de Meent, and Robin Walters. Learning symmetric embeddings for equivariant world models, 2022. URL https://arxiv.org/abs/2204.11371.

Jurgis Pasukonis, Timothy Lillicrap, and Danijar Hafner. Evaluating long-term memory in 3d mazes, 2022. URL https://arxiv.org/abs/2210.13383.

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

Ryan Po, Yotam Nitzan, Richard Zhang, Berlin Chen, Tri Dao, Eli Shechtman, Gordon Wetzstein, and Xun Huang. Long-context state-space video world models, 2025.

Siamak Ravanbakhsh, Jeff Schneider, and Barnabás Póczos. Equivariance through parameter-sharing. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 2892–2901. PMLR, 06–11 Aug 2017. URL https://proceedings.mlr.press/v70/ravanbakhsh17a.html.

Ronan Riochet, Mario Ynocente Castro, Mathieu Bernard, Adam Lerer, Rob Fergus, Véronique Izard, and Emmanuel Dupoux. Intphys: A framework and benchmark for visual intuitive physics reasoning, 2020. URL https://arxiv.org/abs/1803.07616.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2022. URL https://arxiv.org/abs/2112.10752.

Lloyd Russell, Anthony Hu, Lorenzo Bertoni, George Fedoseev, Jamie Shotton, Elahe Arani, and Gianluca Corrado. Gaia-2: A controllable multi-view generative world model for autonomous driving. *arXiv preprint arXiv:2503.20523*, 2025.

Mohammad Reza Samsami, Artem Zholus, Janarthanan Rajendran, and Sarath Chandar. Mastering memory tasks with world models, 2024. URL https://arxiv.org/abs/2403.04253.

Nedko Savov, Naser Kazemi, Deheng Zhang, Danda Pani Paudel, Xi Wang, and Luc Van Gool. Statespacediffuser: Bringing long context to diffusion world models. *arXiv preprint arXiv:2505.22246*, 2025.

Kiwhan Song, Boyuan Chen, Max Simchowitz, Yilun Du, Russ Tedrake, and Vincent Sitzmann. History-guided video diffusion, 2025. URL https://arxiv.org/abs/2502.06764.

Tadahiro Taniguchi, Shingo Murata, Masahiro Suzuki, Dimitri Ognibene, Pablo Lanillos, Emre Ugur, Lorenzo Jamone, Tomoaki Nakamura, Alejandra Ciria, Bruno Lara, et al. World models and predictive coding for cognitive and developmental robotics: frontiers and challenges. *Advanced Robotics*, 37(13):780–806, 2023.

Elise van der Pol, Daniel E. Worrall, Herke van Hoof, Frans A. Oliehoek, and Max Welling. Mdp homomorphic networks: Group symmetries in reinforcement learning, 2021. URL https://arxiv.org/abs/2006.16908.

Yuqi Wang, Jiawei He, Lue Fan, Hongxin Li, Yuntao Chen, and Zhaoxiang Zhang. Driving into the future: Multiview visual forecasting and planning with world model for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14749–14759, 2024.

Daniel E. Worrall, Stephan J. Garbin, Daniyar Turmukhambetov, and Gabriel J. Brostow. Harmonic networks: Deep translation and rotation equivariance, 2017. URL https://arxiv.org/abs/1612.04642.

Jiannan Xiang, Guangyi Liu, Yi Gu, Qiyue Gao, Yuting Ning, Yuheng Zha, Zeyu Feng, Tianhua Tao, Shibo Hao, Yemin Shi, Zhengzhong Liu, Eric P. Xing, and Zhiting Hu. Pandora: Towards general world model with natural language actions and video states, 2024. URL https://arxiv.org/abs/2406.09455.

Zeqi Xiao, Yushi Lan, Yifan Zhou, Wenqi Ouyang, Shuai Yang, Yanhong Zeng, and Xingang Pan. Worldmem: Long-term consistent world simulation with memory, 2025. URL https://arxiv.org/abs/2504.12369.

Mengjiao Yang, Yilun Du, Kamyar Ghasemipour, Jonathan Tompson, Dale Schuurmans, and Pieter Abbeel. Learning interactive real-world simulators. *arXiv preprint arXiv:2310.06114*, 1(2):6, 2023.

Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, Da Yin, Yuxuan Zhang, Weihan Wang, Yean Cheng, Bin Xu, Xiaotao Gu, Yuxiao Dong, and Jie Tang. Cogvideox: Text-to-video diffusion models with an expert transformer, 2025. URL https://arxiv.org/abs/2408.06072.

Hanrong Ye, Haotian Zhang, Erik Daxberger, Lin Chen, Zongyu Lin, Yanghao Li, Bowen Zhang, Haoxuan You, Dan Xu, Zhe Gan, Jiasen Lu, and Yinfei Yang. Mm-ego: Towards building egocentric multimodal llms for video qa, 2025. URL https://arxiv.org/abs/2410.07177.

Jiwen Yu, Jianhong Bai, Yiran Qin, Quande Liu, Xintao Wang, Pengfei Wan, Di Zhang, and Xihui Liu. Context as memory: Scene-consistent interactive long video generation with memory retrieval. *arXiv preprint arXiv:2506.03141*, 2025.

Qinhong Zhou, Sunli Chen, Yisong Wang, Haozhe Xu, Weihua Du, Hongxin Zhang, Yilun Du, Joshua B. Tenenbaum, and Chuang Gan. Hazard challenge: Embodied decision making in dynamically changing environments, 2024. URL `https://arxiv.org/abs/2401.12975`.

Siyuan Zhou, Yilun Du, Yuncong Yang, Lei Han, Peihao Chen, Dit-Yan Yeung, and Chuang Gan. Learning 3d persistent embodied world models. *arXiv preprint arXiv:2505.05495*, 2025.

A Related Work

Generative World Modeling with Memory Generative World Models [Ha and Schmidhuber, 2018, Brooks et al., 2024] aim to simulate and predict how environments evolve over time in frames. They have broad applications in reinforcement learning [Hafner et al., 2024, Samsami et al., 2024, Hansen et al., 2024], autonomous driving [Bar et al., 2025, Yang et al., 2023, Russell et al., 2025, Jia et al., 2023, Wang et al., 2024], robotics [Yang et al., 2023, Agarwal et al., 2025, Ali et al., 2025, Taniguchi et al., 2023], and planning [Hao et al., 2023, Cloos et al., 2024], where agents must anticipate future states in order to reason and act [Gao et al., 2025]. Recent works have moved toward building more generalizable and large-scale world simulators with interactability and persistence [Ball et al., 2025, Agarwal et al., 2025, Xiang et al., 2024], with game engines emerging as a promising area of interest reliant on these aspects [He et al., 2025, Ball et al., 2025, Bruce et al., 2024, Dynamics Lab, 2025, Guo et al., 2025]. However, a critical limitation remains: most generative world models lack a semblance of memory. Autoregressive rollouts with transformer models have limited context windows, and remaining consistent with information outside of the context window requires something beyond naive self attention. As a result, long rollouts often produce contradictions with earlier context or previously generated sequences, undermining temporal coherence. Although recent explorations have introduced memory mechanisms through explicit (with 3D priors) [Zhou et al., 2025, Xiao et al., 2025, Yu et al., 2025] or implicit (based on neural / learned components) [Po et al., 2025, Gu et al., 2025, Savov et al., 2025] memory mechanisms, the focus has been on maintaining physical consistency with a static world. Dynamics are included in some datasets, but are not the main focus; especially the dynamics of objects outside the current field of view has not yet been studied as the main focus with the context of image space world modeling, to our knowledge. Prediction of the world necessarily requires predicting the state evolution of occluded or unseen entities, especially in real-world settings. Addressing this gap is crucial for advancing toward faithful and embodied generative world models.

Partially Observable Environments and Tasks Partial observability is a fundamental challenge in embodied AI and reinforcement learning, and a variety of benchmarks and environments have been developed to study it. Existing tasks can be broadly categorized along three dimensions. The first concerns whether the underlying environment is dynamic or static. The second concerns how partial observability (PO) is introduced: (i) PO within a fixed view, typically through object occlusion or objects moving in and out of view; (ii) PO outside the current view, which requires changes in the ego perspective. The third dimension is the task objective: some benchmarks are designed for question answering under PO, while others target next-state prediction, i.e. world modeling. We posit that the combination of all three dimensions – dynamic environments, with partial observability outside the current view, targeting world modeling – is currently understudied and underdeveloped.

A first branch of existing work lies in the fixed-view setting, where PO arises from occlusion in 3D scenes. Vision benchmarks such as CATER [Girdhar and Ramanan, 2020] and IntPhys [Riochet et al., 2020] capture such occlusion scenarios, but remain focused on passive observation without involving an embodied agent or dynamics outside the current field of view.

A second branch tests out-of-view partial observability, frequently utilizing simulators to control the state of the world. Early work introducing partially observable markov decision processes in environments like partially observable pacman require a rough understanding of the state of the world that may include dynamics [Hausknecht and Stone, 2017], but the task is entangled with decision making on a limited domain. 3D platforms such as DMLab [Beattie et al., 2016], VizDoom [Kempka et al., 2016], and MiniWorld/MiniGrid [Chevalier-Boisvert et al., 2023], probe exploration or combat with compact observations. For example, Pasukonis et al. [2022] introduces a memory maze using DMLab to evaluate the static memory of RL agents. Among these simulators, DMLab is largely static, while VizDoom and MiniGrid can introduce dynamics out-of-view.

Newer benchmarks approach dynamic PO but for the most part, they are primarily focused on QA rather than explicit next-state forecasting. Dynamic House [Kurenkov et al., 2023] includes evolving scenes and asks for future relations (e.g., whether an object moved rooms when hidden), effectively casting dynamics as link prediction. Hazard Challenge [Zhou et al., 2024] evaluates decision-making under evolving disasters (e.g. fire, flood). WorldPredictions [Chen et al., 2025] studies world modeling in a partially observable semi-MDP, but emphasizes action selection and procedural planning. Similarly, first-person embodied QA datasets over long videos [Ye et al., 2025, Das et al., 2018, Kim and Ammanabrolu, 2025] combine partial observability with dynamics but are retrospective: they query past observations rather than test predictive reasoning about where previously dynamic objects might be now.

Although many relevant tasks and environments exist, to our knowledge, there are still no direct evaluations designed to test world models on their ability to understand, encode, and predict dynamics under partial observability. Nevertheless, these existing environments provide valuable building blocks for constructing benchmarks that assess whether generative world models can move beyond static memory to capture the evolving dynamics of objects out of view.

Equivariant World Models Equivariant models respect the symmetry of their data, ensuring that structured transformations in the input induce predictable changes in the model's internal state. This inductive bias has indeed been found to be valuable in prior work on world modeling. Specifically, although not explicitly framed as equivariant, one of the most related world modeling architectures to our proposed self-motion equivariance is the Neural Map [Parisotto and Salakhutdinov, 2017]. This work introduced a spatially organized 2D memory that stores observations at estimated agent coordinates. The storage location of these observations is shifted precisely according to the agent's actions, yielding an effectively equivariant 'allocentric' latent map. In Section 5 of the paper, the authors describe an egocentric version of their model which can in fact be seen as a special case of our FloWM, specifically equivalent to the ablation without velocity channels. The authors demonstrate that their allocentric map enables long-term recall and generalization in navigation tasks. In a similar vein, EgoMap [Beeching et al., 2020] built on this by leveraging inverse perspective transformations to map from observations in 3D environments to a top-down egocentric map. This work also explicitly transforms the latent map in an action-conditioned manner, although the transformation is learned with a Spatial Transformer Network, making it only approximately equivariant. Our work can be seen to formalize these early models in the framework of group theory, allowing us to extend the action space beyond just spatial translation to any Lie group and any world space. For example, our framework can theoretically support full 3-dimensional 'neural maps' without problem, following the framework of flow equivariance. Finally, there are a few other works that discuss equivariant world modeling, but are less precisely related to our own. Specifically, [van der Pol et al., 2021] was one of the first works to build equivariant policy and value networks for reinforcement learning, but not with respect to motion, instead with respect to the symmetries of the environment (such as static rotations or translations). More recent work [Park et al., 2022, Ghaemi et al., 2025] proposes to approach the goal of building equivariant world models in a more approximate manner by conditioning or encouraging equivariance through training losses, rather than our approach which builds it in explicitly. Overall, we find all of these approaches to be complementary to our own and are excited for their combined potential.

B Dataset Details

Here, we describe dataset generation and parameter settings for our ablations on self-motion, dynamics, and partial observability in the MNIST world setting. The subsets are succinctly described in Table 2, and the generation parameters in Table 3. A subset is described as partially observable if the world size is larger than the window size. We also scale the number of digits by the size of the world. Each dataset example has a video of shape `[num_frames, channels, height, width]`, where channels is 1; and an accompanying actions list of shape `[num_frames, 2]`, for the x and y translation of the agent view at each timestep. Each dataset subset contains 180,000 videos in the training set, and 8,000 videos in the validation set. The `dynamic_fo_no_sm` subset just has dynamics and is fully observable; the `dynamic_fo` subset has dynamics and is fully observable, but also has self-motion; the `static_po` subset is partially observable, and the agent has self-motion, but the digits do not move; and finally, the `dynamic_po` subset includes partial observability, agent movement, and dynamics. In the main text, we report all results on just the `dynamic_po` subset. For all subsets with dynamics, each digit is given an integer velocity for x and y in the digit velocity range (e.g., -2 to 2). For each dataset with self-motion, at each step during the observation and prediction phase, a random integer is chosen in x and y to be the agent's view translation, bounded by the self-motion range (e.g., -10 to 10). For each dataset, objects that move across the boundary reappear on the other side as a circular pad. Results on each of these data subsets for FloWM and baseline models are described in Appendix C.

Table 2: MNIST world data subsets demonstrating scaling difficulty in self-motion, dynamics, and partial observability.

Data Subset	Self-Motion	Dynamics	Partially Observable
`dynamic_fo_no_sm`	No	Yes	No
`dynamic_fo`	Yes	Yes	No
`static_po`	Yes	No	Yes
`dynamic_po`	Yes	Yes	Yes

Table 3: Generation parameters for dataset subsets.

Data Subset	World Size	Window Size	# Digits	Self-motion Range	Digit Velocity Range, x and y
`dynamic_fo_no_sm`	32	32	3	0	-2 to +2
`dynamic_fo`	32	32	3	10	-2 to +2
`static_po`	50	32	5	10	0
`dynamic_po`	50	32	5	10	-2 to +2

C Additional Results

Here we report additional results on the MNIST World subsets described in Appendix B. We evaluate the FloWM, DFoT, and FloWM ablations described in Appendix D.

The error (MSE) between the predicted future observations (rollout) and the ground truth is plotted for each baseline in Figure 4 as a function of forward prediction timestep (x-axis). The average MSE over the first 20 timesteps (the training length) and over the full 150 timesteps (length generalization) is summarized in Table 4. Due to being constructed with a different number of digits, MSE between the data subsets are not necessarily directly comparable. We provide the All-Black Baseline (model that only predicts **0** for future observations) as a form of normalization for comparison.

All models are able to do reasonably well on the simplest fully observable dataset with no self-motion; note here the DFoT is doing latent diffusion, so there is a small amount of MSE error from the decoding step, around 0.02, see Appendix E.4 for more details. This setup aligns with the typical setting of world modeling, where the information that the model needs is expected to be in the attention window. The other dataset splits do not follow this assumption, and the results align with expectations about the model's capabilities. The DFoT does relatively better on the static `static_po` compared to the `dynamic_po` dataset, due to not having to model dynamics, but the model's outputs still diverge from the ground truth quickly.

For a dataset where the velocity channels are redundant, i.e. `static_po`, the FloWM (no VC) does slightly better than FloWM. Further note that the FloWM (no VC) is able to have low error on most of the tasks, though with a much higher value than FloWM as errors accumulate due to not having the velocity channels to encode flow equivariantly. Taken together, the ablations suggest that self-motion equivariance is key to solving the problem, and that input flow equivariance via velocity channels helps with exactness and convergence time, with the tradeoff of a larger hidden state activation size.

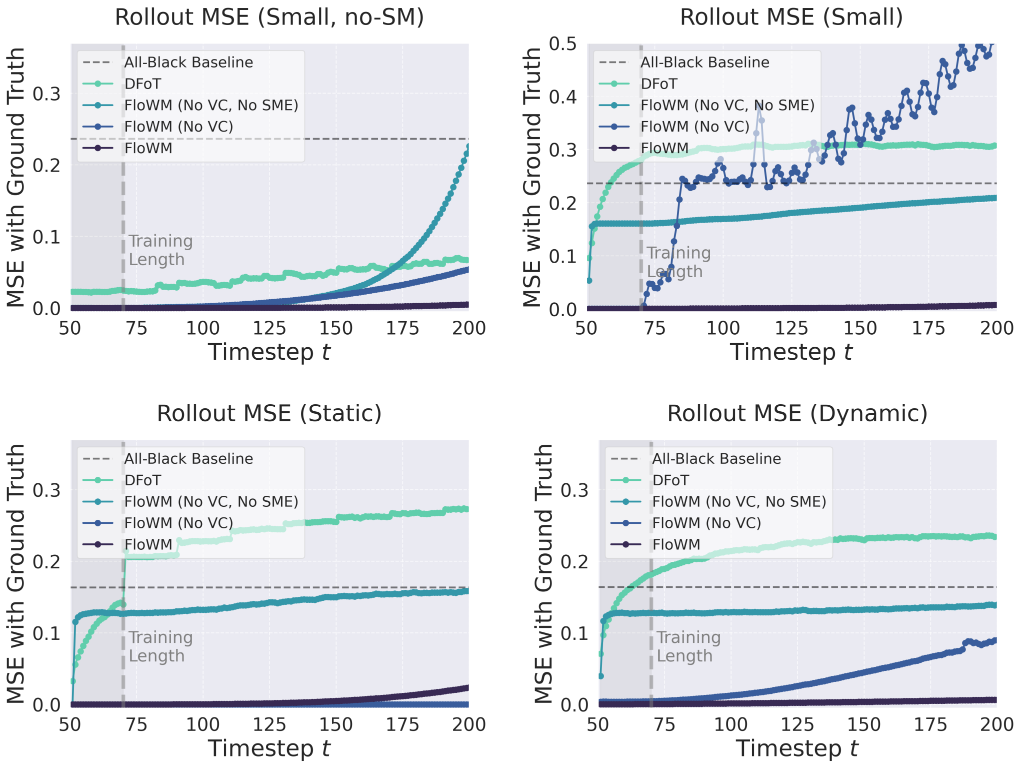

Figure 4: Rollout Error (MSE) vs. Forward Prediction Steps for all data subsets. The dynamic subset is replicated from the main text for ease of comparison.

Table 4: Validation MSE. Columns show mean MSE over the first 20 frames (matches training distribution) vs. 150 frames (length generalization). Models that 'solve' the task are in **bold** (MSE ≤ 0.05).

Model	`dynamic_fo_no_sm`		`dynamic_fo`		`static_po`		`dynamic_po`	
	20	150	20	150	20	150	20	150
All-Black Baseline	0.2372	0.2372	0.2363	0.2363	0.1636	0.1636	0.1641	0.1641
FloWM (Ours)	**0.00013**	**0.00117**	**0.00035**	**0.00249**	**0.00013**	**0.00509**	**0.00058**	**0.00294**
(no VC)	**0.00028**	**0.01460**	**0.00075**	0.26493	**0.00002**	**0.00004**	**0.00410**	**0.03380**
(no SME)	**0.00007**	**0.00179**	0.15534	0.16063	0.12035	0.12947	0.12250	0.12909
(no SME, no VC)	**0.00032**	**0.03198**	0.15537	0.18079	0.12037	0.14079	0.12242	0.13072
(action concat)	**0.00021**	**0.02070**	0.13402	0.15452	0.09449	0.11827	0.11161	0.13438
DFoT	**0.02321**	**0.04285**	0.22805	0.29419	0.10658	0.22723	0.14874	0.21281

D FloWM Experiment Details

In this work, we introduce the Flow Equivariant World Model (FloWM). The model is built as a simple sequence-to-sequence RNN with small CNN encoders/decoders to model MNIST digit features.

For completeness, we repeat the FloWM recurrence relation below:

$$h_{t+1}(\nu) = \sigma\big(\psi_1(\nu - a_t) \cdot \mathcal{W} \star h_t(\nu) + \mathrm{pad}(\mathcal{U} \star f_t)\big). \tag{2}$$

D.1 Recurrence

The hidden state $h_t \in \mathbb{R}^{|V| \times C_{hid} \times H_{world} \times W_{world}}$ has $|V|$ velocity channels (indexed by the elements $\nu \in V$), and $C_{hid} = 64$ hidden state channels. The spatial dimensions of the hidden state are set to match the world size for each dataset. For the partially observed world, this means $H_{world} = W_{world} = 50$ (where the window size is set to 32×32), while for the fully observed world, $H_{world} = W_{world} = 32$. The hidden state is initialized to all zeros for the first timestep, i.e. $h_0 = \mathbf{0}$.

The hidden state is processed between timesteps by a convolutional kernel $\mathcal{W}$. This kernel has the potential to span between velocity channels, and therefore model acceleration or more complex dynamics than static velocities. In this work, since our dataset has no such dynamics (we only have constant object velocities), we safely ignore the inter-velocity convolution terms, and simply set $\mathcal{W}$ to be a 3 × 3 convolutional kernel, with 64 input and output channels, circular padding, and no bias. We refer the interested reader to Keller [2025] for details on the form of the full flow-equivariant convolution that could be equally used in this model. The hidden state is finally passed through a non-linearity σ to complete the update to the next timestep. In this work, we use a ReLU.

D.2 Velocity Channels

In this work, we add velocity channels up to ±2 in both the X and Y dimensions of the image. Explicitly, $V = \{(-2,-2), (-2,-1), \ldots (0,0) \ldots (2,2)\}$. Thus in total, $|V| = 25$ for the FloWM. Each channel is flowed by its corresponding velocity field (defined by $\psi(\nu)$) at each step. This is denoted by $\psi_1(\nu) \cdot h_t(\nu)$.

The actions of the agent then induce an additional flow of the visual stimulus, as depicted in Figure 2. In order to be equivariant with respect to this flow in addition to the flows in V, we simply additionally flow each hidden state by the corresponding inverse of the action flow $\psi(-a_t)$. In total this gives the combined flow for each flow channel $\psi_1(\nu - a_t)$. In practice, this is implemented by performing a roll operation on the hidden state by exactly $(\nu - a_t)$ pixels.

D.3 Encoder

The 'encoder' is simply a single convolutional layer, with 3 × 3 kernel $\mathcal{U}$, 1 input channel, and 64 output channels. The convolution uses circular padding, and no bias. The observation at timestep t (f_t), is thus processed by the encoder ($\mathcal{U} \star f_t$) yielding the processed observation of the agent. Given this observation is only a partial observation of the full world, we must pad this observation to match the world size, and the size of the hidden state. We denote this operation as 'pad' in the recurrence relation, and simply pad the boundary of the output of the encoder with 0 to match the world-size (size of the hidden state).

D.4 Decoder

We learn the parameters of the FloWM by training it to predict future observations from its hidden state and the corresponding sequence of future actions. To compute this prediction, we take a consistent $\mathrm{window\_size}$ crop from the center of the hidden state, corresponding to the same location where the encoder 'writes-in'. We denote this crop $\mathrm{window}(h_{t+1})$. To then enable the model to predict each pixel's velocity independently, we perform a pixel-wise max-pool over the V dimension ('velocity channels') before passing the result to a decoder g_θ. Specifically: $\hat{f}_{t+1} = g_\theta\big(\max_\nu (\mathrm{window}(h_{t+1}))\big)$.

The decoder g_θ is a simple 2 layer convolutional neural network with 3 × 3 convolutional kernels, 64 hidden channels, and a ReLU non-linearity between the layers.

D.5 Ablation: No Velocity Channels

To construct the ablated version of the FloWM with no velocity channels, we simply set $V = \{(0,0)\}$. Since the original FloWM model simply max-pools over velocity channels, the decoder already only takes a single velocity channel as input, so no other portions of the model need to change. We note that this model is identical to a simple convolutional recurrent neural network with self-motion equivariance. Explicitly:

$$h_{t+1} = \sigma\big(\psi_1(-a_t) \cdot \mathcal{W} \star h_t + \mathrm{pad}(\mathcal{U} \star f_t)\big). \tag{3}$$

D.6 Ablation: No Self-Motion Equivariance

To construct the ablated version of the FloWM with no self-motion equivariance, we simply remove the term $-a_t$ from the flow of the recurrence relation. Explicitly:

$$h_{t+1}(\nu) = \sigma\big(\psi_1(\nu) \cdot \mathcal{W} \star h_t(\nu) + \mathrm{pad}(\mathcal{U} \star f_t)\big). \tag{4}$$

We note that this is equivalent to the original FERNN model with the addition of the partial-observability modifications (padding the input and windowing the hidden state for readout).

D.7 Ablation: No Velocity Channels + No Self-Motion Equivariance

To ablate both velocity channels and self-motion equivariance, we reach a simple convolutional RNN:

$$h_{t+1} = \sigma\big(\mathcal{W} \star h_t + \mathrm{pad}(\mathcal{U} \star f_t)\big). \tag{5}$$

D.8 Ablation: Conv-RNN + Action Concat

In the appendix, we additionally include a version of the model with no velocity channels, no self-motion equivariance, but with action conditioning for both the input and hidden state. Specifically, we concatenate the current action to the hidden state vector and the input image as two additional channels (corresponding to the x and y components of the action translation vector), and change the number of input channels for both convolutions correspondingly. Explicitly:

$$h_{t+1} = \sigma\big(\mathcal{W} \star \mathrm{concat}(h_t, a_t) + \mathrm{pad}(\mathcal{U} \star \mathrm{concat}(f_t, a_t))\big). \tag{6}$$

Empirically, we find that this additional conditioning marginally improves the model performance; however, the model is still clearly unable to learn the precise equivariance that the FloWM has built-in.

D.9 Training Details

To train the FloWM, as well as the ablated versions, we provide the model with 50 observation frames as input, and train the model to predict the next 20 observations conditioned on the corresponding action sequence. Specifically, we minimize the mean squared error (MSE) between the output of the model and the ground truth sequence, averaged over the 20 frames (from frame 50 to 70):

$$\mathcal{L}_{MSE} = \frac{1}{20} \sum_{t=50}^{70} ||f_t - \hat{f}_t||_2^2. \tag{7}$$

The models are trained with the Adam optimizer with a learning rate of $1e-4$, a batch size of 32, and gradient clipping by norm with a value of 1.0. They are each trained for 50 epochs, or until converged. Some models, such as the FloWM with self-motion equivariance but no velocity channels, took longer than 50 epochs to converge, and thus training was extended to 100 epochs. All FloWM models (and ablations) have roughly 75K trainable parameters.

E Video Diffusion Transformer Baseline Details

E.1 Video Diffusion Transformers

Diffusion Transformer based video generation models are the most prominent so-called world models today [Peebles and Xie, 2023, Brooks et al., 2024]. Training follows a similar formula with diffusion image generation pipelines, requiring attention over the temporal dimension to retain temporal consistency. For video data, diffusion models are typically trained within the latent space of a variational autoencoder (VAE) [Rombach et al., 2022, Gupta et al., 2023], where raw video frames are first compressed into compact latent representation.

The ability for these video diffusion models to generate impressively realistic videos has led to an increased interest for their use as world models, and there is a growing focus in ensuring the spatiotemporal consistency of these models as world simulators. Due to the size complexity of the input token space, to generate long videos, researchers have turned to autoregressive sampling and sliding window attention; though ubiquitously used, we speculate that the drawbacks of this method for inference, where there is no hidden state passed between generation rounds after the window shifts, is a major reason that DiT baselines fails on the simple task presented in this work.

E.2 Diffusion Forcing Transformer Baseline

Due to its claims of long term consistency and flexible inference abilities, for our baseline we chose a History-guided Diffusion Forcing training scheme, using latent diffusion with a CogVideoX-style transformer backbone, which we will call here DFoT [Song et al., 2025, Chen et al., 2024, Yang et al., 2025, Rombach et al., 2022]. Models for state of the art video world modeling today have similar training formulas and architectures for the backbone [Xiang et al., 2024, Ball et al., 2025, Decart et al., 2024, Agarwal et al., 2025]. We first trained a spatial downsampling VAE on frames of the MNIST-world data subsets, then pass input video frames through the VAE to form a latent representation before it reaches the diffusion model.

Following the standard diffusion forcing training scheme, each frame during training is corrupted with independent gaussian noise, and the training target is to predict some form of the ground truth from these noisy frames. Song et al. [2025] showed that using this training schedule allows for the history image frames to be prepended to the noisy frames as context in the same self attention window, with zero (or some minimal) noise level, called History Guidance.

For DFoT models, unlike FloWM recurrent models, during training we make no distinction between observation and prediction frames, and train on length 70 sequences in the self-attention window, where each frame's tokens receive independent gaussian noise. During inference, we utilize History Guidance with 70 frames in the attention window to provide image context for consistent generation. Specifically, the 50 observation frames are given minimal noise, and the 20 prediction frames all begin at full noise; then the entire set of frames is passed through the model multiple times according to the scheduler to complete denoising the target frames to get clean frames as outputs. Specifically, each latent frame in the sequence $\mathbf{x}_t \in \mathbf{x}_\mathcal{T}$ is assigned an independent noise level $k_t \in [0, 1]$. Each frame (more precisely, each collection of spatial tokens corresponding to a single frame) is noised according to the following equation:

$$\mathbf{x}_t^{k_t} = \alpha_{k_t} \mathbf{x}_t^0 + \sigma_{k_t} \epsilon_t, \quad \epsilon_t \sim \mathcal{N}(0, 1), \tag{8}$$

where α_{k_t} and σ_{k_t} denote the signal and noise scaling factors, respectively, determined by the chosen variance schedule. The diffusion model ϵ_θ takes in as input a sequence of noise levels, $k_\mathcal{T}$, and the sequence of independently noised inputs $\mathbf{x}_\mathcal{T}^{k_\mathcal{T}}$. The model is trained to minimize the following diffusion loss:

$$\mathbb{E}_{k_\mathcal{T}, \mathbf{x}_\mathcal{T}, \epsilon_\mathcal{T}} \left[\left\| \epsilon_\mathcal{T} - \epsilon_\theta\big(\mathbf{x}_\mathcal{T}^{k_\mathcal{T}},\, k_\mathcal{T}\big) \right\|^2 \right]. \tag{9}$$

For more information on diffusion models in general, please see Chan [2025].

To run inference for generation of longer videos (as in the length extrapolation experiments), we use a sliding window approach, matching the number of frames seen during training in the self-attention window. Specifically, we keep 50 context frames, shift the window ahead by 20 frames after each chunk is done denoising, and use the newly generated frames as context for the next generation round.

E.3 DFoT Training Details

We train a separate DFoT model for each data subset to separate out its abilities. We embed actions, which are dimension 2, using a simple MLP embedder, and concatenate it to the video tokens, following CogVideoX. Our 96M parameter DFoT's validation loss and validation metrics converge after 240k steps on 1 NVIDIA L40S 48GB GPU with a batch size of 128. More training hyperparameters are reported in Table 5.

Table 5: DFoT configurations. Classifier-free guidance [Ho and Salimans, 2022] for conditions is not used during inference; though the models have been trained to allow for it, we find their instruction following ability not to be a limiting factor. Loss weighting uses sigmoid reweighting proposed by Kingma and Gao [2023] and adopted by Hoogeboom et al. [2024]. History guidance follows the stablized conditional method (level = 0.02) from Song et al. [2025]; please refer to their code base for details.

Section	Key	Value
Training	Effective batch size	128
	Learning rate	2e-4 with linear warmup
	Warmup steps	2,000
	Weight decay	1e-3
	Training epochs	175
	GPU usage	1×L40S
	Optimizer	Adam, betas=(0.9, 0.99)
	Training strategy	Distributed Data Parallel
	Precision	Bfloat16
Diffusion	Objective	v-prediction
	Sampling steps	50
	Noise schedule	cosine
	Loss weighting	sigmoid
Model	Total parameters	95.3 M
	# attention heads	12
	Head dimension	64
	# layers	10
	Time embed dimension	256
	Condition embed dimension	768
Inference	History guidance	stablized conditional (level = 0.02)
	Context frames	50
	Sampler	DDIM

E.4 VAE Training Details

Following standard practice, we use a VAE to perform latent diffusion; doing diffusion on pixels instead could offer perceptually different results, but we do not believe it would alter the results of the model. We train our 8x spatial downsampling VAE on sample frames from a mix of all of the data subsets, such that all combinations of overlapping MNIST digits are within the training distribution. Our 20M parameter VAE's validation loss converges at about 90k steps, using an effective batch size of 256 across 4 NVIDIA L40S 48GB GPUs with a learning rate of 4e-4. We utilize a Masked Autoencoder Vision Transformer based VAE [He et al., 2021]. We directly apply the VAE code from Oasis [Decart et al., 2024], including an additional discriminator loss that helps with visual quality; please refer to their work for more details. The reconstruction MSE accuracy reaches 0.02, so any DFoT MSE can be expected to be 0.02 higher than if trained on pixels; we believe this should not affect convergence behavior of the DFoT models on the downstream task. During diffusion training, for our VAE with latent dimension 4, and spatial downsampling ratio 8, input videos of shape `[num_frames, channels, height, width]` are converted to shape `[num_frames, 4, height // 8, width // 8]`. More training hyperparameters are reported in Table 6.

Table 6: VAE configurations. The input size from the dataset is 32×32.

Component	**Option**	**Value**
Training	Learning rate	4e-4
	Effective batch size	256
	Precision	Float16 mixed precision
	Strategy	Distributed Data Parallel
	Warmup steps	10,000
	Training epochs	172
	GPU usage	4×L40S
	Optimizer (AE)	Adam, betas=(0.5, 0.9)
	Optimizer (Disc)	Adam, betas=(0.5, 0.9)
Model	Total parameters	19.7 M
	Encoder dim	384
	Encoder depth	4
	Encoder heads	12
	Decoder dim	384
	Decoder depth	7
	Decoder heads	12
	Patch size	8
Latent	Latent dim	4
	Temporal downsample	1