MultiGen: Level-Design for Editable Multiplayer Worlds in Diffusion Game Engines

https://ryanpo.com/multigen

Ryan Po[1], David Junhao Zhang[2], Amir Hertz[2],
Gordon Wetzstein*[1], Neal Wadhwa*[2], and Nataniel Ruiz*[2]

[1] Stanford University
[2] Google

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 the generation roll-out. 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 an editable memory representation, and it naturally extends to real-time multiplayer rollouts with coherent viewpoints and consistent cross-player interactions.

Keywords: generative game engines · game design · video generation

1 Introduction

Recent advances in generative models have made real-time, action-controllable video generation increasingly practical, enabling interactive rollouts that resemble explorable [2, 7, 21–24, 48, 52]. However, most video world models are still fundamentally single-user experiences [9, 41, 83, 90]: 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

* Equal contributions.

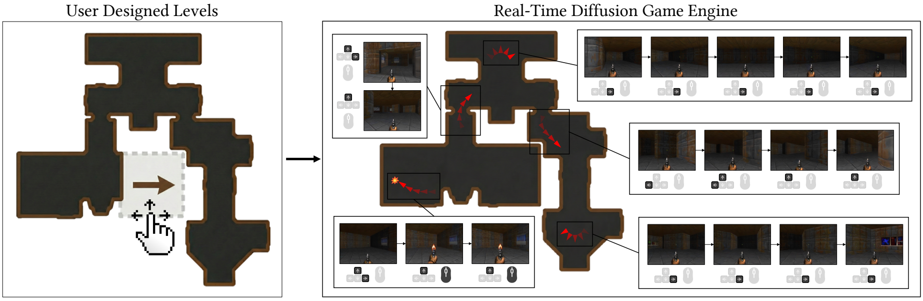


Fig. 1: Level Design via Editable Memory. Users define a level through coarse 2D geometry (left). During inference, the diffusion model generates first-person observations consistent with the top-down level layout (right).

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

We use Doom [34] 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 [2,7,9,48,52,68,83,90]. 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 interactive 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 external 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 in video generation models have mostly been set by large-scale bidirectional diffusion transformers [6, 53], demonstrating remarkable capabilities in synthesizing high-quality, complex video clips. The core mechanism relies on full spatiotemporal attention applied to all tokens [6], as every video token is denoised simultaneously [3–6, 8, 18, 20, 25, 27, 28, 38, 43, 55, 69, 71, 80, 85], generating videos of fixed lengths. However, these models are inherently non-causal and constrained to fixed-length generation.

Autoregressive Video Models. Autoregressive approaches in video generation factorizes the joint distribution over all frames into, $p(x^{1:N}) = \prod_{i=1}^{N} p(x^i|x^{<i})$. This formulation naturally aligns with the causality of time, as video frames are generated sequentially, making autoregressive models well suited for interactive simulation [33, 59]. Conventional autoregressive video models rely on direct next-token prediction of discrete video tokens [7, 37, 56, 72, 73, 79]. However, the performance of discretized AR models often lag behind diffusion models. To address this, recent works have explored training strategies combining autoregression and diffusion [15–17, 31, 35, 44, 45, 47, 74, 89, 91]. Some works train conditional diffusion models that denoise next frames condition on past clean frames [33, 86], while other approaches introduce per-frame independent noise levels during training [10, 61, 68], allowing for AR inference.

Diffusion Game Engines. Our work is most closely related to recent efforts that repurpose diffusion models as real-time game engines or world simulator [2, 7, 9, 41, 48, 52, 83, 90]. GameNGen [68] generates gameplay by conditioning an image diffusion model on a short history of previous frames and the next action, using diffusion as an 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. While existing video frameworks typically construct memory using explicit 3D representations [14, 76, 92] or manage input conditions through compressed context windows and KV caching

[16, 29, 32, 42, 75, 77, 82], 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. Our work is also related to the field of Game Generation as a whole. A wide range of approaches have been proposed for directly generating game content [64]. Recent works have utilized Generative Adversarial Networks (GANs) to great effect for generating game levels and interactiive environments [36, 39, 58, 70]. While earlier works explore a wide range of methods for generating game content [19, 60, 63, 67]. Other works also explore the use of diffusion models for game generation [68, 93] and LLMs for designing game mechanics [1,11,12,30,49,62,66,84]. These approaches primarily focus on generating the game content itself such as world layouts, assets, or mechanics. Our model instead operates like a game engine and gives users additional control over the environment while generating every observed frame online through a diffusion model.

3 Method

We model an interactive game rollout as a sequence of actions and observations over discrete timesteps $t \in \{0, 1, \dots, T\}$. Let $a_t \in \mathcal{A}$ denote the agent action at timestep t, and let $o_t \in \mathbb{R}^{H\times W\times C}$ denote the rendered observation (image frame) at timestep t. A rollout is therefore represented as an alternating sequence

$$\tau = \Big(o_0, a_0, o_1, a_1, \dots, a_{T-1}, o_T\Big). \tag{1}$$

Given a history of past observations and actions, the goal of a diffusion game engine is to predict the distribution of the next observation,

$$p(o_{t+1} \mid o_{\le t}, a_{\le t}), \tag{2}$$

and to enable closed-loop simulation by repeatedly sampling o_{t+1} and feeding it back as input at the next step.

Following GameNGen [68], we instantiate this predictor as a conditional generative model that maps a representation of the current "state" together with the next action to a distribution over the next frame. In the original GameNGen [68] formulation, the state is implicit and is represented solely by a fixed-length window of the most recent observed frames. Concretely, letting

$$S_t = o_{t-L+1:t} = \big(o_{t-L+1}, \dots, o_t\big) \tag{3}$$

denote an L-frame context, the next observation is generated by querying a diffusion model conditioned on the current state S_t and action a_t:

$$o_{t+1} \sim p_\theta(o \mid S_t, a_t). \tag{4}$$

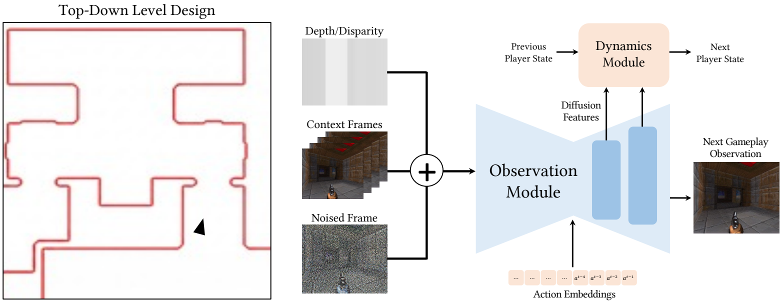


Fig. 2: Method overview. We introduce an explicit external memory and factor the diffusion game engine into three modules: ***Memory*** (map geometry and pose; Sec. 3.1), ***Observation*** (next-frame generation conditioned on history and memory readouts; Sec. 3.2), and ***Dynamics*** (pose update for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

Intuitively, S_t provides the visual history needed to maintain appearance and short-term temporal continuity, while a_t specifies the agent control signal that drives the transition.

In our work, we retain this interactive generative modeling setup but go beyond the conventional "frames-as-state" design [2, 7, 21–24, 48, 52] by introducing an explicitly controlled external memory [14, 76, 92]. 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

While GameNGen [68] represents the simulator state implicitly as a window of recent frames, we instead maintain an explicit state that separates persistent, low-dimensional game information from high-dimensional observations. Concretely, at timestep t we define the state as

$$S_t = \big(M,\ p_t,\ o_{t-L+1:t}\big), \tag{5}$$

where $o_{t-L+1:t} = (o_{t-L+1}, \dots, o_t)$ 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 a set of 2D vertices and line segments defining the walkable layout and walls:

$$\begin{aligned} M &= (V, E), \\ V &= \{v_i\}_{i=1}^{N_v}, \qquad v_i \in \mathbb{R}^2, \\ E &= \{e_j\}_{j=1}^{N_e}, \qquad e_j \in \{1, \dots, N_v\}^2, \end{aligned} \tag{6}$$

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,

$$p_t = (x_t, y_t, \theta_t) \in \mathbb{R}^2 \times \mathbb{S}^1, \tag{7}$$

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 visual observation conditioned on recent context, a geometric readout from external memory, and the next action. At timestep t, we model

$$o_{t+1} \sim p_\phi(o \mid o_{t-L+1:t},\, r_t,\, a_t)\,, \tag{8}$$

where r_t is a geometric conditioning signal derived 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 [14] 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 conditioning 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 [26] trained with the standard velocity-parameterization objective [57]. Given a ground-truth next frame o_{t+1}, we sample a diffusion timestep τ, construct a noised version according to the forward process, and train the UNet to match the corresponding velocity target. Concretely, we optimize

$$\mathcal{L}_{\text{obs}} = \mathbb{E}_{t,\tau}\left[\|v_\phi(\cdot \mid o_{t-L+1:t}, r_t, a_t; \tau) - v^\star(o_{t+1}, \tau)\|_2^2\right], \tag{9}$$

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 [13,33,46,54], we follow prior work [10,68] 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-dimensional representation (e.g., via pooling) 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:

$$\mathit{\Delta}\hat{p}_t = \mathcal{D}_\psi(p_t,\, a_t,\, r_t,\, f_t)\,, \tag{10}$$

where f_t denotes the aggregated UNet feature representation. We apply this increment to obtain $\hat{p}_{t+1}$, using angle wrapping for the orientation component.

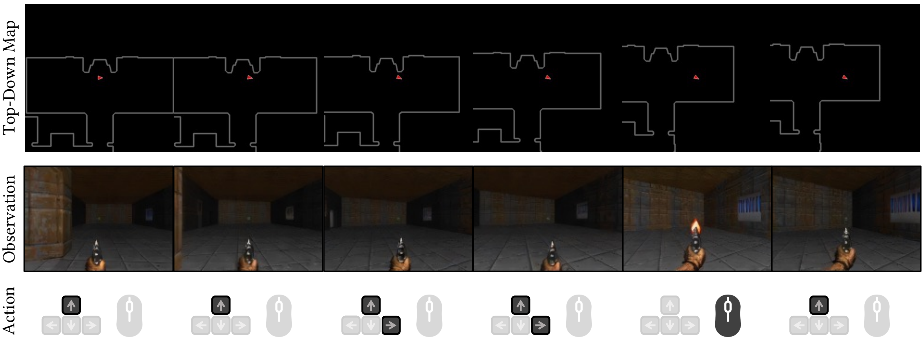


Fig. 3: Example rollouts under an authored map and action sequence. Top: minimap M with pose p_t (red arrow). Middle: generated first-person observations $\hat{o}_t$. Bottom: actions a_t. The viewpoint evolves coherently with the action inputs while adhering with the designed layout.

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 time, the system functions as an interactive simulator that repeatedly maps the current state and an action to the next observation and updated state:

$$(S_t, a_t) \mapsto (\hat{o}_{t+1}, S_{t+1}), \tag{11}$$

where $S_t = (M, p_t, o_{t-L+1:t})$ contains the static map M, the current pose p_t, and an L-frame context window.

For each timestep, we (1) query external memory to obtain the geometric readout r_t from the current pose and map, (2) sample the next frame $\hat{o}_{t+1}$ using the diffusion observation model conditioned on the context, geometry, and action, and (3) update the pose with the dynamics module using the action, geometry, and intermediate UNet features. To stabilize long rollouts, we use history guidance [61]: the conditional branch receives the clean context frames, while the unconditional branch receives a noised version of the context, encouraging fidelity to recent history while retaining robustness to imperfect inputs. The state is then advanced by shifting the context window and updating the pose for the next step.

4 Application I: Level Design

A primary advantage of an external memory is that it provides a direct handle for modifying the underlying structure of the world. By defining the world

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 [81]	0.415	**0.433**	0.396	18.74	**20.30**	17.19	0.488	0.397	0.578
ControlNet [87]	0.411	0.422	0.401	18.51	19.58	17.45	0.524	0.453	0.596
GameNGen [68]	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**

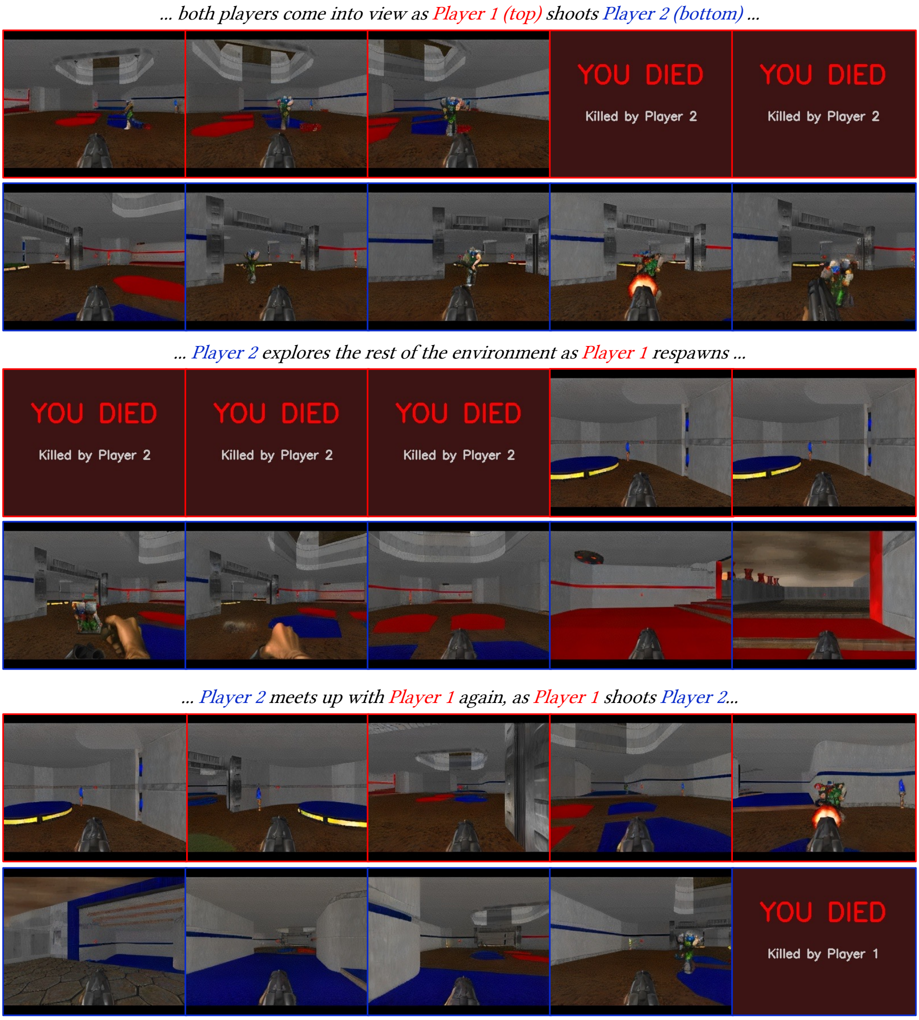
explicitly with course map structures, users can directly influence the structure of the environment before inference even begins. We evaluate this capability on ***level-conditioned gameplay generation***: synthesizing interactive first-person rollouts that remain consistent with a user-authored level layout. In this task, the model is given (i) a top-down map M specified as coarse line geometry (vertices and wall segments), (ii) an initial player pose p_0, and (iii) a sequence of actions $\{a_t\}_{t=0}^{T-1}$, and generates a corresponding sequence of observations $\{\hat{o}_t\}_{t=1}^{T}$. Unlike implicit-state diffusion game engines that must infer global structure from a finite history, our approach can query M at every timestep through an external-memory readout, 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 [50], which produces varied layouts while preserving valid Doom geometry. We then deploy a pre-trained Doom agent [40] 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. 1 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.



... both players come into view as Player 1 (top) shoots Player 2 (bottom) ...

... Player 2 explores the rest of the environment as Player 1 respawns ...

... Player 2 meets up with Player 1 again, as Player 1 shoots Player 2...

Fig. 4: Example Two-Player Gameplay Roll-out. Our method generates consistent first-person views for both players by maintaining a shared world memory. The roll-out shows a short two-player interaction: the players meet, and Player 1 kills Player 2, after which Player 2 is removed from the shared state. Player 1 then explores the map while Player 2 respawns and is re-added to the shared state. The players meet again, and Player 1 kills Player 2 once more. Note that both views are are consistent with each other, as actions from one player directly effects the next-frame observation generated by the other model. All game play frames are generated using the observation module. Frames shown during player death are not part of the model output, and are only shown for illustrative purposes.

4.3 Evaluations

We compare against relevant baselines. First, we include GameNGen [68], which models the entire engine as a single diffusion network without external memory, representing environment state implicitly through a finite window of past frames. We also compare against alternative approaches for conditioning on external state, specifically ControlNet [88] and IP-Adapter [81], which condition on the top-down minimap. For all methods, we initialize from the same initial observation and pose and roll out under the same action sequence for T steps. We then measure similarity to the ground-truth observations from the underlying simulator using SSIM (structural similarity) and LPIPS (perceptual distance), reporting averages over early and late rollout ranges to assess long-horizon stability.

As shown in Tab. 1, conditioning on external memory improves structural consistency, with larger gains in later rollout segments where implicit-state baselines are more prone to drift. Access to the authored map provides a stable geometric reference that anchors generation to the true layout over time, whereas the memory-free baseline increasingly hallucinates layout changes that compound under autoregressive sampling.

5 Application II: Multiplayer Interaction

The external memory not only enables editability of the environment, it also naturally extends to act as a shared state that multiple agents can condition on and update in during generated roll-outs. We evaluate this capability on ***multi-player gameplay generation***: synthesizing synchronized first-person rollouts for multiple players interacting within the same environment. In this application, the model is given (i) a shared top-down map M, (ii) initial player poses $\{p_0^{(i)}\}_{i=1}^{N}$, and (iii) per-player action sequences $\{a_t^{(i)}\}_{t=0}^{T-1}$, and generates per-player observations $\{\hat{o}_t^{(i)}\}_{t=1}^{T}$. During inference, all players query and modify the same external memory state, so that one player's actions can influence what other players observe. This is in direct contrast with "frames-as-state" models [2, 7, 21–24, 48, 52, 65], where state is entangled with the local observation history of every player, 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

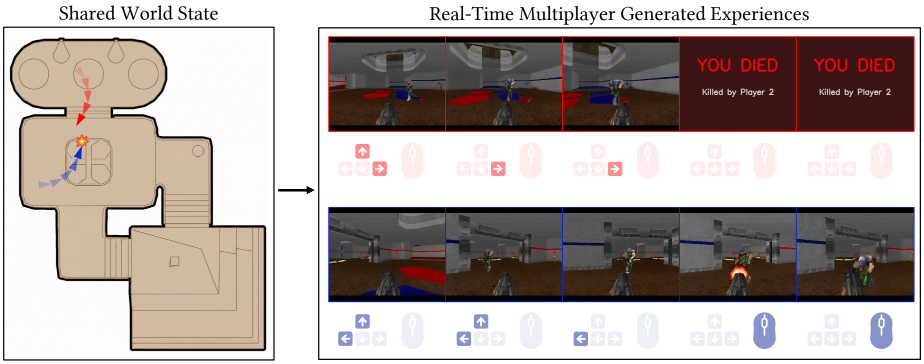


Fig. 5: Real-Time Interactive Multiplayer Generative Experiences. A shared consistent world state (left) enables consistent multiplayer generative experiences (right). Our method leverages a diffusion model conditioned on past frame observations, the next player action, and the external world state to generate gameplay roll-outs in real-time. The shared world state enables meaningful interactions between players, such as one player killing another (right).

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 viewpointsin a single "split-screen" observation [65] 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.

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 [78] and collect simulated Doom deathmatch sequences in which one pre-trained agent [40] 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. 4 shows a two-player rollout with a typical gameplay loop: the players start at different map locations, approach until they come into each other's view, and

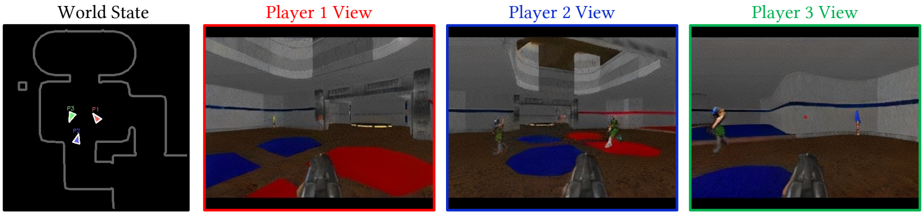


Fig. 6: Consistent Views in Three-Player. Our distributed model system supports an arbitrary number of players in the shared state despite only being trained on data from a single viewpoint. Player positions are accurately represented in all three viewpoints in the example above.

Player 1 kills Player 2, removing Player 2 from the shared state and triggering a death animation from Player 2's viewpoint while Player 1 continues moving. After a delay, Player 2 respawns (re-entering the shared state), the players meet again, and Player 2 is killed in a second encounter. Beyond visual fidelity, the striking result is that these interactions remain **mutually consistent** across viewpoints despite being generated autoregressively: when a player should be visible, they appear with the correct pose and location, and when they are dead or out of view, they disappear from the other player's camera while the victim still observes a coherent death/respawn sequence.

As described above, our method supports an arbitrary number of players by running separate Observation/Dynamics instances that share a common external world state. Importantly, each player's model runs independently, so adding more players does not slow down inference—we simply run additional per-player instances that read/write the same shared state. In our implementation, the full system runs at approximately **20 FPS** using a single NVIDIA A100 per player, making the resulting multiplayer rollouts not only consistent, but also interactive in real time. Fig. 6 shows consistent generations from three viewpoints, each aligned with the shared state: Player 1 sees no other players, Player 2 sees both Players 1 and 3, and Player 3 sees only Player 2.

5.4 Evaluations

For quantitative comparisons, we compare to CotrolNet [88] and IP-Adapter [81]. Another natural baseline for multiplayer is to jointly model all player views with a single network, resembling a "split-screen" game [65]. To highlight the role of external memory, we compare our approach against a two-player split-screen diffusion model trained to predict both viewpoints on the same map without any explicit memory mechanism. In this baseline, multiplayer consistency must be learned purely from the limited observation histories of both players, since there is no shared structured state that persists beyond the context window.

We generate a set of random multiplayer gameplay trajectories from the environment and use them as ground truth. For each trajectory, we roll out all models by forcing each player to follow the ground-truth action stream, producing paired

Table 2: Multiplayer consistency evaluation using opponent-presence detection. We report accuracy, precision, and recall of a VLM-based judge on generated frames against ground-truth visibility labels. MultiGen consistently outperforms all baselines.

Method	Accuracy ↑	Precision ↑	Recall ↑
IP-Adapter [81]	62.12%	84.62%	40.73%
ControlNet [88]	60.71%	82.16%	39.20%
Split-screen [65]	65.31%	86.86%	44.59%
Ours (MultiGen)	**75.38%**	**88.12%**	**65.07%**

generated views for Player 1 and Player 2 over T steps. We then measure multiplayer consistency via ***opponent presence accuracy***: whether the generated frame contains the other player when they should be visible. We use a pre-trained vision-language model (VLM) as an automated judge [51]. Concretely, for each timestep and each player's viewpoint, we query the VLM to determine whether the opponent is present in the generated frame, and compute accuracy by agreement with the same VLM judgement on the corresponding ground-truth frame. We report **accuracy**, **precision**, and **recall** (opponent visible = positive). Precision penalizes hallucinated opponents (false positives), while recall penalizes missed opponents when they should be visible (false negatives).

6 Discussion and Conclusions

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. 3, increasing the context length consistently improves fidelity, reflected by higher SSIM/PSNR and lower LPIPS.

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

Table 3: Context frame ablation.

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 a MultiGen, 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.

Acknowledgments.

We would like to thank Yang Zheng, Hadi Alzayer, Zizhang Li, Lior Yariv, Ilker Oguz, Seung-Woo Nam, Julian Quevedo, Bryan Chiang, Eric Li, and Alec Yiu for help with testing real-time demo and fruitful discussions. R.P. is supported by the Stanford Graduate Fellowship.

References

1. Anjum, A., Li, Y., Law, N., Charity, M., Togelius, J.: 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), `https://api.semanticscholar.org/CorpusID:268249064`
2. Ball, P.J., Bauer, J., Belletti, F., Brownfield, B., Ephrat, A., Fruchter, S., Gupta, A., Holsheimer, K., Holynski, A., Hron, J., Kaplanis, C., Limont, M., McGill, M., Oliveira, Y., Parker-Holder, J., Perbet, F., Scully, G., Shar, J., Spencer, S., Tov, O., Villegas, R., Wang, E., Yung, J., Baetu, C., Berbel, J., Bridson, D., Bruce, J., Buttimore, G., Chakera, S., Chandra, B., Collins, P., Cullum, A., Damoc, B., Dasagi, V., Gazeau, M., Gbadamosi, C., Han, W., Hirst, E., Kachra, A., Kerley, L., Kjems, K., Knoepfel, E., Koriakin, V., Lo, J., Lu, C., Mehring, Z., Moufarek, A., Nandwani, H., Oliveira, V., Pardo, F., Park, J., Pierson, A., Poole, B., Ran, H., Salimans, T., Sanchez, M., Saprykin, I., Shen, A., Sidhwani, S., Smith, D., Stanton, J., Tomlinson, H., Vijaykumar, D., Wang, L., Wingfield, P., Wong, N., Xu, K., Yew, C., Young, N., Zubov, V., Eck, D., Erhan, D., Kavukcuoglu, K., Hassabis, D., Gharamani, Z., Hadsell, R., van den Oord, A., Mosseri, I., Bolton, A., Singh, S., Rocktäschel, T.: Genie 3: A new frontier for world models (2025)
3. Bar-Tal, O., Chefer, H., Tov, O., Herrmann, C., Paiss, R., Zada, S., Ephrat, A., Hur, J., Li, Y., Michaeli, T., Wang, O., Sun, D., Dekel, T., Mosseri, I.: Lumiere: A space-time diffusion model for video generation. SIGGRAPH Asia 2024 Conference Papers (2024), `https://api.semanticscholar.org/CorpusID:267095113`
4. Blattmann, A., Dockhorn, T., Kulal, S., Mendelevitch, D., Kilian, M., Lorenz, D.: Stable video diffusion: Scaling latent video diffusion models to large datasets. ArXiv **abs/2311.15127** (2023), `https://api.semanticscholar.org/CorpusID:265312551`
5. Blattmann, A., Rombach, R., Ling, H., Dockhorn, T., Kim, S.W., Fidler, S., Kreis, K.: 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 (2023), `https://api.semanticscholar.org/CorpusID:258187553`

6. Brooks, T., Peebles, B., Holmes, C., DePue, W., Guo, Y., Jing, L., Schnurr, D., Taylor, J., Luhman, T., Luhman, E., Ng, C., Wang, R., Ramesh, A.: Video generation models as world simulators (2024), `https://openai.com/research/video-generation-models-as-world-simulators`
7. Bruce, J., Dennis, M., Edwards, A., Parker-Holder, J., Shi, Y., Hughes, E., Lai, M., Mavalankar, A., Steigerwald, R., Apps, C., Aytar, Y., Bechtle, S., Behbahani, F.M.P., Chan, S., Heess, N.M.O., Gonzalez, L., Osindero, S., Ozair, S., Reed, S., Zhang, J., Zolna, K., Clune, J., de Freitas, N., Singh, S., Rocktaschel, T.: Genie: Generative interactive environments. ArXiv **abs/2402.15391** (2024), `https://api.semanticscholar.org/CorpusID:267897982`
8. Cai, S., Yang, C., Zhang, L., Guo, Y., Xiao, J., Yang, Z., Xu, Y., Yang, Z., Yuille, A., Guibas, L.J., Agrawala, M., Jiang, L., Wetzstein, G.: Mixture of contexts for long video generation. ArXiv **abs/2508.21058** (2025), `https://api.semanticscholar.org/CorpusID:280950315`
9. Che, H., He, X., Liu, Q., Jin, C., Chen, H.: Gamegen-x: Interactive open-world game video generation. ArXiv **abs/2411.00769** (2024), `https://api.semanticscholar.org/CorpusID:273798141`
10. Chen, B., Monso, D.M., Du, Y., Simchowitz, M., Tedrake, R., Sitzmann, V.: Diffusion forcing: Next-token prediction meets full-sequence diffusion. ArXiv **abs/2407.01392** (2024), `https://api.semanticscholar.org/CorpusID:270869622`
11. Chung, J.J.Y., Kreminski, M.: Patchview: Llm-powered worldbuilding with generative dust and magnet visualization. Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology (2024), `https://api.semanticscholar.org/CorpusID:271769283`
12. Chung, J.J.Y., Roemmele, M., Kreminski, M.: 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), `https://api.semanticscholar.org/CorpusID:273290335`
13. Cui, J., Wu, J., Li, M., Yang, T., Li, X., Wang, R., Bai, A., Ban, Y., jui Hsieh, C.: Self-forcing++: Towards minute-scale high-quality video generation. ArXiv **abs/2510.02283** (2025), `https://api.semanticscholar.org/CorpusID:281724666`
14. Deng, B., Tucker, R., Li, Z., Guibas, L.J., Snavely, N., Wetzstein, G.: Streetscapes: Large-scale consistent street view generation using autoregressive video diffusion. ACM SIGGRAPH 2024 Conference Papers (2024), `https://api.semanticscholar.org/CorpusID:271165095`
15. Gao, K., Shi, J., Zhang, H., Wang, C., Xiao, J., Chen, L.: Ca2-vdm: Efficient autoregressive video diffusion model with causal generation and cache sharing. ArXiv **abs/2411.16375** (2024), `https://api.semanticscholar.org/CorpusID:274235095`
16. Gu, Y., Mao, W., Shou, M.Z.: Long-context autoregressive video modeling with next-frame prediction. ArXiv **abs/2503.19325** (2025), `https://api.semanticscholar.org/CorpusID:277313237`
17. Guo, Y., Yang, C., Yang, Z., Ma, Z., Lin, Z., Yang, Z., Lin, D., Jiang, L.: Long context tuning for video generation. ArXiv **abs/2503.10589** (2025), `https://api.semanticscholar.org/CorpusID:276961453`
18. Gupta, A., Yu, L., Sohn, K., Gu, X., Hahn, M., Li, F.F., Essa, I., Jiang, L., Lezama, J.: Photorealistic video generation with diffusion models. In: European Conference on Computer Vision (2023), `https://api.semanticscholar.org/CorpusID:266163109`

19. Guzdial, M.J., Riedl, M.O.: Game level generation from gameplay videos. In: Artificial Intelligence and Interactive Digital Entertainment Conference (2021), `https://api.semanticscholar.org/CorpusID:37001924`
20. HaCohen, Y., Chiprut, N., Brazowski, B., Shalem, D., Moshe, D.P., Richardson, E., Levin, E.I., Shiran, G., Zabari, N., Gordon, O., Panet, P., Weissbuch, S., Kulikov, V., Bitterman, Y., Melumian, Z., Bibi, O.: Ltx-video: Realtime video latent diffusion. ArXiv **abs/2501.00103** (2024), `https://api.semanticscholar.org/CorpusID:275212083`
21. Hafner, D., Lillicrap, T., Ba, J., Norouzi, M.: Dream to control: Learning behaviors by latent imagination (2020), `https://arxiv.org/abs/1912.01603`
22. Hafner, D., Lillicrap, T., Norouzi, M., Ba, J.: Mastering atari with discrete world models (2022), `https://arxiv.org/abs/2010.02193`
23. Hafner, D., Pasukonis, J., Ba, J., Lillicrap, T.: Mastering diverse domains through world models (2024), `https://arxiv.org/abs/2301.04104`
24. Hafner, D., Yan, W., Lillicrap, T.: Training agents inside of scalable world models (2025), `https://arxiv.org/abs/2509.24527`
25. Ho, J., Chan, W., Saharia, C., Whang, J., Gao, R., Gritsenko, A.A., Kingma, D.P., Poole, B., Norouzi, M., Fleet, D.J., Salimans, T.: Imagen video: High definition video generation with diffusion models. ArXiv **abs/2210.02303** (2022), `https://api.semanticscholar.org/CorpusID:252715883`
26. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models (2020), `https://arxiv.org/abs/2006.11239`
27. Ho, J., Salimans, T., Gritsenko, A., Chan, W., Norouzi, M., Fleet, D.J.: Video diffusion models. ArXiv **abs/2204.03458** (2022), `https://api.semanticscholar.org/CorpusID:248006185`
28. Hong, W., Ding, M., Zheng, W., Liu, X., Tang, J.: Cogvideo: Large-scale pretraining for text-to-video generation via transformers. ArXiv **abs/2205.15868** (2022), `https://api.semanticscholar.org/CorpusID:249209614`
29. Hong, Y., Mei, Y., Ge, C., Xu, Y., Zhou, Y., Bi, S., Hold-Geoffroy, Y., Roberts, M., Fisher, M., Shechtman, E., et al.: Relic: Interactive video world model with long-horizon memory. arXiv preprint arXiv:2512.04040 (2025)
30. Hu, C., Zhao, Y., Liu, J.: Game generation via large language models. 2024 IEEE Conference on Games (CoG) pp. 1–4 (2024), `https://api.semanticscholar.org/CorpusID:269149102`
31. Hu, J., Hu, S., Song, Y., Huang, Y., Wang, M., Zhou, H., Liu, Z., Ma, W.Y., Sun, M.: Acdit: Interpolating autoregressive conditional modeling and diffusion transformer. ArXiv **abs/2412.07720** (2024), `https://api.semanticscholar.org/CorpusID:274610804`
32. Huang, J., Hu, X., Han, B., Shi, S., Tian, Z., He, T., Jiang, L.: Memory forcing: Spatio-temporal memory for consistent scene generation on minecraft (2025), `https://arxiv.org/abs/2510.03198`
33. Huang, X., Li, Z., He, G., Zhou, M., Shechtman, E.: Self forcing: Bridging the train-test gap in autoregressive video diffusion. ArXiv **abs/2506.08009** (2025), `https://api.semanticscholar.org/CorpusID:279251392`
34. id Software: Doom (1993), mS-DOS game
35. Jin, Y., Sun, Z., Li, N., Xu, K., Jiang, H., Nan, Z., Huang, Q., Song, Y., Mu, Y., Lin, Z.: Pyramidal flow matching for efficient video generative modeling. ArXiv **abs/2410.05954** (2024), `https://api.semanticscholar.org/CorpusID:273228937`

36. Kim, S.W., Zhou, Y., Philion, J., Torralba, A., Fidler, S.: Learning to simulate dynamic environments with gamegan. 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) pp. 1228–1237 (2020), https://api.semanticscholar.org/CorpusID:218869555
37. Kondratyuk, D., Yu, L., Gu, X., Lezama, J., Huang, J., Hornung, R., Adam, H., Akbari, H., Alon, Y., Birodkar, V., Cheng, Y., Chiu, M.C., Dillon, J., Essa, I., Gupta, A., Hahn, M., Hauth, A., Hendon, D., Martinez, A., Minnen, D.C., Ross, D.A., Schindler, G., Sirotenko, M., Sohn, K., Somandepalli, K., Wang, H., Yan, J., Yang, M., Yang, X., Seybold, B., Jiang, L.: Videopoet: A large language model for zero-shot video generation. ArXiv **abs/2312.14125** (2023), https://api.semanticscholar.org/CorpusID:266435847
38. Kong, W., Tian, Q., Zhang, Z., Min, R., Dai, Z., Zhou, J., Xiong, J.L., Li, X., Wu, B., Zhang, J., Wu, K., Lin, Q., Yuan, J., Long, Y., Wang, A., Wang, A., Li, C., Huang, D., Yang, F., Tan, H., Wang, H., Song, J., Bai, J., Wu, J., Xue, J., Wang, J., Wang, K., Liu, M., Li, P.Y., Li, S., Wang, W., Yu, W., Deng, X., Li, Y., Chen, Y., Cui, Y., Peng, Y., Yu, Z., He, Z., Xu, Z., Zhou, Z., Xu, Z., Tao, Y.D., Lu, Q., Liu, S., Zhou, D., Wang, H., Yang, Y., Wang, D., Liu, Y., Jiang, J., Zhong, C.: Hunyuanvideo: A systematic framework for large video generative models. ArXiv **abs/2412.03603** (2024), https://api.semanticscholar.org/CorpusID:274514554
39. Kumaran, V., Mott, B.W., Lester, J.C.: Generating game levels for multiple distinct games with a common latent space. In: Artificial Intelligence and Interactive Digital Entertainment Conference (2020), https://api.semanticscholar.org/CorpusID:221737009
40. Lample, G., Chaplot, D.S.: Playing fps games with deep reinforcement learning (2018), https://arxiv.org/abs/1609.05521
41. Li, J., Tang, J., Xu, Z.T., Wu, L., Zhou, Y., Shao, S., Yu, T., Cao, Z., Lu, Q.: Hunyuan-gamecraft: High-dynamic interactive game video generation with hybrid history condition. ArXiv **abs/2506.17201** (2025), https://api.semanticscholar.org/CorpusID:279465382
42. Li, R., Torr, P., Vedaldi, A., Jakab, T.: Vmem: Consistent interactive video scene generation with surfel-indexed view memory (2025), https://arxiv.org/abs/2506.18903
43. Li, T., Tian, Y., Li, H., Deng, M., He, K.: Autoregressive image generation without vector quantization. ArXiv **abs/2406.11838** (2024), https://api.semanticscholar.org/CorpusID:270560593
44. Li, Z., Hu, S., Liu, S., Zhou, L., Choi, J., Meng, L., Guo, X., Li, J., Ling, H., Wei, F.: Arlon: Boosting diffusion transformers with autoregressive models for long video generation. ArXiv **abs/2410.20502** (2024), https://api.semanticscholar.org/CorpusID:273653802
45. Liu, H., Liu, S., Zhou, Z., Xu, M., Xie, Y., Han, X., P'erez, J.C., Liu, D., Kahatapitiya, K., Jia, M., Wu, J.C., He, S., Xiang, T., Schmidhuber, J., P'erez-R'ua, J.M.: Mardini: Masked autoregressive diffusion for video generation at scale. Trans. Mach. Learn. Res. **2025** (2024), https://api.semanticscholar.org/CorpusID:273655157
46. Liu, K., Hu, W., Xu, J., Shan, Y., Lu, S.: Rolling forcing: Autoregressive long video diffusion in real time. ArXiv **abs/2509.25161** (2025), https://api.semanticscholar.org/CorpusID:281676207
47. Liu, Y., Ren, Y., Cun, X., Artola, A., Liu, Y., Zeng, T., Chan, R.H., Morel, J.M.: Redefining temporal modeling in video diffusion: The vectorized timestep

approach. ArXiv **abs/2410.03160** (2024), `https://api.semanticscholar.org/CorpusID:273162851`

48. Mao, X., Lin, S., Li, Z., Li, C., Peng, W., He, T., Pang, J., Chi, M., Qiao, Y., Zhang, K.: Yume: An interactive world generation model. arXiv preprint arXiv:2507.17744 (2025)
49. Nasir, M.U., Togelius, J.: Practical pcg through large language models. 2023 IEEE Conference on Games (CoG) pp. 1–4 (2023), `https://api.semanticscholar.org/CorpusID:258960371`
50. Obsidian Community: Obsidian level generator. `https://obsidian-level-maker.github.io/`, accessed: 2026-03-03
51. OpenAI, :, Jaech, A., Kalai, A., Lerer, A., Richardson, A., El-Kishky, A., Low, A., Helyar, A., Madry, A., Beutel, A., Carney, A., Iftimie, A., Karpenko, A., Passos, A.T., Neitz, A., Prokofiev, A., Wei, A., Tam, A., Bennett, A., Kumar, A., Saraiva, A., Vallone, A., Duberstein, A., Kondrich, A., Mishchenko, A., Applebaum, A., Jiang, A., Nair, A., Zoph, B., Ghorbani, B., Rossen, B., Sokolowsky, B., Barak, B., McGrew, B., Minaiev, B., Hao, B., Baker, B., Houghton, B., McKinzie, B., Eastman, B., Lugaresi, C., Bassin, C., Hudson, C., Li, C.M., de Bourcy, C., Voss, C., Shen, C., Zhang, C., Koch, C., Orsinger, C., Hesse, C., Fischer, C., Chan, C., Roberts, D., Kappler, D., Levy, D., Selsam, D., Dohan, D., Farhi, D., Mely, D., Robinson, D., Tsipras, D., Li, D., Oprica, D., Freeman, E., Zhang, E., Wong, E., Proehl, E., Cheung, E., Mitchell, E., Wallace, E., Ritter, E., Mays, E., Wang, F., Such, F.P., Raso, F., Leoni, F., Tsimpourlas, F., Song, F., von Lohmann, F., Sulit, F., Salmon, G., Parascandolo, G., Chabot, G., Zhao, G., Brockman, G., Leclerc, G., Salman, H., Bao, H., Sheng, H., Andrin, H., Bagherinezhad, H., Ren, H., Lightman, H., Chung, H.W., Kivlichan, I., O'Connell, I., Osband, I., Gilaberte, I.C., Akkaya, I., Kostrikov, I., Sutskever, I., Kofman, I., Pachocki, J., Lennon, J., Wei, J., Harb, J., Twore, J., Feng, J., Yu, J., Weng, J., Tang, J., Yu, J., Candela, J.Q., Palermo, J., Parish, J., Heidecke, J., Hallman, J., Rizzo, J., Gordon, J., Uesato, J., Ward, J., Huizinga, J., Wang, J., Chen, K., Xiao, K., Singhal, K., Nguyen, K., Cobbe, K., Shi, K., Wood, K., Rimbach, K., Gu-Lemberg, K., Liu, K., Lu, K., Stone, K., Yu, K., Ahmad, L., Yang, L., Liu, L., Maksin, L., Ho, L., Fedus, L., Weng, L., Li, L., McCallum, L., Held, L., Kuhn, L., Kondraciuk, L., Kaiser, L., Metz, L., Boyd, M., Trebacz, M., Joglekar, M., Chen, M., Tintor, M., Meyer, M., Jones, M., Kaufer, M., Schwarzer, M., Shah, M., Yatbaz, M., Guan, M.Y., Xu, M., Yan, M., Glaese, M., Chen, M., Lampe, M., Malek, M., Wang, M., Fradin, M., McClay, M., Pavlov, M., Wang, M., Wang, M., Murati, M., Bavarian, M., Rohaninejad, M., McAleese, N., Chowdhury, N., Chowdhury, N., Ryder, N., Tezak, N., Brown, N., Nachum, O., Boiko, O., Murk, O., Watkins, O., Chao, P., Ashbourne, P., Izmailov, P., Zhokhov, P., Dias, R., Arora, R., Lin, R., Lopes, R.G., Gaon, R., Miyara, R., Leike, R., Hwang, R., Garg, R., Brown, R., James, R., Shu, R., Cheu, R., Greene, R., Jain, S., Altman, S., Toizer, S., Toyer, S., Miserendino, S., Agarwal, S., Hernandez, S., Baker, S., McKinney, S., Yan, S., Zhao, S., Hu, S., Santurkar, S., Chaudhuri, S.R., Zhang, S., Fu, S., Papay, S., Lin, S., Balaji, S., Sanjeev, S., Sidor, S., Broda, T., Clark, A., Wang, T., Gordon, T., Sanders, T., Patwardhan, T., Sottiaux, T., Degry, T., Dimson, T., Zheng, T., Garipov, T., Stasi, T., Bansal, T., Creech, T., Peterson, T., Eloundou, T., Qi, V., Kosaraju, V., Monaco, V., Pong, V., Fomenko, V., Zheng, W., Zhou, W., McCabe, W., Zaremba, W., Dubois, Y., Lu, Y., Chen, Y., Cha, Y., Bai, Y., He, Y., Zhang, Y., Wang, Y., Shao, Z., Li, Z.: Openai o1 system card (2024), `https://arxiv.org/abs/2412.16720`

52. Parker-Holder, J., Ball, P., Bruce, J., Dasagi, V., Holsheimer, K., Kaplanis, C., Moufarek, A., Scully, G., Shar, J., Shi, J., Spencer, S., Yung, J., Dennis, M., Kenjeyev, S., Long, S., Mnih, V., Chan, H., Gazeau, M., Li, B., Pardo, F., Wang, L., Zhang, L., Besse, F., Harley, T., Mitenkova, A., Wang, J., Clune, J., Hassabis, D., Hadsell, R., Bolton, A., Singh, S., Rocktäschel, T.: Genie 2: A large-scale foundation world model (2024), https://deepmind.google/discover/blog/genie-2-a-large-scale-foundation-world-model/
53. Peebles, W.S., Xie, S.: Scalable diffusion models with transformers. 2023 IEEE/CVF International Conference on Computer Vision (ICCV) pp. 4172–4182 (2022), https://api.semanticscholar.org/CorpusID:254854389
54. Po, R., Chan, E.R., Chen, C., Wetzstein, G.: Bagger: Backwards aggregation for mitigating drift in autoregressive video diffusion models (2025), https://arxiv.org/abs/2512.12080
55. Polyak, A., Zohar, A., Brown, A., Tjandra, A., Sinha, A., Lee, A., Vyas, A., Shi, B., Ma, C.Y., Chuang, C.Y., Yan, D., Choudhary, D., Wang, D., Sethi, G., Pang, G., Ma, H., Misra, I., Hou, J., Wang, J., ran Jagadeesh, K., Li, K., Zhang, L., Singh, M., Williamson, M., Le, M., Yu, M., Singh, M.K., Zhang, P., Vajda, P., Duval, Q., Girdhar, R., Sumbaly, R., Rambhatla, S.S., Tsai, S.S., Azadi, S., Datta, S., Chen, S., Bell, S., Ramaswamy, S., Sheynin, S., Bhattacharya, S., Motwani, S., Xu, T., Li, T., Hou, T., Hsu, W.N., Yin, X., Dai, X., Taigman, Y., Luo, Y., Liu, Y.C., Wu, Y.C., Zhao, Y., Kirstain, Y., He, Z., He, Z., Pumarola, A., Thabet, A.K., Sanakoyeu, A., Mallya, A., Guo, B., Araya, B., Kerr, B., Wood, C., Liu, C., Peng, C., Vengertsev, D., Schonfeld, E., Blanchard, E., Juefei-Xu, F., Nord, F., Liang, J., Hoffman, J., Kohler, J., Fire, K., Sivakumar, K., Chen, L., Yu, L., Gao, L., Georgopoulos, M., Moritz, R., Sampson, S.K., Li, S., Parmeggiani, S., Fine, S., Fowler, T., Petrovic, V., Du, Y.: Movie gen: A cast of media foundation models (2024), https://api.semanticscholar.org/CorpusID:273403698
56. Ren, S., Ma, S., Sun, X., Wei, F.: Next block prediction: Video generation via semi-autoregressive modeling. ArXiv **abs/2502.07737** (2025), https://api.semanticscholar.org/CorpusID:276258605
57. Salimans, T., Ho, J.: Progressive distillation for fast sampling of diffusion models (2022), https://arxiv.org/abs/2202.00512
58. Schubert, F., Awiszus, M., Rosenhahn, B.: Toad-gan: A flexible framework for few-shot level generation in token-based games. IEEE Transactions on Games **14**, 284–293 (2021), https://api.semanticscholar.org/CorpusID:233520445
59. Shin, J., Li, Z., Zhang, R., Zhu, J.Y., Park, J., Schechtman, E., Huang, X.: Motionstream: Real-time video generation with interactive motion controls (2025), https://api.semanticscholar.org/CorpusID:282739800
60. Snodgrass, S., Ontañón, S.: Experiments in map generation using markov chains. In: International Conference on Foundations of Digital Games (2014), https://api.semanticscholar.org/CorpusID:15083922
61. Song, K., Chen, B., Simchowitz, M., Du, Y., Tedrake, R., Sitzmann, V.: History-guided video diffusion. ArXiv **abs/2502.06764** (2025), https://api.semanticscholar.org/CorpusID:276249479
62. Sudhakaran, S., Gonz'alez-Duque, M., Glanois, C., Freiberger, M.A., Najarro, E., Risi, S.: Mariogpt: Open-ended text2level generation through large language models. ArXiv **abs/2302.05981** (2023), https://api.semanticscholar.org/CorpusID:256827347
63. Summerville, A.J., Mateas, M.: Super mario as a string: Platformer level generation via lstms. ArXiv **abs/1603.00930** (2016), https://api.semanticscholar.org/CorpusID:16077591

64. Summerville, A.J., Snodgrass, S., Guzdial, M.J., Holmgård, C., Hoover, A.K., Isaksen, A., Nealen, A., Togelius, J.: Procedural content generation via machine learning (pcgml). IEEE Transactions on Games **10**, 257–270 (2017), https://api.semanticscholar.org/CorpusID:9950600
65. team, E.: Introducing multiverse: The first ai multiplayer world model (2025), https://enigma.inc/blog
66. Todd, G., Earle, S., Nasir, M.U., Green, M.C., Togelius, J.: Level generation through large language models. Proceedings of the 18th International Conference on the Foundations of Digital Games (2023), https://api.semanticscholar.org/CorpusID:256826896
67. Treanor, M., Zook, A., Eladhari, M.P., Togelius, J., Smith, G., Cook, M., Thompson, T., Magerko, B., Levine, J., Smith, A.M.: Ai-based game design patterns. In: International Conference on Foundations of Digital Games (2015), https://api.semanticscholar.org/CorpusID:3099158
68. Valevski, D., Leviathan, Y., Arar, M., Fruchter, S.: Diffusion models are real-time game engines. ArXiv **abs/2408.14837** (2024), https://api.semanticscholar.org/CorpusID:271962839
69. Villegas, R., Babaeizadeh, M., Kindermans, P.J., Moraldo, H., Zhang, H., Saffar, M.T., Castro, S., Kunze, J., Erhan, D.: Phenaki: Variable length video generation from open domain textual description. ArXiv **abs/2210.02399** (2022), https://api.semanticscholar.org/CorpusID:252715594
70. Volz, V., Schrum, J., Liu, J., Lucas, S.M.M., Smith, A.M., Risi, S.: Evolving mario levels in the latent space of a deep convolutional generative adversarial network. Proceedings of the Genetic and Evolutionary Computation Conference (2018), https://api.semanticscholar.org/CorpusID:13676024
71. Wang, A., Ai, B., Wen, B., Mao, C., Xie, C.W., Chen, D., Yu, F., Zhao, H., Yang, J., Zeng, J., Wang, J., Zhang, J., Zhou, J., Wang, J., Chen, J., Zhu, K., Zhao, K., Yan, K., Huang, L., Meng, X., Zhang, N., Li, P., Wu, P., Chu, R., Feng, R., Zhang, S., Sun, S., Fang, T., Wang, T., Gui, T., Weng, T., Shen, T., Lin, W., Wang, W., Wang, W., Zhou, W.C., Wang, W., Shen, W., Yu, W., Shi, X., Huang, X., Xu, X., Kou, Y., Lv, Y.M., Li, Y., Liu, Y., Wang, Y., Zhang, Y., Huang, Y., Li, Y., Wu, Y., Liu, Y., Pan, Y., Zheng, Y., Hong, Y., Shi, Y., Feng, Y., Jiang, Z., Han, Z., Wu, Z., Liu, Z.: Wan: Open and advanced large-scale video generative models. ArXiv **abs/2503.20314** (2025), https://api.semanticscholar.org/CorpusID:277321639
72. Wang, Y., Xiong, T., Zhou, D., Lin, Z., Zhao, Y., Kang, B., Feng, J., Liu, X.: Loong: Generating minute-level long videos with autoregressive language models. ArXiv **abs/2410.02757** (2024), https://api.semanticscholar.org/CorpusID:273098341
73. Weissenborn, D., Täckström, O., Uszkoreit, J.: Scaling autoregressive video models. ArXiv **abs/1906.02634** (2019), https://api.semanticscholar.org/CorpusID:174802916
74. Weng, W., Feng, R., Wang, Y., Dai, Q., Wang, C., Yin, D., Zhao, Z., Qiu, K., Bao, J., Yuan, Y., Luo, C., Zhang, Y., Xiong, Z.: 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), https://api.semanticscholar.org/CorpusID:265506663
75. Wu, R., He, X., Cheng, M., Yang, T., Zhang, Y., Kang, Z., Cai, X., Wei, X., Guo, C., Li, C., Cheng, M.M.: Infinite-world: Scaling interactive world models to 1000-frame horizons via pose-free hierarchical memory (2026), https://arxiv.org/abs/2602.02393

76. Wu, T., Yang, S., Po, R., Xu, Y., Liu, Z., Lin, D., Wetzstein, G.: Video world models with long-term spatial memory. ArXiv **abs/2506.05284** (2025), `https://api.semanticscholar.org/CorpusID:279244178`
77. Wu, X., Zhang, G., Xu, Z., Zhou, Y., Lu, Q., He, X.: Pack and force your memory: Long-form and consistent video generation. arXiv preprint arXiv:2510.01784 (2025)
78. Wydmuch, M., Kempka, M., Jaśkowski, W.: ViZDoom Competitions: Playing Doom from Pixels. IEEE Transactions on Games **11**(3), 248–259 (2019). `https://doi.org/10.1109/TG.2018.2877047`, the 2022 IEEE Transactions on Games Outstanding Paper Award
79. Yan, W., Zhang, Y., Abbeel, P., Srinivas, A.: Videogpt: Video generation using vq-vae and transformers. ArXiv **abs/2104.10157** (2021), `https://api.semanticscholar.org/CorpusID:233307257`
80. Yang, Z., Teng, J., Zheng, W., Ding, M., Huang, S., Xu, J., Yang, Y., Hong, W., Zhang, X., Feng, G., Yin, D., Gu, X., Zhang, Y., Wang, W., Cheng, Y., Liu, T., Xu, B., Dong, Y., Tang, J.: Cogvideox: Text-to-video diffusion models with an expert transformer. ArXiv **abs/2408.06072** (2024), `https://api.semanticscholar.org/CorpusID:271855655`
81. Ye, H., Zhang, J., Liu, S., Han, X., Yang, W.: Ip-adapter: Text compatible image prompt adapter for text-to-image diffusion models (2023), `https://arxiv.org/abs/2308.06721`
82. Yu, J., Bai, J., Qin, Y., Liu, Q., Wang, X., Wan, P., Zhang, D., Liu, X.: Context as memory: Scene-consistent interactive long video generation with memory retrieval (2025), `https://arxiv.org/abs/2506.03141`
83. Yu, J., Qin, Y., Wang, X., Wan, P., Zhang, D., Liu, X.: Gamefactory: Creating new games with generative interactive videos. ArXiv **abs/2501.08325** (2025), `https://api.semanticscholar.org/CorpusID:275515409`
84. Zala, A., Cho, J., Lin, H., Yoon, J., Bansal, M.: Envgen: Generating and adapting environments via llms for training embodied agents. ArXiv **abs/2403.12014** (2024), `https://api.semanticscholar.org/CorpusID:268531207`
85. Zhang, D.J., Wu, J.Z., Liu, J.W., Zhao, R., Ran, L., Gu, Y., Gao, D., Shou, M.Z.: Show-1: Marrying pixel and latent diffusion models for text-to-video generation. International Journal of Computer Vision **133**(4), 1879–1893 (2025)
86. Zhang, L., Cai, S., Li, M., Wetzstein, G., Agrawala, M.: Frame context packing and drift prevention in next-frame-prediction video diffusion models (2025), `https://api.semanticscholar.org/CorpusID:277857265`
87. Zhang, L., Rao, A., Agrawala, M.: Adding conditional control to text-to-image diffusion models. 2023 IEEE/CVF International Conference on Computer Vision (ICCV) pp. 3813–3824 (2023), `https://api.semanticscholar.org/CorpusID:256827727`
88. Zhang, L., Rao, A., Agrawala, M.: Adding conditional control to text-to-image diffusion models (2023), `https://arxiv.org/abs/2302.05543`
89. Zhang, T., Bi, S., Hong, Y., Zhang, K., Luan, F., Yang, S., Sunkavalli, K., Freeman, W.T., Tan, H.: Test-time training done right. ArXiv **abs/2505.23884** (2025), `https://api.semanticscholar.org/CorpusID:279071244`
90. Zhang, Y., Peng, C., Wang, B., Wang, P., Zhu, Q., Kang, F., Jiang, B., Gao, Z., Li, E., Liu, Y., et al.: Matrix-game: Interactive world foundation model. arXiv preprint arXiv:2506.18701 (2025)
91. Zhang, Y., Jiang, J., Ma, G., Lu, Z., Huang, H., min Yuan, J., Duan, N.: Generative pre-trained autoregressive diffusion transformer. ArXiv **abs/2505.07344** (2025), `https://api.semanticscholar.org/CorpusID:278501597`

92. Zhao, J., Wei, F., Liu, Z., Zhang, H., Xu, C., Lu, Y.: Spatia: Video generation with updatable spatial memory (2025), https://arxiv.org/abs/2512.15716
93. Zhou, H., Zhu, J., Mateas, M., Wardrip-Fruin, N.: 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), https://api.semanticscholar.org/CorpusID:270963065