WorldMAR: Experiments in Improving Speed and Consistency in World Models

Praccho Muna-McQuay Brown University
Tanish Makadia Brown University
Noah Rousell Brown University
Heon Lee Brown University

Abstract

World Models are increasingly popular for modeling complex environments where environment interaction is costly. It's common to use video data paired with actions to train action-conditioned next-frame prediction models to act as these world models, yet these often suffer from (i) temporal inconsistency and (ii) high latency. A recent architecture used for image-generation that combines a heavy autoregressive architecture with a lighweight patch-based diffusion model shows impressive speedups compared to standard DiT. A recent strategy to improve temporal consistency relies on the insight that visual coverage of a frustrum can be achieved with a constant amount of frames given intelligent retrieval over a memory bank. We introduce **WORLDMAR**, a world model for the Minecraft environment with a masked autoregression architecture equipped with world memory to validate recent approaches to improving latency and temporal consistency in video modeling. While bottlenecked by compute, we see initial signs that this architecture has the capacity to model the environment dynamics and visual information, and provide profiling to show the theoretical inference-time speedup achievable from this architecture.

1 Introduction

Model-based agents in open-world environments require accurate forward models of visual dynamics to plan and act effectively (Hafner et al. [2024]). Existing video prediction architectures are (i) temporally inconsistent, especially over long time-horizons and (ii) slow at rolling out long sequences. We address these challenges with **WORLDMAR**, a masked autoregression architecture that operates on frame latents and conditions generation on pose and actions. Although the architecture is task-agnostic, we instantiate it on the Minecraft environment with the OASIS VAE tokenizer (Decart et al. [2024]). OASIS is a diffusion video generator conditioned on player actions. However, it loses temporal coherence quickly. WorldMem (Xiao et al. [2025]) addresses this by maintaining a memory bank of past frames and associated states (poses and timestamps), which it queries using a spatial overlap heuristic, achieving faithful scene recall over large viewpoint or temporal gaps. However, it is at the cost of high latency, due to the need for full attention over the video inputs for each diffusion step. WORLDMAR inherits the memory concept yet replaces the expensive every-step processing with lighter-weight masked autoregressive (MAR) denoising of latent tokens, delivering both fidelity and long-horizon consistency. Under equal $\sim$500M parameter budgets on two NVIDIA A5000 GPUs, WORLDMAR samples video $\sim 4\times$ faster than OASIS and $\sim 10\times$ faster than WorldMEM.

2 Related Works

Video Diffusion Models. Spatio-temporal diffusion in latent space has recently emerged as a leading paradigm for controllable video generation. OASIS (Decart et al. [2024]) pioneers action-conditioned diffusion for Minecraft, while VideoLDM (Blattmann et al. [2023]), Diffusion-4k (Zhang et al. [2025]), and History-Guided Video Diffusion (Song et al. [2025]) push fidelity on natural video. MARDini (Liu et al. [2024]) scales masked autoregressive diffusion to minute-long clips, and our work adopts a similar masked denoising scheme in the latent domain.

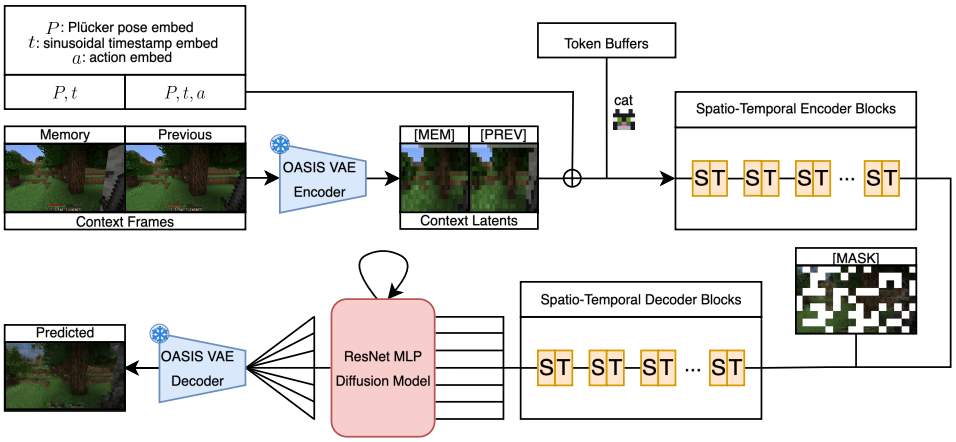

Figure 1: WORLDMAR Architecture: Overview of the WORLDMAR architecture. Memory (pose-retrieved) and previous (frames immediately prior to the one bring predicted) frames make up the context, and are encoded to latent space with the pre-trained OASIS encoder. They are given embeddings related to action, timestep, and pose, and concatenated with a buffer. These tokens are encoded with attention between non-masked tokens, then decoded with masked positions being replaced with a mask token the model learns to predict on. The outputs for these tokens serve as conditioning for a light-weight per-token DDPM with an MLP backbone, and the whole frame can be subsequently decoded.

Diffusion-Autoregressive Hybrids. Diffusion Forcing (Chen et al. [2024]) and Autoregressive Image Generation without Vector Quantization (Li et al. [2024]) demonstrate that next-token language training objectives can be blended with full-sequence diffusion to accelerate sampling. WORLDMAR extends this idea to video by masking subsets of latent tokens and denoising them in parallel, reducing wall-clock generation over diffusion models, while retaining the capacity to model a stochastic distribution over tokens.

Memory-Based World Simulation. WorldMem (Xiao et al. [2025]) introduces a memory bank that stores frames and their pose to recover distant views reliably. Our retrieval strategy borrows this design as outlined in Section 3.1.

Spatio-Temporal Self-Attention. Transformer backbones such as ViT (Dosovitskiy et al. [2021]), TimeSformer (Bertasius et al. [2021]), and separable ST attention networks (Zhang et al. [2021]) are now widely adopted and achieve leading performance for video understanding. We inherit their factorized attention kernel but operate in a compressed latent space to reduce computational demands.

Latent Autoencoders. Variational autoencoders and vector-quantized models (VQ-VAE, VQGAN) compress high-resolution video before generative modeling. OASIS employs a spatial VAE, while latent diffusion models (Rombach et al. [2022]) popularized the approach for images. WORLDMAR leverages the frozen OASIS encoder as its tokenizer and focuses its modeling capacity on environmental dynamics rather than pixel fidelity.

3 Methods

We model the video as an autoregressive, Kth-order Markov process over frames. In other words, only the most informative K past frames—and the most recent action—affect the next state. Concretely,

$$p(x_{1:T}|a_{1:T-1}) = \prod_{t=1}^{T} p(x_t|x_{1:t-1}, a_{t-1}),$$

but conditioning on the full history is intractable for long sequences. We therefore impose the Markov assumption and restrict the conditioning set to

$$p(x_t|x_{1:t-1}, a_{t-1}) \approx p(x_t|x_{S_t}, a_{t-1}),$$

where $S_t \subseteq \{1, \ldots, t-1\}$ and $|S_t| \leq K \ll t$. The index set S_t contains up to K frames judged most salient for predicting x_t.

To define the frames most salient for prediction, we define a relevance score $R(S, t)$ that quantifies how informative the collection of frames indexed by S is for the target frame x_t. Then

$$S_t = \underset{S \subseteq \{1,\ldots,t-1\}, |S| \leq K}{\arg\max} R(S, t).$$

Under this Kth-order Markov approximation, the model scales with K rather than t, making training and inference feasible on realistic video lengths while still capturing long-range dependencies through the relevance function R.

For the choice of what frames are relevant, the basis of our plans was to make use of 3D information through pose (the (x, y, z, θ, ϕ) position and viewing direction associated with each frame) to determine these. In the middle of our discussions, WorldMem (Xiao et al. [2025]) was released who also shared this same approach when it came to using pose to determine relevance. We adapt their algorithm for our purposes.

3.1 Memory Retrieval via Pose

As in WorldMem (Xiao et al. [2025]), we maintain a memory bank of historical latent frames and their associated state embeddings $(\mathcal{I}_i, p_i, \tau_i)_{i=1}^{T}$. Given the current state $(\mathcal{I}_t, p_t, \tau_t)$, we iteratively select up to K context frames that maximize field-of-view (FOV) overlap while avoiding redundancy. We partition the K context frames into $M, (\leq K)$ *previous* frames and $K - M$ *memory* frames. The previous set is deterministically chosen as the M most recent entries in the memory bank, providing a strong local prior and anchoring temporal continuity. The remaining memory frames are sampled using the greedy retrieval procedure below.

1. **Inductive bias from previous frames:** The M most recent frames are automatically included in the context window, providing a strong local prior and anchor temporal continuity.
2. **Greedy coverage:** For each subsequent slot $i \in M+1, \ldots, K$ we draw a Monte-Carlo set of 3-D points s_i inside the current camera frustum of $\mathcal{I}_t$. For every candidate index j not yet selected we compute the overlap set $o_{i,j} = s_i \cap \text{proj}(j)$ and define the confidence score where η controls the recency penalty. We pick $j^\star = \arg\max_j \alpha_{i,j}$, add it to the context list, and update the residual region by subtracting the newly covered points: $s_{i+1} = s_i \setminus o_{i,j^\star}$.

The resulting set S contains the full context frames used as input to the OASIS VAE encoder to retrieve the context latents.

Memory bank maintenance. Instead of storing all past frames—which would grow without bound and become intractable—we enforce a fixed capacity L for the memory bank. At inference time, each new frame $(\mathcal{I}_t, p_t, \tau_t)$ is appended, and one existing entry is pruned. We assign each stored frame i a freshness weight

$$w_i = e^{-\gamma\,(t-i)},$$

so that older frames decay exponentially. To choose which frame to remove, we first shortlist those with the highest FOV overlap with the incoming frame (to avoid redundancy), and among them, remove the one with the smallest w_i. This exponential-decay removal scheme biases removal toward redundant and stale frames, preserving both diversity and temporal continuity.

3.2 Embedding Parameterizations

We employ traditional embedding schemes such as sinusoidal timestamp embeddings. Additionally, we use the following specialized embedding schemes to condition each latent token.

Plücker Pose. The 6-DoF camera pose is represented in Plücker coordinates (Sitzmann et al. [2022]), which enables dense pose representation and relative pose embeddings. Concatenating the ray direction $\mathbf{d}$ with its moment $\mathbf{m}$ yields a 6-vector that a two-layer MLP projects into the latent dimension.

Patchification. Each latent feature map is partitioned into non-overlapping $p \times p$ patches (p=2). Flattening every patch into a single token shortens sequences by p^2 and reduces memory cost.

Two-Dimensional RoPE. Self-attention is permutation-invariant, so we inject spatial structure using 2-D Rotary Positional Embeddings (RoPE). By rotating query and key vectors according to each token's (x, y) patch coordinates, RoPE equips the model with relative positional awareness. As highlighted by Su et al. [2023], RoPE offers three advantages: (i) it is sequence-length agnostic, supporting arbitrarily long rollouts; (ii) inter-token influence decays smoothly with larger relative offsets; and (iii) it remains compatible with linearized self-attention, providing relative position encoding at constant memory and computation cost.

3.3 Architecture

Inspired by the speed of MAR (Li et al. [2024])—which for single-frame generation exhibits a $\sim 4\times$ speedup compared to traditional full-frame diffusion, for the same quality—we wanted to effectively apply the same technique for *conditional* frame generation, where the conditioning is salient prior frames and the current action. To be clear, there is the more general autoregressive problem of predicting the next frame which we are modeling at the highest level, but our application of MAR here is to autoregressively predict (latent) tokens within this predicted frame. Since this autoregression over frame tokens happens in latent space, we require some VAE to encode our raw images. To avoid the difficulties of training this ourself, we make use of the ViT autoencoder used in Oasis (Decart et al. [2024]).

Because this VAE is not quantized, we cannot use a softmax to define a distribution over possible tokens as in typical autoregression. We can, however, use the same principle as in MAR (Li et al. [2024]) of instead defining this distribution with a diffusion model, which takes a single conditioning vector produced by a network that can see all previous tokens. In our case, this network is a ViT that must do spatio-temporal attention over the tokens across every frame in our set of memory frames, previous frames, and the prediction frame.

The way this functionally works is that, for a given sequence, there may be some set of unmasked tokens (either available during training, or already predicted through iterative sampling), and a set of masked tokens for which we want to make predictions. Unmasked tokens start in the latent space of the VAE's encoder with some dimension d_{VAE}, but are then projected to a higher embedding dimension d_{model} for the Transformer blocks. An encoder allows the unmasked positions to attend to each other only, with the masked positions being attention masked. The output of the encoder is then fed into a decoder, which replaces the embeddings in masked positions with a learned [MASK] token. The output of the decoder at each masked position is then the vector used as conditioning for the patch-level diffusion model, responsible for turning noise at the original d_{VAE} size into the predicted patch token.

The benefit of this approach is that full-frame diffusion with a Transformer backbone (DiT), as in Oasis and most next frame generation in this area, can be extremely slow for both training and sampling. When this happens at the patch/token level with a much more lightweight MLP-based backbone, while still allowing a Transformer-based part of the architecture to produce the conditioning for it instead, we enjoy much better efficiency when sampling whole frames (as seen in the results section). Unfortunately, full spatio-temporal attention at the Transformer level would be prohibitively expensive, therefore we adopt spatio-temporal (ST) blocks that separate this into strictly temporal and strictly spatial steps, as used in works like Genie (Bruce et al. [2024]).

A special note for the ST Transformer, we choose to concatenate a buffer for each frame along one of the spatial dimensions. A similar technique was done by the original MAR authors (Li et al. [2024]) where they prepend a 64-sized sequence of [CLS] tokens to their spatial token sequence. The motivation for this is that it gives the model a dedicated space where it can learn to pool global information, rather than arbitrarily among regular image tokens it deems less salient, an observed behavior from Darcet et al. [2024].

For a full architecture overview, see Figure 1.

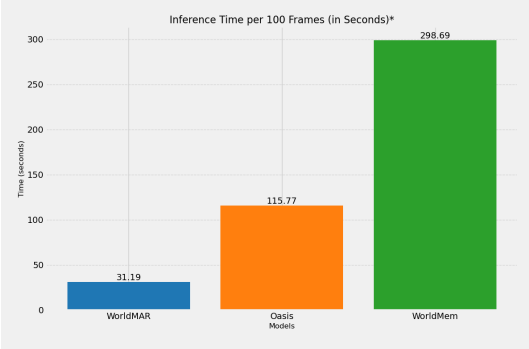

Figure 2: Inference Time per 100 Frames (in Seconds) for WORLDMAR, Oasis, and WorldMem models using two A5000 GPUs. All models have approximately 500 million parameters. WORLDMAR demonstrates significantly lower inference time compared to Oasis and WorldMem.

3.4 Training

Training this model purely comes from the diffusion loss in training our patch-level DDPM to correctly predict the noise added at randomly sampled diffusion steps k, or more explicitly

$$\mathbb{E}_{M,\epsilon,k}\left[\sum_{(i,j)\in M} \|\epsilon - \epsilon_\theta(x^k_{i,j}|k, z_{i,j})\|^2\right]$$

for masked positions (i, j) from a randomly sampled mask M in the frame we are predicting, $x^k_{i,j}$ being the ground truth token $x_{i,j}$ noised at step k, and $z_{i,j}$ being the conditioning vector from the output of the decoder.

At training, we generate the mask M by choosing a fraction of m random (i, j) positions. This m is generated using a truncated normal about a mean masking rate of 1.0 (being fully masked), with a left min of 0.7. The objective is then correctly predicting the noise added for each of these masked tokens, in one go. We make ourselves more sample efficient by effectively duplicating the batch before the diffusion step four times, sampling and predicting for four different values of k for each batch item.

One thing we noticed during training was that the model was learning an early optimal solution of effectively learning the identity for the previous frame, being able to see it completely. This in general works well for predicting the next frame, since the immediately prior frame is going to naturally be very similar. In an effort to make it learn to account for the action and the resulting dynamics however, we also added a form of regularization in attention masking parts of the previous frame in the Transformer block, preventing a direct identity mapping for some tokens and forcing the model to better learn how to "fill in the gaps."

The largest model we trained was $\sim$150M parameters, which we trained on one GH200 for around 8 epochs. Samples can be found in Figures 3a and 3b.

4 Results

We compare our model with OASIS and WorldMem in inference time. As seen in Figure 2, WORLDMAR attains approximately $4\times$ speedup over OASIS and $10\times$ speedup over WorldMem. That being said, there is still work left to be done in reaching a similar level of fidelity as OASIS and WorldMem.

Specific instances of next frame generation can be found in Figure 3. We have yet to conduct experiments in generating full-length videos, as we are still in the process of training the model. Since our preliminary experiments were conducted via next frame generation, it is difficult to judge whether the model is learning to utilize the context frames and the action embeddings. In Figure 3c, however, we show precursory signs that our model is learning to use actions.

(a) From left to right: two padding memory frames, the previous frame, the ground truth, and the predicted frame.

(b) From left to right: the previous frame, the ground truth, and the predicted frame.

(c) From left to right: the previous frame, the ground truth, and the predicted frame.

(d) From left to right: the previous frame, the ground truth, and the predicted frame.

(e) From left to right: the previous frame, the ground truth, and the predicted frame.

(f) From left to right: the previous frame, the ground truth, and the predicted frame.

(g) From left to right: the previous frame, the ground truth, and the predicted frame.

Figure 3: A display of sample generations. Subfigure 3c indicates learning of the action.

5 Discussion

Efficiency and temporal consistency are two obstacles to employing world models in 3D environments as aids to agents. Architectural changes to reduce the need for expensive diffusion operations such as MAR and constant-frame consistency solutions such as WorldMem will allow world models trained using the rich signal present in abundant video data to be used in new applications, such as training in imagination or even model-predictive control. An exciting direction for future work is exploring how to flexibly trade off low-level reconstruction quality for efficiency. Pushing the envelope in the speed-quality frontier and controlling which information and dynamics present in video are learned by world models will bring this even further.

A Division of Labor

- **Praccho:** Worked heavily on MAR implementation. Implemented the ST Transformer, some diffusion modifications, as well as the logic for training and iterative sampling with MAR. Also abstracted model configuration for ease of use, and handled the training runs.
- **Tanish:** Did significant work on the logic for pose, action, timestep, and categorical embeddings for the Transformer blocks. Abstracted masking mechanism to be multi-frame while preserving predictions to one. Implemented the image logging and animation capabilities for iterative sampling.
- **Noah:** Handled the dataset collection and parsing logic, implemented latent caching with the Oasis VAE, and worked on and optimized the pose retrieval mechanism for gathering memory frames for each batch.
- **Heon:** Implemented diffloss. Helped with memory retrieval implementation and logic for optimization and with 2D RoPE. Implemented memory cache for inference time.

References

Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding?, 2021. URL https://arxiv.org/abs/2102.05095.

Andreas Blattmann, Robin Rombach, Huan Ling, Tim Dockhorn, Seung Wook Kim, Sanja Fidler, and Karsten Kreis. Align your latents: High-resolution video synthesis with latent diffusion models, 2023. URL https://arxiv.org/abs/2304.08818.

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.

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.

Timothée Darcet, Maxime Oquab, Julien Mairal, and Piotr Bojanowski. Vision transformers need registers, 2024. URL https://arxiv.org/abs/2309.16588.

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

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale, 2021. URL https://arxiv.org/abs/2010.11929.

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

Tianhong Li, Yonglong Tian, He Li, Mingyang Deng, and Kaiming He. Autoregressive image generation without vector quantization, 2024. URL https://arxiv.org/abs/2406.11838.

Haozhe Liu, Shikun Liu, Zijian Zhou, Mengmeng Xu, Yanping Xie, Xiao Han, Juan C. Pérez, Ding Liu, Kumara Kahatapitiya, Menglin Jia, Jui-Chieh Wu, Sen He, Tao Xiang, Jürgen Schmidhuber, and Juan-Manuel Pérez-Rúa. Mardini: Masked autoregressive diffusion for video generation at scale, 2024. URL https://arxiv.org/abs/2410.20280v1.

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.

Vincent Sitzmann, Semon Rezchikov, William T. Freeman, Joshua B. Tenenbaum, and Fredo Durand. Light field networks: Neural scene representations with single-evaluation rendering, 2022. URL https://arxiv.org/abs/2106.02634.

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.

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.

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.

Jinjin Zhang, Qiuyu Huang, Junjie Liu, Xiefan Guo1, and Di Huang. Diffusion-4k: Ultra-high-resolution image synthesis with latent diffusion models, 2025. URL https://arxiv.org/abs/2503.18352.

Yanyi Zhang, Xinyu Li, Chunhui Liu, Bing Shuai, Yi Zhu, Biagio Brattoli, Hao Chen, Ivan Marsic, and Joseph Tighe. Vidtr: Video transformer without convolutions, 2021. URL https://arxiv.org/abs/2104.11746.