MULTIGEN: LEVEL-DESIGN FOR EDITABLE MULTIPLAYER WORLDS IN DIFFUSION GAME ENGINES

Anonymous authors
Paper under double-blind review

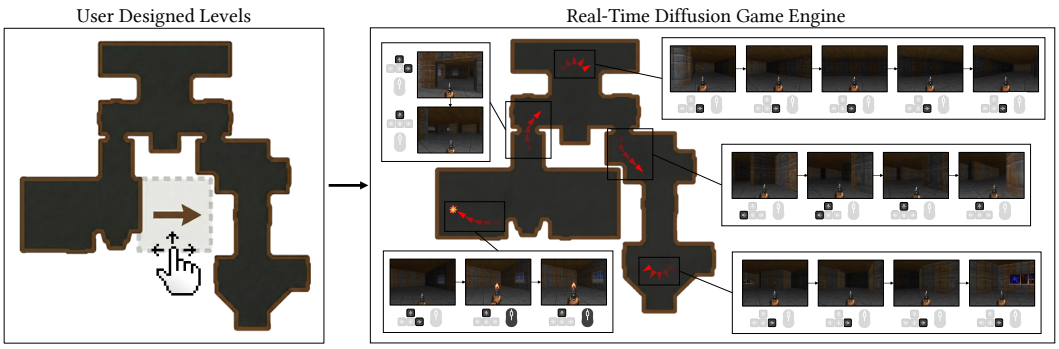


Figure 1: **Level Design via Editable Memory.** Users define a level through coarse 2D geometry (left). During inference, the diffusion model generates first-person observations consistent with the top-down level layout (right).

ABSTRACT

Video world models have shown immense promise for interactive simulation and entertainment, but current systems still struggle with two important aspects of interactivity: user control over the environment for reproducible, editable experiences, and shared inference where players hold influence over a common world. To address these limitations, we introduce an explicit external memory into the system, a persistent state operating independent of the model's context window, that is continually updated by user actions and queried throughout generation. Unlike conventional diffusion game engines that operate as next-frame predictors, our approach decomposes generation into Memory, Observation, and Dynamics modules. This design gives users direct, editable control over environment structure via the memory representation, and it naturally extends to real-time multiplayer rollouts with coherent viewpoints and consistent cross-player interactions.

1 INTRODUCTION

Recent advances in generative models have made real-time, action-controllable video generation increasingly practical, enabling interactive rollouts that resemble explorable worlds (Hafner et al., 2020; 2022; 2024; 2025; Mao et al., 2025; Ball et al., 2025; Parker-Holder et al., 2024; Bruce et al., 2024). However, most video world models are still fundamentally single-user experiences (Li et al., 2025a; Yu et al., 2025b; Zhang et al., 2025d; Che et al., 2024): they generate the world on the fly with only implicit internal state, making it difficult to support **shared** worlds where multiple players can reliably interact through a common underlying state. In parallel, the same limitation restricts **control**: creators have little ability to specify the structure of an environment up front, and long rollouts can become hard to steer, hard to reproduce, and misaligned with user intent.

We argue that both multiplayer interactivity and environment authoring point to a missing primitive: an explicit external memory that persists beyond the model's context window and is updated by user actions. With external memory, multiple agents can read and write a shared state to support consistent cross-viewpoint interaction, and users can directly edit the state to control global structure.

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.

In this paper, we introduce a memory-augmented diffusion world model that leverages this idea for real-time interactive generation.

We use Doom (id Software, 1993) as a controlled but expressive testbed. Doom offers rich first-person dynamics while retaining a clear notion of level layout, making it well suited for studying shared, long-horizon rollouts. To provide a simple and general interface for environment control, we represent world structure as a top-down 2D minimap that can be authored or edited before interaction begins. This minimap serves as an external memory blueprint that anchors generation while leaving the model free to synthesize detailed observations and moment-to-moment interactions.

Implementing external memory requires moving beyond the single-model paradigm used by prior diffusion game engines (Valevski et al., 2024; Yu et al., 2025b; Zhang et al., 2025d; Che et al., 2024; Mao et al., 2025; Ball et al., 2025; Parker-Holder et al., 2024; Bruce et al., 2024). Instead, we decompose the system into three specialized modules: **Memory**, **Observation**, and **Dynamics**. The memory module maintains a persistent state (including the minimap and agent states). The observation module generates the next visual observation conditioned on the memory readout and recent history. The dynamics module updates state given actions and observations. This separation makes long-horizon structure easier to maintain and, critically, enables multiplayer in a natural way: multiple agents act on the same shared memory, and the model can render coherent observations from one or more viewpoints with interaction effects between players.

Together, these choices support (i) **real-time multiplayer rollouts** grounded in shared external memory, and (ii) a simple **level-design workflow** where users specify or edit a minimap and obtain consistent, reproducible interactive sessions. In summary, our contributions are:

- We introduce an external-memory-based formulation for diffusion world models that supports shared state updates from user actions, enabling consistent long-horizon rollouts.
- We propose a modular architecture with Memory, Observation, and Dynamics modules, replacing the single-model paradigm and providing a clean interface for read/write memory.
- We demonstrate two applications enabled by external memory: editable environment design via a minimap blueprint, and real-time multiplayer interaction with coherent cross-viewpoint behavior, and evaluate each against relevant baselines.

2 Related Work

Video Diffusion Models. Current state-of-the-art video generation models are large-scale bidirectional diffusion transformers (Peebles & Xie, 2022; Brooks et al., 2024) that denoise all video tokens simultaneously under full spatiotemporal attention (Blattmann et al., 2023a;b; Brooks et al., 2024; Li et al., 2024a; Gupta et al., 2023; HaCohen et al., 2024; Ho et al., 2022a;b; Kong et al., 2024; Polyak et al., 2024; Villegas et al., 2022; Wang et al., 2025; Hong et al., 2022; Yang et al., 2024; Bar-Tal et al., 2024; Cai et al., 2025; Zhang et al., 2025a), achieving remarkable quality but remaining inherently non-causal and constrained to fixed-length generation.

Autoregressive Video Models. Autoregressive approaches instead factorize the distribution over frames as $p(x^{1:N}) = \prod_{i=1}^{N} p(x^i \mid x^{<i})$, aligning naturally with the causality of time and making them well suited for interactive simulation (Shin et al., 2025; Huang et al., 2025b). Because direct next-token prediction of discrete video tokens (Bruce et al., 2024; Kondratyuk et al., 2023; Ren et al., 2025; Wang et al., 2024; Weissenborn et al., 2019; Yan et al., 2021) lags behind diffusion in quality, recent works combine autoregression and diffusion (Gao et al., 2024; Gu et al., 2025; Guo et al., 2025; Hu et al., 2024b; Jin et al., 2024; Li et al., 2024b; Liu et al., 2024a;b; Weng et al., 2023; Zhang et al., 2025c;e), either denoising next frames conditioned on past clean frames (Zhang et al., 2025b; Huang et al., 2025b) or training with per-frame independent noise levels (Chen et al., 2024; Song et al., 2025; Valevski et al., 2024).

Diffusion Game Engines. Our work is most closely related to recent efforts that repurpose diffusion models as real-time game engines or world simulator (Li et al., 2025a; Yu et al., 2025b; Zhang et al., 2025d; Che et al., 2024; Mao et al., 2025; Ball et al., 2025; Parker-Holder et al., 2024; Bruce et al., 2024). GameNGen (Valevski et al., 2024) generates gameplay by conditioning an image diffusion model on a short history of previous frames and the next action, using diffusion as an

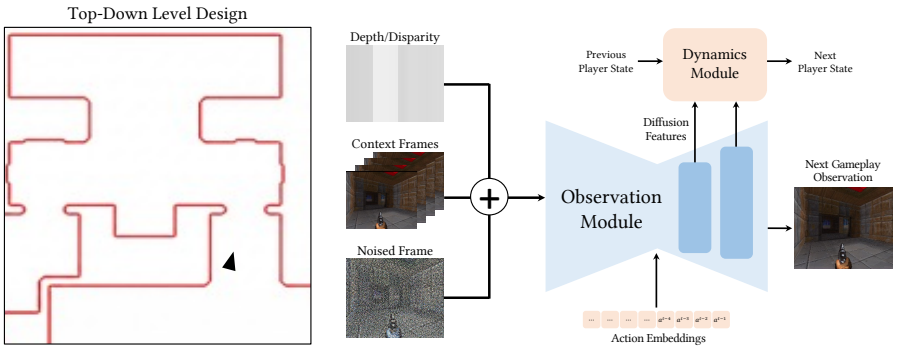


Figure 2: **Method overview.** We introduce an explicit external memory and factor the diffusion game engine into three modules: ***Memory*** (map geometry and pose; Sec. 3.1), ***Observation*** (next-frame generation conditioned on history and memory readouts; Sec. 3.2), and ***Dynamics*** (pose update for state progression; Sec. 3.3).

autoregressive next-frame generator. We build on this by introducing a modular diffusion game engine with explicit external memory. This design provides a persistent reference to the authored level layout, improving structural adherence over long horizons and enabling practical level-conditioned generation from coarse user edits.

Video Models with External Memory. External memory predates diffusion models, appearing in generative temporal models (Gemici et al., 2017), GAN-based game and driving simulators (Kim et al., 2020; 2021), and GAN-based image synthesis (Kim et al., 2018; Zhu et al., 2019). While existing video frameworks typically construct memory using explicit 3D representations (Wu et al., 2025a; Deng et al., 2024; Zhao et al., 2025) or manage input conditions through compressed context windows and KV caching (Huang et al., 2025a; Wu et al., 2026; Gu et al., 2025; Yu et al., 2025a; Li et al., 2025b; Hong et al., 2025; Wu et al., 2025b), our approach develops an external memory that persists beyond the model's context window. Updated by user actions, this memory provides a shared state for multiple agents to read and write, ensuring consistent cross-viewpoint interaction while allowing users to directly edit the state to govern the global structure.

Game Generation. A complementary line of work generates game content itself—levels, assets, or mechanics—using procedural methods (Summerville et al., 2017; Treanor et al., 2015; Snodgrass & Ontañón, 2014; Guzdial & Riedl, 2021; Summerville & Mateas, 2016), GANs (Volz et al., 2018; Kumaran et al., 2020; Schubert et al., 2021; Kim et al., 2020), diffusion models (Valevski et al., 2024; Zhou et al., 2024), and LLMs (Sudhakaran et al., 2023; Todd et al., 2023; Nasir & Togelius, 2023; Hu et al., 2024a; Zala et al., 2024; Anjum et al., 2024; Chung & Kreminski, 2024; Chung et al., 2024). Our model instead operates as a game engine: it generates every observed frame online through a diffusion model while giving users direct control over the environment.

3 METHOD

We model an interactive game rollout as an alternating sequence of observations and actions $\tau = (o_0, a_0, o_1, a_1, \ldots, a_{T-1}, o_T)$, where $a_t \in \mathcal{A}$ denotes the agent action and $o_t \in \mathbb{R}^{H \times W \times C}$ the rendered observation at timestep t. A diffusion game engine predicts $p(o_{t+1} \mid o_{\leq t}, a_{\leq t})$, enabling closed-loop simulation by repeatedly sampling o_{t+1} and feeding it back as input. Following GameNGen (Valevski et al., 2024), this predictor is instantiated as a conditional model $o_{t+1} \sim p_\theta(o \mid S_t, a_t)$, where the state $S_t = o_{t-L+1:t}$ is represented implicitly by a fixed window of the L most recent frames, providing the visual history needed for appearance and short-term continuity, while a_t drives the transition.

In our work, we retain this interactive generative modeling setup but go beyond the conventional "frames-as-state" design (Hafner et al., 2020; 2022; 2024; 2025; Mao et al., 2025; Ball et al., 2025; Parker-Holder et al., 2024; Bruce et al., 2024) by introducing an explicitly controlled external memory (Wu et al., 2025a; Deng et al., 2024; Zhao et al., 2025). Rather than requiring a single network

to both (i) maintain long-horizon information about the environment and agent configuration and (ii) generate high-dimensional observations, we factor the diffusion game engine into three components: a ***memory module*** that stores structured map geometry and the player pose (3.1), an ***observation module*** that predicts the next frame conditioned on the visual context and memory-derived geometric signals (3.2), and a ***dynamics module*** that updates the player pose to advance the state over time (3.3). This modularization separates persistent, low-dimensional game state from high-dimensional image generation, leading to a more controllable and interpretable simulation process.

3.1 Memory Module

We maintain an explicit state that separates persistent, low-dimensional game information from high-dimensional observations. Concretely, at timestep t we define the state as

$$S_t = \big(M,\ p_t,\ o_{t-L+1:t}\big), \tag{1}$$

where $o_{t-L+1:t} = (o_{t-L+1}, \ldots, o_t)$ denotes an L-frame visual context, M denotes the (static) level map, and p_t denotes the player pose.

We represent the map M as 2D vertices and line segments defining the walkable layout and walls:

$$\begin{aligned} M &= (V, E), \\ V &= \{v_i\}_{i=1}^{N_v}, \qquad v_i \in \mathbb{R}^2, \\ E &= \{e_j\}_{j=1}^{N_e}, \qquad e_j \in \{1, \ldots, N_v\}^2, \end{aligned} \tag{2}$$

where each $e_j = (u_j, w_j)$ indexes an (undirected) line segment connecting vertices v_{u_j} and v_{w_j}.

Intuitively, M serves as a reliable, persistent external reference for the generative game engine. In a "frames-as-state" design, all information about the environment must be preserved implicitly within a finite visual context $o_{t-L+1:t}$; as rollouts grow longer, relevant layout cues may fall out of the buffer, forcing the model to hallucinate or re-infer structure from incomplete evidence. In contrast, the map M is time-invariant and can always be consulted to provide a coarse but stable description of the current level. This persistent signal simplifies long-horizon consistency by giving the model an explicit representation of global geometry that does not degrade with context length.

Player information is parameterized by their Euclidean coordinates and yaw angle, $p_t = (x_t, y_t, \theta_t) \in \mathbb{R}^2 \times \mathbb{S}^1$, with (x_t, y_t) denoting the player location in map coordinates and θ_t denoting the facing direction. The memory module maintains the static map M and the evolving pose p_t throughout the rollout, and provides these quantities to the observation and dynamics modules as part of S_t. In the following sections, we describe how the map and pose are used to compute auxiliary geometric signals (e.g., a ray-traced 1D depth observation) for frame prediction, and how player information is updated over time.

3.2 Observation Module

The observation module generates the next observation conditioned on recent context, a geometric readout from external memory, and the next action. At timestep t, we model $o_{t+1} \sim p_\phi(o \mid o_{t-L+1:t},\ r_t,\ a_t)$, where r_t is the geometric readout from the memory module (Sec. 3.1).

Geometric conditioning via disparity. Given the current pose and map, the memory module ray-traces a 1D depth vector within the agent's field of view and converts it to disparity (inverse depth) to emphasize near-field structure. We then feed geometry to the UNet by mapping the 1D disparity (Deng et al., 2024) to a spatial tensor at the UNet input resolution and concatenating it as additional channels alongside the L context frames.

Action conditioning. We represent the discrete action a_t with a learned embedding and inject it into the observation UNet through cross-attention tokens. This allows the denoiser to modulate generation according to the agent's control input without changing the convolutional input interface.

Diffusion objective. We instantiate p_ϕ as a diffusion model (Ho et al., 2020) trained with the standard velocity-parameterization objective (Salimans & Ho, 2022). Given a ground-truth next frame o_{t+1}, we sample a diffusion timestep τ, construct a noised version according to the forward

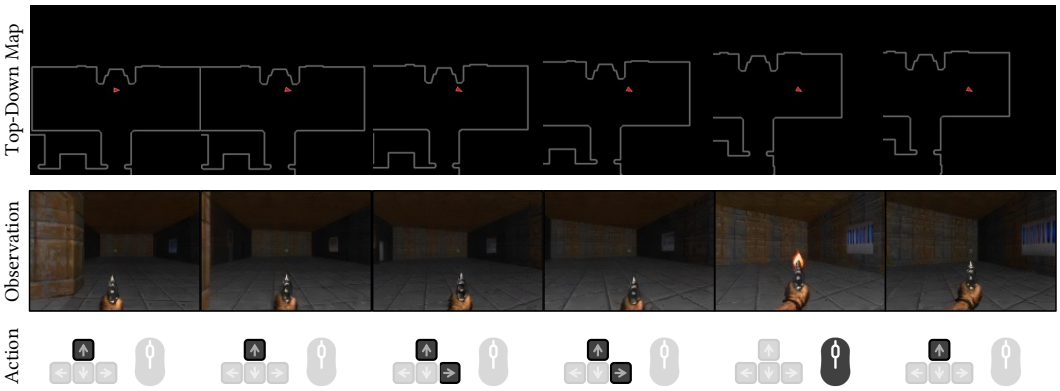


Figure 3: Example rollouts under an authored map and action sequence: minimap M with pose p_t (top, red arrow), generated first-person observations $\hat{o}_t$ (middle), and actions a_t (bottom).

process, and train the UNet to match the corresponding velocity target. Concretely, we optimize

$$\mathcal{L}_{\text{obs}} = \mathbb{E}_{t,\tau}\left[\left\|v_\phi(\cdot \mid o_{t-L+1:t}, r_t, a_t; \tau) - v^\star(o_{t+1}, \tau)\right\|_2^2\right], \tag{3}$$

where $v^\star$ denotes the standard velocity target associated with the chosen diffusion parameterization.

Noised-context training for drift robustness. During training, the observation module conditions on ground-truth context frames, whereas at test time it conditions on its own generated history. To reduce this train–test mismatch (Huang et al., 2025b; Cui et al., 2025; Liu et al., 2025; Po et al., 2025), we follow prior work (Chen et al., 2024; Valevski et al., 2024) and corrupt all context frames with Gaussian noise at a randomly sampled noise scale during training. This exposes the model to imperfect histories and improves robustness under long autoregressive rollouts.

3.3 Dynamics Module

To advance an interactive rollout beyond single-step prediction, the system must update the agent state used by external memory. In our setting, this reduces to predicting the next player pose $p_t = (x_t, y_t, \theta_t)$, where θ_t is wrapped to a canonical range.

Inputs. The dynamics module consumes (i) the action a_t, (ii) the same geometric conditioning signal r_t used by the observation module, and (iii) intermediate UNet features produced while denoising the next frame (e.g., a bottleneck feature map). We aggregate these UNet features into a fixed-size representation and concatenate them with embeddings of pose, action, and geometry.

Lightweight transformer dynamics. We implement dynamics as a small transformer encoder that predicts an incremental pose update $\Delta\hat{p}_t = \mathcal{D}_\psi(p_t,\, a_t,\, r_t,\, f_t)$, where f_t denotes the aggregated UNet features. We apply this increment to obtain $\hat{p}_{t+1}$, using angle wrapping for orientation.

Training objective and state update. We supervise dynamics using ground-truth poses from the environment. In practice, we use an ℓ_2 loss on translation and a wrapped-angle error on orientation. After predicting $\hat{p}_{t+1}$, the external memory state is advanced by updating the pose and shifting the visual context window to include the newly generated frame.

3.4 Inference

At inference, the system acts as an interactive simulator mapping $(S_t, a_t) \mapsto (\hat{o}_{t+1}, S_{t+1})$: memory provides the geometric readout r_t, the observation model samples $\hat{o}_{t+1}$, and the dynamics module updates the pose. To stabilize long rollouts, we use history guidance (Song et al., 2025): the conditional branch receives clean context frames while the unconditional branch receives a noised version, encouraging fidelity to recent history while retaining robustness to imperfect inputs.

Table 1: SSIM/PSNR/LPIPS between generated frames and ground truth. Our method consistently outperforms baselines, with the largest gains appearing in later rollout stages where consistent memory is the most important.

	SSIM ↑			PSNR ↑			LPIPS ↓		
Method	All	1–128	128–256	All	1–128	128–256	All	1–128	128–256
IP-Adapter (Ye et al., 2023)	0.415	**0.433**	0.396	18.74	**20.30**	17.19	0.488	0.397	0.578
ControlNet (Zhang et al., 2023a)	0.411	0.422	0.401	18.51	19.58	17.45	0.524	0.453	0.596
GameNGen (Valevski et al., 2024)	0.405	0.427	0.384	18.77	20.23	17.33	0.471	**0.379**	0.562
Ours (MultiGen)	**0.418**	0.429	**0.406**	**19.32**	20.06	**18.59**	**0.453**	0.400	**0.505**

4 Application I: Level Design

External memory provides a direct handle for modifying the underlying structure of the world before inference even begins. We evaluate this capability on ***level-conditioned gameplay generation***: given a top-down map M specified as coarse line geometry (vertices and wall segments), an initial player pose p_0, and actions $\{a_t\}_{t=0}^{T-1}$, the model synthesizes first-person observations $\{\hat{o}_t\}_{t=1}^{T}$ consistent with the user-authored layout. Unlike implicit-state diffusion game engines that must infer global structure from a finite history, our approach can query M at every timestep, providing a persistent geometric reference throughout the roll-out.

4.1 Level Design Dataset

To train our model to generalize across a diverse set of level layouts, we generate gameplay sequences on 100 procedurally generated maps with randomized structure. We create these maps using the Obsidian map generator (Obsidian Community, 2026), which produces varied layouts while preserving valid Doom geometry. We then deploy a pre-trained Doom agent (Lample & Chaplot, 2018) to explore the resulting maps, collecting over 10 million gameplay frames paired with the corresponding actions and player poses. This dataset exposes the model to a wide range of corridor and room configurations, encouraging it to rely on external memory for global structure rather than memorizing a small set of fixed levels.

4.2 Results

Fig. 3 shows representative rollouts conditioned on a designed map and action sequence. The top row visualizes the authored layout and the evolving player pose, the middle row shows generated first-person observations, and the bottom row shows the applied action inputs. Across these examples, the viewpoint evolves consistently with the action inputs, while adhering to the general structure of the layout specified by M. Qualitatively, this demonstrates that coarse user geometry is sufficient to anchor long rollouts: the model maintains stable corridor structure, respects turns represented by the map, and avoids any structural drift that occurs when global layout must be inferred from a limited frame-based visual history.

4.3 Evaluations

We compare against GameNGen (Valevski et al., 2024), which models the entire engine as a single diffusion network with state represented implicitly by a finite window of past frames, and against ControlNet (Zhang et al., 2023b) and IP-Adapter (Ye et al., 2023), which condition on the top-down minimap. All methods start from the same initial observation and pose and roll out under the same action sequence for T steps; we measure SSIM (structural similarity) and LPIPS (perceptual distance) against ground-truth observations, averaged over early and late rollout ranges to assess long-horizon stability.

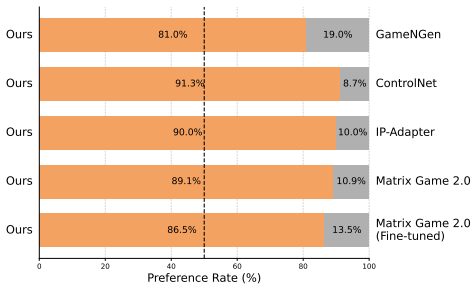


Figure 4: **Level-Design User Study.** 2AFC preference rates over 200+ A/B pairs per method; ours is consistently preferred.

As shown in Tab. 1, external memory improves structural consistency, with the largest gains in later rollout segments where implicit-state baselines drift: the authored map anchors generation to the

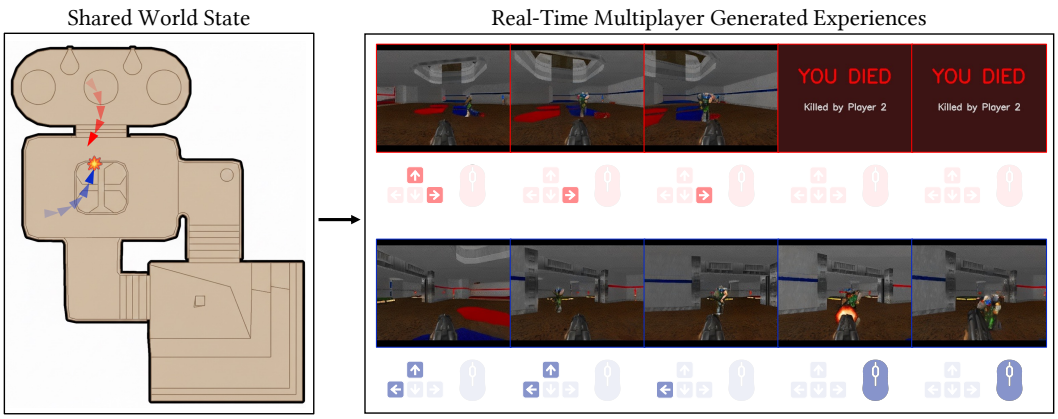


Figure 5: **Real-Time Interactive Multiplayer Generative Experiences.** A shared world state (left) enables consistent multiplayer roll-outs (right), including meaningful cross-player interactions such as one player killing another.

true layout, whereas the memory-free baseline hallucinates layout changes that compound under autoregressive sampling.

We supplement these metrics with a user study measuring layout adherence: in a two-alternative forced-choice protocol, participants are shown a target map and select which of two generated videos better aligns with it. Across over 200 A/B pairs per method, ours is consistently preferred (Fig. 4). Under the same protocol, ours is also favored over the state-of-the-art open-source world model Matrix-Game 2.0 (He et al., 2026), with and without fine-tuning on our data (Fig. 9, in the supplement).

5 Application II: Multiplayer Interaction

The external memory also naturally acts as a shared state that multiple agents can condition on and update during generated roll-outs. We evaluate this capability on ***multiplayer gameplay generation***: given a shared map M, initial poses $\{p_0^{(i)}\}_{i=1}^{N}$, and per-player actions $\{a_t^{(i)}\}_{t=0}^{T-1}$, the model generates per-player observations $\{\hat{o}_t^{(i)}\}_{t=1}^{T}$, with all players querying and modifying the same memory so that one player's actions influence what other players observe. This is in direct contrast with "frames-as-state" models (Hafner et al., 2020; 2022; 2024; 2025; Mao et al., 2025; Ball et al., 2025; Parker-Holder et al., 2024; Bruce et al., 2024; team, 2025), where state is entangled with each player's local observation history, making cross-player consistency difficult to maintain over long horizons.

5.1 Shared Memory for Multiplayer Roll-outs

In our multiplayer setting, the shared world state is represented explicitly by the external memory: the static map layout M together with the set of active player poses $\{p_t^{(i)}\}_{i=1}^{N_t}$. Generation is **distributed**: each player runs their own copy of the Observation and Dynamics modules, while all players read from and write to the same shared memory. At each timestep, every player submits an action $a_t^{(i)}$ and queries the shared state to obtain viewpoint-specific conditioning signals, including geometric depth/disparity from the map and information about other players that are visible from that pose. Conditioned on these shared-memory readouts, each player's Observation module generates the next first-person frame $\hat{o}_{t+1}^{(i)}$ for that player. After all players have generated their next frames, we update the shared state by applying each player's action through their Dynamics module, advancing the set of poses (and updating which players are active).

Crucially, this distributed design supports an arbitrary number of players without changing the model interface. In contrast, a simple joint baseline that models all viewpoints in a single "split-screen" observation (team, 2025) must fix the number of players at training time. Our approach naturally supports gameplay events such as player death and respawn by removing or reintroducing player poses in the shared memory.

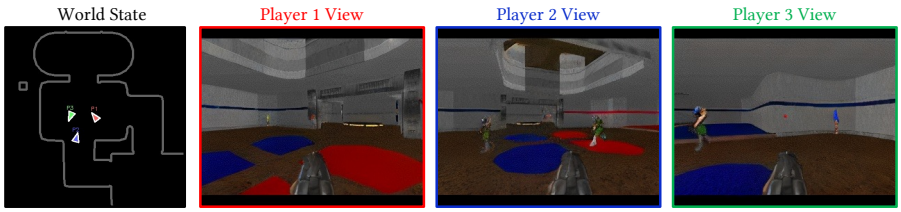


Figure 7: **Consistent Views in Three-Player.** Our distributed model system supports an arbitrary number of players in the shared state despite only being trained on data from a single viewpoint. Player positions are accurately represented in all three viewpoints in the example above.

5.2 Multiplayer Dataset

Because each model instance only generates its own first-person viewpoint, we can train on standard single-view gameplay data: consistent multiplayer viewpoints arise from shared external memory, without requiring explicit multi-view supervision. We build on ViZDoom (Wydmuch et al., 2019) and collect simulated Doom deathmatch sequences in which one pre-trained agent (Lample & Chaplot, 2018) plays against four identical agents. In total, we gather over 10 million gameplay frames. Each sequence records the ego-agent's action, the poses of all active players, and the map layout M. For this multiplayer study, we train and evaluate on a single map.

5.3 Results

Fig. 6 shows a two-player rollout with a typical gameplay loop: the players meet, Player 1 kills Player 2 (removing them from the shared state and triggering a death animation), and Player 2 respawns, re-enters the shared state, and is killed again in a second encounter. These interactions remain **mutually consistent** across viewpoints despite being generated autoregressively: players appear with the correct pose and location when visible, and disappear from the other player's camera when dead or out of view, while the victim observes a coherent death/respawn sequence.

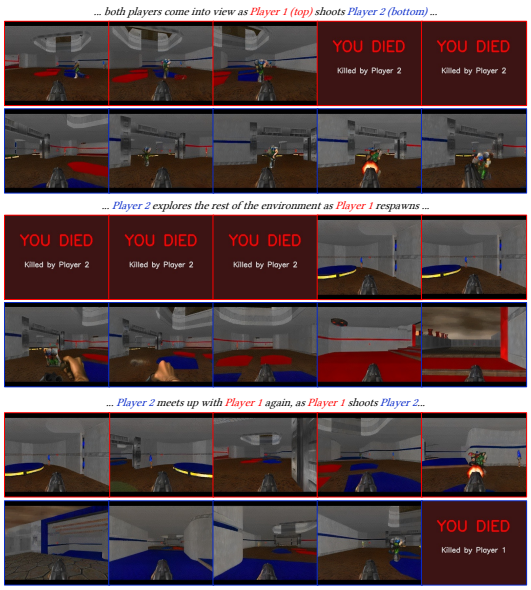


Figure 6: **Two-Player Roll-out.** Kill, respawn, and re-encounter events remain mutually consistent across both players' viewpoints. Death frames are shown for illustration only.

Because each player's model runs independently, adding more players does not slow down inference: the full system runs at approximately **20 FPS** using a single NVIDIA A100 per player, making multiplayer rollouts interactive in real time. Fig. 7 shows three consistent viewpoints aligned with the shared state: Player 1 sees no other players, Player 2 sees both other players, and Player 3 sees only Player 2. Real-time rollouts with up to eight players are shown in the supplement (Fig. 8).

5.4 Evaluations

We compare to ControlNet (Zhang et al., 2023b), IP-Adapter (Ye et al., 2023), and a "split-screen" baseline (team, 2025): a two-player diffusion model trained to jointly predict both viewpoints on the same map without an explicit memory mechanism, so that cross-player consistency must be learned purely from the players' limited observation histories.

We roll out all models on random multiplayer trajectories from the environment, forcing each player to follow the ground-truth action stream, and measure multiplayer consistency via ***opponent presence accuracy***: whether a generated frame contains the other player when they should be visible. A pre-trained vision-language model (VLM) (OpenAI et al., 2024) judges opponent presence in each generated frame, and accuracy is computed by agreement with the same judgement on the

Table 2: Multiplayer consistency evaluation. We report accuracy, precision, and recall of a VLM-based judge on generated frames against ground-truth visibility labels. We additionally report spatial consistency as the mean IoU between user-annotated and ground-truth opponent bounding boxes (250 annotated frames per method). MultiGen consistently outperforms all baselines.

Method	Accuracy ↑	Precision ↑	Recall ↑	Mean IoU ↑
IP-Adapter (Ye et al., 2023)	62.12%	84.62%	40.73%	0.246
ControlNet (Zhang et al., 2023b)	60.71%	82.16%	39.20%	0.166
Split-screen (team, 2025)	65.31%	86.86%	44.59%	0.207
Ours (MultiGen)	**75.38%**	**88.12%**	**65.07%**	**0.604**

Table 3: Module ablation. Removing the Memory module reduces the model to a frames-as-state architecture analogous to GameNGen, while removing the Dynamics module leaves the player pose static, causing a mismatch between generated frames and ray-traced depth. The full model outperforms both variants.

	SSIM ↑			PSNR ↑			LPIPS ↓		
Method	All	1–128	128–256	All	1–128	128–256	All	1–128	128–256
Ours (MultiGen)	**0.418**	**0.429**	**0.406**	**19.32**	20.06	**18.59**	**0.453**	0.400	**0.505**
w/o Memory	0.405	0.427	0.384	18.77	**20.23**	17.33	0.471	**0.379**	0.562
w/o Dynamics	0.394	0.410	0.389	18.36	19.08	18.50	0.535	0.487	0.541

corresponding ground-truth frame. We report **accuracy**, **precision**, and **recall** (opponent visible = positive): precision penalizes hallucinated opponents, recall missed ones.

Presence accuracy does not assess whether opponents appear at the correct location. We therefore also collect 250 user-annotated bounding boxes per method and compare them against ground-truth boxes using mean intersection-over-union (IoU). As shown in Tab. 2, MultiGen substantially outperforms all baselines, indicating that geometric conditioning from the shared memory places visible opponents at the correct position and distance (a reference comparison at $\approx$0.5 IoU is shown in Fig. 11, in the supplement).

6 Discussion and Conclusions

Ablation. We ablate the individual modules of our architecture (Tab. 3). Without the Memory module, the model reduces to a frames-as-state architecture analogous to GameNGen (Valevski et al., 2024); without the Dynamics module, the player pose remains static, causing a mismatch between generated frames and the ray-traced depth conditioning. The full model outperforms both variants. We do not ablate the Observation module, since no frames can be generated without it. An additional ablation over the number of context frames is provided in the supplement (Tab. 4).

Limitations. Our approach relies on the explicit state for long-horizon consistency. Consequently, scene properties not represented in the map M (e.g., textures or small objects) are not explicitly preserved when revisiting the same region, which can lead to appearance inconsistencies. The dynamics model is also imperfect, so small pose errors may accumulate over long rollouts. However, actions remain aligned with plausible motion and the overall gameplay experience is preserved. Finally, visual appearance is bounded by the training distribution and may not generalize to styles far outside the collected VizDoom trajectories.

Conclusion. We present MultiGen, a diffusion game engine built around explicit external memory, enabling level-conditioned gameplay generation and real-time multiplayer interaction in a shared world. Our system combines (i) external memory that stores map geometry and the evolving set of player poses, (ii) an observation model conditioned on ray-traced disparity and actions, and (iii) a lightweight dynamics model that updates pose to advance the roll-out. This decomposition supports controllable level design from coarse layouts and improves structural adherence, while also providing a scalable interface for multi-player roll-outs where each player generates a consistent first-person view conditioned on the same underlying state. We believe this modular, memory-centric formulation is a step toward more controllable and extensible generative game engines.

REFERENCES

Asad Anjum, Yuting Li, Noelle Law, M Charity, and Julian Togelius. The ink splotch effect: A case study on chatgpt as a co-creative game designer. *Proceedings of the 19th International Conference on the Foundations of Digital Games*, 2024. URL https://api.semanticscholar.org/CorpusID:268249064.

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

Omer Bar-Tal, Hila Chefer, Omer Tov, Charles Herrmann, Roni Paiss, Shiran Zada, Ariel Ephrat, Junhwa Hur, Yuanzhen Li, Tomer Michaeli, Oliver Wang, Deqing Sun, Tali Dekel, and Inbar Mosseri. Lumiere: A space-time diffusion model for video generation. *SIGGRAPH Asia 2024 Conference Papers*, 2024. URL https://api.semanticscholar.org/CorpusID:267095113.

A. Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, and Dominik Lorenz. Stable video diffusion: Scaling latent video diffusion models to large datasets. *ArXiv*, abs/2311.15127, 2023a. URL https://api.semanticscholar.org/CorpusID:265312551.

A. Blattmann, Robin Rombach, Huan Ling, Tim Dockhorn, Seung Wook Kim, Sanja Fidler, and Karsten Kreis. Align your latents: High-resolution video synthesis with latent diffusion models. *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 22563–22575, 2023b. URL https://api.semanticscholar.org/CorpusID:258187553.

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

Jake Bruce, Michael Dennis, Ashley Edwards, Jack Parker-Holder, Yuge Shi, Edward Hughes, Matthew Lai, Aditi Mavalankar, Richie Steigerwald, Chris Apps, Yusuf Aytar, Sarah Bechtle, Feryal M. P. Behbahani, Stephanie Chan, Nicolas Manfred Otto Heess, Lucy Gonzalez, Simon Osindero, Sherjil Ozair, Scott Reed, Jingwei Zhang, Konrad Zolna, Jeff Clune, Nando de Freitas, Satinder Singh, and Tim Rocktaschel. Genie: Generative interactive environments. *ArXiv*, abs/2402.15391, 2024. URL https://api.semanticscholar.org/CorpusID:267897982.

Shengqu Cai, Ceyuan Yang, Lvmin Zhang, Yuwei Guo, Junfei Xiao, Ziyan Yang, Yinghao Xu, Zhenheng Yang, Alan Yuille, Leonidas J. Guibas, Maneesh Agrawala, Lu Jiang, and Gordon Wetzstein. Mixture of contexts for long video generation. *ArXiv*, abs/2508.21058, 2025. URL https://api.semanticscholar.org/CorpusID:280950315.

Haoxuan Che, Xuanhua He, Quande Liu, Cheng Jin, and Hao Chen. Gamegen-x: Interactive open-world game video generation. *ArXiv*, abs/2411.00769, 2024. URL https://api.semanticscholar.org/CorpusID:273798141.

Boyuan Chen, Diego Marti Monso, Yilun Du, Max Simchowitz, Russ Tedrake, and Vincent Sitzmann. Diffusion forcing: Next-token prediction meets full-sequence diffusion. *ArXiv*,

abs/2407.01392, 2024. URL https://api.semanticscholar.org/CorpusID:270869622.

John Joon Young Chung and Max Kreminski. Patchview: Llm-powered worldbuilding with generative dust and magnet visualization. *Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology*, 2024. URL https://api.semanticscholar.org/CorpusID:271769283.

John Joon Young Chung, Melissa Roemmele, and Max Kreminski. Toyteller: Toy-playing with character symbols for ai-powered visual storytelling. *Adjunct Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology*, 2024. URL https://api.semanticscholar.org/CorpusID:273290335.

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*, abs/2510.02283, 2025. URL https://api.semanticscholar.org/CorpusID:281724666.

Boyang Deng, Richard Tucker, Zhengqi Li, Leonidas J. Guibas, Noah Snavely, and Gordon Wetzstein. Streetscapes: Large-scale consistent street view generation using autoregressive video diffusion. *ACM SIGGRAPH 2024 Conference Papers*, 2024. URL https://api.semanticscholar.org/CorpusID:271165095.

Kaifeng Gao, Jiaxin Shi, Hanwang Zhang, Chunping Wang, Jun Xiao, and Long Chen. Ca2-vdm: Efficient autoregressive video diffusion model with causal generation and cache sharing. *ArXiv*, abs/2411.16375, 2024. URL https://api.semanticscholar.org/CorpusID:274235095.

Mevlana Gemici, Chia-Chun Hung, Adam Santoro, Greg Wayne, Shakir Mohamed, Danilo Jimenez Rezende, David Amos, and Timothy P. Lillicrap. Generative temporal models with memory. *ArXiv*, abs/1702.04649, 2017. URL https://api.semanticscholar.org/CorpusID:2801340.

Yuchao Gu, Weijia Mao, and Mike Zheng Shou. Long-context autoregressive video modeling with next-frame prediction. *ArXiv*, abs/2503.19325, 2025. URL https://api.semanticscholar.org/CorpusID:277313237.

Yuwei Guo, Ceyuan Yang, Ziyan Yang, Zhibei Ma, Zhijie Lin, Zhenheng Yang, Dahua Lin, and Lu Jiang. Long context tuning for video generation. *ArXiv*, abs/2503.10589, 2025. URL https://api.semanticscholar.org/CorpusID:276961453.

Agrim Gupta, Lijun Yu, Kihyuk Sohn, Xiuye Gu, Meera Hahn, Fei-Fei Li, Irfan Essa, Lu Jiang, and José Lezama. Photorealistic video generation with diffusion models. In *European Conference on Computer Vision*, 2023. URL https://api.semanticscholar.org/CorpusID:266163109.

Matthew J. Guzdial and Mark O. Riedl. Game level generation from gameplay videos. In *Artificial Intelligence and Interactive Digital Entertainment Conference*, 2021. URL https://api.semanticscholar.org/CorpusID:37001924.

Yoav HaCohen, Nisan Chiprut, Benny Brazowski, Daniel Shalem, David-Pur Moshe, Eitan Richardson, E. I. Levin, Guy Shiran, Nir Zabari, Ori Gordon, Poriya Panet, Sapir Weissbuch, Victor Kulikov, Yaki Bitterman, Zeev Melumian, and Ofir Bibi. Ltx-video: Realtime video latent diffusion. *ArXiv*, abs/2501.00103, 2024. URL https://api.semanticscholar.org/CorpusID:275212083.

Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination, 2020. URL https://arxiv.org/abs/1912.01603.

Danijar Hafner, Timothy Lillicrap, Mohammad Norouzi, and Jimmy Ba. Mastering atari with discrete world models, 2022. URL https://arxiv.org/abs/2010.02193.

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

Danijar Hafner, Wilson Yan, and Timothy Lillicrap. Training agents inside of scalable world models, 2025. URL https://arxiv.org/abs/2509.24527.

Xianglong He, Chunli Peng, Zexiang Liu, Boyang Wang, Yifan Zhang, Qi Cui, Fei Kang, Biao Jiang, Mengyin An, Yangyang Ren, Baixin Xu, Hao-Xiang Guo, Kaixiong Gong, Size Wu, Wei Li, Xuchen Song, Yang Liu, Yangguang Li, and Yahui Zhou. Matrix-game 2.0: An open-source real-time and streaming interactive world model, 2026. URL https://arxiv.org/abs/2508.13009.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models, 2020. URL https://arxiv.org/abs/2006.11239.

Jonathan Ho, William Chan, Chitwan Saharia, Jay Whang, Ruiqi Gao, Alexey A. Gritsenko, Diederik P. Kingma, Ben Poole, Mohammad Norouzi, David J. Fleet, and Tim Salimans. Imagen video: High definition video generation with diffusion models. *ArXiv*, abs/2210.02303, 2022a. URL https://api.semanticscholar.org/CorpusID:252715883.

Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J. Fleet. Video diffusion models. *ArXiv*, abs/2204.03458, 2022b. URL https://api.semanticscholar.org/CorpusID:248006185.

Wenyi Hong, Ming Ding, Wendi Zheng, Xinghan Liu, and Jie Tang. Cogvideo: Large-scale pretraining for text-to-video generation via transformers. *ArXiv*, abs/2205.15868, 2022. URL https://api.semanticscholar.org/CorpusID:249209614.

Yicong Hong, Yiqun Mei, Chongjian Ge, Yiran Xu, Yang Zhou, Sai Bi, Yannick Hold-Geoffroy, Mike Roberts, Matthew Fisher, Eli Shechtman, et al. Relic: Interactive video world model with long-horizon memory. *arXiv preprint arXiv:2512.04040*, 2025.

Chengpeng Hu, Yunlong Zhao, and Jialin Liu. Game generation via large language models. *2024 IEEE Conference on Games (CoG)*, pp. 1–4, 2024a. URL https://api.semanticscholar.org/CorpusID:269149102.

Jinyi Hu, Shengding Hu, Yuxuan Song, Yufei Huang, Mingxuan Wang, Hao Zhou, Zhiyuan Liu, Wei-Ying Ma, and Maosong Sun. Acdit: Interpolating autoregressive conditional modeling and diffusion transformer. *ArXiv*, abs/2412.07720, 2024b. URL https://api.semanticscholar.org/CorpusID:274610804.

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, 2025a. URL https://arxiv.org/abs/2510.03198.

Xun Huang, Zhengqi Li, Guande He, Mingyuan Zhou, and Eli Shechtman. Self forcing: Bridging the train-test gap in autoregressive video diffusion. *ArXiv*, abs/2506.08009, 2025b. URL https://api.semanticscholar.org/CorpusID:279251392.

id Software. Doom, 1993. MS-DOS game.

Yang Jin, Zhicheng Sun, Ningyuan Li, Kun Xu, Hao Jiang, Zhuang Nan, Quzhe Huang, Yang Song, Yadong Mu, and Zhouchen Lin. Pyramidal flow matching for efficient video generative modeling. *ArXiv*, abs/2410.05954, 2024. URL https://api.semanticscholar.org/CorpusID:273228937.

Seung Wook Kim, Yuhao Zhou, Jonah Philion, Antonio Torralba, and Sanja Fidler. Learning to simulate dynamic environments with gamegan. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1228–1237, 2020. URL https://api.semanticscholar.org/CorpusID:218869555.

Seung Wook Kim, Jonah Philion, Antonio Torralba, and Sanja Fidler. Drivegan: Towards a controllable high-quality neural simulation. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5816–5825, 2021. URL https://api.semanticscholar.org/CorpusID:233476334.

Youngjin Kim, Minjung Kim, and Gunhee Kim. Memorization precedes generation: Learning unsupervised gans with memory networks, 2018. URL https://arxiv.org/abs/1803.01500.

D. Kondratyuk, Lijun Yu, Xiuye Gu, José Lezama, Jonathan Huang, Rachel Hornung, Hartwig Adam, Hassan Akbari, Yair Alon, Vighnesh Birodkar, Yong Cheng, Ming-Chang Chiu, Josh Dillon, Irfan Essa, Agrim Gupta, Meera Hahn, Anja Hauth, David Hendon, Alonso Martinez, David C. Minnen, David A. Ross, Grant Schindler, Mikhail Sirotenko, Kihyuk Sohn, Krishna Somandepalli, Huisheng Wang, Jimmy Yan, Ming Yang, Xuan Yang, Bryan Seybold, and Lu Jiang. Videopoet: A large language model for zero-shot video generation. *ArXiv*, abs/2312.14125, 2023. URL https://api.semanticscholar.org/CorpusID:266435847.

Weijie Kong, Qi Tian, Zijian Zhang, Rox Min, Zuozhuo Dai, Jin Zhou, Jia-Liang Xiong, Xin Li, Bo Wu, Jianwei Zhang, Kathrina Wu, Qin Lin, Junkun Yuan, Yanxin Long, Aladdin Wang, Andong Wang, Changlin Li, Duojun Huang, Fan Yang, Hao Tan, Hongmei Wang, Jacob Song, Jiawang Bai, Jianbing Wu, Jinbao Xue, Joey Wang, Kai Wang, Mengyang Liu, Peng-Yu Li, Shuai Li, Weiyan Wang, Wenqing Yu, Xi Deng, Yang Li, Yi Chen, Yutao Cui, Yuanbo Peng, Zhen Yu, Zhiyu He, Zhiyong Xu, Zixiang Zhou, Z. Xu, Yang-Dan Tao, Qinglin Lu, Songtao Liu, Daquan Zhou, Hongfa Wang, Yong Yang, Di Wang, Yuhong Liu, Jie Jiang, and Caesar Zhong. Hunyuanvideo: A systematic framework for large video generative models. *ArXiv*, abs/2412.03603, 2024. URL https://api.semanticscholar.org/CorpusID:274514554.

Vikram Kumaran, Bradford W. Mott, and James C. Lester. Generating game levels for multiple distinct games with a common latent space. In *Artificial Intelligence and Interactive Digital Entertainment Conference*, 2020. URL https://api.semanticscholar.org/CorpusID:221737009.

Guillaume Lample and Devendra Singh Chaplot. Playing fps games with deep reinforcement learning, 2018. URL https://arxiv.org/abs/1609.05521.

Jiaqi Li, Junshu Tang, Zhi-Ting Xu, Longhuang Wu, Yuan Zhou, Shuai Shao, Tianbao Yu, Zhiguo Cao, and Qinglin Lu. Hunyuan-gamecraft: High-dynamic interactive game video generation with hybrid history condition. *ArXiv*, abs/2506.17201, 2025a. URL https://api.semanticscholar.org/CorpusID:279465382.

Runjia Li, Philip Torr, Andrea Vedaldi, and Tomas Jakab. Vmem: Consistent interactive video scene generation with surfel-indexed view memory, 2025b. URL https://arxiv.org/abs/2506.18903.

Tianhong Li, Yonglong Tian, He Li, Mingyang Deng, and Kaiming He. Autoregressive image generation without vector quantization. *ArXiv*, abs/2406.11838, 2024a. URL https://api.semanticscholar.org/CorpusID:270560593.

Zongyi Li, Shujie Hu, Shujie Liu, Long Zhou, Jeongsoo Choi, Lingwei Meng, Xun Guo, Jinyu Li, Hefei Ling, and Furu Wei. Arlon: Boosting diffusion transformers with autoregressive models for long video generation. *ArXiv*, abs/2410.20502, 2024b. URL https://api.semanticscholar.org/CorpusID:273653802.

Haozhe Liu, Shikun Liu, Zijian Zhou, Mengmeng Xu, Yanping Xie, Xiao Han, Juan C. P'erez, Ding Liu, Kumara Kahatapitiya, Menglin Jia, Jui-Chieh Wu, Sen He, Tao Xiang, Jürgen Schmidhuber, and Juan-Manuel P'erez-R'ua. Mardini: Masked autoregressive diffusion for video generation at scale. *Trans. Mach. Learn. Res.*, 2025, 2024a. URL https://api.semanticscholar.org/CorpusID:273655157.

Kunhao Liu, Wenbo Hu, Jiale Xu, Ying Shan, and Shijian Lu. Rolling forcing: Autoregressive long video diffusion in real time. *ArXiv*, abs/2509.25161, 2025. URL https://api.semanticscholar.org/CorpusID:281676207.

Yaofang Liu, Yumeng Ren, Xiaodong Cun, Aitor Artola, Yang Liu, Tieyong Zeng, Raymond H. Chan, and Jean-Michel Morel. Redefining temporal modeling in video diffusion: The vectorized timestep approach. *ArXiv*, abs/2410.03160, 2024b. URL https://api.semanticscholar.org/CorpusID:273162851.

Xiaofeng Mao, Shaoheng Lin, Zhen Li, Chuanhao Li, Wenshuo Peng, Tong He, Jiangmiao Pang, Mingmin Chi, Yu Qiao, and Kaipeng Zhang. Yume: An interactive world generation model. *arXiv preprint arXiv:2507.17744*, 2025.

Muhammad Umair Nasir and Julian Togelius. Practical pcg through large language models. *2023 IEEE Conference on Games (CoG)*, pp. 1–4, 2023. URL https://api.semanticscholar.org/CorpusID:258960371.

Obsidian Community. Obsidian level generator. https://obsidian-level-maker.github.io/, 2026. Accessed: 2026-03-03.

OpenAI, :, Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, Alex Iftimie, Alex Karpenko, Alex Tachard Passos, Alexander Neitz, Alexander Prokofiev, Alexander Wei, Allison Tam, Ally Bennett, Ananya Kumar, Andre Saraiva, Andrea Vallone, Andrew Duberstein, Andrew Kondrich, Andrey Mishchenko, Andy Applebaum, Angela Jiang, Ashvin Nair, Barret Zoph, Behrooz Ghorbani, Ben Rossen, Benjamin Sokolowsky, Boaz Barak, Bob McGrew, Borys Minaiev, Botao Hao, Bowen Baker, Brandon Houghton, Brandon McKinzie, Brydon Eastman, Camillo Lugaresi, Cary Bassin, Cary Hudson, Chak Ming Li, Charles de Bourcy, Chelsea Voss, Chen Shen, Chong Zhang, Chris Koch, Chris Orsinger, Christopher Hesse, Claudia Fischer, Clive Chan, Dan Roberts, Daniel Kappler, Daniel Levy, Daniel Selsam, David Dohan, David Farhi, David Mely, David Robinson, Dimitris Tsipras, Doug Li, Dragos Oprica, Eben Freeman, Eddie Zhang, Edmund Wong, Elizabeth Proehl, Enoch Cheung, Eric Mitchell, Eric Wallace, Erik Ritter, Evan Mays, Fan Wang, Felipe Petroski Such, Filippo Raso, Florencia Leoni, Foivos Tsimpourlas, Francis Song, Fred von Lohmann, Freddie Sulit, Geoff Salmon, Giambattista Parascandolo, Gildas Chabot, Grace Zhao, Greg Brockman, Guillaume Leclerc, Hadi Salman, Haiming Bao, Hao Sheng, Hart Andrin, Hessam Bagherinezhad, Hongyu Ren, Hunter Lightman, Hyung Won Chung, Ian Kivlichan, Ian O'Connell, Ian Osband, Ignasi Clavera Gilaberte, Ilge Akkaya, Ilya Kostrikov, Ilya Sutskever, Irina Kofman, Jakub Pachocki, James Lennon, Jason Wei, Jean Harb, Jerry Twore, Jiacheng Feng, Jiahui Yu, Jiayi Weng, Jie Tang, Jieqi Yu, Joaquin Quiñonero Candela, Joe Palermo, Joel Parish, Johannes Heidecke, John Hallman, John Rizzo, Jonathan Gordon, Jonathan Uesato, Jonathan Ward, Joost Huizinga, Julie Wang, Kai Chen, Kai Xiao, Karan Singhal, Karina Nguyen, Karl Cobbe, Katy Shi, Kayla Wood, Kendra Rimbach, Keren Gu-Lemberg, Kevin Liu, Kevin Lu, Kevin Stone, Kevin Yu, Lama Ahmad, Lauren Yang, Leo Liu, Leon Maksin, Leyton Ho, Liam Fedus, Lilian Weng, Linden Li, Lindsay McCallum, Lindsey Held, Lorenz Kuhn, Lukas Kondraciuk, Lukasz Kaiser, Luke Metz, Madelaine Boyd, Maja Trebacz, Manas Joglekar, Mark Chen, Marko Tintor, Mason Meyer, Matt Jones, Matt Kaufer, Max Schwarzer, Meghan Shah, Mehmet Yatbaz, Melody Y. Guan, Mengyuan Xu, Mengyuan Yan, Mia Glaese, Mianna Chen, Michael Lampe, Michael Malek, Michele Wang, Michelle Fradin, Mike McClay, Mikhail Pavlov, Miles Wang, Mingxuan Wang, Mira Murati, Mo Bavarian, Mostafa Rohaninejad, Nat McAleese, Neil Chowdhury, Neil Chowdhury, Nick Ryder, Nikolas Tezak, Noam Brown, Ofir Nachum, Oleg Boiko, Oleg Murk, Olivia Watkins, Patrick Chao, Paul Ashbourne, Pavel Izmailov, Peter Zhokhov, Rachel Dias, Rahul Arora, Randall Lin, Rapha Gontijo Lopes, Raz Gaon, Reah Miyara, Reimar Leike, Renny Hwang, Rhythm Garg, Robin Brown, Roshan James, Rui Shu, Ryan Cheu, Ryan Greene, Saachi Jain, Sam Altman, Sam Toizer, Sam Toyer, Samuel Miserendino, Sandhini Agarwal, Santiago Hernandez, Sasha Baker, Scott McKinney, Scottie Yan, Shengjia Zhao, Shengli Hu, Shibani Santurkar, Shraman Ray Chaudhuri, Shuyuan Zhang, Siyuan Fu, Spencer Papay, Steph Lin, Suchir Balaji, Suvansh Sanjeev, Szymon Sidor, Tal Broda, Aidan Clark, Tao Wang, Taylor Gordon, Ted Sanders, Tejal Patwardhan, Thibault Sottiaux, Thomas Degry, Thomas Dimson, Tianhao Zheng, Timur Garipov, Tom Stasi, Trapit Bansal, Trevor Creech, Troy Peterson, Tyna Eloundou, Valerie Qi, Vineet Kosaraju, Vinnie Monaco, Vitchyr Pong, Vlad Fomenko, Weiyi Zheng, Wenda Zhou, Wes McCabe, Wojciech Zaremba, Yann Dubois, Yinghai Lu, Yining Chen, Young Cha, Yu Bai, Yuchen He, Yuchen Zhang, Yunyun Wang, Zheng Shao, and Zhuohan Li. Openai o1 system card, 2024. URL https://arxiv.org/abs/2412.16720.

Jack Parker-Holder, Philip Ball, Jake Bruce, Vibhavari Dasagi, Kristian Holsheimer, Christos Kaplanis, Alexandre Moufarek, Guy Scully, Jeremy Shar, Jimmy Shi, Stephen Spencer, Jessica Yung, Michael Dennis, Sultan Kenjeyev, Shangbang Long, Vlad Mnih, Harris Chan, Maxime Gazeau, Bonnie Li, Fabio Pardo, Luyu Wang, Lei Zhang, Frederic Besse, Tim Harley, Anna Mitenkova, Jane Wang, Jeff Clune, Demis Hassabis, Raia Hadsell,

Adrian Bolton, Satinder Singh, and Tim Rocktäschel. Genie 2: A large-scale foundation world model. 2024. URL https://deepmind.google/discover/blog/genie-2-a-large-scale-foundation-world-model/.

William S. Peebles and Saining Xie. Scalable diffusion models with transformers. *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4172–4182, 2022. URL https://api.semanticscholar.org/CorpusID:254854389.

Ryan Po, Eric Ryan Chan, Changan Chen, and Gordon Wetzstein. Bagger: Backwards aggregation for mitigating drift in autoregressive video diffusion models, 2025. URL https://arxiv.org/abs/2512.12080.

Adam Polyak, Amit Zohar, Andrew Brown, Andros Tjandra, Animesh Sinha, Ann Lee, Apoorv Vyas, Bowen Shi, Chih-Yao Ma, Ching-Yao Chuang, David Yan, Dhruv Choudhary, Dingkang Wang, Geet Sethi, Guan Pang, Haoyu Ma, Ishan Misra, Ji Hou, Jialiang Wang, Ki ran Jagadeesh, Kunpeng Li, Luxin Zhang, Mannat Singh, Mary Williamson, Matt Le, Matthew Yu, Mitesh Kumar Singh, Peizhao Zhang, Peter Vajda, Quentin Duval, Rohit Girdhar, Roshan Sumbaly, Sai Saketh Rambhatla, Sam S. Tsai, Samaneh Azadi, Samyak Datta, Sanyuan Chen, Sean Bell, Sharadh Ramaswamy, Shelly Sheynin, Siddharth Bhattacharya, Simran Motwani, Tao Xu, Tianhe Li, Tingbo Hou, Wei-Ning Hsu, Xi Yin, Xiaoliang Dai, Yaniv Taigman, Yaqiao Luo, Yen-Cheng Liu, Yi-Chiao Wu, Yue Zhao, Yuval Kirstain, Zecheng He, Zijian He, Albert Pumarola, Ali K. Thabet, Artsiom Sanakoyeu, Arun Mallya, Baishan Guo, Boris Araya, Breena Kerr, Carleigh Wood, Ce Liu, Cen Peng, Dimitry Vengertsev, Edgar Schonfeld, Elliot Blanchard, Felix Juefei-Xu, Fraylie Nord, Jeff Liang, John Hoffman, Jonas Kohler, Kaolin Fire, Karthik Sivakumar, Lawrence Chen, Licheng Yu, Luya Gao, Markos Georgopoulos, Rashel Moritz, Sara K. Sampson, Shikai Li, Simone Parmeggiani, Steven Fine, Tara Fowler, Vladan Petrovic, and Yuming Du. Movie gen: A cast of media foundation models. 2024. URL https://api.semanticscholar.org/CorpusID:273403698.

Shuhuai Ren, Shuming Ma, Xu Sun, and Furu Wei. Next block prediction: Video generation via semi-autoregressive modeling. *ArXiv*, abs/2502.07737, 2025. URL https://api.semanticscholar.org/CorpusID:276258605.

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

Frederik Schubert, Maren Awiszus, and Bodo Rosenhahn. Toad-gan: A flexible framework for few-shot level generation in token-based games. *IEEE Transactions on Games*, 14:284–293, 2021. URL https://api.semanticscholar.org/CorpusID:233520445.

Joonghyuk Shin, Zhengqi Li, Richard Zhang, Jun-Yan Zhu, Jaesik Park, Eli Schechtman, and Xun Huang. Motionstream: Real-time video generation with interactive motion controls. 2025. URL https://api.semanticscholar.org/CorpusID:282739800.

Sam Snodgrass and Santiago Ontañón. Experiments in map generation using markov chains. In *International Conference on Foundations of Digital Games*, 2014. URL https://api.semanticscholar.org/CorpusID:15083922.

Kiwhan Song, Boyuan Chen, Max Simchowitz, Yilun Du, Russ Tedrake, and Vincent Sitzmann. History-guided video diffusion. *ArXiv*, abs/2502.06764, 2025. URL https://api.semanticscholar.org/CorpusID:276249479.

Shyam Sudhakaran, Miguel Gonz'alez-Duque, Claire Glanois, Matthias Anton Freiberger, Elias Najarro, and Sebastian Risi. Mariogpt: Open-ended text2level generation through large language models. *ArXiv*, abs/2302.05981, 2023. URL https://api.semanticscholar.org/CorpusID:256827347.

Adam James Summerville and Michael Mateas. Super mario as a string: Platformer level generation via lstms. *ArXiv*, abs/1603.00930, 2016. URL https://api.semanticscholar.org/CorpusID:16077591.

Adam James Summerville, Sam Snodgrass, Matthew J. Guzdial, Christoffer Holmgård, Amy K. Hoover, Aaron Isaksen, Andy Nealen, and Julian Togelius. Procedural content generation via machine learning (pcgml). *IEEE Transactions on Games*, 10:257–270, 2017. URL https://api.semanticscholar.org/CorpusID:9950600.

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

Graham Todd, Sam Earle, Muhammad Umair Nasir, Michael Cerny Green, and Julian Togelius. Level generation through large language models. *Proceedings of the 18th International Conference on the Foundations of Digital Games*, 2023. URL https://api.semanticscholar.org/CorpusID:256826896.

Mike Treanor, Alexander Zook, Mirjam Palosaari Eladhari, Julian Togelius, Gillian Smith, Michael Cook, Tommy Thompson, Brian Magerko, John Levine, and Adam M. Smith. Ai-based game design patterns. In *International Conference on Foundations of Digital Games*, 2015. URL https://api.semanticscholar.org/CorpusID:3099158.

Dani Valevski, Yaniv Leviathan, Moab Arar, and Shlomi Fruchter. Diffusion models are real-time game engines. *ArXiv*, abs/2408.14837, 2024. URL https://api.semanticscholar.org/CorpusID:271962839.

Ruben Villegas, Mohammad Babaeizadeh, Pieter-Jan Kindermans, Hernan Moraldo, Han Zhang, Mohammad Taghi Saffar, Santiago Castro, Julius Kunze, and D. Erhan. Phenaki: Variable length video generation from open domain textual description. *ArXiv*, abs/2210.02399, 2022. URL https://api.semanticscholar.org/CorpusID:252715594.

Vanessa Volz, Jacob Schrum, Jialin Liu, Simon M. M. Lucas, Adam M. Smith, and Sebastian Risi. Evolving mario levels in the latent space of a deep convolutional generative adversarial network. *Proceedings of the Genetic and Evolutionary Computation Conference*, 2018. URL https://api.semanticscholar.org/CorpusID:13676024.

Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, Jianyuan Zeng, Jiayu Wang, Jingfeng Zhang, Jingren Zhou, Jinkai Wang, Jixuan Chen, Kai Zhu, Kang Zhao, Keyu Yan, Lianghua Huang, Xiaofeng Meng, Ningying Zhang, Pandeng Li, Ping Wu, Ruihang Chu, Rui Feng, Shiwei Zhang, Siyang Sun, Tao Fang, Tianxing Wang, Tianyi Gui, Tingyu Weng, Tong Shen, Wei Lin, Wei Wang, Wei Wang, Wen-Chao Zhou, Wente Wang, Wen Shen, Wenyuan Yu, Xianzhong Shi, Xiaomin Huang, Xin Xu, Yan Kou, Yan-Mei Lv, Yifei Li, Yijing Liu, Yiming Wang, Yingya Zhang, Yitong Huang, Yong Li, You Wu, Yu Liu, Yulin Pan, Yun Zheng, Yuntao Hong, Yupeng Shi, Yutong Feng, Zeyinzi Jiang, Zhengbin Han, Zhigang Wu, and Ziyu Liu. Wan: Open and advanced large-scale video generative models. *ArXiv*, abs/2503.20314, 2025. URL https://api.semanticscholar.org/CorpusID:277321639.

Yuqing Wang, Tianwei Xiong, Daquan Zhou, Zhijie Lin, Yang Zhao, Bingyi Kang, Jiashi Feng, and Xihui Liu. Loong: Generating minute-level long videos with autoregressive language models. *ArXiv*, abs/2410.02757, 2024. URL https://api.semanticscholar.org/CorpusID:273098341.

Dirk Weissenborn, Oscar Täckström, and Jakob Uszkoreit. Scaling autoregressive video models. *ArXiv*, abs/1906.02634, 2019. URL https://api.semanticscholar.org/CorpusID:174802916.

Wenming Weng, Ruoyu Feng, Yanhui Wang, Qi Dai, Chunyu Wang, Dacheng Yin, Zhiyuan Zhao, Kai Qiu, Jianmin Bao, Yuhui Yuan, Chong Luo, Yueyi Zhang, and Zhiwei Xiong. Art•v: Auto-regressive text-to-video generation with diffusion models. *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 7395–7405, 2023. URL https://api.semanticscholar.org/CorpusID:265506663.

Ruiqi Wu, Xuanhua He, Meng Cheng, Tianyu Yang, Yong Zhang, Zhuoliang Kang, Xunliang Cai, Xiaoming Wei, Chunle Guo, Chongyi Li, and Ming-Ming Cheng. Infinite-world: Scaling interactive world models to 1000-frame horizons via pose-free hierarchical memory, 2026. URL https://arxiv.org/abs/2602.02393.

Tong Wu, Shuai Yang, Ryan Po, Yinghao Xu, Ziwei Liu, Dahua Lin, and Gordon Wetzstein. Video world models with long-term spatial memory. *ArXiv*, abs/2506.05284, 2025a. URL https://api.semanticscholar.org/CorpusID:279244178.

Xiaofei Wu, Guozhen Zhang, Zhiyong Xu, Yuan Zhou, Qinglin Lu, and Xuming He. Pack and force your memory: Long-form and consistent video generation. *arXiv preprint arXiv:2510.01784*, 2025b.

Marek Wydmuch, Michał Kempka, and Wojciech Jaśkowski. ViZDoom Competitions: Playing Doom from Pixels. *IEEE Transactions on Games*, 11(3):248–259, 2019. doi: 10.1109/TG.2018.2877047. The 2022 IEEE Transactions on Games Outstanding Paper Award.

Wilson Yan, Yunzhi Zhang, P. Abbeel, and A. Srinivas. Videogpt: Video generation using vq-vae and transformers. *ArXiv*, abs/2104.10157, 2021. URL https://api.semanticscholar.org/CorpusID:233307257.

Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, Da Yin, Xiaotao Gu, Yuxuan Zhang, Weihan Wang, Yean Cheng, Ting Liu, Bin Xu, Yuxiao Dong, and Jie Tang. Cogvideox: Text-to-video diffusion models with an expert transformer. *ArXiv*, abs/2408.06072, 2024. URL https://api.semanticscholar.org/CorpusID:271855655.

Hu Ye, Jun Zhang, Sibo Liu, Xiao Han, and Wei Yang. Ip-adapter: Text compatible image prompt adapter for text-to-image diffusion models, 2023. URL https://arxiv.org/abs/2308.06721.

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, 2025a. URL https://arxiv.org/abs/2506.03141.

Jiwen Yu, Yiran Qin, Xintao Wang, Pengfei Wan, Di Zhang, and Xihui Liu. Gamefactory: Creating new games with generative interactive videos. *ArXiv*, abs/2501.08325, 2025b. URL https://api.semanticscholar.org/CorpusID:275515409.

Abhaysinh Zala, Jaemin Cho, Han Lin, Jaehong Yoon, and Mohit Bansal. Envgen: Generating and adapting environments via llms for training embodied agents. *ArXiv*, abs/2403.12014, 2024. URL https://api.semanticscholar.org/CorpusID:268531207.

David Junhao Zhang, Jay Zhangjie Wu, Jia-Wei Liu, Rui Zhao, Lingmin Ran, Yuchao Gu, Difei Gao, and Mike Zheng Shou. Show-1: Marrying pixel and latent diffusion models for text-to-video generation. *International Journal of Computer Vision*, 133(4):1879–1893, 2025a.

Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 3813–3824, 2023a. URL https://api.semanticscholar.org/CorpusID:256827727.

Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models, 2023b. URL https://arxiv.org/abs/2302.05543.

Lvmin Zhang, Shengqu Cai, Muyang Li, Gordon Wetzstein, and Maneesh Agrawala. Frame context packing and drift prevention in next-frame-prediction video diffusion models. 2025b. URL https://api.semanticscholar.org/CorpusID:277857265.

Tianyuan Zhang, Sai Bi, Yicong Hong, Kai Zhang, Fujun Luan, Songlin Yang, Kalyan Sunkavalli, William T. Freeman, and Hao Tan. Test-time training done right. *ArXiv*, abs/2505.23884, 2025c. URL https://api.semanticscholar.org/CorpusID:279071244.

Yifan Zhang, Chunli Peng, Boyang Wang, Puyi Wang, Qingcheng Zhu, Fei Kang, Biao Jiang, Zedong Gao, Eric Li, Yang Liu, et al. Matrix-game: Interactive world foundation model. *arXiv preprint arXiv:2506.18701*, 2025d.

Yuan Zhang, Jiacheng Jiang, Guoqing Ma, Zhiying Lu, Haoyang Huang, Jian min Yuan, and Nan Duan. Generative pre-trained autoregressive diffusion transformer. *ArXiv*, abs/2505.07344, 2025e. URL https://api.semanticscholar.org/CorpusID:278501597.

Jinjing Zhao, Fangyun Wei, Zhening Liu, Hongyang Zhang, Chang Xu, and Yan Lu. Spatia: Video generation with updatable spatial memory, 2025. URL https://arxiv.org/abs/2512.15716.

Hongwei Zhou, Jichen Zhu, Michael Mateas, and Noah Wardrip-Fruin. The eyes, the hands and the brain: What can text-to-image models offer for game design and visual creativity? *Proceedings of the 19th International Conference on the Foundations of Digital Games*, 2024. URL https://api.semanticscholar.org/CorpusID:270963065.

Minfeng Zhu, Pingbo Pan, Wei Chen, and Yi Yang. Dm-gan: Dynamic memory generative adversarial networks for text-to-image synthesis. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5795–5803, 2019. URL https://api.semanticscholar.org/CorpusID:91183909.

7 SUPPLEMENTARY MATERIALS

The supplementary website accompanying this document provides a broad set of additional results in video form. These include representative multiplayer gameplay samples, examples demonstrating interactive level editing, and a 30-minute timelapse of real-time four-player gameplay. Together, these videos illustrate the qualitative behavior of the system across extended rollouts, multi-agent interactions, and user-edited environments.

S1. MODEL ARCHITECTURE

Our system consists of two sequentially trained components built on Stable Diffusion v1.4: a visual world model that generates first-person game frames, and a dynamics model that predicts player state transitions.

S1.1 OBSERVATION MODEL

The observation model is a fine-tuned StableDiffusionV1.4 conditioned on context frames, player actions, and geometric depth. To support multi-frame input, the UNet input layer is expanded from 4 channels to 33. 32 latent channels from 8 frames plus 1 depth channel. The original 4-channel weights are copied into the new layer, and the remaining channels are zero-initialized.

Conditioning At each training step, the current player pose is used to cast 320 rays against map geometry. Depth values d are transformed as $80/d$ and arranged into a single-channel spatial map aligned with the latent resolution. Player actions and context noise timestep is embedded with a learned embedding module. These embeddings are concatenated and provided to the UNet cross-attention layers, replacing CLIP text embeddings.

Training The target is always the most recent frame in an 32-frame window. The other 31 context frames are noised using timesteps sampled from $[0, 0.7T]$ with $T = 1000$. The scheduler uses v-prediction. Latents are precomputed offline with the SD v1.4 VAE encoder. The UNet and embedding modules are trained jointly with AdamW at learning rate 2×10^{-5} using mixed precision.

S1.2 DYNAMICS MODEL

The dynamics model predicts the player's next position and heading. The Stage 1 UNet is loaded and kept frozen during Stage 2, serving only as a feature extractor.

Feature Extraction Forward hooks are attached to the four decoder up-blocks of the frozen UNet. A single forward pass captures multi-scale feature maps at resolutions 8×10, 16×20, 32×40, and 32×40.

Readout Network These features are projected to a common 8×10 resolution with 320 channels each, concatenated, and processed by convolution layers to produce 80 visual patch tokens of dimension 768. These are combined with a CLS token, 8 position-history tokens, 9 action-history tokens, and 16 omnidirectional depth tokens. The full sequence is passed through an 8-layer, 8-head Transformer encoder. The CLS output is mapped by an MLP to representing relative displacement and player direction.

Training Only the readout and embedding modules are optimized. Training uses AdamW with learning rate 4×10^{-6}, gradient clipping at 5.0, mixed precision, and mean squared error loss on the motion targets.

S2. INFERENCE

Inference runs as an autoregressive game loop with a rolling context window of 32 latent frames initialized from a held-out sequence.

Depth and Denoising At each step, 320 forward-facing rays are cast to compute the visual depth map, and 16 omnidirectional rays are cast for the dynamics model. Frame generation uses a DDIM scheduler with v-prediction and 3 denoising steps. Classifier-free guidance blends a conditional pass with clean context and an unconditional pass with noisy context, with guidance scale $w = 1.3$.

State Update To reduce overhead, only the final denoising step uses the eager UNet with hooks. Earlier steps use a compiled UNet without hooks. The captured features are passed to the dynamics model to predict the next player state, update position and heading, and roll the context window forward. The final latent is decoded with TAESD for faster image reconstruction.

S3. Multiplayer Changes

Extending the system to multiple players requires changes to the observation model, the dynamics model, and the inference pipeline so that the opponent is represented explicitly and all player states remain synchronized.

S3.1 Observation Model

The main multiplayer modification to the observation model is opponent-aware geometric conditioning. Opponents are represented as circles of radius 16 in the 2D map plane. During training, ray-circle intersections are computed alongside the standard wall ray-casting. If a wall is hit before an opponent, the opponent is treated as occluded.

This produces three additional per-ray signals: inverse opponent depth, opponent alive/dead status (encoded as 1.0 for dead, 0.0 for alive or no hit), and the 2D facing direction of the nearest visible opponent in the player's local frame. These signals are reshaped into latent-resolution spatial maps using the same procedure as the wall depth channel, giving 5 total geometric conditioning channels: wall depth, opponent depth, opponent alive/dead status, and two opponent-direction channels. The UNet input is therefore expanded match the dimensions needed for these additional channels.

S3.2 Dynamics Model

The dynamics model is extended with a kill-prediction head: a two-layer MLP attached to the same CLS token used by the position head. This head outputs a scalar logit indicating whether the player scores a kill at the current step. Because kill events are rare, the model is trained with a weighted binary cross-entropy loss using a positive class weight of 100.

The 320-ray field-of-view features, previously unused, now encode opponent alive/dead status using the same ray-circle intersection logic as the observation model. The 16 omnidirectional wall-depth tokens are unchanged. Two separate action embedding modules are used: one for the action history and one for the current predicted action. Their outputs are concatenated before being passed to the Transformer. The embedding modules and readout network are trained jointly with AdamW at a learning rate of 4×10^{-6}.

S3.3 Multiplayer Inference

At each step, the main process sends the current opponent state to each worker. The worker casts 320 forward-facing rays and 16 omnidirectional rays from the player's current pose, computes opponent depth, alive/dead status, and facing direction with wall occlusion, and assembles the 133-channel UNet input from the noisy target latent, 5 geometric conditioning channels, and 31 context latents. The context latents are minimally noised at timestep 0, and this noise level is communicated through the context noise embedding.

The observation model then performs 4 DDIM denoising steps. Both the conditional branch, which uses the current action embedding, and the unconditional branch, which uses an empty action context, receive the same geometric conditioning. Classifier-free guidance is applied with scale $w = 1.0$. The resulting features are passed to the dynamics Transformer, which predicts player position deltas together with a kill logit. The player's position is then updated. The final latent is

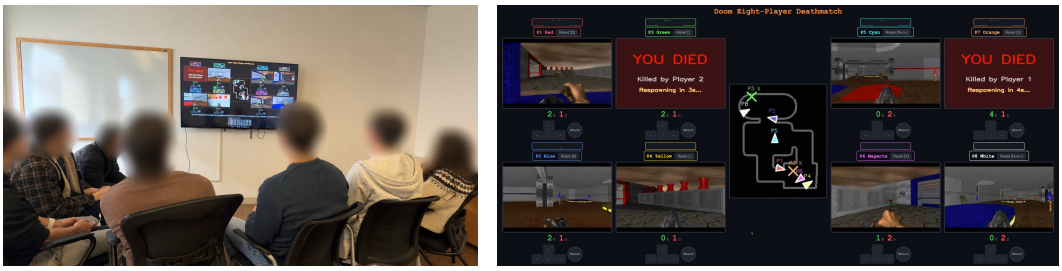


Figure 8: **Eight-Player Real-Time Gameplay.** Stills from a real-time rollout with eight simultaneous players sharing the same external world state. Each player's observation model runs on a separate GPU, and the system runs at 20 FPS irrespective of the number of players; the only cross-player communication is the synchronization of player positions in the shared state, which is negligible.



Figure 9: **Comparison with Matrix-Game 2.0.** Using the preference protocol of the level-design user study, we compare against Matrix-Game 2.0 (He et al., 2026) with and without fine-tuning on our data; the model was fine-tuned for 10K iterations at batch size 8 and a learning rate of 2×10^{-6}. Without fine-tuning, the video domain drifts during rollout. Our method is favored over both variants.

decoded with TAESD, JPEG-encoded, and returned to the main process together with the updated pose and kill probability. A kill is registered when the predicted kill probability exceeds 0.995.

S4. Context-Length Ablation

We perform an ablation study over the number of conditioning frames, varying $L \in 2, 4, 8, 16, 32$ while keeping all other settings fixed. As shown in Tab. 4, increasing the context length consistently improves fidelity, reflected by higher SSIM/PSNR and lower LPIPS.

Table 4: Context frame ablation.

L	SSIM ↑	PSNR ↑	LPIPS ↓
2	0.709	27.6	0.121
4	0.775	29.5	0.097
8	0.783	29.8	0.094
16	0.782	29.8	0.093
32	0.789	30.0	0.089

S5. VLM-Based Evaluation of Opponent Visibility

We evaluated whether generated frames preserve opponent visibility using a vision–language model (VLM) judge. For each generated frame, the evaluation script pairs it with the corresponding

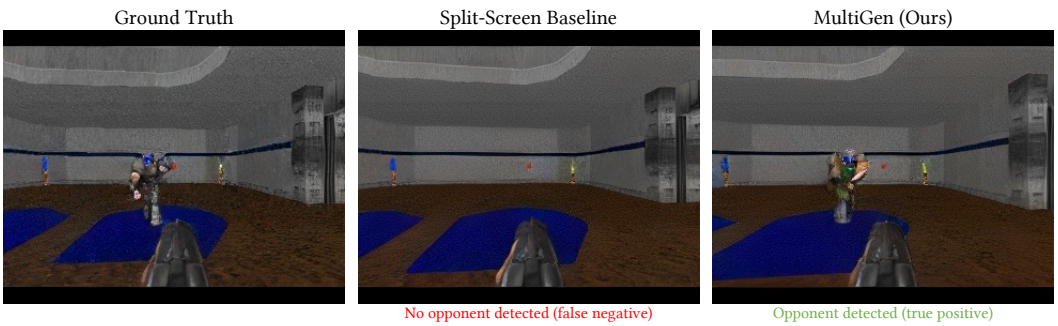


Figure 10: Example comparisons for multiplayer VLM-based evaluation of opponent visibility. Ground truth (left), is compared against each method (middle and right). In this example, we expect the opposing player to be visible in the current view. Our method successfully reflects this in the generated frame, while the opponent is absent in the baseline.

ground-truth frame from the same timestep and player viewpoint, and submits both images to GPT-4o. The model is asked, independently for each image, whether an opposing player is visible in the frame. We pass both images along with the following system prompt to the VLM:

```
You are evaluating frames from a first-person shooter game (Doom).
You will see two images side by side: a ground-truth frame (left)
and a generated frame (right). Both frames show the same moment
from the same player's perspective.

For EACH image, answer: Is there another player (opponent) visible
in the frame? The opponent appears as a humanoid figure holding a
weapon, distinct from walls/floor/ceiling.

Respond ONLY with a JSON object (no markdown, no explanation):
{"gt_opponent_visible": true/false,
"gen_opponent_visible": true/false}

Left image: Ground Truth. Right image: Generated.
```

S5.1 Bounding-Box Annotation Protocol

For the spatial-consistency evaluation in Sec. 5.4, human annotators drew a bounding box around the visible opponent in each generated frame, with 250 frames annotated per method. Each annotated box is compared against the ground-truth bounding box from the underlying simulator at the same timestep, and we report the mean IoU over all annotated frames. Fig. 11 shows a representative comparison with an IoU of approximately 0.5 to calibrate the reported scores.

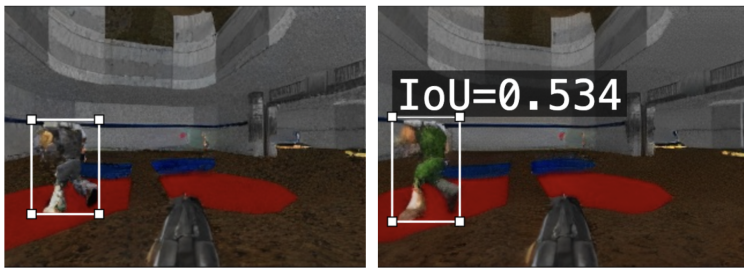


Figure 11: Representative bounding-box comparison with IoU ≈ 0.5, shown for reference when interpreting Tab. 2.

S6. DISCLOSURE OF LLM USAGE

In this work, we used generative AI tools to assist with paper writing, including polishing the text for readability and drafting parts of sections. We also used generative AI tools for drafting the templates used in our user studies and for identifying relevant related work. We did not use generative AI tools for any of the tasks requiring disclosure under the ICLR LLM-usage policy. No generative AI was used to generate data, develop theoretical models or mathematical claims, assist with proofs, design research methodology or experiments, implement methods, clean or analyze data, or interpret results. We have reviewed all AI-assisted content. In particular, all references surfaced through AI-assisted literature discovery were manually checked for relevance and verified against their original sources to ensure none were hallucinated. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.