# STREAMING MULTI-AGENT AUTOREGRESSIVE DIFFUSION MODEL WITH WORLD STATE REGISTERS

**Sicheng Mo**[*1] **Yuheng Li**[*2] **Ziyang Leng**[1] **Krishna Kumar Singh**[2] **Bolei Zhou**[1]

[1]University of California, Los Angeles [2]Adobe Research

arXiv:2607.21594v1 [cs.CV] 23 Jul 2026

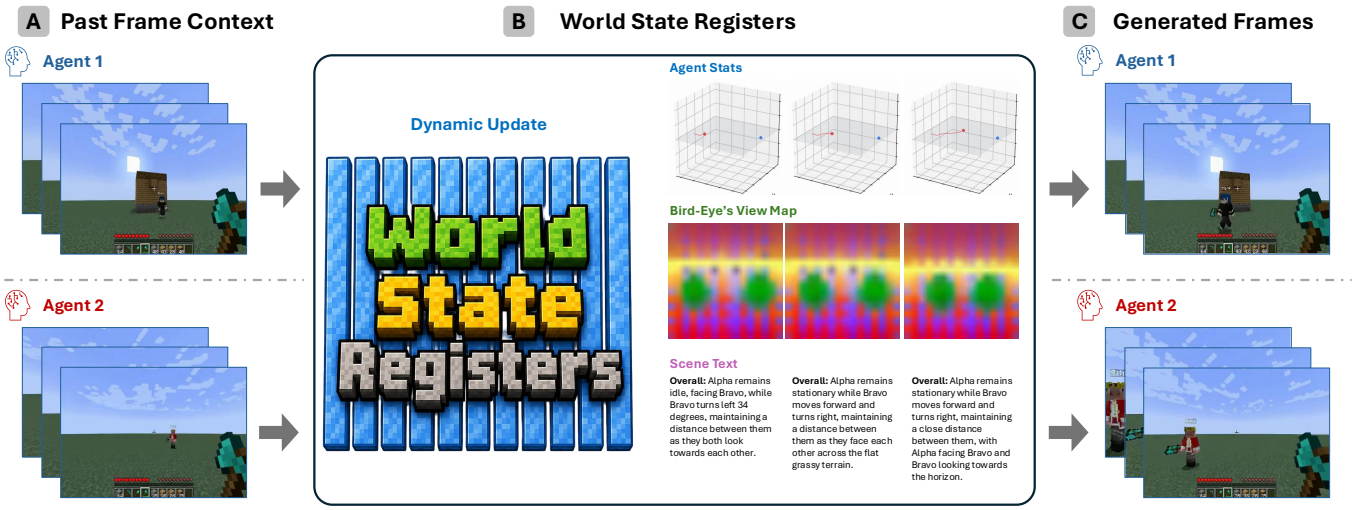


Figure 1: Our model enables interactive agents acting in a shared world. During rollout, a set of register tokens represents the evolving world state shared across agents. This state captures agent status, bird's-eye views, and scene text, and is being dynamically updated at each rollout step while new frames are generated.

## ABSTRACT

Multi-agent interactive world models should not only generate consistent observations, but also maintain world states that persist across agents and evolve across views. Existing autoregressive video diffusion pipelines carry forward observation history as conditioning context, which makes shared state difficult to maintain in multi-agent and multi-view settings. We present WorldWeaver ($\mathbf{W^2}$), a streaming multi-agent video diffusion model that augments rollout with cross-agent world state registers: learnable tokens that store shared world information, track individual agent status, and are dynamically updated after each generated chunk. We ground these registers with supervision signals spanning individual agent status, global state views including bird's-eye views, and scene text. We further improve the architecture with a Mixture-of-Transformers design that uses separate weights for world state modeling and visual frame modeling. Extensive experiments in two-agent Minecraft video generation show that explicit world-state modeling improves logical consistency and generation quality. Project Page: https://vail-ucla.github.io/worldweaver/

[*]Equal contribution.

## 1 INTRODUCTION

Recent advances in generative modeling have enabled increasingly powerful interactive and streaming world models (WMs) (ope; Savva et al., 2026). Most video world models simulate a scene from the view of a single agent. While this design can produce visually plausible frames, it is challenging to maintain logical and geometric consistency. This issue is further amplified in multi-agent settings (Savva et al., 2026), where each observer only sees a partial 2D projection of the same underlying 3D world, while observations across agents must remain compatible with that shared state. Beyond these cross-agent constraints, the state of the world continues to evolve no matter a particular observer sees it or not. This perspective motivates us to model the underlying world state directly, rather than treating consistency as a byproduct of per-agent video generation.

Temporally consistent video can be generated by chunk-based autoregressive video diffusion models (Chen et al., 2024; Yin et al., 2024). In such a pipeline, past frames are treated as conditioning context, and the model generates the next video chunk with joint spatiotemporal attention. However, because each chunk must be decoded into pixel space, the model needs to re-infer world information at every generation step. As a result, it carries forward limited observation history rather than explicit world states that can account for both observed and unobserved changes. In multi-view settings, this further introduces a mismatch between each agent's observation and the shared world state.

To tackle this issue, we explicitly model world states inside the generation process, instead of leaving them to be repeatedly inferred from frame history. We introduce a world state register, a group of learnable tokens that captures shared world information and is being incrementally updated during generation. The register is designed to carry persistent global information and individual agent status. We define two key properties for the world state register: (1) being persistent across agents and rollout steps, and (2) being dynamically updateable when new observations arrive. In this way, the observations from all agents contribute to the world state update, encouraging temporal and cross-agent consistency at the level of the underlying world state.

To facilitate long-horizon video generation, we adapt the Self-Forcing paradigm (Huang et al., 2025) to update both frames and world registers during rollout. After each denoised video chunk, the model refreshes the world register from the previous register and the newly generated observations, so the updated register can condition the next chunk. We further ground the register with complementary supervision signals: agent status encourages local motion consistency, bird's-eye views constrain global geometry, and scene text provides semantic grounding. We also explore the effect of semi-supervised world-state training, where a small labeled subset can anchor register semantics while additional unlabeled videos continue to improve the generative rollout.

We also study architectures for world-state modeling in diffusion models. When dense transformer weights must generate pixels and update world registers, the frame and state objectives can compete, especially once the register is pushed toward richer supervised semantics. We therefore use a Mixture-of-Transformers (MoT) architecture (Liang et al., 2024) that treats the world register branch as a distinct state pathway while keeping it coupled to the visual rollout. This separation improves supervised register learning and stabilizes frame register self-forcing during streaming generation.

We summarize our contributions as follows:

1. We introduce a new design of world state registers, a persistent and dynamically updateable set of tokens for carrying shared world information across agents and rollout steps.
2. We explore the supervision space for grounding world registers, ranging from individual agent status to global world-state signals such as bird's-eye views and scene text.
3. We propose an improved MoT-based architecture for video diffusion models that supports joint generation of world states and video frames. Our experiments show that this design improves performance and reduces conflicts between state updating and frame generation.

## 2 RELATED WORK

**Streaming video diffusion models.** Large-scale diffusion models provide a flexible foundation for image and video generation. Video diffusion models (ope; Ma et al., 2024; Wang et al., 2023) use bidirectional attention to synthesize coherent visual content and smooth temporal transitions under a

shared text condition. Building on these backbones, autoregressive video diffusion methods (Chen et al., 2024; Yin et al., 2024) introduce chunk level causal attention and KV caching to support streaming generation. Self-Forcing (Huang et al., 2025) further reduces the train test mismatch by rolling out the model on its own generated frames during training, while recent extensions (Shin et al., 2025; Cui et al., 2025; Zhu et al., 2026; Zhao et al., 2026) improve minute level streaming with attention sinks and stronger positional encoding. These advances make long horizon generation increasingly practical, but their main objective remains generation quality and rollout stability. In contrast, our WorldWeaver builds on streaming video diffusion for interactive generation and uses persistent world state registers to generate a logically consistent world, where temporal and cross agent consistency are grounded in an explicit state beyond the local context window.

**Memory-based video generation.** Memory mechanisms have been widely explored in video generation to preserve information that is not fully specified by the current frame or local context, including appearance, identity, geometry, object states, and scene layout. Explicit memory methods introduce retrieval buffers, spatial maps, 3D representations, or 4D scene memories that the generator can consult Xiao et al. (2026); Yu et al. (2025; 2026). This design is closely related to 3D novel view synthesis because both use a queryable scene representation to preserve geometry, appearance, and layout across viewpoints Team et al. (2026); Yang et al. (2026b); Sun et al. (2025). Existing explicit modeling, however, remains limited by scalability and by the need to update dynamic object states over time. Implicit memory methods store history inside the generative model through past-frame conditioning, recurrent rollout, or attention KV caches Chen et al. (2024); Yin et al. (2024); Huang et al. (2025); Zhao et al. (2026); Gao et al. (2026), but they provide weaker constraints because the memory is entangled with visual tokens and biased toward recent observations. Another line of work also values the role of state modeling in generative videos to ensure logical correctness (Liu et al., 2026; Po et al., 2025; Chen et al., 2025; Po et al., 2026). In contrast, our method keeps the memory inside the generator as persistent world state registers, but trains it as implicit memory with explicit supervision from individual agent states and global scene states, so the stored state receives direct grounding for world generation beyond frame history alone.

**World understanding and generation in unified models.** Visual generation and understanding are not fully decoupled. Text-to-image diffusion models have shown useful visual representations for discriminative tasks such as classification (Li et al., 2023; Clark & Jaini, 2023), detection (Xu et al., 2024), and segmentation (Xu et al., 2023; Tian et al., 2024; Namekata et al., 2024). A line of unified multimodal models further makes this connection explicit by training a single model for both text-to-image generation and image-to-text understanding (Zhou et al., 2024; Wang et al., 2024; Xie et al., 2024). Another line of work (Mo et al., 2025; Shi et al., 2024; Deng et al., 2025; Yang et al., 2026a), including Mixture-of-Transformers (Liang et al., 2024), explores the use of separate weights or dedicated fusion modules for different modalities and token roles, extending unified multimodal pretraining to broader interleaved and context rich settings. Meanwhile, MetaMorph (Tong et al., 2025) and X-Fusion (Mo et al., 2025) suggest that understanding tasks can improve generation by shaping stronger multimodal representations. Our WorldWeaver builds on these observations and further explores the synergy between world understanding and generation through world state registers. These registers provide a structured and verifiable representation with additional grounding from individual agent states and global scene states.

# 3 METHOD

## 3.1 OVERVIEW

We adopt the multi-agent world-model setting of Solaris (Savva et al., 2026). Multiple agents act in a shared world, and the model predicts each agent's future video stream from past observations and actions. We extend a single-player latent video diffusion pipeline by attaching a player axis to every generated tensor. This gives a compact tensor setting for synchronized multi-agent generation. With $P$ agents, the per-step observation and action are $\mathbf{x}^t = (\mathbf{x}_1^t, \ldots, \mathbf{x}_P^t)$ and $\mathbf{a}^t = (\mathbf{a}_1^t, \ldots, \mathbf{a}_P^t)$. A clip of $N$ latent frames forms $\mathbf{x} \in \mathbb{R}^{P \times N \times H \times W \times C}$. The conditioning $c$ bundles a first-frame visual embedding, a masked first-frame latent, and each agent's action sequence, with concrete dimensions deferred to the experiments. We introduce the stage-specific objectives in the subsections below.

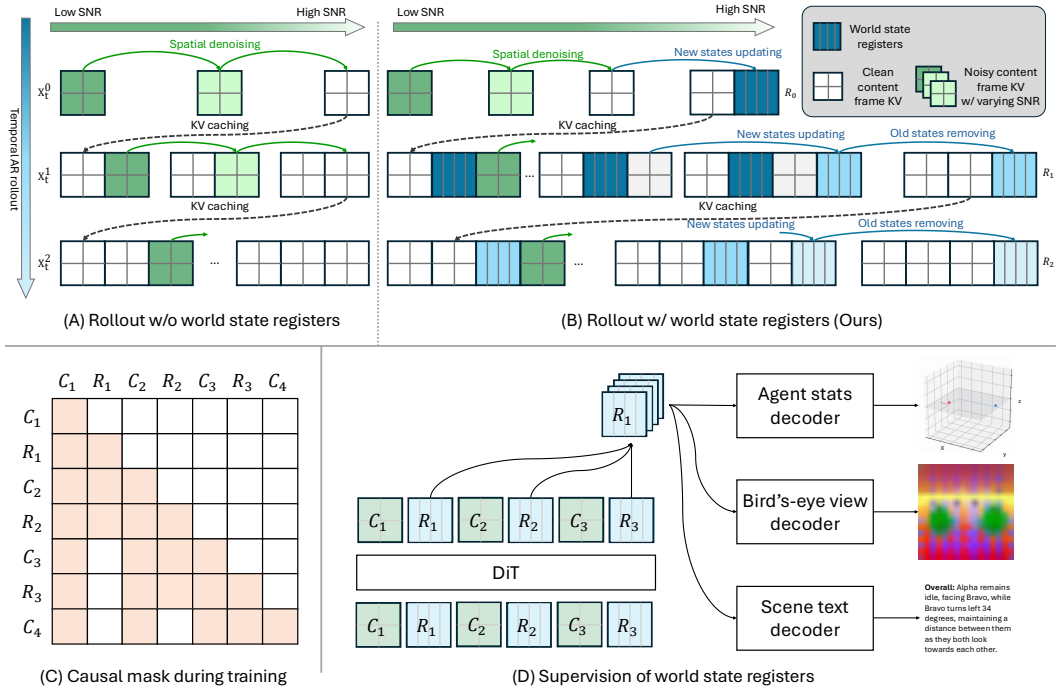


Figure 2: Pipeline overview. (A) A standard streaming autoregressive diffusion model denoises each new frame from a local KV cache window. (B) Our model streams with world state registers, which carry global state information and individual agent status. (C) Causal attention mask during Stage-2 training. Frame tokens attend to the local window and the latest register. (D) Auxiliary decoders ground the registers with agent status, bird's-eye-view maps, and scene text, encouraging the world state registers to encode global scene information.

We next introduce *world state registers* as a separate cross-agent state pathway. These registers are learnable tokens that capture hidden information about the shared world. They are shared across agents and updated incrementally as new frames are generated. The registers live alongside frame tokens in a causal transformer, so a single forward pass can denoise the next frame and refresh the cross-agent world state at the same time. We further use a Mixture-of-Transformers (MoT) design that assigns separate weights to register and frame tokens while keeping joint attention over the interleaved sequence. Detailed implementation is provided in Appendix B.2.

Our model follows a three-stage training curriculum on top of standard single-agent pretraining. *Stage 1 (Bidirectional Training)* finetunes a multi-player teacher on synchronized multi-agent clips. The teacher has full temporal context and learns cross-agent scene structure with bidirectional attention. *Stage 2 (Causal Training)* converts this teacher into a causal student by continuing the flow matching objective with causal attention and register supervision. *Stage 3 (Self-Forcing)* rolls out the student on its own predictions. After each generated frame, the model commits the updated register, so register drift is exposed together with frame drift. This stage uses DMD style distribution matching from the bidirectional teacher while retaining state supervision.

### 3.2 Stage 1: Bidirectional Training

Stage 1 trains a multi-player bidirectional teacher that jointly denoises the latent sequences of all players under bidirectional and cross-player attention. This teacher provides a full-time estimation of synchronized scene structure, including shared layout, relative player motion, and cross-view consistency. We initialize it from the pretrained single-player model to preserve the visual generation prior while adapting the model to synchronized multi-player data. We train this teacher with the conditional flow matching objective that predicts the velocity field $\mathbf{v}_\theta(\mathbf{x}_t, c, t)$:

$$\mathcal{L}_{\text{flow}} = \mathbb{E}\left[\|\mathbf{v}_\theta(\mathbf{x}_t, c, t) - (\epsilon - \mathbf{x}_0)\|_2^2\right]. \tag{1}$$

### 3.3 Stage 2: Causal Training with World State Registers

In this stage, we initialize the causal model from the bidirectional Stage 1 checkpoint and continue training with the same flow matching loss defined above. Following Solaris (Savva et al., 2026) and Diffusion Forcing (Chen et al., 2024), we sample an independent noise level for each player and latent frame, apply a block-level causal mask, and maintain a sliding window of only the latest $W$ past latent frames. This training gives the student autoregressive generation capability, where each new frame is conditioned on the action input and past frames. However, this design does not ground world modeling beyond the local context window, and therefore can fail to preserve a logically consistent world shared across agents.

To address this issue, we explicitly model the world state rather than relying on local frame history alone. The world state represents persistent information at a specific time step and summarizes the shared scene, including global layout, object configuration, and agent status that may not be visible from every player's current view. By carrying this state across rollout steps, the world model can synchronize predictions across agents and preserve logically consistent cross-agent interactions.

In practice, we define the world state registers (**WSR**) as a group of learnable $K$ tokens $\mathbf{r}_i \in \mathbb{R}^{K\times d}$. We formally define the streaming mechanism of world models with WSR as follows:

$$\mathbf{r}_i = G_\theta(\mathbf{r}_{i-1}, \mathbf{x}_{i-W+1}, \ldots, \mathbf{x}_i, a_i), \qquad p_\theta(\mathbf{x}_{i+1} \mid \mathbf{x}_{i-W+1}, \ldots, \mathbf{x}_i, a_{i+1}, \mathbf{r}_i).$$

During rollout, each register $\mathbf{r}_i$ is committed after the context frame $\mathbf{x}_i$ has been generated, thereby serving as the state summary through frame $i$ that conditions the next context frame.

Inspired by causal masking in Diffusion Forcing (Chen et al., 2024), we interleave frame tokens and register groups as $[\mathbf{x}_0, \mathbf{r}_0, \mathbf{x}_1, \mathbf{r}_1, \ldots, \mathbf{x}_{F-1}, \mathbf{r}_{F-1}]$. This ordering makes the rollout causal at the state level: the committed register precedes the next frame and the frame is conditioned on that register. Concretely, $\mathbf{r}_i$ precedes and conditions $\mathbf{x}_{i+1}$, so state update and frame prediction alternate across the sequence. Combined with the local attention window of size $W$, each register query attends to the same local context frames ending at its commit step, along with the immediately preceding register. A register query for $\mathbf{r}_i$ thus attends to $[\mathbf{x}_{i-W+1}, \ldots, \mathbf{x}_i]$ and $\mathbf{r}_{i-1}$, as shown in Figure 2(C).

Stage 2 keeps the flow matching loss and register supervision within a single training objective, where the flow matching term gives the causal student frame-level supervision under the committed register. However, this signal alone does not specify what information the WSR pathway should retain. We therefore ground each committed register with auxiliary world state predictions, as shown in Figure 2(D). For each supervision type $m \in \mathcal{M}$, the prediction head output $h_m(\mathbf{r}_i)$, the same step target $\mathbf{y}_i^m$, and the metric $d_m$ make the prediction, target, and distance calculation explicit. In our experiments, we use three supervision signals:

1. **Agent status.** The head predicts a simulator state vector $\hat{\mathbf{y}}_i^{\text{agent}} = h_{\text{agent}}(\mathbf{r}_i)$, and the target $\mathbf{y}_i^{\text{agent}}$ contains position, velocity, and orientation. We use $d_{\text{agent}} = \|\hat{\mathbf{y}}_i^{\text{agent}} - \mathbf{y}_i^{\text{agent}}\|_2^2$.
2. **Bird's-eye-view map.** The head predicts a bird's-eye view $\hat{\mathbf{y}}_i^{\text{bev}} = h_{\text{bev}}(\mathbf{r}_i)$, supervised by the aligned bird's-eye target $\mathbf{y}_i^{\text{bev}}$. We use $d_{\text{bev}} = 1 - \cos(\hat{\mathbf{y}}_i^{\text{bev}}, \phi_{\text{DINOv2}}(\mathbf{y}_i^{\text{bev}}))$, where $\phi_{\text{DINOv2}}$ denotes a frozen DINOv2 encoder (Oquab et al., 2024).
3. **Scene text.** The head predicts token logits $\hat{\mathbf{y}}_i^{\text{text}} = h_{\text{text}}(\mathbf{r}_i)$, and the target $\mathbf{y}_i^{\text{text}}$ is a textual scene description. We use $d_{\text{text}} = \text{CE}(\hat{\mathbf{y}}_i^{\text{text}}, \mathbf{y}_i^{\text{text}})$.

Together with the flow matching loss, we train the student with the following objective:

$$\mathcal{L}_{\text{reg}} = \sum_{i,m} \lambda_m d_m \left(h_m(\mathbf{r}_i), \mathbf{y}_i^m\right), \qquad \mathcal{L}_{\text{S2}} = \mathcal{L}_{\text{flow}} + \lambda_{\text{state}} \mathcal{L}_{\text{reg}}.$$

Here the sum ranges over rollout steps and the three target types, and we set $\lambda_{\text{state}}$ to one by default in Stage 2. In the Stage 2 objective, $\mathcal{L}_{\text{flow}}$ remains the frame-generation flow matching loss, while $\mathcal{L}_{\text{reg}}$ assigns explicit state targets to the committed registers. The prediction heads are training-only modules and are discarded at inference, so this supervision does not change the rollout cost. We ablate these supervision types in Section 4.3.

**Algorithm 1** Autoregressive Diffusion Inference with World State Registers

**Require:** Frame KV cache size $W$
**Require:** Conditioning $C$
**Require:** Denoising time steps $\{t_1, \dots, t_T\}$
**Require:** Number of generated frames $M$
**Require:** Causal AR diffusion model $G_\theta$, returning KV via $G_\theta$

**Initialize** generated latents $\mathbf{X}_\theta \leftarrow [\,]$
**Initialize** frame KV cache $\mathrm{KV}_C \leftarrow [\,]$
**Initialize** register KV cache $\mathrm{KV}_R \leftarrow [\,]$
**Initialize** world state $\mathbf{r}_0$
$\mathrm{KV}_R.\mathrm{append}(\mathbf{r}_0)$
**for** $i = 1, \dots, M$ **do**
  **Initialize** $\mathbf{x}^i_{t_T} \sim \mathcal{N}(0, I)$
  **for** $j = T, \dots, 1$ **do**
    Set $\mathbf{KV} \leftarrow (\mathrm{KV}_C, \mathrm{KV}_R)$
    Set $\hat{\mathbf{x}}^i_0 \leftarrow G_\theta(\mathbf{x}^i_{t_j}; t_j, \mathbf{KV}, c_i)$
    **if** $j = 1$ **then**
      $\mathbf{X}_\theta.\mathrm{append}(\hat{\mathbf{x}}^i_0)$
      Set $\mathrm{KV}^i_C \leftarrow G_\theta(\hat{\mathbf{x}}^i_0; 0, \mathbf{KV}, c_i)$
      **if** $|\mathrm{KV}_C| = W$ **then**
        $\mathrm{KV}_C.\mathrm{pop}(0)$
      **end if**
      $\mathrm{KV}_C.\mathrm{append}(\mathrm{KV}^i_C)$
      Set $\mathbf{r}_i \leftarrow G_\theta(\mathrm{KV}_R, \hat{\mathbf{x}}^i_0, c_i)$
      Set $\mathrm{KV}_R \leftarrow [\mathbf{r}_0, \mathbf{r}_i]$
    **else**
      sample $\epsilon \sim \mathcal{N}(0, I)$
      Set $\mathbf{x}^i_{t_{j-1}} \leftarrow \Psi(\hat{\mathbf{x}}^i_0, \epsilon, t_{j-1})$
    **end if**
  **end for**
**end for**
**return** $\mathbf{X}_\theta$

**Algorithm 2** Self-Forcing Training with Frame and State Register Rollout

**Require:** Conditioning $C$
**Require:** Denoising time steps $\{t_1, \dots, t_T\}$
**Require:** Number of video frames $N$
**Require:** Causal AR diffusion model $G_\theta$, returning KV via $G_\theta$

**loop**
  **Initialize** generated latents $\mathbf{X}_\theta \leftarrow [\,]$
  **Initialize** frame KV cache $\mathrm{KV}_C \leftarrow [\,]$
  **Initialize** register KV cache $\mathrm{KV}_R \leftarrow [\,]$
  **Initialize** world state $\mathbf{r}_0$
  $\mathrm{KV}_R.\mathrm{append}(\mathbf{r}_0)$
  sample $s \sim \mathrm{Uniform}(1, \dots, T)$
  **for** $i = 1, \dots, N$ **do**
    **Initialize** $\mathbf{x}^i_{t_T} \sim \mathcal{N}(0, I)$
    **for** $j = T, \dots, s$ **do**
      Set $\mathbf{KV} \leftarrow (\mathrm{KV}_C, \mathrm{KV}_R)$
      Set $\hat{\mathbf{x}}^i_0 \leftarrow G_\theta(\mathbf{x}^i_{t_j}; t_j, \mathbf{KV}, c_i)$
      **if** $j = s$ **then**
        $\mathbf{X}_\theta.\mathrm{append}(\hat{\mathbf{x}}^i_0)$
        Set $\mathrm{KV}^i_C \leftarrow G_\theta(\hat{\mathbf{x}}^i_0; 0, \mathbf{KV}, c_i)$
        $\mathrm{KV}_C.\mathrm{append}(\mathrm{KV}^i_C)$
        Set $\mathbf{r}_i \leftarrow G_\theta(0, \mathbf{KV}, c_i)$
        Set $\mathrm{KV}_R \leftarrow [\mathbf{r}_0, \mathbf{r}_i]$
      **else**
        sample $\epsilon \sim \mathcal{N}(0, I)$
        Set $\mathbf{x}^i_{t_{j-1}} \leftarrow \Psi(\hat{\mathbf{x}}^i_0, \epsilon, t_{j-1})$
      **end if**
    **end for**
  **end for**
  Update $\theta$ with $\mathcal{L}_{\mathrm{S3}}$, and update $s_{\mathrm{fake}}$ on student rollouts
**end loop**

### 3.4 Stage 3: Self-Forcing with Context Frame and State Rollout

We define the inference rollout as the autoregressive procedure used by our world model at deployment. At each step, the model denoises the next latent frame from the current frame KV cache, the latest committed register, and the action input. It then appends the generated frame to the cache and refreshes the world state register to summarize the updated world state. Algorithm 1 gives this inference rollout, and Figure 2(B) visualizes the same frame register loop.

However, the Stage 2 model is not yet ready for long-horizon rollout. Teacher forcing trains the student on ground truth causal context, while inference uses its own generated frames and committed registers, allowing frame and state errors to propagate across rollout. Following Self-Forcing, Stage 3 addresses the mismatch by simulating this rollout during training and exposing the causal student to its own generated history. We define the self-forcing training loop as a rollout that first runs the student on generated context frames and committed registers, then applies the self-forcing objective to the resulting clean prediction. Algorithm 2 summarizes this loop.

Applying the DMD generator objective from Self-Forcing (Huang et al., 2025), we perturb the rollout prediction with fresh noise, $\hat{\mathbf{x}}_t = (1-t)\hat{\mathbf{x}}_0 + t\epsilon$ for $\epsilon \sim \mathcal{N}(0, I)$, and train the student with

$$\mathcal{L}_{\mathrm{sf}} = \mathbb{E}_{t,\epsilon}\left[\frac{1}{2}\left\|\hat{\mathbf{x}}_0 - \mathrm{sg}\left(\hat{\mathbf{x}}_0 - (s_{\mathrm{fake}}(\hat{\mathbf{x}}_t, c, t) - s_{\mathrm{real}}(\hat{\mathbf{x}}_t, c, t))\right)\right\|_2^2\right]. \tag{2}$$

Here $\mathrm{sg}(\cdot)$ denotes stop gradient, $s_{\mathrm{real}}$ is a frozen score network initialized from the Stage 1 teacher, and $s_{\mathrm{fake}}$ is a trainable critic on student rollouts. Because frame $\mathbf{x}_i$ commits $\mathbf{r}_i$ during rollout, Stage 3 reuses the same register loss on registers that encode world states of completed context frames:

$$\mathcal{L}_{\mathrm{S3}} = \mathcal{L}_{\mathrm{sf}} + \lambda_{\mathrm{state}}\mathcal{L}_{\mathrm{reg}}.$$

We likewise set $\lambda_{\text{state}}$ to one by default in Stage 3. In practice, we follow Solaris (Savva et al., 2026) by using a four step denoising schedule with timesteps $1000 \to 750 \to 500 \to 250 \to 0$ and by sampling an early exit denoising step during rollout for stochastic gradient truncation. At this stage, we find that the DMD loss alone is not sufficient to improve stability, since WSR modeling also suffers from drift. Thus, we increase selected per-head register supervision weights relative to Stage 2, which helps train better registers and stabilize long horizon rollout.

# 4 Experiments

In this section, we first present the main results in Sec. 4.2, then analyze the contribution of each supervision signal in Sec. 4.3, study the model architecture in Sec. 4.4, and discuss semi-supervised training in Sec. 4.5.

## 4.1 Experimental Setup

**Training data.** We build on our data collection pipeline based on SolarisEngine (Savva et al., 2026). We collect approximately 126 hours of synchronized two-player videos with extensive agent status information and additional bird's-eye view camera footage. We further annotate the scene text of each 4-frame clip with Qwen2.5-VL-72B-Instruct (Bai et al., 2025). Detailed data collection and annotation are provided in Appendix A.

**Evaluation.** We evaluate our model on the Solaris original test split (Savva et al., 2026). For each clip, the model receives both players' first-frame observations and full action sequences, then generates the remaining rollout. We report VLM accuracy and FID for each evaluation category. VLM accuracy measures whether the generated video preserves the queried world relation, such as relative object position and cross-player consistency, while FID measures visual realism and distributional fidelity. To summarize these two axes without replacing either, we also report an auxiliary **world score**:

$$\text{WorldScore} = \frac{\frac{1}{K}\sum_{k=1}^{K} \text{VLM}_k}{\frac{1}{K}\sum_{k=1}^{K} \text{FID}_k}, \tag{3}$$

where $K = 5$. We report VLM accuracy in percentage points and multiply the ratio by 100 for readability. Since WorldScore is unbounded and sensitive to the scale of FID, we use it only as an auxiliary ranking metric. VLM accuracy and FID remain the primary metrics.

## 4.2 Main Results

We compare our WorldWeaver multi-agent world model with the following baselines: (1) *Frame concat*: a world model that concatenates the two players' observations along the channel dimension, following Multiverse (team, 2025); (2) *Solaris* (Savva et al., 2026): a multi-agent world model that synchronizes multi-agent observations by extending the sequence of frame tokens and using bidirectional attention across players. To ensure a fair comparison, we retrain the Solaris model with the same training data as our model.

As shown in Table 1, WorldWeaver improves the aggregate world score to 105.1, surpassing the previous best by a large margin. The VLM accuracy gains are strongest on state-sensitive categories. Compared with the Solaris baseline, Grounding rises from 81.3 to 93.8, Building rises from 9.4 to 28.1, and Consistency rises from 57.8 to 76.6. These gains suggest that the persistent registers do more than improve visual fidelity: they help the model maintain a logically coherent and verifiable world model across players and rollout steps.

Figure 3 visualizes the generated world together with decoded world states, including agent statistics, bird's-eye-view maps, and scene text. For agent statistics, we selectively visualize the positions and trajectories. For the bird's-eye-view maps, we use PCA to project the predicted DINOv2 (Oquab et al., 2024) features to RGB images. WSR rollout allows WorldWeaver to generate a logically coherent world and better synchronize cross-agent information. Additionally, WSR offers a more interpretable way to inspect the state represented by each generated chunk.

Table 1: Comparison with multi-agent world model baselines. We report category-level VLM accuracy, FID (Heusel et al., 2017), and the aggregate world score. Best result in each column is highlighted in **bold**.

| | Movement | | Grounding | | Memory | | Building | | Consistency | | World |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Method | VLM ↑ | FID ↓ | VLM ↑ | FID ↓ | VLM ↑ | FID ↓ | VLM ↑ | FID ↓ | VLM ↑ | FID ↓ | Score ↑ |
| Frame concat | 77.1 | 68.9 | 53.1 | 66.6 | 37.5 | 74.4 | 0.0 | 103.2 | 49.5 | 129.4 | 49.1 |
| Solaris* | 79.7 | 43.3 | 81.3 | 37.2 | 43.8 | **61.2** | 9.4 | 83.4 | 57.8 | 110.7 | 81.0 |
| **W² (Ours)** | **82.8** | **34.0** | **93.8** | **36.8** | **46.9** | 64.8 | **28.1** | **75.9** | **76.6** | **100.7** | **105.1** |

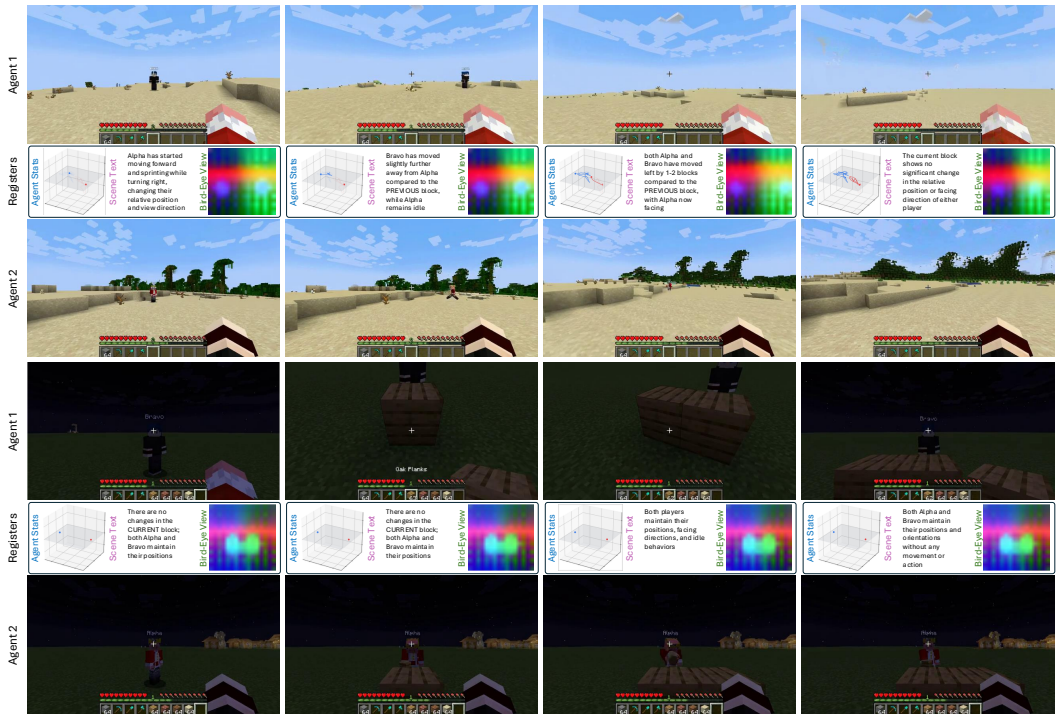


Figure 3: **Qualitative visualization.** We show generated multi-agent rollouts with grounded world states: agent coordinates, bird's-eye-view maps, and scene text. These signals make the persistent world state inspectable and help verify cross-player consistency over time. Please refer to the project page for more detailed video demo.

### 4.3 Analysis of Supervision Signals

A core problem is not only how to supervise the world state registers, but also why explicit supervision is needed. Ideally, the registers could learn both agent-specific state and global scene information from the diffusion loss alone. However, in practice, this signal does not specify which information should be stored as world state. We therefore add complementary supervision signals that encourage the registers to encode meaningful and useful world state information.

We hypothesize that the world state registers should encode local dynamics, global geometry, and cross-modal semantics. We therefore instantiate the three supervision types as follows. (1) For individual-state supervision, the register predicts *agent statistics*, including each player's position, velocity, and orientation. These targets give the register an explicit per-agent coordinate signal, allowing us to test whether the persistent state carries verifiable motion information for each player. (2) For global 3D supervision, we use a *bird's-eye view* that exposes scene layout and object distribution from an allocentric viewpoint. This signal asks the register to preserve spatial structure shared by both players, rather than only the egocentric evidence visible in the current frame. (3) For global cross-modal supervision, we attach *scene text* that describes the visual state of the current frame in language. This target complements local dynamics and global geometry, asking the register to retain categories, attributes, and high-level content across rollouts.

Table 2: Supervision ablation for world state registers. We compare registers without explicit targets and registers with different signals; their combination further improves the world state registers.

| | Movement | | Grounding | | Memory | | Building | | Consistency | | World |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Variant | VLM ↑ | FID ↓ | VLM ↑ | FID ↓ | VLM ↑ | FID ↓ | VLM ↑ | FID ↓ | VLM ↑ | FID ↓ | Score ↑ |
| Baseline | 79.7 | 43.3 | 81.3 | 37.2 | 43.8 | 61.2 | 9.4 | 83.4 | 57.8 | 110.7 | 81.0 |
| Registers only | 90.6 | 43.2 | 81.3 | 44.3 | **62.5** | 66.1 | 21.9 | 84.3 | 62.5 | 101.9 | 93.8 |
| + Agent stats | **95.3** | 41.0 | 59.4 | 45.5 | 56.3 | **60.1** | 9.4 | 80.8 | 75.0 | 107.9 | 88.1 |
| + Bird's-eye view | 82.8 | 39.1 | **96.9** | 40.7 | 46.9 | 64.7 | **31.3** | **74.2** | 71.9 | 103.4 | 102.4 |
| + Scene text | 85.9 | 40.2 | 84.4 | 38.4 | **62.5** | 62.1 | 25.0 | 78.8 | 73.4 | 101.6 | 103.2 |
| + All | 82.8 | **34.0** | 93.8 | **36.8** | 46.9 | 64.8 | 28.1 | 75.9 | **76.6** | **100.7** | **105.1** |

Starting from the Solaris (Savva et al., 2026) baseline, we first enable registers without explicit supervision, then add each signal, and finally evaluate the combined setting, as reported in Table 2. All other hyperparameters stay at the defaults of Section 4.1. First, we observe that registers alone already help: without any explicit state target, world score improves from 81.0 to 93.8. We attribute this gain to registers implicitly building a persistent world state, because they provide a dedicated slot to carry cross-agent information instead of recomputing it from the local window at every step.

However, explicitly modeling world-state targets can push performance further compared with the baseline. All three supervised variants outperform the baseline (81.0) because they specify what the registers should encode and keep verifiable across rollout steps, instead of leaving that choice entirely to the pixel loss. We find that scene text helps the most among single supervision signals, raising world score to 103.2. Combining all three signals further lifts world score from 93.8 to 105.1, with higher mean VLM accuracy and lower mean FID. These results corroborate our hypothesis and highlight the importance of enabling world state registers in the multi-agent world model setting.

### 4.4 Analysis of Model Architecture

Motivated by recent studies in multimodal generation (Mo et al., 2025), we view the architecture as a possible source of conflict between different token roles. A dense transformer can in principle use the same parameters to render pixel context frames and generate accurate world state registers. However, register tokens are intended to represent hidden information about the world, while frame tokens mainly carry local visual information. This difference in function becomes harder to balance once the register encodes richer state information. We follow the MoT design by using separate modality-specific weights to process register and frame tokens, while allowing them to interact through the shared attention mechanism.

Table 3 compares dense and MoT backbones under two settings, using registers without explicit supervision and using registers with scene text supervision. The results suggest that MoT is not automatically beneficial. Without explicit supervision, MoT does not improve the aggregate world score over the dense backbone. The pattern changes under scene text supervision: the dense model drops to 91.3 world score, while MoT reaches 103.2. This contrast suggests that richer semantic representations in world state registers are harder to process with the same parameterization as visual tokens. Separating register and frame computation therefore becomes useful when the register carries a distinct semantic world state.

Additionally, we investigate the training dynamics of the MoT model during Stage 3 training. Separate weight parameters also allow a more controlled learning process. We empirically find that the register loss in Stage 3 training converges more slowly than the DMD distillation loss, while the register loss mainly updates the register transformer and the DMD distillation loss mainly updates the visual transformer. Therefore, after step 1500, we apply the separate LR schedule, where we freeze the weight parameters of the visual transformer and only update the weight parameters of the register transformer during Stage 3 training. As shown in Figure 4, the mean VLM accuracy improves from 63.8 to 68.1. This result suggests that the MoT is important for WSR-based multi-agent world modeling.

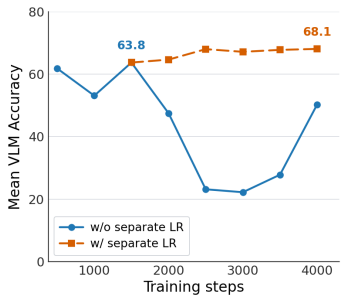


Figure 4: **VLM accuracy over Stage 3 self-forcing.**

Table 3: Dense and MoT backbone ablation. We compare a dense transformer with a Mixture of Transformers under no register supervision and scene text supervision.

| | Movement | | Grounding | | Memory | | Building | | Consistency | | World |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Model | VLM ↑ | FID ↓ | VLM ↑ | FID ↓ | VLM ↑ | FID ↓ | VLM ↑ | FID ↓ | VLM ↑ | FID ↓ | Score ↑ |
| *No supervision* | | | | | | | | | | | |
| Dense | 76.6 | 41.7 | 87.5 | 40.4 | 65.6 | 54.0 | 12.5 | 87.2 | 73.4 | 107.4 | 95.5 |
| MoT | 90.6 | 43.2 | 81.3 | 44.3 | 62.5 | 66.1 | 21.9 | 84.3 | 62.5 | 101.9 | 93.8 |
| *Scene text supervision* | | | | | | | | | | | |
| Dense | 81.3 | 37.3 | 81.3 | 53.1 | 53.1 | 54.8 | 9.4 | 73.9 | 71.9 | 106.4 | 91.3 |
| MoT | 85.9 | 40.2 | 84.4 | 38.4 | 62.5 | 62.1 | 25.0 | 78.8 | 73.4 | 101.6 | 103.2 |

Table 4: Semi-supervised training. Each row varies the number of data clips without bird's-eye view annotation (unlabeled data), where "1K / N K" denotes 1K labeled clips and N K unlabeled clips.

| | Movement | | Grounding | | Memory | | Building | | Consistency | | World |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Data Split | VLM ↑ | FID ↓ | VLM ↑ | FID ↓ | VLM ↑ | FID ↓ | VLM ↑ | FID ↓ | VLM ↑ | FID ↓ | Score ↑ |
| 1K / 0K | 78.1 | 60.0 | 75.0 | 72.6 | 21.9 | 95.0 | 15.6 | 76.9 | 71.9 | 111.0 | 63.2 |
| 1K / 2.5K | 68.8 | 58.8 | 56.3 | 60.2 | 65.6 | 96.4 | 15.6 | 79.7 | 56.3 | 112.4 | 64.4 |
| 1K / 5.0K | 85.9 | 60.4 | 56.3 | 58.3 | 81.3 | 79.8 | 28.1 | 76.6 | 71.9 | 117.7 | 82.3 |
| 1K / 10.0K | 81.3 | 44.3 | 65.6 | 53.2 | 84.3 | 75.8 | 25.0 | 73.6 | 68.8 | 112.9 | 90.3 |

### 4.5 Extension: Semi-Supervised Training

A natural question is whether this pipeline can extend beyond fully instrumented simulator data toward more realistic world modeling. It remains challenging to collect real-world data with explicit world-state annotations *e.g.* agent stats and bird's-eye-view maps. We therefore adopt a semi-supervised setup: clips with bird's-eye-view maps are labeled, and clips without them are unlabeled. This setup serves as a proxy for future realistic data regimes, where raw videos may be abundant but explicit world-state labels are limited.

In our experimental setting, we fix the labeled set to 1K video clips and gradually increase the amount of unlabeled video clips. Labeled samples receive both the diffusion loss and the register supervision losses, while unlabeled samples train the video generator through the diffusion objective only. We apply bird's-eye view supervision and use the default training hyperparameters except for the Stage-2 training length, where we use 20K instead of 60K steps to prevent overfitting to the labeled data. Table 4 shows that adding unlabeled data improves the aggregate world score from 63.2 with no unlabeled data to 82.3 with 5K unlabeled clips, and further to 90.3 with 10K unlabeled clips. We observe a generally consistent improvement in both VLM accuracy and FID across the table, suggesting that unlabeled rollout data can still improve generation quality when a small supervised subset anchors the hidden world state in the registers. Overall, our approach provides a promising direction for settings where explicit world-state labels are hard to collect.

## 5 Conclusion

We introduce WorldWeaver, a streaming multi-agent video diffusion model with cross-agent world state registers for interactive video generation. The core idea is to maintain and update a shared world state across agents. Our model then refines remembered scene information, rather than repeatedly relying on an expanding frame history. Across multi-agent Minecraft experiments, this design improves both visual quality and logical consistency, and the ablations show that explicit world-state supervision further shapes the registers into a more verifiable state representation.

**Limitations.** While our experiments highlight the potential of cross-agent world state registers, there are still several limitations. Our key improvement comes from the additional state supervision, but the state in the real world is much more complex and is not directly available. However, we believe this is a promising direction for future work, where a world model could be aware of more complex state information, e.g., from the low-level 3D consistent visual detail to the high-level semantic relationships across agents and players. We hope this work could shed light on future work on world models with cross-agent capabilities.

## REFERENCES


Sora: Creating video from text — openai.com. https://openai.com/index/sora/. [Accessed 30-04-2025]. 2

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-vl technical report, 2025. URL https://arxiv.org/abs/2502.13923. 7, 18

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. 2, 3, 5

Taiye Chen, Zihan Ding, Anjian Li, Christina Zhang, Zeqi Xiao, Yisen Wang, and Chi Jin. Recurrent autoregressive diffusion: Global memory meets local attention. *arXiv preprint arXiv:2511.12940*, 2025. 3

Kevin Clark and Priyank Jaini. Text-to-image diffusion models are zero shot classifiers. *Advances in Neural Information Processing Systems*, 36:58921–58937, 2023. 3

Justin Cui, Jie Wu, Ming Li, Tao Yang, Xiaojie Li, Rui Wang, Andrew Bai, Yuanhao Ban, and Cho-Jui Hsieh. Self-forcing++: Towards minute-scale high-quality video generation. *arXiv preprint arXiv:2510.02283*, 2025. 3

Chaorui Deng, Deyao Zhu, Kunchang Li, Chenhui Gou, Feng Li, Zeyu Wang, Shu Zhong, Weihao Yu, Xiaonan Nie, Ziang Song, et al. Emerging properties in unified multimodal pretraining. *arXiv preprint arXiv:2505.14683*, 2025. 3

Quankai Gao, Jiawei Yang, Qiangeng Xu, Le Chen, and Yue Wang. Lome: Learning human-object manipulation with action-conditioned egocentric world model. *arXiv preprint arXiv:2603.27449*, 2026. 3

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024. 18

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017. 8

Xun Huang, Zhengqi Li, Guande He, Mingyuan Zhou, and Eli Shechtman. Self forcing: Bridging the train-test gap in autoregressive video diffusion. *arXiv preprint arXiv:2506.08009*, 2025. 2, 3, 6

Alexander C Li, Mihir Prabhudesai, Shivam Duggal, Ellis Brown, and Deepak Pathak. Your diffusion model is secretly a zero-shot classifier. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2206–2217, 2023. 3

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. 2, 3

Fangfu Liu, Kai He, Tianchang Shen, Tianshi Cao, Sanja Fidler, Yueqi Duan, Jun Gao, Igor Gilitschenski, Zian Wang, and Xuanchi Ren. Gamma-world: Generative multi-agent world modeling beyond two players. *arXiv preprint arXiv:2605.28816*, 2026. 3

Xin Ma, Yaohui Wang, Gengyun Jia, Xinyuan Chen, Ziwei Liu, Yuan-Fang Li, Cunjian Chen, and Yu Qiao. Latte: Latent diffusion transformer for video generation. *arXiv preprint arXiv:2401.03048*, 2024. 2

Sicheng Mo, Thao Nguyen, Xun Huang, Siddharth Srinivasan Iyer, Yijun Li, Yuchen Liu, Abhishek Tandon, Eli Shechtman, Krishna Kumar Singh, Yong Jae Lee, et al. X-fusion: Introducing new modality to frozen large language models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 228–238, 2025. 3, 9

Koichi Namekata, Amirmojtaba Sabour, Sanja Fidler, and Seung Wook Kim. Emerdiff: Emerging pixel-level semantic knowledge in diffusion models. *arXiv preprint arXiv:2401.11739*, 2024. 3

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024. 5, 7

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 *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 8733–8744, 2025. 3

Ryan Po, David Junhao Zhang, Amir Hertz, Gordon Wetzstein, Neal Wadhwa, and Nataniel Ruiz. Multigen: Level-design for editable multiplayer worlds in diffusion game engines. *arXiv preprint arXiv:2603.06679*, 2026. 3

Georgy Savva, Oscar Michel, Daohan Lu, Suppakit Waiwitlikhit, Timothy Meehan, Dhairya Mishra, Srivats Poddar, Jack Lu, and Saining Xie. Solaris: Building a multiplayer video world model in minecraft. *arXiv preprint arXiv:2602.22208*, 2026. 2, 3, 5, 7, 9, 15, 18, 19

Weijia Shi, Xiaochuang Han, Chunting Zhou, Weixin Liang, Xi Victoria Lin, Luke Zettlemoyer, and Lili Yu. Llamafusion: Adapting pretrained language models for multimodal generation. *arXiv preprint arXiv:2412.15188*, 2024. 3

Joonghyuk Shin, Zhengqi Li, Richard Zhang, Jun-Yan Zhu, Jaesik Park, Eli Shechtman, and Xun Huang. Motionstream: Real-time video generation with interactive motion controls. *arXiv preprint arXiv:2511.01266*, 2025. 3

Wenqiang Sun, Haiyu Zhang, Haoyuan Wang, Junta Wu, Zehan Wang, Zhenwei Wang, Yunhong Wang, Jun Zhang, Tengfei Wang, and Chunchao Guo. Worldplay: Towards long-term geometric consistency for real-time interactive world modeling. *arXiv preprint arXiv:2512.14614*, 2025. 3

Enigma team. Introducing multiverse: The first ai multiplayer world model, 2025. URL https://enigma.inc/blog. 7

InSpatio Team, Donghui Shen, Guofeng Zhang, Haomin Liu, Haoyu Ji, Hujun Bao, Hongjia Zhai, Jialin Liu, Jing Guo, Nan Wang, et al. Inspatio-world: A real-time 4d world simulator via spatiotemporal autoregressive modeling. *arXiv preprint arXiv:2604.07209*, 2026. 3

Junjiao Tian, Lavisha Aggarwal, Andrea Colaco, Zsolt Kira, and Mar Gonzalez-Franco. Diffuse attend and segment: Unsupervised zero-shot segmentation using stable diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3554–3563, 2024. 3

Shengbang Tong, David Fan, Jiachen Li, Yunyang Xiong, Xinlei Chen, Koustuv Sinha, Michael Rabbat, Yann LeCun, Saining Xie, and Zhuang Liu. Metamorph: Multimodal understanding and generation via instruction tuning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 17001–17012, 2025. 3

Xinlong Wang, Xiaosong Zhang, Zhengxiong Luo, Quan Sun, Yufeng Cui, Jinsheng Wang, Fan Zhang, Yueze Wang, Zhen Li, Qiying Yu, et al. Emu3: Next-token prediction is all you need. *arXiv preprint arXiv:2409.18869*, 2024. 3

Yaohui Wang, Xinyuan Chen, Xin Ma, Shangchen Zhou, Ziqi Huang, Yi Wang, Ceyuan Yang, Yinan He, Jiashuo Yu, Peiqing Yang, et al. Lavie: High-quality video generation with cascaded latent diffusion models. *arXiv preprint arXiv:2309.15103*, 2023. 2

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. 3

Jinheng Xie, Weijia Mao, Zechen Bai, David Junhao Zhang, Weihao Wang, Kevin Qinghong Lin, Yuchao Gu, Zhijie Chen, Zhenheng Yang, and Mike Zheng Shou. Show-o: One single transformer to unify multimodal understanding and generation. *arXiv preprint arXiv:2408.12528*, 2024. 3

Chenfeng Xu, Huan Ling, Sanja Fidler, and Or Litany. 3difftection: 3d object detection with geometry-aware diffusion features. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10617–10627, 2024. 3

Jiarui Xu, Sifei Liu, Arash Vahdat, Wonmin Byeon, Xiaolong Wang, and Shalini De Mello. Open-vocabulary panoptic segmentation with text-to-image diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 2955–2966, 2023. 3

Ceyuan Yang, Zhijie Lin, Yang Zhao, Fei Xiao, Hao He, Qi Zhao, Chaorui Deng, Kunchang Li, Zihan Ding, Yuwei Guo, et al. Context unrolling in omni models. *arXiv preprint arXiv:2604.21921*, 2026a. 3

Yuxue Yang, Lue Fan, Ziqi Shi, Junran Peng, Feng Wang, and Zhaoxiang Zhang. Neoverse: Enhancing 4d world model with in-the-wild monocular videos. *arXiv preprint arXiv:2601.00393*, 2026b. 3

Tianwei Yin, Qiang Zhang, Richard Zhang, William T Freeman, Fredo Durand, Eli Shechtman, and Xun Huang. From slow bidirectional to fast causal video generators. *arXiv e-prints*, pp. arXiv–2412, 2024. 2, 3

Jiwen Yu, Jianhong Bai, Yiran Qin, Quande Liu, Xintao Wang, Pengfei Wan, Di Zhang, and Xihui Liu. Context as memory: Scene-consistent interactive long video generation with memory retrieval. In *Proceedings of the SIGGRAPH Asia 2025 Conference Papers*, pp. 1–11, 2025. 3

Wei Yu, Runjia Qian, Yumeng Li, Liquan Wang, Songheng Yin, Dennis Anthony, Yang Ye, Yidi Li, Weiwei Wan, Animesh Garg, et al. Mosaicmem: Hybrid spatial memory for controllable video world models. *arXiv preprint arXiv:2603.17117*, 2026. 3

Zengqun Zhao, Yanzuo Lu, Ziquan Liu, Jifei Song, Jiankang Deng, and Ioannis Patras. Relax forcing: Relaxed kv-memory for consistent long video generation. *arXiv preprint arXiv:2603.21366*, 2026. 3

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. *arXiv preprint arXiv:2408.11039*, 2024. 3

Hongzhou Zhu, Min Zhao, Guande He, Hang Su, Chongxuan Li, and Jun Zhu. Causal forcing: Autoregressive diffusion distillation done right for high-quality real-time interactive video generation. *arXiv preprint arXiv:2602.02214*, 2026. 3

# APPENDIX

**A Data Collection** . . . . . **15**
A.1 Overview . . . . . 15
A.2 Individual states: agent statistics . . . . . 15
A.3 Global state: bird's-eye view . . . . . 15
A.4 Cross-modal states: scene text . . . . . 15

**B Pipeline and implementation details** . . . . . **17**
B.1 State decoding . . . . . 17
B.1.1 Agent statistics decoding . . . . . 17
B.1.2 Bird's-eye view decoding . . . . . 18
B.1.3 Scene text decoding . . . . . 18
B.2 Mixture of Transformers . . . . . 18
B.3 Training . . . . . 18
B.3.1 Stage 1: Bidirectional Training . . . . . 18
B.3.2 Stage 2: Causal Training with World State Registers . . . . . 19
B.3.3 Stage 3: Self-Forcing with Context Frame and State Rollout . . . . . 19

# A Data Collection

## A.1 Overview

We build our dataset on top of the SolarisEngine simulator (Savva et al., 2026). The dataset contains approximately 126 hours of synchronized two-agent interaction recordings at a resolution of $832 \times 480$ and a frame rate of 20 fps. Each recording session pairs two agents, denoted Alpha and Bravo, acting concurrently in the same episode, and each agent's state is logged at 20 Hz alongside the video stream. Every session is paired with three complementary streams of world-state annotations: per-agent statistics that capture individual dynamics, a shared bird's-eye view that exposes the global scene layout, and per-chunk scene text that provides cross-modal semantics. These signals supply the individual, global, and cross-modal supervision used to ground the world-state registers, and we detail how each stream is collected in the following subsections.

## A.2 Individual states: agent statistics

For the individual state stream, we log each agent's position, velocity, and orientation, along with ground-truth controller inputs at 20 Hz, giving a compact state record that captures how each agent moves through the episode. Figure 5 visualizes these statistics for Alpha and Bravo over time.

## A.3 Global state: bird's-eye view

In addition to each agent's egocentric camera, every recording session is filmed by a shared fixed overhead camera positioned above the scene. Because this camera is shared rather than attached to either agent, both agents appear together in a single frame, giving a global view of their relative position and spatial relationship that complements the two first-person streams. Figure 5 shows representative frames from this overhead camera over the course of one episode.

## A.4 Cross-modal states: scene text

The scene text collection process is separate from the game engine. To provide language-grounded supervision for cross-modal scene understanding, we annotate every recording with Qwen2.5-VL-72B-Instruct. Each episode is divided into temporal chunks of four consecutive frames, corresponding to 200 ms at 20 fps and matching the temporal downsampling factor of our video tokenizer. For each chunk, we assemble synchronized inputs from the shared overhead view and the two egocentric views. We sample three frames from each view for the current chunk, and we include the immediately preceding chunk as temporal context when it is available. Generation is capped at 256 tokens.

To ground the caption in observed behavior rather than asking the model to infer actions from pixels alone, we additionally condition the prompt on each agent's ground-truth controller input summary for the current chunk, including movement keys, action keys, hotbar or tool selection, and net camera rotation. We use the following question template when captioning chunk $t$ and provide chunk $t-1$ as temporal context whenever it exists:

> Synchronized short blocks of a two-agent session are shown in time order: first the previous block, then the current block. For each block you see an overhead view, then Alpha's egocentric view, then Bravo's egocentric view. Ignore corner text, chat, heart and hunger bars; the bottom bar in an egocentric view is that player's hotbar.
>
> Ground-truth controller inputs for the current block: Alpha [current input summary], Bravo [current input summary]. Ground-truth controller inputs for the previous block: Alpha [previous input summary], Bravo [previous input summary].
>
> Using both the views and these inputs, fill in this template exactly, one sentence per line:
>
> Overall: what each agent does in the current block, their relative position and distance, and where each is facing or looking.
>
> Change: how the current block differs from the previous one in relative position, facing direction, view direction, and each agent's behavior.

For an episode's first chunk, which has no previous chunk, we omit the temporal context and use a shorter prompt whose change line is fixed to "first block, no previous, none."

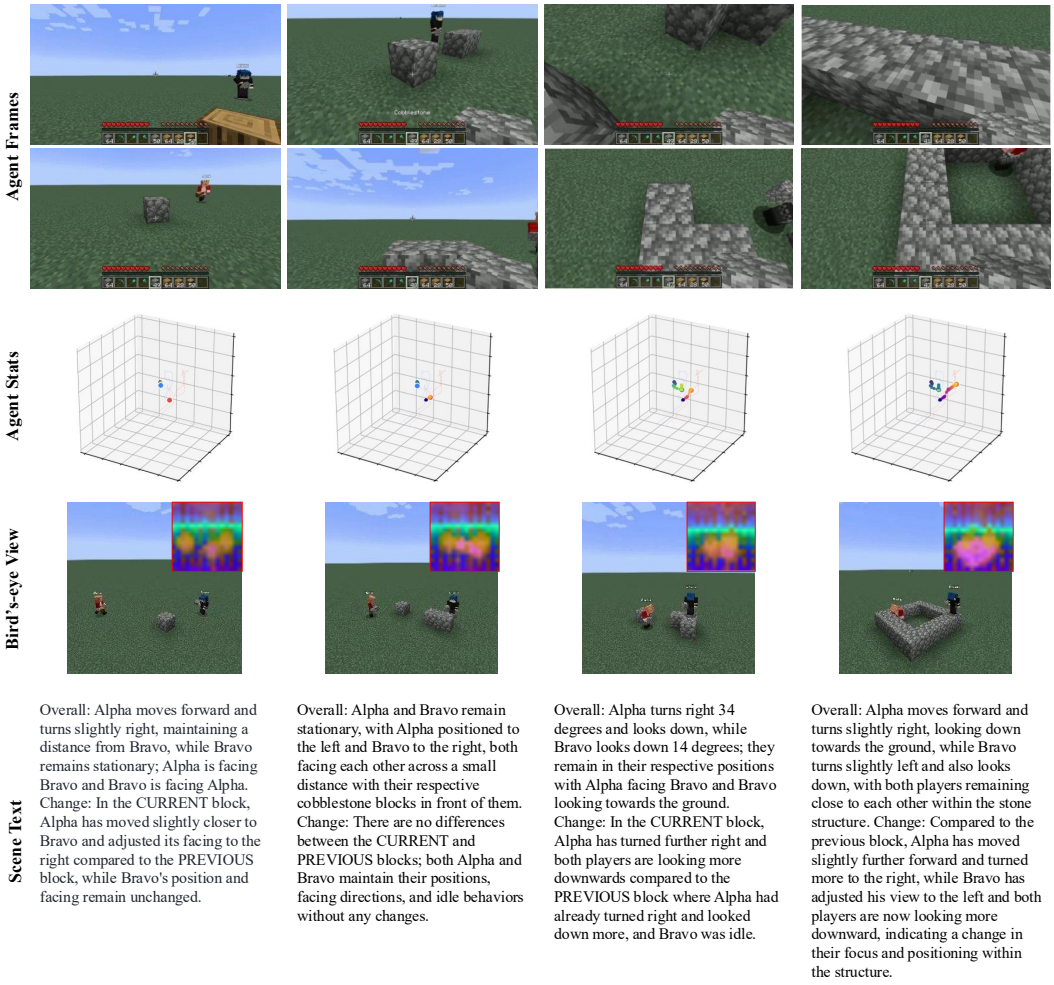


Figure 5: Visualization of the data collection and annotation streams used in our multi-agent Minecraft setting.

Figure 5 shows representative chunk inputs alongside their generated captions. Because this pipeline is built around each agent's local motion and controller inputs, the resulting captions describe per-agent behavior faithfully but do not capture broader scene context beyond the two agents, such as surrounding terrain or events outside either agent's immediate vicinity.

# B PIPELINE AND IMPLEMENTATION DETAILS

WorldWeaver turns a single player video diffusion prior into a streaming two player world model by adding an explicit persistent state pathway to the autoregressive rollout. The pipeline has three implementation components. First, committed world state registers are decoded during training with auxiliary heads for agent statistics, bird's-eye view features, and scene text. Second, a Mixture of Transformers backbone routes register tokens and player frame tokens through role specific parameter branches while preserving joint attention over the interleaved sequence. Third, training follows a three stage curriculum that first adapts the video prior to synchronized two player data, then trains a causal register student, and finally exposes the student to its own generated frames and committed registers under self-forcing.

## B.1 STATE DECODING

Table 5: Configuration of the three world state register supervision signals. Each signal has its own training only prediction head $h_m$, target $\mathbf{y}^m$, distance metric $d_m$, and loss weight $\lambda_m$. The combined setting activates all heads jointly and sums the per signal losses. Heads are discarded at inference.

| | Agent states | Bird's-eye view | Scene text | Combined |
|---|---|---|---|---|
| **Decoder Architecture** | | | | |
| Layers | 4 | 12 | 16 | - |
| Channels | 256 | 768 | 2048 | - |
| Query Number | 2 | 256 | - | - |
| WSR number | 256 | 256 | 256 | 64/64/128 |
| Weight Init | Random | Random | Pretrained | - |
| Learnable | True | True | False | - |
| **Optimization** | | | | |
| Distance metric $d_m$ | MSE | Cos. Sim. | Cross Entropy | - |
| S2: Loss weight $\lambda_m$ | 0.02 | 2.0 | 0.1 | 0.02/2.0/0.1 |
| S3: Loss weight $\lambda_m$ | 0.2 | 2.0 | 0.5 | 0.2/2.0/0.5 |

Table 5 summarizes the numeric configuration, and this subsection provides a more careful description of the state decoding supervision used to train the world state registers. The common pattern is to project committed world state register (WSR) tokens into the decoder space and place them in an attention sequence together with the tokens that will receive supervision. For the transformer based heads, this sequence is processed by full self attention; for scene text, the projected registers play the same role as prefix embeddings for the frozen language model. In this view, the projected registers serve as prefix like conditioning tokens: the remaining query, patch, or caption tokens attend to this state representation and are then scored by the corresponding target. The following paragraphs first describe this shared decoding pattern, then give the concrete targets and losses for agent statistics, bird's-eye view features, and scene text.

The objective in the main text already defines the weighted register loss, so we focus here on implementation details that are not explicit in the main description. Each head decodes every committed register step, with no pooling over WSR tokens. Single signal ablations may read the full register bank, whereas the combined setting uses disjoint $64/64/128$ slices. Missing targets are skipped for the affected head, and the implementation uses the per head weights in Table 5 directly, with no separate global state weight.

### B.1.1 AGENT STATISTICS DECODING

The appendix specifies the state vector used by the agent head. Each player is represented by position, velocity, and orientation, matching the agent-state target in the main text. Targets are selected at the latent aligned simulator frame and kept in raw units. We do not apply per field normalization, clipping, or temporal smoothing.

#### B.1.2 BIRD'S-EYE VIEW DECODING

The bird's-eye view signal is feature supervision rather than pixel reconstruction. The decoder predicts a $16\times16$ grid of 768 dimensional DINOv2 ViT-B/14 patch features from 256 learnable patch queries. The target top down frame is processed by the data pipeline and then by the frozen DINOv2 preprocessing path to $224\times224$; we use dense patch tokens rather than the CLS token or a pooled feature. PCA to RGB is used only for qualitative visualization.

#### B.1.3 SCENE TEXT DECODING

The scene text head should not be read as a direct vocabulary logit head. The trainable module projects the selected WSR slice into 2048 dimensional prefix embeddings for a frozen Llama-3.2-1B (Grattafiori et al., 2024) scorer; the frozen language model, not the trainable head, produces the next token logits. Captions are tokenized to 128 training slots, with padding and BOS ignored in the cross entropy. The caption targets are authored offline by Qwen2.5-VL-72B-Instruct (Bai et al., 2025), which is separate from the frozen Llama model used for supervision.

### B.2 MIXTURE OF TRANSFORMERS

Alongside the world state registers, we introduce a Mixture of Transformers (MoT) backbone into the autoregressive diffusion design. Importantly, MoT changes only how the generator's parameters are shared while the streaming rollout, the interleaved token layout, and the causal KV-cache all stay the same. Concretely, WSR tokens use the state branch, while the content frame tokens use the visual branch. In our implementation, one branch includes the transformer projection layers, feed forward, conditioning, and normalization layers. We initialize the state branch weights using the pretrained visual branch weights. It is worth noting that for self attention, although the register tokens and content frame tokens are encoded by different projection matrices, the attention weights are computed jointly over the interleaved sequence. Since the branches are shape matched, every token still activates only a single parameter copy, so per-token FLOPs are unchanged apart from routing overhead even as the total parameter count grows.

### B.3 TRAINING

Table 6: Hyperparameter setup for the three stage training curriculum used by the primary combined world state model. All stages use $P{=}2$ players. Entries marked "–" do not apply to that stage.

| | Stage 1<br>Bidirectional | Stage 2<br>Causal Training | Stage 3<br>Self-Forcing |
|---|---|---|---|
| **Architecture** | | | |
| Sliding window $W$ | – | 6 | 6 |
| Register tokens $K$ | – | 256 | 256 |
| Backbone | Dense | MoT | MoT |
| **Optimization** | | | |
| Training steps | 120K | 80K | 3-4K |
| Batch size | 32 | 32 | 32 |
| Optimizer | AdamW | AdamW | AdamW |
| $\beta$ | $(0.9, 0.95)$ | $(0.9, 0.95)$ | $(0.9, 0.95)$ |
| Learning rate | $1\times10^{-4}$ | $1\times10^{-4}$ | $3\times10^{-6}$ |

Table 6 lists the hyperparameters of the primary combined setting. Shared choices are $P{=}2$, $K{=}256$ registers with the $64/64/128$ head grouping, $W{=}6$ in causal stages, AdamW with bf16 mixed precision, and a batch size of 32.

#### B.3.1 STAGE 1: BIDIRECTIONAL TRAINING

We initialize from the official Solaris bidirectional checkpoint Savva et al. (2026). A player axis is added to observations, actions, VAE latents, and first-frame conditioning, and no register pathway is used. Training runs for 120K steps under Table 6.

### B.3.2 Stage 2: Causal Training with World State Registers

We first train a causal student without registers for 60K steps, then continue with WSR, MoT, and the three state heads for 20K steps. This 60K+20K schedule matches training with WSR for the full 80K steps in our checks. Following Solaris (Savva et al., 2026),we train this model with flow matching loss and register loss on ground-truth latents under Diffusion Forcing noise, with no live teacher or ODE rollout. Register weights follow Table 5 with no extra global multiplier.

### B.3.3 Stage 3: Self-Forcing with Context Frame and State Rollout

The generator loads the matching Stage-2 combined checkpoint. Both $s_{\text{real}}$ and $s_{\text{fake}}$ start from the Stage-1 bidirectional checkpoint; $s_{\text{real}}$ stays frozen and $s_{\text{fake}}$ is trained separately. Each update runs a no-gradient rollout on $1000 \to 750 \to 500 \to 250 \to 0$ with a shared early exit, then a gradient pass on the generated context for the DMD and register losses. Relative to Stage 2, agent-statistics and scene-text weights are raised and the BEV weight is unchanged (Table 5). Training lasts 3–4K steps. The MoT freeze schedule in Section 4.4 is a separate continuation that locks all non-register generator parameter groups.