MoTWorld: Geometry-Grounded World Modeling with Joint Video-Voxel Generation

Anonymous authors
Paper under double-blind review

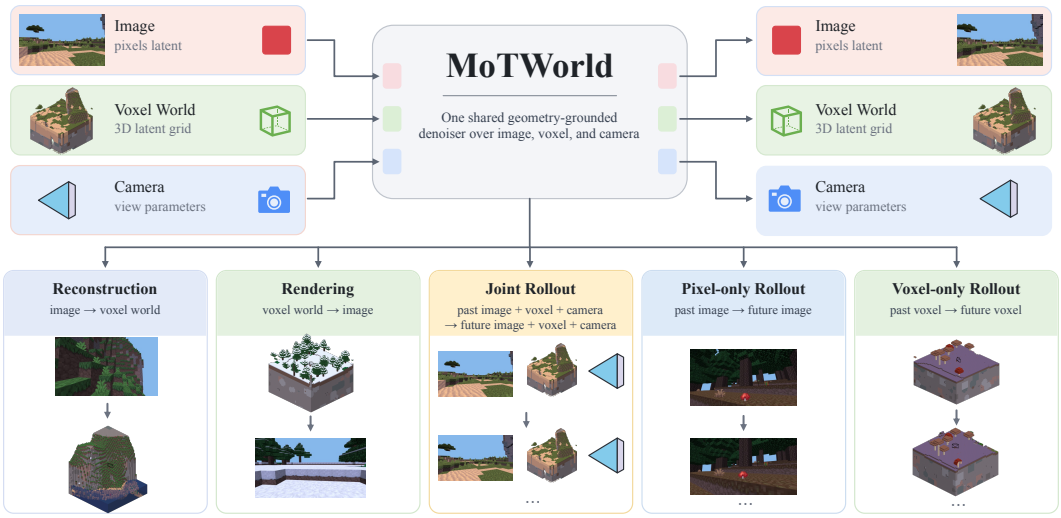


Figure 1: Illustration of MoTWorld. We model video frames, semantic voxel grids, and camera parameters with a unified Mixture-of-Transformers denoiser. By specifying which modalities are observed or generated, we use the same model for a variety of tasks, including image-to-voxel reconstruction, voxel-to-image rendering, joint video-voxel rollout, video-only rollout, and voxel-only rollout.

Abstract

Interactive world models predict future observations from actions of an agent, but maintaining spatial consistency over long rollouts remains challenging. Explicit 3D states provide a persistent representation of the scene, yet cascaded approaches typically predict geometry first and then use it to condition video generation, limiting information exchange between the two modalities. We propose MoTWorld, a world model that jointly generates video frames and semantic voxel grids within a Mixture-of-Transformers framework. Specifically, we represent images, voxels, and camera parameters as separate streams with modality-specific weights and shared attention. By changing which streams are observed or generated, we use the same model for image-to-voxel reconstruction, voxel-to-image rendering, and action-conditioned rollout. We further identify that shared attention alone does not explicitly establish correspondence between image and voxel tokens, which occupy different coordinate systems. We propose a novel rotary positional encoding that uses camera geometry and predicted depth to represent both modalities in a common 3D coordinate system. We account for voxel extent and depth uncertainty in the encoding to guide cross-modal attention. Experiments on two Minecraft-style test sets with exact 3D ground truth shows that MoTWorld significantly improves single-image voxel reconstruction and video quality over 600-frame rollouts compared with state-of-the-art methods, while supporting reconstruction, rendering, and generation within a unified architecture.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

1 INTRODUCTION

Interactive world models predict how an environment evolves in response to an agent's actions. Recent video-based models generate interactive environments by predicting future frames from visual observations and actions (Ha & Schmidhuber, 2018; Alonso et al., 2024; Valevski et al., 2025; Bruce et al., 2024; Decart et al., 2024; Feng et al., 2026; Guo et al., 2025; Parker-Holder et al., 2025; Wang et al., 2026b; Agarwal et al., 2026; Zhang et al., 2026b). However, maintaining a consistent scene over long rollouts remains challenging (Wang et al., 2025; Huang et al., 2026a; Xiao et al., 2026). When a model represents the environment through a limited history of images, information about regions outside the current view may become unavailable, and errors can accumulate as generated frames are reused as context. Memory-based approaches retain and retrieve earlier observations to improve consistency (Xiao et al., 2026), but the stored information remains tied to individual viewpoints rather than an explicit 3D scene representation.

An explicit 3D state provides a spatial representation that can persist across changes in viewpoint. PERSIST (Garcin et al., 2026), for example, forecasts a semantic voxel grid and rasterizes it to condition a video denoiser, and other models condition the video on a 3D cache (Huang et al., 2025b;a; Ren et al., 2025; Zhou et al., 2026), a geometry-field regularizer (Dai et al., 2026), generated depth or point clouds (Liao et al., 2026; Lu et al., 2026), or a cascaded occupancy forecast (Zheng et al., 2024; Li et al., 2025; Yang et al., 2026b; Li et al., 2026a). In all of these, geometry conditions the video but the video never corrects the geometry, and no single model can both infer the grid from an image and render an image from the grid (Li et al., 2024; Wu et al., 2026a).

We argue that the two modalities should correct each other: images provide visual cues for estimating scene geometry, while voxel grids retain spatial information beyond the current view. To combine these representations, we propose MoTWorld (as shown in Figure 1), a flow-matching model based on the Mixture-of-Transformers (MoT) architecture (Liang et al., 2024). We process images, voxels, and camera parameters with modality-specific weights and exchange information through shared attention. We assign independent noise levels across modalities and frames and train under different conditioning configurations, allowing the same model to support reconstruction, rendering, and rollout (Zhou et al., 2025; Esser et al., 2024; Agarwal et al., 2026; Bi et al., 2026). However, shared attention alone does not specify the correspondence between image and voxel tokens, which depends on camera geometry and scene depth.

To provide this geometric information, we introduce SpRoPE (SpatialRoPE), a rotary positional encoding for image-voxel attention. Building on geometry-aware attention for multi-view images (Miyato et al., 2024; Li et al., 2026c; Wu et al., 2026c), we encode image and voxel tokens in a common metric coordinate system. We use a shared layout of temporal, image-plane, and metric 3D components, with coordinates computed for each modality. We also project voxel centers onto the image plane to provide correspondence cues in image coordinates. These encodings incorporate spatial relationships into cross-modal attention while accounting for voxel extent and depth uncertainty.

We evaluate on a Minecraft-style dataset from Craftium (Malagón et al., 2024), which provides exact voxel states, camera poses, and actions. Our contributions are three-fold:

- MoTWorld, a unified Mixture-of-Transformer flow matching model over video, voxels, and camera parameters that supports reconstruction, rendering, and rollout with shared parameters.
- SpRoPE, a rotary positional encoding method that uses modality-specific spatial coordinates, per-block depth estimation, and voxel-to-image projection to guide cross-modal attention while accounting for voxel extent and depth uncertainty.
- We evaluate MoTWorld against simulator ground truth, demonstrating improved single-image voxel reconstruction and long-horizon video quality over state-of-the-art methods. We further analyze the effects of geometric encoding and joint denoising through ablations.

2 RELATED WORK

World models with a persistent 3D state. Occupancy world models tokenize a semantic voxel grid and roll it forward from the ego trajectory, with autoregressive, diffusion, Gaussian, triplane and point-map variants (Zheng et al., 2024; Gu et al., 2024; Yang et al., 2025a; Wang et al., 2026a; Wei et al., 2024; Zuo et al., 2025; Xu et al., 2025; Yan et al., 2025; Zhang et al., 2024; Ma et al.,

2024; Lu et al., 2026). They are scored as geometry, which is the evaluation we want, but they emit no pixels, so no image can correct the grid. A second family conditions video generation on a 3D cache of what it has already rendered (Zhou et al., 2026; Huang et al., 2025b;a; Ren et al., 2025) or keeps the 3D structure implicit (Parker-Holder et al., 2025). Driving world models cascade a fixed occupancy generator into multi-view video (Li et al., 2025; Yang et al., 2026b; Li et al., 2026a; Yang et al., 2026a), and unified models use geometry as a regularizer, as a voxel-only generator, or as a native depth state under known cameras (Dai et al., 2026; Merino et al., 2026; Liao et al., 2026; Huang et al., 2026b). Closest to us is PERSIST (Garcin et al., 2026), which forecasts a voxel grid with one DiT and renders it with a second DiT through a rasterize, with a separate regressor for the camera. But the cascading structure neglect the mutual benefits across 2D and 3D modalities during prediction. Ours treats the voxel grid as a generated modality equally with the video, read and written by the same attention, which leverage mutual information across modalities.

Interactive video world models. Action-driven video world models range from latent-space models (Ha & Schmidhuber, 2018; Hafner et al., 2025) to pixel-space diffusion and autoregressive generators for games and open worlds (Alonso et al., 2024; Valevski et al., 2025; Bruce et al., 2024; Che et al., 2025; Feng et al., 2026; Yu et al., 2025b; Bar et al., 2025), including Oasis, MineWorld and Matrix-Game on Minecraft (Decart et al., 2024; Guo et al., 2025; Wang et al., 2026b). Their finite context is extended by history guidance, state-space backbones, self-forcing, key-value memory and pose-based retrieval (Song et al., 2025; Po et al., 2025; Huang et al., 2026a; Wu et al., 2026b; Zhang et al., 2026a; Xiong et al., 2026; Xiao et al., 2026; Zhang et al., 2026b), all of which keep pixels for longer (Wang et al., 2025; Yu et al., 2025a). However, without the geometry, the long-horizon consistency is hard to preserve. Therefore, PERSIST (Garcin et al., 2026) is our primary baseline.

Mixture-of-Transformers denoisers. MoT (Liang et al., 2024) gives each modality its own non-embedding parameters under a global attention, a pattern shared by Transfusion, MM-DiT and Cosmos 3 (Zhou et al., 2025; Esser et al., 2024; Agarwal et al., 2026), by two-tower models that attach a language model to a generator (Jain et al., 2026; Gao et al., 2026), and by recent world-action models (Bi et al., 2026; Li et al., 2026b; Hu et al., 2026; Team et al., 2026; Fang et al., 2026; Mo et al., 2026). None is geometry-grounded. Their modalities share no coordinate system, and their cross-modal attention must learn correspondence from data. We keep the MoT interface and add the positional code under which the 2D and 3D towers can find each other.

Rotary Position Embedding. RoPE (Su et al., 2024) encodes positions by rotating pairs of feature channels, and video DiTs extend this encoding to time, height, and width (Yang et al., 2025b; Wan et al., 2025; Wang et al., 2024; Wei et al., 2025). For multi-view images with known camera parameters, GTA, PRoPE, SPAD, and PAIWorld inject relative camera transforms into attention (Miyato et al., 2024; Li et al., 2026c; Kant et al., 2024; Huang et al., 2026b), while RayRoPE (Wu et al., 2026c) computes the expected rotary encoding over a depth interval along each token's projected ray. All of these methods model relationships between 2D tokens. SpRoPE shares RayRoPE's expected-RoPE formulation but relates 2D image tokens to 3D voxel tokens in a common metric coordinate frame, averages rotary encodings over voxel extents, and uses exact first-hit depths for supervision.

3 METHOD

MOTWORLD models a 3D scene as three coupled streams: a semantic voxel grid, the image the agent sees, and its camera. Section 3.1 casts them as one denoising problem with run-time roles, Section 3.2 describes the MoT denoiser, Section 3.3 introduces SpRoPE, which lets 2D and 3D streams attend to each other geometrically, and Section 3.4 covers training and inference.

3.1 WORLD STATE AND PROBLEM FORMULATION

At every frame τ the environment provides a semantic voxel grid $\mathbf{v}_\tau$ of G^3 blocks centered on the agent with K block classes, the rendered image $\mathbf{p}_\tau$, the camera $\mathbf{c}_\tau$ (rotation, translation and field of view) and the action $\mathbf{a}_\tau$. Following PERSIST (Garcin et al., 2026), frozen autoencoders map the grid and the image to latents $\mathbf{z}_\tau^{\mathrm{V}}$ and $\mathbf{z}_\tau^{\mathrm{P}}$, and the camera is standardized to $\mathbf{z}_\tau^{\mathrm{C}}$. We write $m \in \{\mathrm{V}, \mathrm{P}, \mathrm{C}\}$ for the modalities and call a modality at a frame a group (m, τ). A training window is T consecutive frames at a random stride, conditioned on the actions of the window.

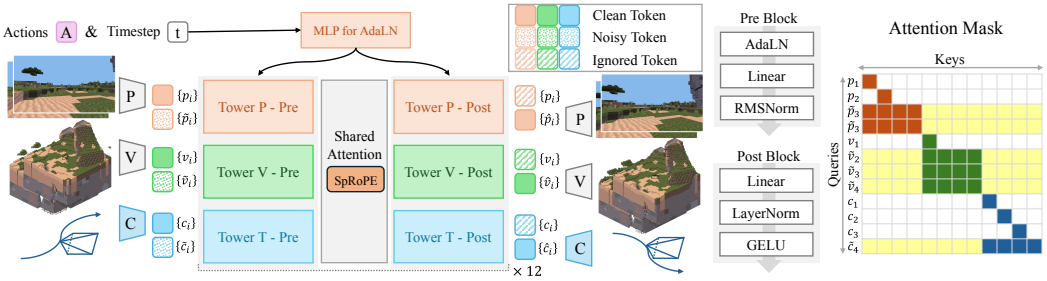


Figure 2: Overview of MoTWorld denoiser. Frozen encoders tokenise each frame's image (P), voxel grid (V) and camera (C) into clean condition tokens and noisy target tokens. Each of the 12 blocks has one tower per modality, conditioned on the actions and the noise level through AdaLN, and the towers meet only in one shared attention under SpRoPE. *Attention mask:* a clean token attends only to itself, so generated tokens cannot alter a given input, whereas a noisy token attends to all tokens. Yellow marks cross-modality entries.

Every group has a role. The architecture does not fix which groups are given and which are predicted. An *observed* group ($r_\tau^m = \text{obs}$) is held at a small noise level t_{clean}, a *generated* group ($r_\tau^m = \text{gen}$) is noised and scored, and an *absent* group ($\varnothing$) contributes no tokens, with $\mathcal{M} \subseteq \{\text{V}, \text{P}, \text{C}\}$ the set of modalities present in a window. With t_τ^m the noise level of a group and $\boldsymbol{\varepsilon}$ Gaussian noise, every group follows the same flow-matching interpolant and the model $\mathbf{f}_\theta$ predicts its velocity,

$$\mathbf{x}_\tau^m(t_\tau^m) = (1 - t_\tau^m)\,\mathbf{z}_\tau^m + t_\tau^m\,\boldsymbol{\varepsilon}, \qquad \mathcal{L}_{\text{FM}} = \sum_{m,\tau} \lambda_m\, \mathbb{1}[r_\tau^m = \text{gen}]\, \left\|\mathbf{f}_\theta^m(\mathbf{X}) - (\boldsymbol{\varepsilon} - \mathbf{z}_\tau^m)\right\|_2^2, \quad (1)$$

where $\mathbf{X}$ is the token sequence of the window and λ_m weights the modalities. Only generated groups contribute to the loss. The assignment of roles over a window is a *role matrix*, and a task is nothing more than a role matrix: the network receives no task identifier. We train on five tasks, reconstruction P→V, rendering V→P, joint rollout PV→PV, video-only rollout P→P and voxel-only rollout V→V. The reconstruction tasks observe the camera, the rollout tasks observe the earlier frames of the window and generate the last, and a stream absent from the task name is absent from the sequence. Because roles are chosen at run time, the same weights render voxels into pixels and infer pixels into voxels, and absent groups cost nothing since their tokens are dropped rather than masked.

3.2 Mixture-of-Transformers denoiser

Framework. The denoiser takes one token sequence $\mathbf{X}$ in which we concatenate, over the present groups, the patchified latents of the voxel stream, of the image stream, and the camera vector, and project each into a common width with its own input layer (Figure 2). Voxel tokens are the patches of the voxel latent, so each token stands for a cube of W blocks. Image tokens are the patches of the image latent, arranged on a lattice of N_h rows and N_w columns. The camera is one token per frame.

The model is a Mixture-of-Transformers (Liang et al., 2024). Every block contains three *towers*, one per modality, that are identical in shape and disjoint in parameters: each tower owns its adaptive layer norm, its query, key and value projections, its output projection and its feed-forward network. The towers interact only through a single attention over the concatenation of the tokens of all present modalities $\mathcal{M}$, under one mask: a noisy token attends to every token of every modality, whereas a clean condition token attends only to itself, so that the features of a given input are never altered by the tokens being generated (Figure 2, right). With $\mathbf{h}^m$ the hidden states of tower m, $\mathbf{q}^m, \mathbf{k}^m, \boldsymbol{\nu}^m$ its queries, keys and values, and SpRoPE the rotation of Section 3.3, a block computes

$$\mathbf{h}^m \leftarrow \mathbf{h}^m + \text{proj}_m\Big(\text{Attn}\big(\text{SpRoPE}(\mathbf{q}^m);\ \{\text{SpRoPE}(\mathbf{k}^{m'}), \boldsymbol{\nu}^{m'}\}_{m' \in \mathcal{M}}\big)\Big), \quad (2)$$

with adaptive layer norms and their gates omitted for clarity (Appendix B gives the full block). We condition each tower, in every block, on a per-frame vector that sums an embedding of the group's noise level t_τ^m, an embedding of the action $\mathbf{a}_\tau$ shared by all towers, and an embedding of the group's role. The noise level tells a tower how much to trust each frame, the action tells it what the agent did, and the role tells it whether a frame is context or a target.

Design choices. Modality-specific weights let each latent keep a representation suited to its own vocabulary, while the shared attention keeps the exchange between them dense. We place no

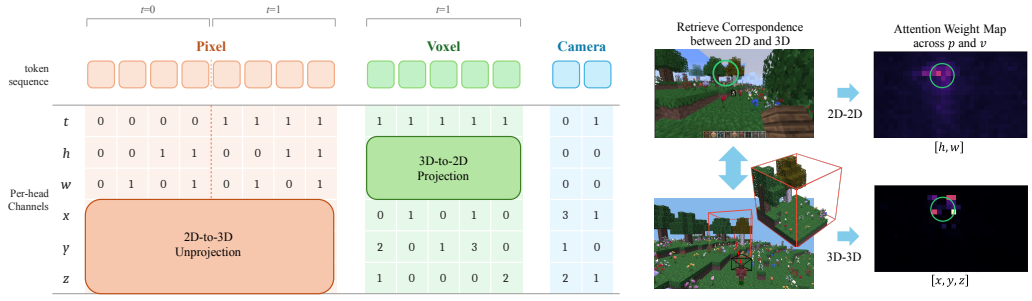


Figure 3: Illustration of SpRoPE. *(Left)* We partition each attention head into six rotary groups (t, h, w, x, y, z). Pixel tokens encode (t, h, w) and obtain (x, y, z) from camera rays and per-block depth predictions. Voxel tokens encode (t, x, y, z) and obtain (h, w) by projecting their cell centers through the camera. Camera tokens encode t and their centers (x, y, z). *(Right)* For a query pixel (green circle), attention over pixel and voxel keys highlights corresponding image regions and 3D cells, illustrating correspondence across modalities.

learned gate on the cross-modal attention, since a gate is a scalar the optimizer can drive to zero whenever ignoring the other modality lowers the loss faster, and the positional code already supplies the geometry. All towers share the same head layout, so the rotary layout of Section 3.3 applies identically to every modality.

3.3 SpRoPE: one metric frame for 2D and 3D tokens

A joint attention between image and voxel tokens is only as good as its positional code. The camera determines which image token should attend to which voxel, but a transformer with image-lattice positions on one side and grid indices on the other must learn that projective correspondence from data, and ours did not. SpRoPE gives every token, whatever its modality, a phase in one shared metric frame, so a cross-modal attention score becomes a analytical statement about geometry (Figure 3).

One rotary layout, three fills. Rotary encodings (Su et al., 2024) rotate two-dimensional slices of a query and a key by an angle that depends on position, so that their dot product depends only on the difference of the two angles. We partition every head into six groups $g \in \{t, h, w, x, y, z\}$: the frame index, the image row and column, and the three world axes in blocks. The layout is the same for every modality, and only the fill differs (Figure 3, left). A pixel token carries (t, h, w) and obtains (x, y, z) by unprojecting its ray with a predicted depth, from 2D to 3D. A voxel token carries (t, x, y, z) and obtains (h, w) by projecting its cell center through the camera, from 3D to 2D. A camera token is a point. For token i, group g and frequency $\omega_{g,k}$ (Appendix A), the encoding supplies a phase $\varphi^g_{i,k}$, where the token is, and an attenuation $\beta^g_{i,k} \in [0, 1]$, how precisely it knows, and the logit between a query i and a key j becomes

$$\mathcal{A}_{ij} = \frac{1}{\sqrt{D_h}} \sum_g \sum_k \beta^g_{i,k}\, \beta^g_{j,k}\, \left(\mathbf{q}^{g,k}_i\right)^\top \mathbf{R}\left(\varphi^g_{j,k} - \varphi^g_{i,k}\right) \mathbf{k}^{g,k}_j, \tag{3}$$

where $\mathbf{q}^{g,k}_i, \mathbf{k}^{g,k}_j$ are the two-dimensional slices, $\mathbf{R}(\theta)$ is the rotation by θ and D_h is the head dimension. Channel k of a voxel key meets channel k of an image query, and the scales a token is sure of dominate the logit.

Expected codes. All fills follow one rule: the code of a token is the expected rotary factor under the uncertainty about its position. A camera token is a point, the camera centre $\mathbf{e}_\tau$. A voxel token is a cube of width W around its centre $\mathbf{b}_n$. An image token is a ray from the camera centre $\mathbf{o}_i$ in the direction $\hat{\mathbf{r}}_i$, with depth in an interval of centre d^{mid}_i and length Δ_i. Averaging the rotation over the cube and over the interval gives, for a metric axis a with frequency ω_k,

$$\begin{aligned}
&\text{camera (point):} && \varphi_k = \omega_k\, e_{\tau,a}, && \beta_k = 1, \\
&\text{voxel (cell):} && \varphi_k = \omega_k\, b_{n,a}, && \beta_k = \operatorname{sinc}\big(\omega_k W/2\big), \\
&\text{image (ray):} && \varphi_k = \omega_k \big(o_{i,a} + d^{\text{mid}}_i\, \hat{r}_{i,a}\big), && \beta_k = \operatorname{sinc}\big(\omega_k\, \hat{r}_{i,a}\, \Delta_i/2\big),
\end{aligned} \tag{4}$$

with $\operatorname{sinc}(z) = \sin z / z$. The sinc factors suppress exactly the frequencies whose wavelength is shorter than the token's extent and leave the coarse channels intact, so a wide depth interval degrades

gracefully instead of asserting a wrong depth with full confidence. The metric term between a voxel and an image token is therefore large only when the voxel lies where the ray's surface estimate is, which makes the cross-modal logit an analytic test of whether the ray passes through the voxel. The same rule gives every other pair its natural relation, a 3D displacement between voxels, a motion-cancelling term between images of different frames, and ego motion between cameras (Appendix A).

Sharpening the ray with a depth head. Without a depth estimate an image token's depth interval is the whole segment of its ray inside the voxel volume, which is long and leaves the fine metric channels dead. A small linear head in every block therefore reads each image token and emits a log depth $\hat{\delta}_i$ and a log-spread η_i, from which

$$d_i^{\mathrm{lo}} = e^{\hat{\delta}_i - \eta_i}, \qquad d_i^{\mathrm{hi}} = e^{\hat{\delta}_i + \eta_i}, \qquad d_i^{\mathrm{mid}} = e^{\hat{\delta}_i}, \tag{5}$$

clipped to the part of the ray inside the volume, so the geometry sharpens as the features improve from block to block. The head is supervised with the simulator's first-hit depth (Appendix A).

Projected image address. A voxel's (h, w) depends on which camera looks at it, so across a window with several cameras it cannot be a single per-token phase. We evaluate the voxel-query rows of the attention per camera, with a visibility gate that removes voxels behind the camera, and stitch one softmax across them, which is exact (Appendix A). SpRoPE assumes that the camera of each frame is known, as in all experiments of Section 4. Appendix C describes an extension in which the camera stream produces the pose of a generated frame.

3.4 Training and inference

Objective. We train the model on the flow-matching loss of Eq. (1) together with the depth-head loss and three auxiliary terms: a pose loss on a per-block head that regresses the camera from the pooled image features, a Gram-anchoring term that ties the image tower's features to the frozen autoencoder's latents, and a contrastive term between voxel and image features,

$$\mathcal{L} = \mathcal{L}_{\mathrm{FM}} + \lambda_{\mathrm{depth}}\mathcal{L}_{\mathrm{depth}} + \lambda_{\mathrm{pose}}\mathcal{L}_{\mathrm{pose}} + \lambda_{\mathrm{gram}}\mathcal{L}_{\mathrm{gram}} + \lambda_{\mathrm{nce}}\mathcal{L}_{\mathrm{nce}}. \tag{6}$$

The last three are regularisers. Their weights and the values of all other constants are in Appendix B.

History corruption. During a rollout the history is the model's own output rather than ground truth. We therefore train with Diffusion Forcing (Chen et al., 2024), which gives every group an independent noise level, and additionally corrupt the observed history frames with manually designed noise in the spirit of Helios (Yuan et al., 2026), while keeping their clean noise label. Details are in Appendix B.

Task mixture and inference. Each training step draws one of the five tasks, the same on every rank, and a batch from the matching loader. At inference we integrate the generated groups from noise to data with a fixed number of Euler steps while the observed groups stay at t_{clean}. For a long rollout we slide the window and feedback the historical predicted frames.

4 Experiments

4.1 Setup

Simulator and datasets. We generate data with Craftium (Malagón et al., 2024) in the Minecraft-style Luanti engine used by PERSIST. Every frame comes with the 640×360 image, the exact 48^3 semantic block grid around the agent ($K = 2138$ classes), the camera and a 23-bit action. Episodes are freshly generated worlds of 600 frames at 24 fps. We train on MoT-train (30,000 episodes) and evaluate on MoT-eval (200 held-out episodes) and PERSIST-eval (the 256 trajectories released with PERSIST). Details are in Appendix D.

Model and training. MoTWorld has 688 M parameters ($D = 1024$, $L = 12$, $H = 16$), trained from scratch on 32 H200 GPUs with the frozen PERSIST tokenizers, the Muon optimizer (Jordan et al., 2024; Liu et al., 2025), and the history corruption and task mixture of Section 3.4. Hyperparameters are in Appendix B, and throughput and memory in Appendix D.

Baselines. We compare with Oasis (Decart et al., 2024), a pixel-history world model retrained on our dataset with the same tokenizers, and PERSIST (Garcin et al., 2026), the cascade with 3D voxels, which runs from its released PERSIST-XL checkpoint.

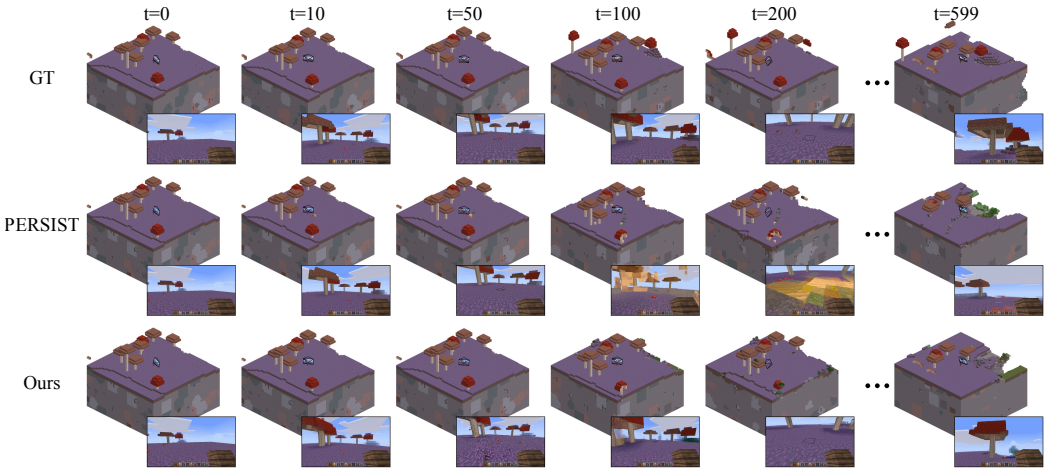


Figure 5: **Rollout results.** A 600-frame rollout on MoT-eval, showing the carried voxel world with the camera marked and the generated frame (inset) at selected steps. Ours stays close to the ground truth in both, whereas PERSIST's frames develop clear color and texture artifacts from $t = 100$ onwards and its voxel world drifts, gaining wrong block classes and losing the trees and terrain it started with.

Table 1: **Long-horizon video quality.** FVD (↓) at 200, 400 and 600 rollout frames on PERSIST-eval (256 trajectories) and MoT-eval (200), each rollout started from a single initial observation and driven by the recorded actions. MoTWorld is seeded with $\mathbf{v}_0$ and given the true pose on every frame, and the P→P row is the same checkpoint with the voxel stream switched off. Best in bold, second best underlined.

Method	PERSIST-eval			MoT-eval		
	200	400	600	200	400	600
Oasis	519.6	642.3	788.9	509.5	706.3	859.2
PERSIST w/o $\mathbf{v}_0$	176.2	194.0	199.1	137.1	158.4	173.6
MoTWorld w/o $\mathbf{v}_0$, P→P, voxel tower off	99.3	<u>108.9</u>	<u>111.9</u>	103.9	119.2	128.3
PERSIST w/ $\mathbf{v}_0$	<u>95.6</u>	133.1	148.8	<u>79.5</u>	<u>104.4</u>	<u>123.5</u>
MoTWorld, w/ $\mathbf{v}_0$	**49.6**	**48.1**	**46.8**	**57.9**	**57.5**	**56.3**

4.2 Evaluation protocol

All metrics use exact simulator ground truth on held-out episodes. **Reconstruction**: from the first frame and its pose, the model builds the voxel world in one sampler pass. We report occupancy IoU split by the camera frustum and block-class accuracy split by occlusion-aware visibility (Tables 2 and 4). **Rollout**: 600-frame rollouts on both sets from one initial observation $\langle \mathbf{p}_0, \mathbf{c}_0 \rangle$ and the recorded actions, following PERSIST, scored by FVD at 200, 400 and 600 frames and a per-frame FID curve. **Camera**: for the extension of Appendix C, the rotation and translation error of the predicted pose over the rollout, against PERSIST's camera model and a hold baseline that never moves. An identifiability test is defined in Appendix D.

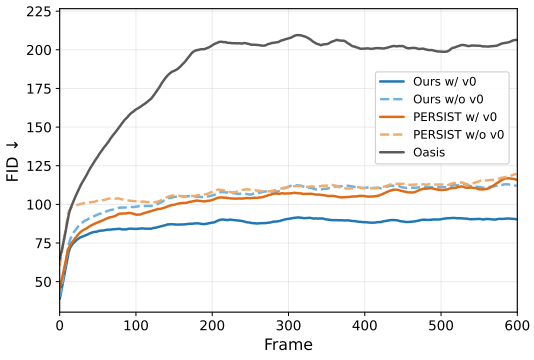


Figure 4: **FID versus rollout frame** (0–600) on MoT-eval for the models of Table 1. MoTWorld's curve stays flat over the whole rollout while the baselines climb.

4.3 Main results

Long-horizon video quality. Table 1 reports FVD at 200, 400 and 600 rollout frames on both held-out sets, Figure 4 the FID curve, and Figure 5 a qualitative rollout (another is in Appendix D,

Table 2: **Voxel from one frame.** Median over episodes, in percent, on PERSIST-eval (256 episodes) and MoT-eval (200), with no $\mathbf{v}_0$ and one sampler pass. IoU is split by the camera frustum with occlusion ignored, and class accuracy by occlusion-aware visibility. seen: the ray-hit surface plus the free space the rays crossed.

	PERSIST-eval				MoT-eval			
	IoU (%)		class acc. (%)		IoU (%)		class acc. (%)	
Method	in-frustum	out-frustum	seen	unseen	in-frustum	out-frustum	seen	unseen
PERSIST-XL, no $\mathbf{v}_0$	69.5	74.0	69.2	59.3	55.4	74.9	74.1	59.2
MoTWorld, P→V	**88.2**	**82.5**	**84.2**	**67.7**	**83.8**	**84.5**	**85.2**	**68.0**

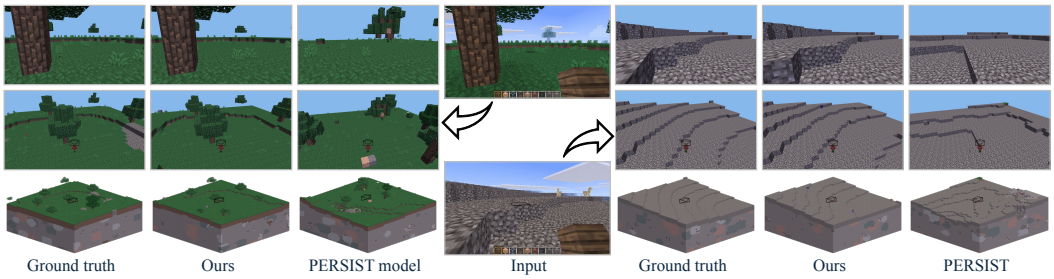


Figure 6: **Voxel world from one frame.** For two input frames (centre), we compare the ground-truth world, our reconstruction, and PERSIST's using rendered first-person views (top), oblique views with the camera marked (middle), and full 48^3 voxel grids (bottom).

Table 3: **Camera pose error in a free-running rollout.** Mean per-frame translation error (blocks) and rotation error (degrees) over the first 200 / 400 / 600 frames, 96 episodes per set. Every pose after frame 0 is predicted from the recorded actions. Hold never moves and never turns. Best in bold, second best underlined.

	PERSIST-eval		MoT-eval	
Method	transl. (blk)	rot. (°)	transl. (blk)	rot. (°)
PERSIST-XL + $\mathbf{v}_0$	**1.67 / 2.62 / 3.38**	<u>9.96 / 17.6 / 24.0</u>	**2.77** / <u>4.19</u> / <u>5.40</u>	<u>6.38 / 10.0 / 13.6</u>
MoTWorld	<u>1.78 / 2.80 / 3.65</u>	**3.40 / 4.19 / 4.81**	<u>2.79</u> / **4.17 / 5.36**	**0.86 / 1.53 / 1.99**

Figure S1). MoTWorld has the lowest FVD at every horizon, and its error stays flat over the rollout while every baseline degrades. A persistent voxel grid anchors the video, so errors in appearance do not compound as they do when each frame is predicted from a window of pixels. The gap to the cascade holds even when PERSIST is given the true initial world state. The P→P row isolates the cause: the same checkpoint with the voxel stream switched off roughly doubles its FVD at 600 frames. Appendix D covers the window split and inference cost, where the joint rollout is several times faster than the cascade at less than half of its memory.

3D voxels from one frame. Table 2 scores the voxel world each model builds from a single image and its pose. Figure 6 shows two examples. MoTWorld is more accurate than the cascade on both sets and in every column, with the largest margin inside the frustum, where the image is the evidence: about twenty IoU points above PERSIST. This is the direct effect of SpRoPE: image and voxel tokens share one metric frame, so attention tests whether a ray passes through a voxel, and the grid is written where the image says a surface is. Outside the frustum MoTWorld still leads, combining what it sees with a learned prior over world layout.

Camera pose. Table 3 scores the camera in a free-running rollout for the extension of Appendix C, where every pose after frame 0 is the model's own. On translation MoTWorld matches PERSIST's dedicated camera model. On rotation MoTWorld's heading stays within a few degrees of the truth over the whole rollout, while PERSIST's drifts to tens of degrees.

The two towers use each other. The identifiability test of Appendix D matches the correct video to its voxel grid well above chance, whereas our earlier architecture without a shared geometric code ignored the video. The attention is also readable geometrically: the metric term of Eq. (S3) scores

Table 4: **SpRoPE ablation on the reconstruction task.** Single-task P→V at window 1, scored at frame 0 of the 200 MoT-eval episodes. Columns are as in Table 2, higher is better, best in bold, second best underlined. The ladder is cumulative and R4 is the full model.

Id	depth head (Eq. 5)	projected (h, w) (Appendix A)	Gram anchoring	IoU (%) in-frustum	IoU (%) out-frustum	class acc. (%) seen	class acc. (%) unseen
R1				81.9	83.7	83.0	66.9
R2	✓			82.1	83.6	83.1	66.1
R3	✓	✓		<u>82.9</u>	**84.7**	<u>85.0</u>	<u>67.7</u>
R4	✓	✓	✓	**83.8**	<u>84.5</u>	**85.2**	**68.0**

Table 5: **Cascade versus joint denoising.** FVD (↓) on the 200 MoT-eval episodes over 600 frames at window $1 + 7$, one checkpoint throughout. The settings differ only in how each window is split into a voxel pass and a pixel pass. ϕ marks an absent voxel stream. Best in bold, second best underlined.

Setting	Voxel pass	Pixel pass	FVD@200	FVD@400	FVD@600
S0	PV→PV	same pass	**57.9**	**57.5**	**56.3**
S1	PV→V	PV→P	<u>72.2</u>	<u>72.8</u>	<u>71.4</u>
S2	PV→V	V→P	120.6	117.7	115.1
S3	V→V	V→P	122.0	119.1	115.4
S4	ϕ	P→P	103.9	119.2	128.3

a ray through a voxel above one that misses it before any training, and on the trained model the highest-scoring voxels for an image token lie along its ray near the first solid hit (Figure 3, right).

4.4 Ablations

SpRoPE ladder. Table 4 adds the components of SpRoPE one at a time on the reconstruction task: R1 averages each ray over its whole segment inside the volume, R2 adds the per-block depth head, R3 the projected image address, and R4 the Gram regulariser, the full model. The two columns that face the camera improve monotonically along the ladder, with the projected address the largest step, while the two that face away barely move: the encoding buys accuracy where the image is evidence.

Cascade versus joint denoising. Since the role matrix is chosen at inference, the trained joint model can run as a cascade of the kind PERSIST uses without retraining. Table 5 compares the joint rollout S0 with four settings: S1 generates voxels and then pixels in two passes, S2 and S3 render each window from voxels produced with or without the pixel history, and S4 switches the voxel tower off. The passes of S1 to S3 were never trained, so those settings run zero-shot. Three things follow. Joint denoising beats every cascade at every horizon, and since every setting that carries a grid produces the same voxel state, the whole difference is in the pixels. Carrying the voxel state at all is worth a factor of two in FVD, as S4 shows.

5 Conclusion

We propose MoTWorld, a Mixture-of-Transformer-based flow matching model for video, semantic voxels, and camera parameters that supports reconstruction, rendering, and rollout within one model. We introduced SpRoPE to encode image-voxel spatial relationships in a common metric frame while accounting for voxel extent and depth uncertainty. Experiments with known camera poses demonstrate improved single-image voxel reconstruction and long-horizon video quality over state-of-the-art methods, and our ablations show the benefits of joint denoising. Our evaluation currently focuses on Minecraft-style environments. Extending the framework to real-world scenes with uncertain geometry and camera poses is a direction for future work.

AI use statement

In this work, we used generative AI tools to help implement the code and edit the paper to improve readability. We have not used generative AI tools to generate synthetic datasets, help develop

theoretical models or conceptual frameworks, propose or refine hypotheses, or clean and reformat datasets. Formulating mathematical claims, providing critical ingredients for proving mathematical claims, assisting in the writing of proofs, designing or providing feedback on research methodology or experiments, assisting with translation, supporting qualitative and thematic data analysis, and interpreting results are not applicable to this work. We have reviewed all AI-assisted work: the AI-generated code for baseline implementations was unit-tested and manually verified by confirming that the trained baseline successfully replicated the qualitative behaviors and physical mechanics demonstrated in the original project's reference media. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

Ethics statement

All data used in this work is generated in an open-source game engine. No human subjects, personal data or real-world imagery are involved in training.

Reproducibility statement

The model, tokenizers, evaluation protocol and all hyperparameters are described in Sections 3 and 4 and in Appendices A and B. Code, training configurations and dataset settings will be released.

References

Niket Agarwal, Arslan Ali, Jon Allen, Martin Antolini, Adeline Aubame, Alisson Azzolini, Junjie Bai, Maciej Bala, Yogesh Balaji, Josh Bapst, et al. Cosmos 3: Omnimodal world models for physical ai. *arXiv preprint arXiv:2606.02800*, 2026.

Michael Albergo, Nicholas M Boffi, and Eric Vanden-Eijnden. Stochastic interpolants: A unifying framework for flows and diffusions. *Journal of Machine Learning Research*, 26(209):1–80, 2025.

Eloi Alonso, Adam Jelley, Vincent Micheli, Anssi Kanervisto, Amos Storkey, Tim Pearce, and François Fleuret. Diffusion for world modeling: Visual details matter in atari. *Advances in Neural Information Processing Systems*, 37:58757–58791, 2024.

Amir Bar, Gaoyue Zhou, Danny Tran, Trevor Darrell, and Yann LeCun. Navigation world models. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15791–15801. IEEE, 2025.

Hongzhe Bi, Hengkai Tan, Shenghao Xie, Zeyuan Wang, Shuhe Huang, Haitian Liu, Ruowen Zhao, Yao Feng, Chendong Xiang, Yinze Rong, et al. Motus: A unified latent action world model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 35101–35113, 2026.

Jake Bruce, Michael D Dennis, Ashley Edwards, Jack Parker-Holder, Yuge Shi, Edward Hughes, Matthew Lai, Aditi Mavalankar, Richie Steigerwald, Chris Apps, et al. Genie: Generative interactive environments. In *Forty-first international conference on machine learning*, 2024.

Haoxuan Che, Xuanhua He, Quande Liu, Cheng Jin, and Hao Chen. Gamegen-x: Interactive open-world game video generation. In *International Conference on Learning Representations*, volume 2025, pp. 37546–37593, 2025.

Boyuan Chen, Diego Martí Monsó, Yilun Du, Max Simchowitz, Russ Tedrake, and Vincent Sitzmann. Diffusion forcing: Next-token prediction meets full-sequence diffusion. *Advances in Neural Information Processing Systems*, 37:24081–24125, 2024.

Yixiang Dai, Fan Jiang, Chiyu Wang, Mu Xu, and Yonggang Qi. Fantasyworld: Geometry-consistent world modeling via unified video and 3d prediction. In *International Conference on Learning Representations*, volume 2026, pp. 103603–103622, 2026.

Etched Decart, Quinn McIntyre, Spruce Campbell, Xinlei Chen, and Robert Wachen. Oasis: A universe in a transformer. *URL: https://oasis-model. github. io*, 2(3):6, 2024.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first international conference on machine learning*, 2024.

Jianjie Fang, Yongyan Xu, Ziyou Wang, Chen Gao, Yuchao Huang, Zhaolu Wang, Rongze Tang, Mingyuan Jia, Baining Zhao, Weichen Zhang, et al. Worldscape-moe: A unified mixture-of-experts world model for scalable heterogeneous action control. *arXiv preprint arXiv:2607.03964*, 2026.

Ruili Feng, Han Zhang, Zhilei Shu, Zhantao Yang, Longxiang Tang, Zhicai Wang, Andy Zheng, Jie Xiao, Zhiheng Liu, Ruihang Chu, et al. The matrix: Infinite-horizon world generation with real-time moving control. *Advances in Neural Information Processing Systems*, 38:87318–87344, 2026.

Chong Gao, Jie Ma, Zhan Peng, Chongxiao Wang, Haoxue Wu, Jun Liang, Guanbin Li, and Jing Li. Moroute: Dynamic routing for in-context multimodal video generation. *arXiv preprint arXiv:2607.29545*, 2026.

Samuel Garcin, Thomas Walker, Steven McDonagh, Tim Pearce, Hakan Bilen, Tianyu He, Kaixin Wang, and Jiang Bian. Beyond pixel histories: World models with persistent 3d state. *arXiv preprint arXiv:2603.03482*, 2026.

Songen Gu, Wei Yin, Bu Jin, Xiaoyang Guo, Junming Wang, Haodong Li, Qian Zhang, and Xiaoxiao Long. Dome: Taming diffusion model into high-fidelity controllable occupancy world model. *arXiv preprint arXiv:2410.10429*, 2024.

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. *arXiv preprint arXiv:2504.08388*, 2025.

David Ha and Jürgen Schmidhuber. World models. *arXiv preprint arXiv:1803.10122*, 2(3):440, 2018.

Danijar Hafner, Wilson Yan, and Timothy Lillicrap. Training agents inside of scalable world models. *arXiv preprint arXiv:2509.24527*, 2025.

Yucheng Hu, Jianke Zhang, Yuanfei Luo, Yanjiang Guo, Xiaoyu Chen, Xinshu Sun, Kun Feng, Qingzhou Lu, Sheng Chen, Yangang Zhang, et al. Bagelvla: Enhancing long-horizon manipulation via interleaved vision-language-action generation. *arXiv preprint arXiv:2602.09849*, 2026.

Junchao Huang, Xinting Hu, Boyao Han, Shaoshuai Shi, Zhuotao Tian, Tianyu He, and Li Jiang. Memory forcing: Spatio-temporal memory for consistent scene generation on minecraft. *arXiv preprint arXiv:2510.03198*, 2025a.

Tianyu Huang, Wangguandong Zheng, Tengfei Wang, Yuhao Liu, Zhenwei Wang, Junta Wu, Jie Jiang, Hui Li, Rynson Lau, Wangmeng Zuo, et al. Voyager: Long-range and world-consistent video diffusion for explorable 3d scene generation. *ACM Transactions on Graphics (TOG)*, 44(6): 1–15, 2025b.

Xun Huang, Zhengqi Li, Guande He, Mingyuan Zhou, and Eli Shechtman. Self forcing: Bridging the train-test gap in autoregressive video diffusion. *Advances in Neural Information Processing Systems*, 38:167283–167308, 2026a.

Yuhang Huang, Xuan Lv, Junyan Xu, Zhiyuan Yu, Jiazhao Zhang, Ruizhen Hu, Wancheng Feng, Shilong Zou, Hewen Xiao, Ziqiao Zhou, et al. Paiworld: A 3d-consistent world foundation model for robotic manipulation. *arXiv preprint arXiv:2606.18375*, 2026b.

Achin Jain, Jie An, Siddharth Chaudhary, and Davide Modolo. Mural: Transferring llm knowledge to image generation via mixture-of-transformers. *arXiv preprint arXiv:2606.29013*, 2026.

Keller Jordan, Yuchen Jin, Vlado Boza, Jiacheng You, Franz Cesista, Laker Newhouse, and Jeremy Bernstein. Muon: An optimizer for hidden layers in neural networks, 2024. *URL https://kellerjordan. github. io/posts/muon*, 6(3):4, 2024.

Yash Kant, Aliaksandr Siarohin, Ziyi Wu, Michael Vasilkovsky, Guocheng Qian, Jian Ren, Riza Alp Guler, Bernard Ghanem, Sergey Tulyakov, and Igor Gilitschenski. Spad: Spatially aware multi-view diffusers. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10026–10038. IEEE, 2024.

Bohan Li, Jiazhe Guo, Hongsi Liu, Yingshuang Zou, Yikang Ding, Xiwu Chen, Hu Zhu, Feiyang Tan, Chi Zhang, Tiancai Wang, et al. Uniscene: Unified occupancy-centric driving scene generation. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11971–11981. IEEE, 2025.

Bohan Li, Xin Jin, Hu Zhu, Hongsi Liu, Ruikai Li, Jiazhe Guo, Kaiwen Cai, Chao Ma, Yueming Jin, Hao Zhao, et al. Scaling up occupancy-centric driving scene generation: Dataset and method. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2026a.

Lin Li, Qihang Zhang, Yiming Luo, Shuai Yang, Ruilin Wang, Fei Han, Mingrui Yu, Zelin Gao, Nan Xue, Xing Zhu, et al. Causal world modeling for robot control. *arXiv preprint arXiv:2601.21998*, 2026b.

Ruilong Li, Brent Yi, Junchen Liu, Hang Gao, Yi Ma, and Angjoo Kanazawa. Cameras as relative positional encoding. *Advances in Neural Information Processing Systems*, 38:15984–16009, 2026c.

Zhiqi Li, Wenhai Wang, Hongyang Li, Enze Xie, Chonghao Sima, Tong Lu, Qiao Yu, and Jifeng Dai. Bevformer: learning bird's-eye-view representation from lidar-camera via spatiotemporal transformers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(3):2020–2036, 2024.

Weixin Liang, Lili Yu, Liang Luo, Srinivasan Iyer, Ning Dong, Chunting Zhou, Gargi Ghosh, Mike Lewis, Wen-tau Yih, Luke Zettlemoyer, et al. Mixture-of-transformers: A sparse and scalable architecture for multi-modal foundation models. *arXiv preprint arXiv:2411.04996*, 2024.

Kang Liao, Yihang Luo, Xiao-Ming Wu, Linyi Jin, Size Wu, Chunyu Lin, Yao Zhao, Fei Wang, Wei Li, and Chen Change Loy. Puffin-world: Scaling a unified multimodal model with native 3d world states. *arXiv preprint arXiv:2609.04196*, 2026.

Jingyuan Liu, Jianlin Su, Xingcheng Yao, Zhejun Jiang, Guokun Lai, Yulun Du, Yidao Qin, Weixin Xu, Enzhe Lu, Junjie Yan, et al. Muon is scalable for llm training. *arXiv preprint arXiv:2502.16982*, 2025.

Yiren Lu, Xin Ye, Jiaming Liu, Philip Jacobson, Jin Yao, Yi-chung Chen, Liam Merino, Dhruva Dixith Kurra, Min Cai, Tom Lampo, et al. Geowam: Visual geometry world action models for autonomous driving. *arXiv preprint arXiv:2608.23486*, 2026.

Junyi Ma, Xieyuanli Chen, Jiawei Huang, Jingyi Xu, Zhen Luo, Jintao Xu, Weihao Gu, Rui Ai, and Hesheng Wang. Cam4docc: Benchmark for camera-only 4d occupancy forecasting in autonomous driving applications. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21486–21495. IEEE, 2024.

Mikel Malagón, Josu Ceberio, and Jose A Lozano. Craftium: An extensible framework for creating reinforcement learning environments. *arXiv preprint arXiv:2407.03969*, 2024.

Tim Merino, Sam Earle, Ryunosuke Iwai, Julian Togelius, and Edoardo Cetin. Dream-cubed: Controllable generative modeling in minecraft by training on billions of cubes. *arXiv preprint arXiv:2604.22847*, 2026.

Takeru Miyato, Bernhard Jaeger, Max Welling, and Andreas Geiger. Gta: A geometry-aware attention mechanism for multi-view transformers. In *International Conference on Learning Representations*, volume 2024, pp. 8172–8208, 2024.

Sicheng Mo, Yuheng Li, Ziyang Leng, Krishna Kumar Singh, and Bolei Zhou. Streaming multi-agent autoregressive diffusion model with world state registers. *arXiv preprint arXiv:2607.21594*, 2026.

Jack Parker-Holder, Shlomi Fruchter, et al. Genie 3: A new frontier for world models. *Google DeepMind Blog*, 4, 2025.

Ryan Po, Yotam Nitzan, Richard Zhang, Berlin Chen, Tri Dao, Eli Shechtman, Gordon Wetzstein, and Xun Huang. Long-context state-space video world models. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 8733–8744. IEEE, 2025.

Xuanchi Ren, Tianchang Shen, Jiahui Huang, Huan Ling, Yifan Lu, Merlin Nimier-David, Thomas Müller, Alexander Keller, Sanja Fidler, and Jun Gao. Gen3c: 3d-informed world-consistent video generation with precise camera control. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6121–6132. IEEE, 2025.

Kiwhan Song, Boyuan Chen, Max Simchowitz, Yilun Du, Russ Tedrake, and Vincent Sitzmann. History-guided video diffusion. *arXiv preprint arXiv:2502.06764*, 2025.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

MotuBrain Team, Chendong Xiang, Fan Bao, Haitian Liu, Hengkai Tan, Hongzhe Bi, James Li, Jiabao Liu, Jingrui Pang, Kiro Jing, et al. Motubrain: An advanced world action model for robot control. *arXiv preprint arXiv:2604.27792*, 2026.

Dani Valevski, Yaniv Leviathan, Moab Arar, and Shlomi Fruchter. Diffusion models are real-time game engines. In *International Conference on Learning Representations*, volume 2025, pp. 73754–73776, 2025.

Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, et al. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025.

Jing Wang, Fengzhuo Zhang, Xiaoli Li, Vincent YF Tan, Tianyu Pang, Chao Du, Aixin Sun, and Zhuoran Yang. Error analyses of auto-regressive video diffusion models: A unified framework. *arXiv preprint arXiv:2503.10704*, 2025.

Lening Wang, Wenzhao Zheng, Yilong Ren, Han Jiang, Zhiyong Cui, Haiyang Yu, and Jiwen Lu. Occsora: 4d occupancy generation models as world simulators for autonomous driving. *IEEE Transactions on Image Processing*, 2026a.

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, et al. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024.

Zile Wang, Zexiang Liu, Jiaxing Li, Kaichen Huang, Baixin Xu, Fei Kang, Mengyin An, Peiyu Wang, Biao Jiang, Yichen Wei, et al. Matrix-game 3.0: Real-time and streaming interactive world model with long-horizon memory. *arXiv preprint arXiv:2604.08995*, 2026b.

Julong Wei, Shanshuai Yuan, Pengfei Li, Qingda Hu, Zhongxue Gan, and Wenchao Ding. Occllama: An occupancy-language-action generative world model for autonomous driving. *arXiv preprint arXiv:2409.03272*, 2024.

Xilin Wei, Xiaoran Liu, Yuhang Zang, Xiaoyi Dong, Pan Zhang, Yuhang Cao, Jian Tong, Haodong Duan, Qipeng Guo, Jiaqi Wang, et al. Videorope: What makes for good video rotary position embedding? *arXiv preprint arXiv:2502.05173*, 2025.

Haoyu Wu, Jiwen Yu, Yingtian Zou, and Xihui Liu. Multiworld: Scalable multi-agent multi-view video world models. *arXiv preprint arXiv:2604.18564*, 2026a.

Xindi Wu, Sven Elflein, James Lucas, Olga Russakovsky, Laura Leal-Taixé, Despoina Paschalidou, Jonathan Lorraine, and Aljoša Ošep. Addressable memory for video world models. *arXiv preprint arXiv:2608.07408*, 2026b.

Yu Wu, Minsik Jeon, Jen-Hao Rick Chang, Oncel Tuzel, and Shubham Tulsiani. Rayrope: Projective ray positional encoding for multi-view attention. In *European Conference on Computer Vision*, pp. 192–210. Springer, 2026c.

Zeqi Xiao, Yushi Lan, Yifan Zhou, Wenqi Ouyang, Shuai Yang, Yanhong Zeng, and Xingang Pan. Worldmem: Long-term consistent world simulation with memory. *Advances in Neural Information Processing Systems*, 38:49632–49652, 2026.

Zhexiao Xiong, Yizhi Song, Hao Kang, Qing Yan, Liming Jiang, Jenson Yang, Zhoujie Fu, Stathi Fotiadis, Angtian Wang, Zichuan Liu, et al. Actworld: From explorable to interactive world model via action-aware memory. *arXiv preprint arXiv:2606.17730*, 2026.

Haoran Xu, Peixi Peng, Guang Tan, Yiqian Chang, Yisen Zhao, and Yonghong Tian. Delta-triplane transformers as occupancy world models. *arXiv preprint arXiv:2503.07338*, 2025.

Ziyang Yan, Wenzhen Dong, Yihua Shao, Yuhang Lu, Haiyang Liu, Jingwen Liu, Haozhe Wang, Zhe Wang, Yan Wang, Fabio Remondino, et al. Renderworld: World model with self-supervised 3d label. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6063–6070. IEEE, 2025.

Xiuyu Yang, Bohan Li, Shaocong Xu, Nan Wang, Chongjie Ye, Zhaoxi Chen, Minghan Qin, Yikang Ding, Zheng Zhu, Xin Jin, Hang Zhao, and Hao Zhao. Orv: 4d occupancy-centric robot video generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1053–1066, June 2026a.

Yu Yang, Jianbiao Mei, Yukai Ma, Siliang Du, Wenqing Chen, Yijie Qian, Yuxiang Feng, and Yong Liu. Driving in the occupancy world: Vision-centric 4d occupancy forecasting and planning via world models for autonomous driving. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 9327–9335, 2025a.

Zhenya Yang, Zhe Liu, Yuxiang Lu, Liping Hou, Chenxuan Miao, Siyi Peng, Bailan Feng, Xiang Bai, and Hengshuang Zhao. Geniedrive: Towards physics-aware driving world model with 4d occupancy guided video generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 35680–35690, 2026b.

Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, et al. Cogvideox: Text-to-video diffusion models with an expert transformer. In *International Conference on Learning Representations*, volume 2025, pp. 83048–83077, 2025b.

Jiwen Yu, Yiran Qin, Haoxuan Che, Quande Liu, Xintao Wang, Pengfei Wan, Di Zhang, Kun Gai, Hao Chen, and Xihui Liu. A survey of interactive generative video. *arXiv preprint arXiv:2504.21853*, 2025a.

Jiwen Yu, Yiran Qin, Xintao Wang, Pengfei Wan, Di Zhang, and Xihui Liu. Gamefactorly: Creating new games with generative interactive videos. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 11590–11599. IEEE, 2025b.

Shenghai Yuan, Yuanyang Yin, Zongjian Li, Xinwei Huang, Xiao Yang, and Li Yuan. Helios: Real real-time long video generation model. *arXiv preprint arXiv:2603.04379*, 2026.

Haiming Zhang, Ying Xue, Xu Yan, Jiacheng Zhang, Weichao Qiu, Dongfeng Bai, Bingbing Liu, Shuguang Cui, and Zhen Li. An efficient occupancy world model via decoupled dynamic flow and image-assisted training. *arXiv preprint arXiv:2412.13772*, 2024.

Maeve Zhang, Rain Sun, Xiang Wang, Cyril Zhang, Shalfun Li, Meng Cao, Howard Lu, Ethan Chen, Harry Jhou, KZ Zheng, et al. Wall-ss: Scaling long-horizon world models via next-scale autoregression. *arXiv preprint arXiv:2608.26239*, 2026a.

Songchun Zhang, Yaowei Li, Junhao Zhuang, Weiyang Jin, Haoyu Wang, Xin Lu, Yilang Sun, Shiyi Zhang, Haoran Li, Xiaoxiao Ma, et al. Echowm: Open and enterable omnimodal world models. *arXiv preprint arXiv:2608.23189*, 2026b.

Wenzhao Zheng, Weiliang Chen, Yuanhui Huang, Borui Zhang, Yueqi Duan, and Jiwen Lu. Occworld: Learning a 3d occupancy world model for autonomous driving. In *European conference on computer vision*, pp. 55–72. Springer, 2024.

Chunting Zhou, Lili Yu, Arun Babu, Kushal Tirumala, Michihiro Yasunaga, Leonid Shamis, Jacob Kahn, Xuezhe Ma, Luke Zettlemoyer, and Omer Levy. Transfusion: Predict the next token and diffuse images with one multi-modal model. In *International Conference on Learning Representations*, volume 2025, pp. 6446–6469, 2025.

Siyuan Zhou, Yilun Du, Yuncong Yang, Lei Han, Peihao Chen, Dit-Yan Yeung, and Chuang Gan. Learning 3d persistent embodied world models. *Advances in Neural Information Processing Systems*, 38:104807–104829, 2026.

Sicheng Zuo, Wenzhao Zheng, Yuanhui Huang, Jie Zhou, and Jiwen Lu. Gaussianworld: Gaussian world model for streaming 3d occupancy prediction. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6772–6781. IEEE, 2025.

Table of Contents in Appendix

A Details of SpRoPE **17**

B Implementation details **18**

B.1 Details of the model framework 18

B.2 Training protocol 19

C The camera pose tower **20**

D Experimental details and additional results **21**

D.1 Data Curation 21

D.2 Training and evaluation details 21

D.3 Model Analysis 22

A Details of SpRoPE

This appendix gives the derivations, the frequency ladders and the two mechanisms that Section 3.3 states in words, and closes with the nine modality pairings. Symbols are those of Section 3.3. The token sequence has N rows, tower m owns M_m of them, and the logit matrix of one head is $\mathbf{A} = [\mathcal{A}_{ij}] \in \mathbb{R}^{N \times N}$.

Three operations: rotate, attenuate, dot. For every token i, group g and pair k the query and key slices are transformed as

$$\tilde{\mathbf{q}}_i^{g,k} = \beta_{i,k}^g \, \mathbf{R}(\varphi_{i,k}^g) \, \mathbf{q}_i^{g,k}, \qquad \tilde{\mathbf{k}}_j^{g,k} = \beta_{j,k}^g \, \mathbf{R}(\varphi_{j,k}^g) \, \mathbf{k}_j^{g,k}, \tag{S1}$$

and values are not rotated. Since $\mathbf{R}(\theta_i)^\top \mathbf{R}(\theta_j) = \mathbf{R}(\theta_j - \theta_i)$, the dot product of the transformed slices is the summand of Eq. (3). Three consequences follow. A group can only express a relative position, so every group must be built in one shared coordinate system. Attenuation multiplies on both sides and is never renormalised, so $\beta = 0$ removes a pair from every logit of that token. And a token has one phase per group, indexed by the token and not by the pair (i, j). Anything that depends on which key is asking cannot be written as Eq. (S1), hence the block-wise evaluation below.

The averaged codes. A voxel token is a cube of width W around $\mathbf{b}_n$. Averaging the rotation uniformly over the cube on axis a,

$$\mathbb{E}_{x \sim \mathcal{U}[b_{n,a} - W/2,\, b_{n,a} + W/2]} \big[\mathbf{R}(\omega_k x) \big] = \mathbf{R}(\omega_k b_{n,a}) \, \mathrm{sinc}(\omega_k W / 2), \tag{S2}$$

which is the voxel line of Eq. (4). The image line is the same integral taken along the ray $\mathbf{o}_i + d\,\hat{\mathbf{r}}_i$ over $d \in [d_i^{\mathrm{lo}}, d_i^{\mathrm{hi}}]$, and it coincides with the closed form that RayRoPE (Wu et al., 2026c) derives for the expectation of a rotary code over an interval. Cell centres are shifted by the integer re-centring $\boldsymbol{\xi}_\tau$ of the agent-centred grid so that voxels of different frames of a window share one frame. Most windows contain at least one re-centring.

The voxel-image metric term. Collecting the metric pairs of one axis, the contribution of axis a to the logit between voxel n and image token i is

$$\mathcal{S}_{ni}^a = \sum_k \mathrm{sinc}\Big(\tfrac{\omega_k W}{2}\Big) \, \mathrm{sinc}\Big(\tfrac{\omega_k \hat{r}_{i,a} \Delta_i}{2}\Big) \, \big(\mathbf{q}_n^{a,k}\big)^\top \mathbf{R}\Big(\omega_k \big(o_{i,a} + d_i^{\mathrm{mid}} \hat{r}_{i,a} - b_{n,a}\big)\Big) \mathbf{k}_i^{a,k}, \tag{S3}$$

the two sinc factors accounting for the cell extent and the depth interval. Three cases arise for any pair of tokens on any group: when both sides carry the identity the group contributes pure content matching. When both are rotated it contributes a relative-position term. When one side is identity and the other rotated, the identity side reads the other's absolute phase. Table S1 at the end of this appendix gives the case for each of the nine modality pairings on each group.

Design notes. Four choices behind Section 3.3 deserve a word. On the time group all three modalities carry the same clock, so that a voxel and an image token of one frame agree on *when* before they can agree on *where*. On the image-plane group only the image tokens carry their own lattice address, since a two-dimensional address between two voxels would compare them by how far apart they *appear*, which conflates depth. We do not renormalise the attenuated code, since a renormalised code would assert an arbitrary position with full confidence. And we average over the depth interval rather than committing to a single depth: the attenuation is anisotropic, a ray travelling mostly along one axis is uncertain along that axis and sharp on the other two, and it is selective, a wide interval kills only the frequencies whose wavelength is shorter than the interval, whereas hard-assigning a wrong depth would bake a false correspondence into every logit of the token. The depth head is per block because early features know little about depth and deep features know a great deal, so a depth re-chosen per block lets the geometry sharpen exactly as fast as the representation does.

The depth-head loss. The head of Eq. (5) is trained with a heteroscedastic negative log-likelihood in log depth against the first-hit depth obtained by ray-marching the ground-truth voxel grid, with a one-sided censored term for the rays that leave the volume without hitting a surface, about half of them: those rays are pushed to have depth beyond the far end of their segment inside the volume and are free to grow their spread. The centre of the interval is the head's own point estimate rather than the midpoint of the interval, which for a log-symmetric interval is $\cosh(\eta_i)$ times it. The head is initialised to the dataset prior of depth and spread (Appendix B). Only the image tokens' metric phases are recomputed per block. Everything else in the geometry is built once per forward.

The borrowed image address and its visibility gate. A voxel's centre is projected through the camera of frame τ' onto the lattice, $(h_{n,\tau'}, w_{n,\tau'}) = \pi_{\tau'}(\mathbf{b}_n + \boldsymbol{\xi}_\tau)$, evaluated on the image-plane ladders, and attenuated by

$$\alpha_{n,\tau'} = \begin{cases} \exp\big(-(\chi_{n,\tau'}/\chi_0)^2\big) & \text{in front of the camera} \\ 0 & \text{behind the camera,} \end{cases} \tag{S4}$$

where $\chi_{n,\tau'}$ is the distance in lattice cells by which the projection falls outside the frame and χ_0 the roll-off width. A per-block learned scalar per ordered modality pair, initialised to one, scales the image-plane attenuation on the query side so that a systematically larger geometric term in one pairing cannot steal attention mass from the others. Because attenuation is per token and shared across all keys, a voxel behind the eye would otherwise have its whole image-plane share of the head contribute zero to every logit, including toward other voxels. Voxel-voxel pairs therefore keep the identity on the image-plane group.

The nine modality pairs. With $\mathcal{S}_g$ the per-group sum in Eq. (3), $\mathcal{A}_{ij} \propto \mathcal{S}_t + \mathcal{S}_u + \mathcal{S}_v + \mathcal{S}_x + \mathcal{S}_y + \mathcal{S}_z$. Which terms carry information depends on the pair (Table S1).

Table S1: **The nine modality pairs.** The contribution of each rotary group to the logit for every modality pair.

Pair	t	h, w	x, y, z
V ↔ V	frame gap	identity both sides, hence content	3D displacement, sinc^2 (cell × cell)
P ↔ P, same frame	constant	exact 2D offset on the lattice	3D, sinc × sinc (depth × depth)
P ↔ P, across frames	frame gap	lattice address	3D, camera motion cancels
V ↔ P	frame gap	borrowed $\pi_{\tau'}(\mathbf{b}_n) \times \alpha_{n,\tau'}$ (block-wise)	3D, sinc(cell) × sinc(depth), Eq. (S3)
V ↔ C	frame gap	identity both sides, hence content	3D against the camera centre
P ↔ C	frame gap	image rotated, camera identity	3D against the camera centre
C ↔ C	frame gap	identity both sides, hence content	ego motion

In one sentence: t is the shared clock, (h, w) is the image's private 2D address, and (x, y, z) is the one coordinate system all three modalities can meet in, which is why every cross-modal relation that works is a metric one, and every attempt to make a 2D address cross-modal runs into the per-token limit of Section 3.3.

B Implementation Details

B.1 Details of the Model Framework

Per-frame conditioning. Each tower receives, in every block, a per-frame vector

$$\mathbf{y}_\tau^m = \mathrm{TE}_m(t_\tau^m) + \mathrm{AE}(\mathbf{a}_\tau) + \mathrm{RE}_m(r_\tau^m), \tag{S5}$$

where TE_m is a sinusoidal noise-level embedding, one per tower so that the camera's level may mean something different from the voxel's. AE is a two-layer MLP over the action, shared by all towers because the agent's action is a per-frame fact every stream is entitled to, and RE_m is a zero-initialised embedding of the role. The action of the last frame of a window points past the window and is replaced by a learned absent vector. Nothing about the camera enters the conditioning: the pose reaches the model only as the camera token and through the positional encoding.

One block. With adaptive-layer-norm vectors $\gamma_0..\gamma_5$ computed from $\mathbf{y}_\tau^m$ per frame and broadcast over that frame's tokens, tower m computes

$$\begin{aligned} \bar{\mathbf{h}}^m &= \mathrm{LN}(\mathbf{h}^m) \odot (1+\gamma_1) + \gamma_0, \\ [\mathbf{q}, \mathbf{k}, \boldsymbol{\nu}] &= \mathrm{QKV}_m(\bar{\mathbf{h}}^m), \qquad \mathbf{q} \leftarrow \mathrm{RMSNorm}(\mathbf{q}), \quad \mathbf{k} \leftarrow \mathrm{RMSNorm}(\mathbf{k}), \\ \mathbf{h}^m &\leftarrow \mathbf{h}^m + \gamma_2 \odot \mathrm{proj}_m\big(\mathrm{Attn}(\mathrm{SpRoPE}(\mathbf{q}^m); \{\mathrm{SpRoPE}(\mathbf{k}^{m'}), \boldsymbol{\nu}^{m'}\}_{m' \in \mathcal{M}})\big), \\ \tilde{\mathbf{h}}^m &= \mathrm{LN}(\mathbf{h}^m) \odot (1+\gamma_4) + \gamma_3, \\ \mathbf{h}^m &\leftarrow \mathbf{h}^m + \gamma_5 \odot \mathrm{MLP}_m(\tilde{\mathbf{h}}^m), \end{aligned} \tag{S6}$$

with affine-free layer norms, zero-initialised adaptive-norm projections and output heads so that every block is the identity at initialisation, and a GELU feed-forward network of four times the width. A shared final norm and per-tower linear heads then un-patchify to each latent shape.

B.2 TRAINING PROTOCOL

Training design details. The reconstruction tasks use a window of one frame, and the next-frame tasks a window of T frames whose number of context frames is drawn per step. Each rank draws the same task per step, since the tasks differ by an order of magnitude in cost and an unsynchronised draw makes every all-reduce wait for the slowest rank. The pose head mean-pools a frame's image tokens and regresses the camera together with three log-spreads under a heteroscedastic loss in geodesic angle, centre distance and field of view. Pooling is motivated by a probe in which a single token recovers rotation poorly and the pooled frame recovers it well. Optimiser, schedule and hardware are given in Appendix D. Table S2 lists every constant.

History corruption. A model that only ever sees exact history learns to copy it and carries its own errors forward. We therefore corrupt the observed image frames of a window with the kinds of error a rollout produces, capped so that no frame is corrupted harder than the model meets at inference. A corrupted frame keeps its clean noise label and receives no loss, so the model learns to cope with a degraded history without being told about it. Each observed image frame of a window draws one operation with the weights in Table S2. The operations are leaving the latent clean, blending in Gaussian noise as $\mathbf{x} \leftarrow s\,\boldsymbol{\varepsilon} + (1-s)\,\mathbf{x}$ with $s \sim \mathcal{U}(0, 0.6)$, rescaling contrast about the per-position channel mean by $f \sim \mathcal{U}(1, 2)$, ghosting the frame by blending it with a clean earlier frame of the window as $(1-a)\,\mathbf{x}_\tau + a\,\mathbf{x}_{\tau-j}$ with $a \sim \mathcal{U}(0.3, 0.8)$ and lag $j \le 2$, blurring it with a Gaussian of width $\sigma \sim \mathcal{U}(0.3, 0.9)$, and translating it by a sub-pixel offset of 0.1 to 0.5 latent pixels drawn per axis. With probability 0.5, noise with $s \sim \mathcal{U}(0, 0.4)$ is stacked on top of a structured operation, since a real rollout error is lag, speckle and contrast at once. Every corrupted frame is then shrunk back toward its clean value until its root-mean-square change is at most 1.0, the 90th percentile of the measured early-rollout error. On half of the corrupted samples the same draw, operation, strength and noise pattern, is applied to every history frame. Static windows, whose ground-truth latent changes by less than 0.3 into the predicted frame and which the model would otherwise solve by copying, are always corrupted and, with probability 0.5, have their history replaced by pure noise. A further 5% of the remaining samples are blanked as well. 10% of samples keep a fully clean history. The voxel history receives only the noise operation, with $s \sim \mathcal{U}(0, 1/3)$.

Constants and hyperparameters. Table S2 lists every constant of the reported runs.

Table S2: **Constants and hyperparameters.** Values of every constant in the reported runs.

Symbol	Value
G (grid extent, blocks)	48
$H_\mathrm{I} \times W_\mathrm{I}$ (image)	360×640
d_a (action bits)	23
C_V, G_ℓ (voxel latent)	48, 12
C_P, $H_\ell \times W_\ell$ (image latent)	16, 36×64
p_V, p_P (patch sizes)	2, 2
$N_h \times N_w$ (image lattice)	18×32
N_V, N_P, N_C (tokens per frame)	216, 576, 1
W (blocks per voxel token)	8
D, L, H, D_h	1024, 12, 16, 64
parameters	688 M
P_g for $g = t, h, w, x, y, z$	8, 6, 6, 4, 4, 4 (16, 12, 12, 8, 8, 8 channels)
$\lambda^{\min}_{x,y,z}$	$2.5W = 20$ blocks
$\sigma_{\min}$	10^{-5}
corruption weights (clean, noise, contrast, ghost, blur, shift)	0.10, 0.25, 0.15, 0.25, 0.10, 0.15
clean-history probability	0.1
π_{task} (P→V, V→P, PV→PV, P→P, V→V)	0.2, 0.3, 0.3, 0.1, 0.1
λ_V, λ_P, λ_C	1, 1.29, 1
λ_{depth}, λ_{pose}, λ_{gram}, λ_{nce}	0.1, 0.1, 0.05, 0.05
Ψ (Euler steps)	16

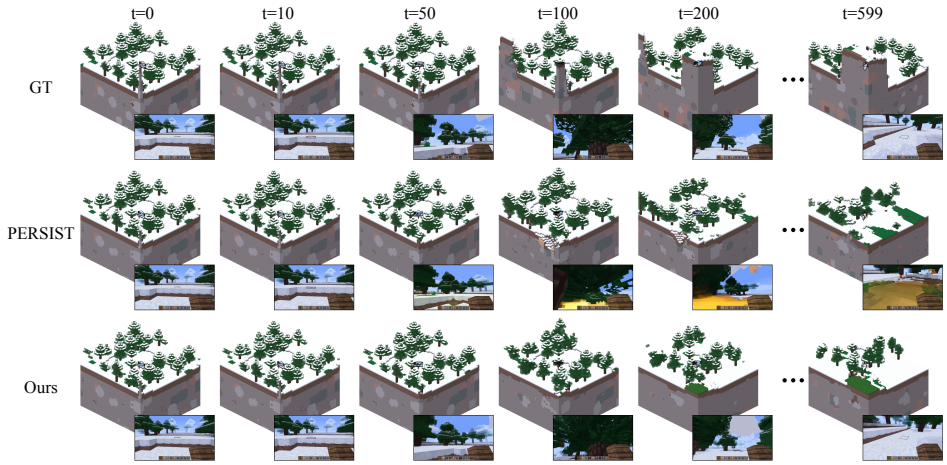


Figure S1: **Additional rollout results.** A second 600-frame rollout on MoT-eval in a snowy biome, with the carried voxel world and the generated frame (inset). Ours keeps the snow cover, the trees and the frame content close to the ground truth over the whole rollout, whereas PERSIST's frames show strong colour artifacts from $t = 100$ and its voxel world turns large areas into the wrong block class by the end.

C THE CAMERA POSE TOWER

Every construction of Section 3.3 needs a camera for each frame: rays for the image tokens, the projection for the borrowed address, and the centre for the camera token. In a multi-view reconstruction setting that camera is an input, which is what RayRoPE assumes. In a world model it need not be: the camera of a frame to be generated is precisely what the model could produce from the agent's actions, and handing the encoding the true future pose during training tells the model where the camera will be while asking it what it will see. We measured that closing this leak tripled the camera stream's one-step rotation error on the same checkpoint. Three pieces make the geometry pose-free.

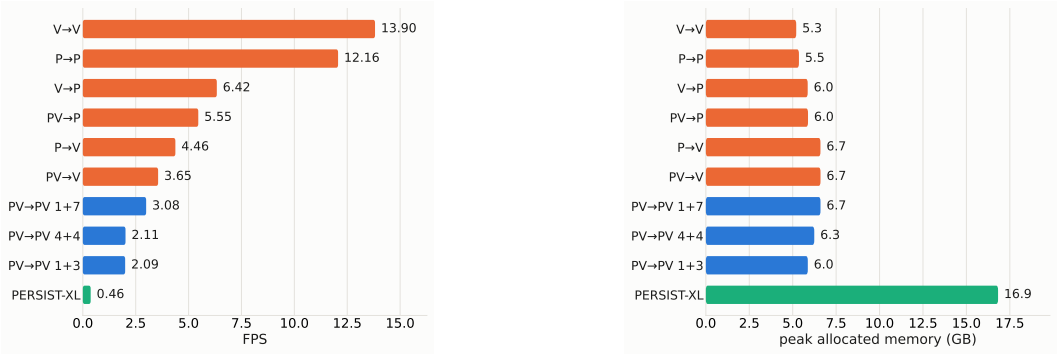
The camera as a flow-matching stream. The camera is the third modality of Eq. (1), one token per frame, and on the next-frame tasks the camera of the predicted frame is generated: the true pose is withheld and the tower must infer it from the observed frames and the action.

Geometry from the stream's own estimate. For a generated frame, the pose that defines its rays, projections and centre is the tower's own current estimate. Let $\hat{\mathbf{z}}^{\mathrm{C}}_{\tau,\psi}$ be the clean-pose estimate implied by the velocity the tower returned at sampler step ψ, with $\hat{\mathbf{z}}^{\mathrm{C}}_{\tau,0}$ the source of the flow. At step ψ the geometry of a generated frame is built from $\tilde{\mathbf{c}}_\tau = \boldsymbol{\sigma}_{\mathrm{C}} \odot \hat{\mathbf{z}}^{\mathrm{C}}_{\tau,\psi-1} + \boldsymbol{\mu}_{\mathrm{C}}$, the de-standardised estimate of the previous step, while an observed frame keeps its true pose. The estimate lags the sampler by one step by construction, and in training, where there is no sampler, the geometry pose of a generated frame is the diffused true pose, so the geometry the model sees is as sharp as its camera token.

The camera bridge. A flow from Gaussian noise starts from a random pose, nearly a right angle from the truth in the median. Rather than patch inference we change the source of the camera flow, following the stochastic-interpolant view that both endpoints of a flow may be arbitrary and dependent distributions (Albergo et al., 2025). Let $\mathbf{z}^{\mathrm{C}}_{\mathrm{anc}}$ be the standardised camera of the last observed frame of the window, after its own history corruption. For generated camera frames

$$\mathbf{x}^{\mathrm{C}}_\tau(t) = (1-t)\,\mathbf{z}^{\mathrm{C}}_\tau + t\,\mathbf{z}^{\mathrm{C}}_{\mathrm{anc}}, \qquad \dot{\mathbf{x}}^{\mathrm{C}}_\tau = \mathbf{z}^{\mathrm{C}}_{\mathrm{anc}} - \mathbf{z}^{\mathrm{C}}_\tau, \tag{S7}$$

so the first sampler step starts from the last known pose, training and sampling share one source, and the construction is stride-independent since the velocity scales with the displacement.



(a) Throughput (frames per second, higher is better)

(b) Peak allocated memory (GB, lower is better)

Figure S2: **Inference cost on one H200 GPU.** Every MoTWorld row is the same checkpoint run under a different role matrix: single-stream and reconstruction tasks (orange), the joint rollout PV→PV at three window splits (blue), and the released PERSIST-XL cascade (green). MoTWorld's joint rollout is about seven times faster than the cascade at less than half of its memory, and its single-modality tasks are cheaper still because absent modalities contribute no tokens.

D Experimental details and additional results

D.1 Data Curation

Simulator settings. Craftium exposes the Luanti engine (formerly Minetest) through a Gymnasium interface. The observation consists of the 640×360 rendered frame with a vertical field of view of 72° (horizontal 104.5°), the 48^3 block grid centred on the agent, exact from the engine, with $K = 2138$ classes (one per distinct block type and orientation), the camera, ten-dimensional as in Section 3.1 and expressed in the grid's frame, and the 23-bit multi-hot action (forward, backward, left, right, jump, sprint, sneak, dig, place, drop, nine hotbar slots, and quantised yaw and pitch steps of ± 0.1). Episodes are 600 frames at 24 fps after 200 discarded warm-up frames during which the terrain emerges, in creative mode with no damage, noon lighting, flat per-face lighting, fog and a 30-block viewing range.

Data pipeline. MoT-train is generated by a navigation policy that mixes straight runs, turns, arcs, strafes and reversals so that the agent keeps moving into unexplored space. MoT-eval holds out 200 further episodes from fresh seeds. PERSIST-eval consists of the 256 evaluation trajectories released with PERSIST, generated by its authors with their own action sampler. We tokenise frames, voxel grids and cameras offline with PERSIST's frozen autoencoders and pre-compute the first-hit depth of every image token by ray-marching the ground-truth grid.

D.2 Training and evaluation details

Training details. The model has $D = 1024$, $L = 12$, $H = 16$ and $D_h = 64$, with 18.9 M parameters per tower per block. Training uses 32 H200 GPUs on 4 nodes, bf16 autocast, batch 12 per GPU, the Muon optimiser with AdamW-matched update RMS for the two-dimensional weights and AdamW for the rest, a peak learning rate of 3×10^{-4} with 1,000 warm-up steps and a cosine schedule, gradient clipping at 1, frame strides drawn uniformly from $\{1, 2, 3, 4\}$, and the history noise and task mixture of Section 3.4. The main run does not use the extension of Appendix C. We materialise the block-wise path of Section 3.3 only for the 216 voxel-query rows per frame. Image and camera queries use the fused kernel. A compute-bound training step at $T = 4$ takes 0.59 s at about 27% model-FLOP utilisation.

Evaluation details. FVD is computed on non-overlapping 16-frame clips of the 360×640 video through a pretrained I3D network, leftover frames discarded, at 200, 400 and 600 rollout frames. FID compares the generated frame with the ground-truth frame at every rollout timestep and is reported as a curve. The camera trajectory is scored by the geodesic angle between the predicted and true

orientation in degrees and by the error of the camera centre in blocks in the frame-0 grid, each averaged per frame over the first 200, 400 and 600 frames. The hold baseline is an agent that never moves and never turns. The identifiability test is a 16×16 cross-matching on the reconstruction task: voxels are decoded for episode i from the video of episode j, and the diagonal $i = j$ must win its row, scored with an exact binomial p-value, which a model that ignores the video cannot pass. The human study follows PERSIST: raters score visual fidelity, 3D consistency, temporal consistency and overall quality of 600-frame rollouts on a 1–5 scale and state head-to-head preferences against each baseline.

D.3 Model Analysis

Effect of the window split. A rollout window of T frames is split into n observed context frames and m generated frames, written $n + m$. Table S3 varies the split on MoT-eval with everything else held fixed, with the PERSIST+$\mathbf{v}_0$ row of Table 1 repeated as the reference. Every split of MoTWorld improves on the cascade by a wide margin, and the ordering among the splits is consistent across horizons: the shortest window, $1 + 3$, gives the best video, followed by $4 + 4$ and then $1 + 7$. Generating fewer frames at once from a short, clean history keeps each frame closer to the carried voxel state, so on worlds that resemble the training distribution a short window is the better choice, while the $1 + 7$ window of the main results trades a little video quality for a longer history.

Table S3: **Effect of the window split.** FVD (↓) on MoT-eval for n context frames and m generated frames. All MoTWorld rows are one checkpoint seeded with $\mathbf{v}_0$. Best in bold, second best underlined.

Method	Split	FVD@200	FVD@400	FVD@600
PERSIST + $\mathbf{v}_0$ (reference)	/	79.5	104.4	123.5
MoTWorld	$1 + 3$	**45.1**	**40.0**	**38.0**
MoTWorld	$4 + 4$	<u>53.2</u>	<u>51.3</u>	<u>50.9</u>
MoTWorld	$1 + 7$	57.9	57.5	56.3

Inference speed and memory. Figure S2 measures the cost of a rollout step on a single H200 GPU for every task the checkpoint serves, against the released PERSIST-XL cascade. The joint rollout PV→PV at the $1 + 7$ window runs at about 3 frames per second within 7 GB, roughly seven times the throughput of PERSIST-XL at less than half of its memory, because MoTWorld generates the grid, the image and the camera in one pass of one denoiser, whereas the cascade runs a voxel denoiser, a rasteriser and a pixel denoiser in sequence. Since an absent modality contributes no tokens, the single-stream and reconstruction tasks are cheaper still: V→P and P→V run at 4 to 6 frames per second and the voxel-only and video-only rollouts at 12 to 14, all within 5 to 7 GB.