VALERANT: AN AUTOMATIC NAVIGABLE GAME MAP GENERATOR VIA ACTION-CONDITIONED WORLD MODEL EXPLORATION

Anonymous authors
Paper under double-blind review

ABSTRACT

World Action Models (WAMs) couple predictive world modeling with action generation, allowing anticipated future states to guide agent behavior. Although WAMs are rapidly advancing embodied AI, general-purpose counterparts remain largely unexplored in games. Existing game-oriented approaches often combine action-conditioned world models with external policies and reward functions to realize WAM-like decision-making, yet they operate mainly in 2D visual observation space and do not instantiate persistent 3D geometry. Extending this paradigm to 3D games introduces a distinct challenge. In autonomous driving and robotics, the physical environment exists independently of the model, providing a persistent 3D world in which selected actions can be executed. Games have no such external substrate; the virtual world itself must be instantiated. Most playable games require a persistent and navigable space, while 3D games additionally require explicit geometry that supports movement and interaction. Action-conditioned video rollouts provide visual observations but not this spatial representation. We present VALERANT, a training-free framework that transforms a pretrained action-conditioned world model into a WAM for exploring and constructing 3D game maps. By coupling predictive visual rollouts with SLAM-based spatial reconstruction and exploration-driven action selection, VALERANT progressively transforms a single image into a persistent 3D game map. This framework extends WAM-based interaction beyond 2D visual simulation and offers a new approach to reducing manual effort in 3D game-map creation.

1 INTRODUCTION

Rapid advances in large-scale generative modeling have established content synthesis as one of the most active areas of modern artificial intelligence, giving rise to the broad paradigm of AI-generated content (AIGC) (Peebles & Xie, 2023; Podell et al., 2024; Wu et al., 2025; Wan et al., 2025). Modern generative models can synthesize high-quality images, videos, and multimodal sequences under increasingly flexible forms of conditioning. Among their dominant technical foundations, diffusion models generate data by learning to reverse a progressive corruption process (Ho et al., 2020; Rombach et al., 2022), whereas flow-matching models learn a continuous velocity field that transports samples from a simple prior distribution to the data distribution (Lipman et al., 2022; Liu et al., 2022). Building on increasingly coherent video generation, recent work has begun to formulate video generators as observation-space world models (WMs) (Brooks et al., 2024; Wang et al., 2024; Alhaija et al., 2025; Qin et al., 2024). By predicting temporally coherent visual trajectories, these models capture regularities in object motion, scene evolution, and the observable outcomes of physical interactions, thereby approximating how a world evolves without requiring an explicit representation of its underlying 3D state. When actions are introduced as conditioning signals, the resulting action-conditioned world models predict future observations under a prescribed action or action sequence (Yang et al., 2023; Valevski et al., 2025; Che et al., 2025; Bar et al., 2025; Parker-Holder et al., 2024; Ball et al., 2025). Crucially, predicting the consequence of a given action does not by itself determine which action should be taken. World Action Models (WAMs) take a further step by coupling predictive world modeling with action generation or selection (Wang et al., 2026a; Zhang et al., 2026). This coupling can be realized through a cascaded pipeline that derives

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.

actions from predicted futures or through a unified model that jointly models future states and actions (Du et al., 2023; Zhao et al., 2025). Rather than merely simulating what may happen under an externally specified action, WAMs use anticipated world evolution to inform what an agent should do. Taken together, this progression extends generative modeling from passive content synthesis, through controllable prediction, to predictive decision-making in dynamic environments.

The practical value of this shift is particularly evident in autonomous driving and robotics, where WAMs use predicted world evolution to guide motion planning and policy generation (Xia et al., 2026; Zhao et al., 2025; Kim et al., 2026; Xiong et al., 2026). The motivation is straightforward: collecting closed-loop experience and evaluating policies in the physical world are expensive, time-consuming, and difficult to scale, while failed trials may damage hardware or create unacceptable safety risks (Li et al., 2024; Quevedo et al., 2026). By coupling prospective world prediction with action generation, WAMs allow possible behaviors and their consequences to inform decisions before costly physical deployment. Game creation presents a fundamentally different setting. General-purpose WAMs remain largely unexplored in games, particularly for applications that require the underlying 3D world itself to be instantiated. Existing approaches instead follow two related routes: model-based agents combine learned WMs with external policies or reward functions to optimize behavior, while interactive game WMs generate action-conditioned visual observations from user inputs (Alonso et al., 2024; Hafner et al., 2025; Valevski et al., 2025; Li et al., 2025; Wang et al., 2026b). Although these systems can realize WAM-like interaction or decision-making at the pipeline level, their learned environments remain represented by 2D frames or latent observation sequences rather than persistent, explicit 3D geometry. Comparable modular WM–policy systems are also used in autonomous driving and robotics (Wang et al., 2024; Bar et al., 2025; Gao et al., 2026). Importantly, however, the worlds modeled in these physical domains already exist independently of the learned system. A vehicle or robot operates in a persistent physical 3D environment, which provides the spatial substrate against which model predictions can be grounded and from which new observations can be obtained after execution. In game creation, by contrast, the world being modeled is also the artifact that must be generated. There is no independently existing physical environment that can provide a persistent 3D scene behind the synthesized observations. Most first-person shooters require a persistent, navigable 3D space in which camera motion, collision, visibility, and object interactions can be resolved consistently over time. Observation-space WMs and existing WM–policy systems therefore cannot directly provide the explicit 3D maps required for constructing such games. Bridging WAM-based exploration with explicit 3D map construction is thus necessary to extend these models from interactive visual simulation to practical game creation.

To address this limitation, we introduce VALERANT, an automatic game-map generator that transforms a single input image into a navigable 3D environment. Without additional training, VALERANT converts a pretrained action-conditioned WM into a WAM by coupling its predictive rollouts with exploration-driven action selection. VALERANT instantiates a virtual agent that queries the WM with a finite set of candidate actions at each exploration step, producing counterfactual video rollouts from a common pre-action state. Each rollout is processed by visual SLAM (Teed & Deng, 2021; Matsuki et al., 2024; Murai et al., 2025) to estimate the camera trajectory and reconstruct the observed 3D structure. An exploration policy scores the resulting candidates according to their mapping utility, after which the agent returns to the pre-action state and executes only the highest-scoring action. The selected observations are integrated into a persistent map, and this process is repeated until the exploration budget is exhausted, yielding a consolidated 3D point-cloud game map. Our contributions are threefold:

- We introduce VALERANT, a unified training-free framework that converts a pretrained action-conditioned WM into a WAM and couples its exploration with SLAM-based reconstruction for automatic game-map generation.
- We develop a counterfactual exploration procedure in which alternative WM rollouts guide action selection and are converted into a persistent 3D environment, bridging the gap between visual world simulation and the explicit spatial instantiation required by games.
- We establish a new direction for game-map authoring in which a single visual concept can be expanded into navigable 3D geometry, thereby reducing game developers' reliance on labor-intensive and cumbersome map-construction workflows.

2 Related Work

2.1 World Model and World Action Model

World models learn predictive representations of environment dynamics to support planning and control. Early approaches compressed high-dimensional observations into latent states and learned action-conditioned transitions within these compact spaces (Ha & Schmidhuber, 2018; Hafner et al., 2019b;a). Subsequent methods improved the scalability, robustness, and generality of latent world modeling, enabling agents to learn across increasingly diverse control tasks through imagined trajectories (Hansen et al., 2024; Hafner et al., 2025). These models are primarily optimized to capture task-relevant dynamics for decision-making, rather than to synthesize photorealistic and spatially persistent environments. Advances in video generation have shifted world modeling toward the direct prediction of visual futures. iVideoGPT (Wu et al., 2024) represents observations, actions, and rewards as interleaved tokens for scalable interactive prediction, while AVID (Rigter et al., 2024) adapts pretrained video diffusion models into action-conditioned simulators using lightweight domain-specific adapters. In robotics, IRASim (Zhu et al., 2025) generates fine-grained robot–object interactions conditioned on action trajectories. These systems are action-conditioned WMs: actions are supplied as conditions, and the models predict their observational consequences rather than generating the actions themselves.

Following recent formalizations (Wang et al., 2026a; Zhang et al., 2026), we reserve the term World Action Model (WAM) for systems that couple forward world prediction with action generation or selection. This coupling can be implemented through cascaded architectures that derive actions from predicted futures or joint architectures that model future states and actions within a unified framework (Du et al., 2023; Zhao et al., 2025; Bi et al., 2026; Kim et al., 2026). Games provide a natural testbed for these models because both observations and actions can be collected at scale. WHAM (Kanervisto et al., 2025) jointly models gameplay visuals and controller actions, representing a game-specific joint WAM for gameplay ideation. By contrast, Oasis (Decart et al., 2024), MineWorld (Guo et al., 2025), and Matrix-Game (Wang et al., 2026b) are action-conditioned WMs that generate future gameplay observations from user controls. Recent work has further extended action-conditioned generation to environments containing reactive agents and complex multi-agent behavior (Agarwal et al., 2026). Despite their increasing visual fidelity and controllability, these methods primarily realize their generated worlds as autoregressive streams of visual observations. Geometry-aware WMs have begun to address spatial forgetting by introducing persistent point-cloud memory or latent 3D scene states (Wu et al., 2026; Garcin et al., 2026). Their primary objective, however, is to improve the spatial consistency and long-horizon stability of visual generation rather than to construct a reusable game map. In contrast, VALERANT starts from a pretrained action-conditioned WM and, without additional training, couples its counterfactual rollouts to an exploration policy, thereby forming a cascaded WAM whose selected observations are reconstructed into an explicit 3D point-cloud game map.

2.2 Visual SLAM and Online Mapping

Simultaneous localization and mapping (SLAM) jointly estimates an agent's trajectory and incrementally constructs a spatial representation of its surroundings from sequential observations. Classical visual SLAM systems rely on geometric correspondences, bundle adjustment, and loop closure to maintain globally consistent camera poses and maps, as exemplified by ORB-SLAM3 (Campos et al., 2021). Learning-based approaches have since incorporated learned components into this pipeline: DROID-SLAM (Teed & Deng, 2021) and iSLAM (Fu et al., 2024) perform recurrent optimization over camera poses and dense depth, while recent systems such as MASt3R-SLAM (Murai et al., 2025), SLAM3R (Liu et al., 2025), and VGGT-SLAM (Maggio et al., 2026) exploit learned 3D reconstruction priors to recover dense geometry from monocular image streams. SLAM has also been tightly coupled with embodied exploration, where an agent repeatedly selects an action, moves through its environment, acquires new observations, and updates its map online (Chaplot et al., 2020). Consequently, both robotic SLAM and VALERANT follow an incremental mapping process in which exploration continually reveals new parts of the environment. Their central distinction lies instead in the source of these observations. Conventional embodied systems obtain sensor measurements by moving through an existing real or simulated 3D environment. In VALERANT, no such environment is initially available. A frozen action-conditioned video WM generates the predicted

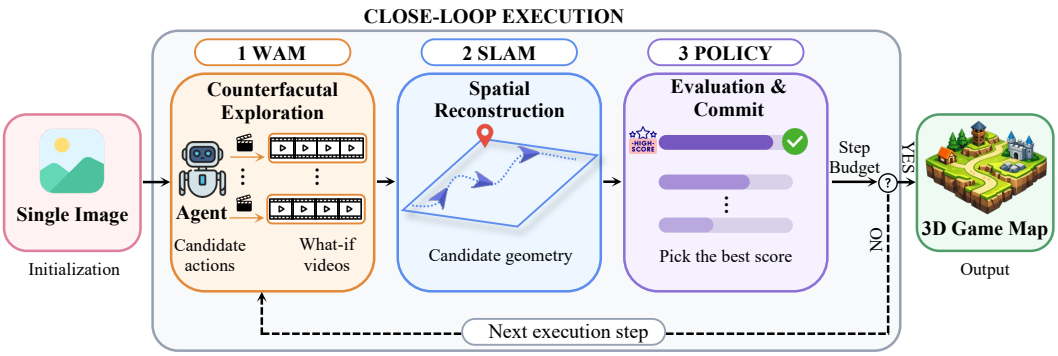


Figure 1: Overview of VALERANT. Starting from a single input image, a virtual agent queries the WAM with candidate actions to generate alternative what-if video rollouts. Visual SLAM reconstructs the corresponding 3D geometry, and an exploration policy selects the branch with the highest mapping utility. The selected observations are committed to a persistent point-cloud map, which is iteratively expanded into the final 3D game map.

visual consequences of externally specified candidate actions, while an exploration policy evaluates the resulting what-if trajectories and selects the action to execute. SLAM then estimates the corresponding camera motion and integrates the selected observations, progressively transforming transient WM rollouts into a persistent and navigable 3D point-cloud map.

2.3 COUNTERFACTUAL REASONING AND WHAT-IF ROLLOUTS

Counterfactual reasoning asks how the future would change under an alternative action while the preceding context remains fixed (Rubin, 1974; Holland, 1986; Pearl et al., 2003). In the physical world, however, only the outcome of the executed action can be observed; the mutually exclusive futures associated with unchosen actions remain inaccessible. Learned world models provide a practical means of exploring these alternatives by simulating candidate actions before execution, an idea widely used in model-based planning (Hansen et al., 2024). Action-conditioned visual WMs further render these alternatives as predicted video trajectories, supporting the comparison of possible maneuvers in embodied navigation and autonomous driving (Yu et al., 2022; Wang et al., 2024; Bar et al., 2025). Such WMs are naturally suited to counterfactual-style reasoning because they can preserve a common observation history while varying the candidate action. This predictive capability becomes part of a WAM when the resulting futures are coupled to action generation or selection. VALERANT realizes this coupling without additional training: it evaluates alternative WM rollouts according to their mapping utility and selects the action that most effectively reveals unobserved 3D space.

3 METHOD

3.1 PRELIMINARIES

World Models. A world model (WM) approximates the transition dynamics of an environment by predicting how its observations evolve under externally specified actions (Wang et al., 2026a). Let $\mathbf{o}_t$ denote the visual observation at environment step t. We define the visual context as $\mathbf{O}_t^{\mathrm{c}} = (\mathbf{o}_{t-K+1}, \ldots, \mathbf{o}_t)$, a candidate action sequence as $\mathbf{A}_t = (\mathbf{a}_t, \ldots, \mathbf{a}_{t+H-1})$, and the corresponding future observations as $\mathbf{O}_t^{\mathrm{f}} = (\mathbf{o}_{t+1}, \ldots, \mathbf{o}_{t+H})$, where K and H denote the context length and prediction horizon, respectively. An observation-space WM parameterized by θ generates a possible future rollout according to

$$\mathbf{O}_t^{\mathrm{f}} \sim p_\theta(\cdot \mid \mathbf{O}_t^{\mathrm{c}}, \mathbf{A}_t), \tag{1}$$

where p_θ denotes the conditional distribution over future visual observations learned by the WM. Modeling a distribution allows the WM to represent multiple plausible futures that may follow the same observation history and action sequence. Importantly, the action sequence is provided as a

condition: the WM predicts its possible visual consequences but does not determine which action should be executed.

Action-Conditioned Video Diffusion Models. An action-conditioned video diffusion WM instantiates the conditional distribution in equation 1 with a latent video generator (Wang et al., 2026b). Let $\mathcal{E}$ and $\mathcal{D}$ denote a video encoder and decoder, respectively. We encode the visual context as $\mathbf{h}_t = \mathcal{E}(\mathbf{O}_t^{\mathrm{c}})$ and the target future video as $\mathbf{z}_0 = \mathcal{E}(\mathbf{O}_t^{\mathrm{f}})$. Following the flow-matching (FM) formulation, a noisy latent at diffusion time $\tau \in [0, 1]$ is constructed as

$$\mathbf{z}_\tau = (1 - \tau)\mathbf{z}_0 + \tau\boldsymbol{\epsilon}, \qquad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \tag{2}$$

and the conditional velocity field v_θ is learned using

$$\mathcal{L}_{\mathrm{FM}} = \mathbb{E}\Big[\|v_\theta(\mathbf{z}_\tau, \tau \mid \mathbf{h}_t, \mathbf{A}_t) - (\boldsymbol{\epsilon} - \mathbf{z}_0)\|_2^2\Big]. \tag{3}$$

At inference time, the learned flow is integrated from Gaussian noise toward the video latent and decoded by $\mathcal{D}$ to obtain $\mathbf{O}_t^{\mathrm{f}}$. Consequently, different candidate action sequences conditioned on the same visual context produce alternative future video rollouts. In VALERANT, the video WM remains frozen and is used only for action-conditioned rollout generation.

Visual Simultaneous Localization and Mapping. Visual simultaneous localization and mapping (SLAM) jointly estimates camera motion and reconstructs 3D scene structure from visual observations (Teed & Deng, 2021; Murai et al., 2025). Given an observation sequence $\mathbf{O}_{1:L} = (\mathbf{o}_1, \ldots, \mathbf{o}_L)$, visual SLAM estimates the camera trajectory $\mathbf{T}_{1:L}$, where $\mathbf{T}_i \in \mathrm{SE}(3)$ is the camera pose at frame i, together with a spatial map $\mathcal{M}$. This process can be expressed abstractly as

$$\left(\hat{\mathbf{T}}_{1:L}, \hat{\mathcal{M}}\right) = \underset{\mathbf{T}_{1:L},\, \mathcal{M}}{\arg\min}\, \mathcal{J}_{\mathrm{SLAM}}(\mathbf{O}_{1:L}; \mathbf{T}_{1:L}, \mathcal{M}), \tag{4}$$

where $\mathcal{J}_{\mathrm{SLAM}}$ measures the visual and geometric consistency between the observations, camera poses, and reconstructed scene. In VALERANT, each generated rollout $\hat{\mathbf{O}}_{t,m}^{\mathrm{f}}$ is processed by visual SLAM to recover its camera trajectory and register its observed geometry in a common 3D coordinate system.

3.2 OVERVIEW

Given a single input image $\mathbf{o}_0$, VALERANT progressively explores the synthesized environment and constructs a navigable 3D game map. At exploration step t, the system maintains a visual context $\mathbf{O}_t^{\mathrm{c}}$, a current camera pose $\mathbf{T}_t$, and a global map $\mathcal{M}_t$. The pretrained video WM remains frozen throughout this process, while visual SLAM and the exploration policy are applied entirely at inference time. By coupling the WM with geometric reconstruction and action selection, VALERANT turns its passive visual predictions into a closed-loop exploration process without additional training.

At each step, the system checkpoints the current WM and SLAM states and evaluates five candidate action sequences:

$$\mathcal{A}_t = \left\{\mathbf{A}_t^{(1)}, \ldots, \mathbf{A}_t^{(5)}\right\} = \{\texttt{uw}, \texttt{jw}, \texttt{lw}, \texttt{jq}, \texttt{lq}\}. \tag{5}$$

Here, `uw` denotes moving straight forward without turning, while `jw` and `lw` denote turning left and right, respectively, followed by forward movement. The remaining actions, `jq` and `lq`, denote in-place rotations to the left and right. The resulting action space therefore contains three forward-moving candidates and two rotation-only candidates.

Starting from the same pre-action context $\mathbf{O}_t^{\mathrm{c}}$, the WM generates a separate future rollout $\mathbf{O}_{t,m}^{\mathrm{f}}$ for each candidate $\mathbf{A}_t^{(m)}$ following equation 1. Because all candidates branch from an identical visual context, their generated rollouts provide directly comparable what-if outcomes of the available actions. Rather than estimating their consequences using an additional learned predictor, VALERANT explicitly generates and reconstructs every candidate future.

Visual SLAM processes each generated rollout $\mathbf{O}_{t,m}^{\mathrm{f}}$ according to equation 4, producing an estimated camera trajectory $\hat{\mathbf{T}}_{t,m}$ and candidate scene geometry $\hat{\mathcal{M}}_{t,m}$. Each candidate is therefore evaluated using its reconstructed motion and observed 3D structure rather than its visual appearance

alone. We define its mapping utility using exploration coverage $\mathcal{C}_{t,m}$ and forward progress $\mathcal{P}_{t,m}$, while $\mathcal{B}_{t,m} \in \{0,1\}$ indicates whether the candidate violates the collision-free constraint. The selected branch is

$$m^\star = \underset{m\in\{1,\dots,5\}:\mathcal{B}_{t,m}=0}{\arg\max} \left(\lambda_{\text{cov}}\mathcal{C}_{t,m} + \lambda_{\text{prog}}\mathcal{P}_{t,m}\right), \tag{6}$$

where λ_{cov} and λ_{prog} balance the two exploration objectives.

After selection, VALERANT restores the pre-action state and commits only $\mathbf{A}_t^{(m^\star)}$. The selected action is replayed from the restored state, while all unsuccessful branches are discarded without modifying the persistent world state. The selected observations and reconstructed geometry are then incorporated into the next visual context and global map:

$$\begin{aligned} \mathbf{O}^{\text{c}}_{t+1} &= \text{Update}\big(\mathbf{O}^{\text{c}}_t, \mathbf{O}^{\text{f}}_{t,m^\star}\big), \\ \mathcal{M}_{t+1} &= \text{Fuse}\Big(\mathcal{M}_t, \hat{\mathcal{M}}_{t,m^\star}\Big). \end{aligned} \tag{7}$$

This branch–reconstruct–evaluate–commit loop is repeated until the exploration budget is exhausted, progressively consolidating the selected observations into a 3D point-cloud game map.

3.3 Robust Exploration under Geometric Drift and Visual Hallucinations

Generated rollouts may contain visual hallucinations, while monocular SLAM can accumulate pose and scale drift, leading to unreliable geometry and unsafe exploration decisions. We define robust exploration as the ability to maintain geometrically consistent mapping and collision-aware progress under these errors, while recovering from locally trapped states.

3.3.1 Robust Floor Estimation

Reliable collision reasoning requires a stable estimate of the floor. In particular, reconstructed points must be distinguished as traversable ground, body-level obstacles, or overhead structures according to their height above the local supporting surface. A floor reference calibrated only once is unreliable under monocular SLAM, whose estimated scale and vertical position may drift over time. As a result, an otherwise valid ground surface can gradually shift into the obstacle height range, producing false collision detections even in open space.

To address this issue, we introduce a robust floor estimator that replaces the fixed global reference with a locally adaptive one. For each SLAM keyframe, we identify candidate support points near the lower envelope of its local reconstruction and fit a dominant plane. The estimate is accepted only when the plane has sufficient geometric support and its normal is consistent with the current vertical direction. When the floor cannot be reliably observed, such as when the camera faces a nearby wall, we propagate the most recent reliable camera-to-floor relation instead of reverting to the initial global calibration. This allows the floor estimate to follow gradual SLAM drift while preventing non-ground surfaces from being adopted as the new reference.

Let $\mathbf{q}_i$ be a point on the estimated floor plane of keyframe i, $\mathbf{n}_i$ its unit normal oriented toward the camera center $\mathbf{c}_i$, and $\mathbf{p}$ a reconstructed 3D point. We define its normalized floor-relative height as

$$\bar{h}_i(\mathbf{p}) = \frac{\mathbf{n}_i^\top(\mathbf{p}-\mathbf{q}_i)}{\mathbf{n}_i^\top(\mathbf{c}_i-\mathbf{q}_i)}. \tag{8}$$

Under this normalization, the local floor has height zero and the camera center has height one. A common translation or scale change in the local reconstruction therefore affects the floor, camera, and scene points together and is canceled by the relative representation. All subsequent collision tests operate on $\bar{h}_i$ rather than absolute global height. Consequently, the ground and body-level occupancy bands remain aligned under reconstruction drift, substantially reducing false obstacles caused by inconsistent floor elevations.

3.3.2 Complementary Collision Detection

Point-Cloud Collision. Action-conditioned video WMs do not explicitly enforce collision constraints and may generate trajectories that pass through scene geometry. We detect such failures

directly from the branch-local geometry reconstructed for each candidate rollout, without relying on a learned occupancy predictor. We treat the candidate map $\hat{\mathcal{M}}_{t,m}$ as a set of reconstructed 3D points and use $\mathbf{p}$ to denote an individual point. Each point is assigned a normalized floor-relative height $\bar{h}(\mathbf{p})$ using the local floor estimate of its source keyframe from equation 8. Let $\mathcal{S}_{t,m}$ denote the set of trajectory samples used for geometric testing, with $\hat{\mathbf{T}}_{t,m,i}$ representing the pose of sample $i \in \mathcal{S}_{t,m}$ and $\kappa_{t,m,i}$ its associated camera-to-floor distance. We detect a point-cloud collision by testing whether the reconstructed geometry intersects a normalized cylindrical body volume along the sampled trajectory:

$$\mathcal{B}^{\mathrm{pc}}_{t,m} = \mathbb{I}\left[\max_{i\in\mathcal{S}_{t,m}} \sum_{\mathbf{p}\in\hat{\mathcal{M}}_{t,m}} \mathbb{I}\left(\frac{\mathrm{dist}_{\perp}(\mathbf{p}, \hat{\mathbf{T}}_{t,m,i})}{\kappa_{t,m,i}} \le \rho \ \wedge\ \eta_{\mathrm{low}} \le \bar{h}(\mathbf{p}) \le \eta_{\mathrm{high}}\right) \ge N_{\mathrm{occ}}\right], \tag{9}$$

where $\mathrm{dist}_{\perp}$ measures the horizontal distance from a point to the agent's vertical body axis, ρ is the normalized body radius, and $[\eta_{\mathrm{low}}, \eta_{\mathrm{high}}]$ specifies the body-level height range. The threshold N_{occ} suppresses isolated reconstruction noise, while the height restriction excludes traversable ground and overhead structures. Rotation-only candidates are exempt because they introduce no additional translational penetration.

Depth-Based Collision. Point-cloud collision detection may become unreliable for textureless or reflective walls, whose correspondences can receive low confidence and be removed during SLAM reconstruction. We therefore introduce a complementary detector that operates directly on the unfiltered depth estimates of all rollout frames, including those not selected as SLAM keyframes. Let $\boldsymbol{\delta}_{t,m,\ell}$ denote the raw depth map at rollout frame ℓ, Ω_{ctr} rectangular region centered in the image, and $\kappa_{t,m,\ell}$ the camera-to-floor distance associated with that frame. A penetration event is identified when a frontal surface appears within a normalized near-field distance but the estimated trajectory nevertheless continues to advance:

$$\mathcal{B}^{\mathrm{dep}}_{t,m} = \mathbb{I}\left[\exists\, \ell : \frac{\mathrm{Avg}_{\mathbf{u}\in\Omega_{\mathrm{ctr}}}\, \boldsymbol{\delta}_{t,m,\ell}(\mathbf{u})}{\kappa_{t,m,\ell}} \le d_{\mathrm{near}} \ \wedge\ \frac{\mathrm{Fwd}\left(\hat{\mathbf{T}}_{t,m}; \ell, L\right)}{\kappa_{t,m,\ell}} \ge s_{\min}(\ell)\right], \tag{10}$$

where $\mathrm{Fwd}(\hat{\mathbf{T}}_{t,m}; \ell, L)$ measures the forward displacement along the estimated trajectory over the video interval from frame ℓ to frame L, and $s_{\min}(\ell)$ accounts for the remaining rollout horizon. The central average provides a robust signature of a nearby frontal surface, while the subsequent motion distinguishes penetration from merely observing a close wall. The collision indicator $\mathcal{B}_{t,m}$ in equation 6 is activated whenever either $\mathcal{B}^{\mathrm{pc}}_{t,m}$ or $\mathcal{B}^{\mathrm{dep}}_{t,m}$ is positive.

3.3.3 Dead-Pocket Rewind

Receding-horizon exploration can lead the agent into a dead pocket in which every translational candidate either violates the collision-free constraint or provides negligible forward progress. Continuing local action selection in this state produces repeated rotations or stagnation rather than revealing new space. We address this failure through a checkpoint-based rewind mechanism that augments local exploration with backtracking. Along the committed trajectory, VALERANT maintains synchronized checkpoints of the WM and SLAM states. When insufficient progress persists, the system selects an earlier checkpoint $r < t$ preceding the stalled segment and restores the visual context, camera pose, and reconstructed map to that state. To prevent the agent from immediately re-entering the same pocket, the spatial cells traversed along the discarded segment are retained in a failed-region memory. Let $\boldsymbol{\Phi}_t$ denote the joint WM–SLAM checkpoint at step t, let $\mathcal{F}_t$ denote the accumulated set of rejected cells, and let $\mathrm{cell}(\mathbf{T}_k)$ map a camera pose to its spatial cell. The rewind operation is summarized as

$$\boldsymbol{\Phi}_{t^+} \leftarrow \boldsymbol{\Phi}_r, \qquad \mathcal{F}_{t^+} \leftarrow \mathcal{F}_t \cup \{\mathrm{cell}(\mathbf{T}_k) \,|\, r < k \le t\}, \tag{11}$$

where t^+ denotes the state immediately after rewinding. Subsequent candidate trajectories that enter $\mathcal{F}_{t^+}$ are treated as inadmissible, encouraging exploration from the restored anchor toward an alternative route. Because the WM and SLAM states are restored jointly, the generated visual context and reconstructed geometry remain synchronized after backtracking. If the restored state remains trapped, the procedure can be repeated with an earlier checkpoint, turning the otherwise greedy exploration process into a spatial search with recoverable decisions.



Figure 2: Qualitative Results of VALERANT for Single-Image 3D Game-Map Generation. Each pair of rows presents one scene. The leftmost column shows the input first frame, the middle columns show six generated frames sampled along the agent's trajectory, and the rightmost column presents a top-down view of the final reconstructed map with the agent's trajectory overlaid.

4 EXPERIMENTS

4.1 EXPERIMENT SETTINGS

Dataset & Baseline All input images used in our experiments are captured directly as screenshots from *VALORANT*. Each screenshot serves as the initial frame and single-view condition. We evaluate all methods using the same set of initial frames. We compare VALERANT with HY-World 2.0 (HY-World et al., 2026), which likewise generates a navigable 3D world from a single image but adopts a substantially different technical approach.

Metric Our task is inherently open-ended and generative: a single input image can correspond to many plausible 3D environments, and no unique ground-truth map is available for reference-based evaluation. We therefore adopt two perceptual metrics: *Authenticity Score* and *User Preference*. The former uses a VLM as an evaluator to assess the perceptual authenticity of the generated results, while the latter reports the proportion of pairwise comparisons in which human evaluators prefer the output of each method. Detailed results are provided in Appendix C. Unlike training-based benchmarks evaluated on held-out data with paired references, our training-free method operates on arbitrary input images without ground-truth 3D scenes. Since different methods may generate distinct yet plausible content, correspondence-based metrics are not meaningful; we therefore rely on VLM evaluation and user preference.

Implementation Details We use Matrix-Game 3.0 (Wang et al., 2026b) as the action-conditioned video WM and MASt3R-SLAM (Murai et al., 2025) for visual localization and 3D reconstruction. We optionally apply video2world (Höllein & Nießner, 2026) as a post-processing stage to globally refine the reconstructed point cloud. All experiments are conducted on a single NVIDIA H100 GPU. Detailed hyperparameter settings are provided in Appendix A.

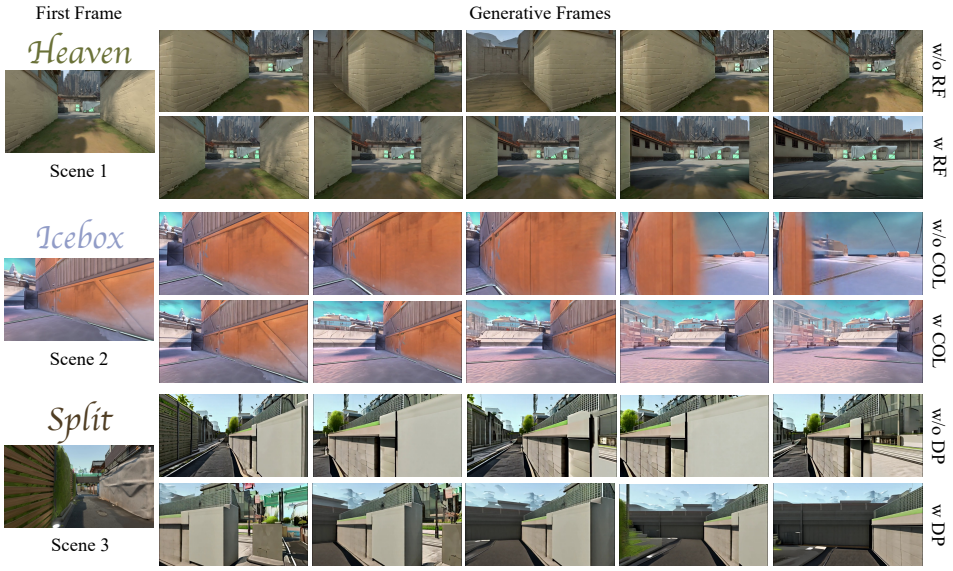


Figure 3: Qualitative ablation of the robust exploration modules. Each pair of rows presents one scene and isolates the effect of a single module by comparing generated frames without (top) and with (bottom) that module. RF, COL, and DP denote robust floor estimation, complementary collision detection, and dead-pocket rewind, respectively.

4.2 Experiment Results

Main Results Figure 2 presents the qualitative results of VALERANT across four scenes. Starting from only the first frame, the entire exploration and reconstruction process proceeds automatically, without manually specified trajectories or intermediate intervention. The generated frames sampled along each trajectory show that the agent moves smoothly through corridors, turns, and open regions while avoiding collisions. Meanwhile, visual SLAM continuously integrates the selected observations into a persistent 3D point-cloud map, with the final top-down views demonstrating substantial spatial coverage along the executed trajectories. These results show that VALERANT can jointly achieve autonomous collision-free exploration and incremental 3D game-map construction from a single image.

Ablation Study Figure 3 qualitatively isolates the contributions of the three robustness components. In Heaven, removing RF destabilizes the estimated ground reference under geometric drift, causing the agent to remain near the surrounding walls and make limited exploration progress; RF restores reliable floor-relative reasoning and enables smooth forward traversal into the open area. In Icebox, the absence of COL allows generated trajectories to penetrate the orange wall, whereas COL rejects geometrically infeasible candidates and keeps exploration within free space. In Split, without DP, the agent becomes trapped in a dead pocket and repeatedly alternates between left and right turns without making spatial progress; DP restores an earlier exploration state and redirects the agent toward an alternative route. These comparisons demonstrate that RF, COL, and DP address complementary failure modes arising from floor-estimation drift, wall penetration, and local exploration traps, respectively.

5 Conclusion

We presented VALERANT, a training-free framework that constructs navigable 3D game maps from a single image. By coupling a frozen action-conditioned video world model with visual SLAM and an exploration policy, VALERANT evaluates alternative action rollouts and incrementally integrates selected observations into a persistent point-cloud map. Experiments show that robust floor estimation, collision detection, and dead-pocket rewind support reliable autonomous exploration under geometric drift and visual hallucinations. Future work will improve geometric consistency and convert the reconstructed point clouds into richer, editable game assets.

AI USE STATEMENT

In this work, we used generative AI tools to improve the clarity, grammar, and presentation of the manuscript, as well as to perform the VLM-based evaluation of generated results under predefined criteria. We did not use generative AI to formulate the research problem, develop the proposed method, implement the system, or generate the experimental results attributed to our method. All AI-assisted revisions were reviewed and edited by the authors, and the evaluation protocol and outputs were manually inspected for consistency and correctness. We take full responsibility for the final content of this work, including all text, claims, and artifacts produced with the aid of generative AI.

ETHICS STATEMENT

Human participants were involved only in the user preference study, in which they compared rendered results produced by different methods. Participation was voluntary, and all participants received appropriate compensation for their time. The study involved only perceptual judgments of generated game scenes and posed no foreseeable risk to participants. No personally identifying information is reported, and the collected responses are presented only in aggregate form.

REPRODUCIBILITY STATEMENT

We provide a detailed description of the VALERANT pipeline and its principal components in the main paper, while implementation details, hyperparameter settings, evaluation protocols, and additional qualitative results are included in the appendices and supplementary materials. These materials are intended to provide the information necessary to reproduce our experiments and reported results. Upon acceptance, we will publicly release the source code, configurations, and evaluation scripts.

REFERENCES

Anmol Agarwal, Pranay Meshram, Sumer Singh, Saurav Suman, Andrew Lapp, Shahbuland Matiana, Louis Castricato, and Spencer Frazier. Combat: Conditional world models for behavioral agent training. *arXiv preprint arXiv:2603.00825*, 2026.

Hassan Abu Alhaija, Jose Alvarez, Maciej Bala, Tiffany Cai, Tianshi Cao, Liz Cha, Joshua Chen, Mike Chen, Francesco Ferroni, Sanja Fidler, et al. Cosmos-transfer1: Conditional world generation with adaptive multimodal control. *arXiv preprint arXiv:2503.14492*, 2025.

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

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.

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

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

Tim Brooks, Bill Peebles, Connor Holmes, Will DePue, Yufei Guo, Leo Jing, David Schnurr, Joe Taylor, Troy Luhman, Eric Luhman, et al. Video generation models as world simulators. *OpenAI Blog*, 1(8):1, 2024.

Carlos Campos, Richard Elvira, Juan J Gómez Rodríguez, José MM Montiel, and Juan D Tardós. Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam. *IEEE transactions on robotics*, 37(6):1874–1890, 2021.

Devendra Singh Chaplot, Dhiraj Gandhi, Saurabh Gupta, Abhinav Gupta, and Ruslan Salakhutdinov. Learning to explore using active neural slam. *arXiv preprint arXiv:2004.05155*, 2020.

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

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

Yilun Du, Sherry Yang, Bo Dai, Hanjun Dai, Ofir Nachum, Josh Tenenbaum, Dale Schuurmans, and Pieter Abbeel. Learning universal policies via text-guided video generation. *Advances in neural information processing systems*, 36:9156–9172, 2023.

Taimeng Fu, Shaoshu Su, Yiren Lu, and Chen Wang. islam: Imperative slam. *IEEE Robotics and Automation Letters*, 9(5):4607–4614, 2024.

Shenyuan Gao, William Liang, Kaiyuan Zheng, Ayaan Malik, Seonghyeon Ye, Sihyun Yu, Wei-Cheng Tseng, Yuzhu Dong, Kaichun Mo, Chen-Hsuan Lin, et al. Dreamdojo: A generalist robot world model from large-scale human videos. *arXiv preprint arXiv:2602.06949*, 2026.

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

Junliang Guo, Yang Ye, Tianyu He, Haoyu Wu, Yushu Jiang, Tim Pearce, and Jiang Bian. Mineworld: a real-time and open-source interactive world model on minecraft. *arXiv preprint arXiv:2504.08388*, 2025.

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

Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. *arXiv preprint arXiv:1912.01603*, 2019a.

Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels. In *International conference on machine learning*, pp. 2555–2565. PMLR, 2019b.

Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse control tasks through world models. *Nature*, 640(8059):647–653, 2025.

Nick Hansen, Hao Su, and Xiaolong Wang. Td-mpc2: Scalable, robust world models for continuous control. In *International Conference on Learning Representations*, volume 2024, pp. 47376–47405, 2024.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Paul W Holland. Statistics and causal inference. *Journal of the American statistical Association*, 81 (396):945–960, 1986.

Lukas Höllein and Matthias Nießner. World reconstruction from inconsistent views. *arXiv preprint arXiv:2603.16736*, 2026.

Team HY-World, Chenjie Cao, Xuhui Zuo, Zhenwei Wang, Yisu Zhang, Junta Wu, Zhenyang Liu, Yuning Gong, Yang Liu, Bo Yuan, et al. Hy-world 2.0: A multi-modal world model for reconstructing, generating, and simulating 3d worlds. *arXiv preprint arXiv:2604.14268*, 2026.

Anssi Kanervisto, Dave Bignell, Linda Yilin Wen, Martin Grayson, Raluca Georgescu, Sergio Valcarcel Macua, Shan Zheng Tan, Tabish Rashid, Tim Pearce, Yuhan Cao, et al. World and human action models towards gameplay ideation. *Nature*, 638(8051):656–663, 2025.

Moo Jin Kim, Yihuai Gao, Tsung-Yi Lin, Yen-Chen Lin, Yunhao Ge, Grace Lam, Percy Liang, Shuran Song, Ming-Yu Liu, Chelsea Finn, et al. Cosmos policy: Fine-tuning video models for visuomotor control and planning. *arXiv preprint arXiv:2601.16163*, 2026.

Jiaqi Li, Junshu Tang, Zhiyong 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 preprint arXiv:2506.17201*, 2(3):6, 2025.

Xuanlin Li, Kyle Hsu, Jiayuan Gu, Karl Pertsch, Oier Mees, Homer Rich Walke, Chuyuan Fu, Ishikaa Lunawat, Isabel Sieh, Sean Kirmani, et al. Evaluating real-world robot manipulation policies in simulation. *arXiv preprint arXiv:2405.05941*, 2024.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022.

Yuzheng Liu, Siyan Dong, Shuzhe Wang, Yingda Yin, Yanchao Yang, Qingnan Fan, and Baoquan Chen. Slam3r: Real-time dense scene reconstruction from monocular rgb videos. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16651–16662. IEEE, 2025.

Dominic Maggio, Hyungtae Lim, and Luca Carlone. Vggt-slam: Dense rgb slam optimized on the sl (4) manifold. *Advances in Neural Information Processing Systems*, 38:129839–129867, 2026.

Hidenobu Matsuki, Riku Murai, Paul HJ Kelly, and Andrew J Davison. Gaussian splatting slam. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18039–18048. IEEE, 2024.

Riku Murai, Eric Dexheimer, and Andrew J Davison. Mast3r-slam: Real-time dense slam with 3d reconstruction priors. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16695–16705. IEEE, 2025.

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/`.

Judea Pearl et al. Causality: models, reasoning, and inference. *Econometric Theory*, 19(675-685): 46, 2003.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4172–4182. IEEE, 2023.

Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. In *International Conference on Learning Representations*, volume 2024, pp. 1862–1874, 2024.

Yiran Qin, Zhelun Shi, Jiwen Yu, Xijun Wang, Enshen Zhou, Lijun Li, Zhenfei Yin, Xihui Liu, Lu Sheng, Jing Shao, et al. Worldsimbench: Towards video generation models as world simulators. *arXiv preprint arXiv:2410.18072*, 2024.

Julian Quevedo, Ansh Kumar Sharma, Yixiang Sun, Varad Suryavanshi, Percy Liang, and Sherry Yang. Worldgym: World model as an environment for policy evaluation. In *International Conference on Learning Representations*, volume 2026, pp. 78932–78957, 2026.

Marc Rigter, Tarun Gupta, Agrin Hilmkil, and Chao Ma. Avid: Adapting video diffusion models to world models. *arXiv preprint arXiv:2410.12822*, 2024.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *2022 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 10674–10685. ieee, 2022.

Donald B Rubin. Estimating causal effects of treatments in randomized and nonrandomized studies. *Journal of educational Psychology*, 66(5):688, 1974.

Zachary Teed and Jia Deng. Droid-slam: Deep visual slam for monocular, stereo, and rgb-d cameras. *Advances in neural information processing systems*, 34:16558–16569, 2021.

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

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

Siyin Wang, Junhao Shi, Zhaoyang Fu, Xinzhe He, Feihong Liu, Chenchen Yang, Yikang Zhou, Zhaoye Fei, Jingjing Gong, Jinlan Fu, et al. World action models: The next frontier in embodied ai. *arXiv preprint arXiv:2605.12090*, 2026a.

Yuqi Wang, Jiawei He, Lue Fan, Hongxin Li, Yuntao Chen, and Zhaoxiang Zhang. Driving into the future: Multiview visual forecasting and planning with world model for autonomous driving. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14749–14759. IEEE, 2024.

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

Jialong Wu, Shaofeng Yin, Ningya Feng, Xu He, Dong Li, Jianye Hao, and Mingsheng Long. ivideogpt: Interactive videogpts are scalable world models. *Advances in Neural Information Processing Systems*, 37:68082–68119, 2024.

Tong Wu, Shuai Yang, Ryan Po, Yinghao Xu, Ziwei Liu, Dahua Lin, and Gordon Wetzstein. Video world models with long-term spatial memory. *Advances in Neural Information Processing Systems*, 38:49371–49393, 2026.

Yecheng Wu, Zhuoyang Zhang, Junyu Chen, Haotian Tang, Dacheng Li, Yunhao Fang, Ligeng Zhu, Enze Xie, Hongxu Yin, Li Yi, et al. Vila-u: a unified foundation model integrating visual understanding and generation. In *International Conference on Learning Representations*, volume 2025, pp. 93620–93638, 2025.

Tianze Xia, Yongkang Li, Lijun Zhou, Jingfeng Yao, Kaixin Xiong, Haiyang Sun, Bing Wang, Kun Ma, Guang Chen, Hangjun Ye, et al. Drivelaw: Unifying planning and video generation in a latent driving world. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 39701–39712, 2026.

Zhexiao Xiong, Xin Ye, Burhan Yaman, Sheng Cheng, Yiren Lu, Jingru Luo, Nathan Jacobs, and Liu Ren. Unidrive-wm: Unified understanding, planning and generation world model for autonomous driving. *arXiv preprint arXiv:2601.04453*, 2026.

Sherry Yang, Yilun Du, Kamyar Ghasemipour, Jonathan Tompson, Leslie Kaelbling, Dale Schuurmans, and Pieter Abbeel. Learning interactive real-world simulators. *arXiv preprint arXiv:2310.06114*, 2023.

Wei Yu, Wenxin Chen, Songheng Yin, Steve Easterbrook, and Animesh Garg. Modular action concept grounding in semantic video prediction. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3595–3604. IEEE, 2022.

Xiaoxiong Zhang, Xiong Zeng, and Wei Zhang. From world models to world action models: A concise tutorial for robotics. *arXiv preprint arXiv:2607.00836*, 2026.

Qingqing Zhao, Yao Lu, Moo Jin Kim, Zipeng Fu, Zhuoyang Zhang, Yecheng Wu, Zhaoshuo Li, Qianli Ma, Song Han, Chelsea Finn, et al. Cot-vla: Visual chain-of-thought reasoning for vision-language-action models. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1702–1713. IEEE, 2025.

Fangqi Zhu, Hongtao Wu, Song Guo, Yuxiao Liu, Chilam Cheang, and Tao Kong. Irasim: A fine-grained world model for robot manipulation. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9834–9844. IEEE, 2025.

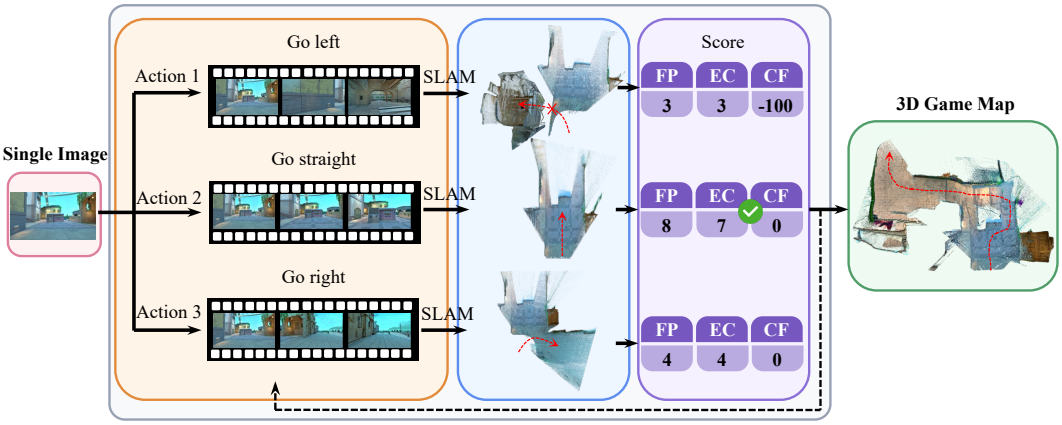


Figure 4: A detailed version of the pipeline, expanding upon Figure 1.

A Implementation Details

World Model and Reconstruction. We instantiate p_θ with the frozen Matrix-Game 3.0 action-conditioned video WM. The visual context length is set to $K = 16$ frames. For the single-stage candidates `uw`, `jq`, and `lq`, the prediction horizon is $H = 57$ frames. The composite candidates `jw` and `lw` append a 40-frame forward-motion segment after the initial turn, resulting in $H = 97$ frames. Flow-matching inference uses three integration steps with a classifier-free guidance scale of 5.0. At every exploration step, all five candidates in $\mathcal{A}_t$ are generated from the same committed visual context. We use MASt3R-SLAM to estimate $\hat{\mathbf{T}}_{t,m}$ and reconstruct $\hat{\mathcal{M}}_{t,m}$ for each rollout. When specified, video2world is applied only as a post-processing stage to refine the final point cloud and does not participate in action selection.

Exploration Policy. The coverage utility is instantiated as $\mathcal{C}_{t,m} = \mathcal{C}^{\text{new}}_{t,m} + 0.5\mathcal{C}^{\text{front}}_{t,m}$, where the two terms measure newly visited cells and proximity to the current exploration frontier, respectively. In equation 6, we set $\lambda_{\text{cov}} = 1.0$ and $\lambda_{\text{prog}} = 5.0$. Both coverage estimation and the failed-region memory $\mathcal{F}_t$ operate on a 128×128 spatial grid whose cell width is one eighth of the nominal displacement of a forward action. The exploration budget is limited to 15 committed steps.

Robust Floor Estimation. For each SLAM keyframe, a candidate floor plane is accepted only when it is supported by at least 300 reconstructed points and satisfies $|\mathbf{n}_i^\top \mathbf{e}_y| \geq 0.7$, where $\mathbf{e}_y$ denotes the vertical axis. If either condition is not met, the most recent reliable camera-to-floor relation is propagated. All subsequent geometric tests use the normalized height $\bar{h}_i(\mathbf{p})$ from equation 8.

Collision Detection. For point-cloud collision detection, the normalized body radius in equation 9 is set to $\rho = 0.21$, and the body-level height interval is $[\eta_{\text{low}}, \eta_{\text{high}}] = [0.18, 1.19]$. We evaluate every pose in $\mathcal{S}_{t,m}$ and retain reconstructed points with a confidence score of at least 4.0. The occupancy threshold is $N_{\text{occ}} = 30$ and must be exceeded at two consecutive trajectory samples; a single sample containing at least 300 occupied points is treated as an immediate deep-penetration event. Any detected collision activates the hard constraint $\mathcal{B}_{t,m} = 1$.

For depth-based collision detection, Ω_{ctr} covers the central one-third of the image along both height and width, and every rollout frame is evaluated. The near-field cutoff is 0.45 in reconstructed depth units, corresponding to $d_{\text{near}} = 0.455$ in the normalized form of equation 10. We set $s_{\min}(\ell) = 0.085\, d_{\text{fwd}}(L - \ell)/L$, where d_{fwd} is the normalized displacement of a nominal forward action. This decay accounts for the amount of rollout remaining after frame ℓ.

Dead-Pocket Rewind. A dead pocket is declared when the policy selects rotation-only actions for three consecutive steps without translational progress. VALERANT retains the three most recent synchronized WM–SLAM checkpoints $\mathbf{\Phi}_t$ and permits at most two rewind operations during one

exploration episode. After rewinding, cells belonging to the discarded trajectory are added to $\mathcal{F}_t$ according to equation 11, preventing the policy from immediately entering the same failed region.

All experiments are conducted at inference time on a single NVIDIA H100 GPU, without additional training or fine-tuning of the WM or SLAM modules.

B METRIC DETAILS

VLM Evaluation Prompt.

> *You are a professional 3D scene evaluator with strong visual judgment. Your task is to compare the quality of 3D Gaussian Splatting (3DGS) scenes generated by two different methods.*
>
> *For each evaluation case, I will provide:*
>
> 1. *A first frame shared by both methods, which serves as the common generation input and defines the initial appearance and visible content of the scene.*
> 2. *Five rendered images of a 3DGS scene generated by Method A.*
> 3. *Five rendered images of a 3DGS scene generated by Method B.*
>
> *Within each method, the five rendered images are presented in their camera-path order and depict different viewpoints of the same generated scene. The method labels are anonymized and arbitrary.*
>
> *You must compare the two generated scenes and make a single-choice decision: select the better result between Method A and Method B. Please base your judgment on the following criteria:*
>
> 1. ***Overall visual quality:*** *Evaluate the immediate visual impression of each result at first glance. Consider whether the scene appears visually appealing, coherent, complete, and free from conspicuous defects. Judge each result as a whole rather than focusing on a single isolated image or detail.*
> 2. ***3D scene consistency:*** *Assess whether the geometry, spatial layout, surfaces, and scene structures remain consistent across the five rendered viewpoints. Scene elements visible in the shared first frame should be preserved plausibly when observed from subsequent viewpoints. Look for distorted geometry, duplicated or missing structures, floating elements, broken surfaces, and viewpoint-dependent artifacts.*
> 3. ***Cross-view and trajectory coherence:*** *Determine whether the shared first frame and the five subsequent renders appear to form a natural sequence captured from one persistent scene along a plausible camera path. Shared structures should remain recognizable, while perspective, visibility, and scene content should change smoothly as the camera moves. Penalize abrupt changes that make the renders appear independently generated or suggest that they belong to different scenes.*
> 4. ***Game-scene realism and rendering quality:*** *Evaluate whether the result resembles a high-quality scene rendered from a 3D game environment. Consider the quality of textures, materials, lighting, geometry, details, and overall rendering, as well as the absence of blur, holes, floaters, or other visible artifacts. Photographic realism is not required; the scene should instead look visually credible and internally coherent as a game environment.*
>
> *After considering all four criteria, choose the single better result between the two methods. Use the shared first frame only as a common reference and evaluate the five rendered images of each method jointly. Do not select a method based on only one view. Your decision should be based solely on the provided visual evidence, without making assumptions about the identities or technical implementations of the methods. Ties are not allowed.*
>
> *Please output only the final choice in the following format:*
>
> `Best Method: [A/B]`



Figure 5: **Qualitative comparison.** Starting from the same input frame, both methods construct a 3DGS scene, from which three views sampled along a camera trajectory are rendered. Compared with HY-World 2.0, VALERANT produces more coherent geometry and appearance across viewpoints, resulting in a more consistent and natural representation of the generated scene.

For the user study, participants compared the two methods following the same instructions and four evaluation criteria used in the VLM-based assessment. Their preferences were collected through a questionnaire, and participation entailed no foreseeable risk.

C BASELINE COMPARISON

To the best of our knowledge, HY-World 2.0 (HY-World et al., 2026) is the only publicly available baseline that shares our objective of constructing an explorable 3D scene from a single image. Given an input image, HY-World 2.0 first employs HY-Pano 2.0 to synthesize a complete 360° panorama. WorldNav then parses the panorama into preliminary geometry, semantic landmarks, and navigable regions, from which it plans camera trajectories intended to maximize scene coverage. Following these trajectories, WorldStereo 2.0 generates camera-controlled keyframes to expand regions that are not sufficiently observed in the initial panorama. Finally, WorldMirror 2.0 reconstructs the generated multi-view observations, after which the resulting geometry is composed and optimized into a 3DGS scene.

HY-World 2.0 produces sparse keyframe observations rather than directly returning a dense video sequence comparable to the action-conditioned rollouts generated by VALERANT. For a fair comparison in a common output space, we therefore represent and render the results of both methods as 3DGS scenes. Consequently, the visual quality of both sets of rendered images is constrained by the reconstruction and optimization of their 3DGS representations and may be lower than that of directly generated video frames. Figure 5 presents a qualitative comparison between the two methods under this unified setting.

As shown in Figure 5, HY-World 2.0 exhibits weaker cross-view consistency and less coherent scene structure. We attribute this difference primarily to its sequential generation process. Its visual content is synthesized in two successive phases: an initial panorama is first generated from the input image, after which a separate world-expansion stage synthesizes additional observations for regions that are missing or insufficiently covered by the panorama. Although HY-World 2.0 introduces geometric and stereo memory to improve consistency, discrepancies introduced during panorama generation can still be propagated or amplified when the missing regions are subsequently completed. The expanded views may therefore disagree with the initial panorama or with one another,

Table 1: Quantitative comparison. AS denotes aesthetic score, and UP denotes user preference.

Method	AS (%)↑	UP (%)↑
HY-World 2.0	20.00	16.67
VALERANT (Ours)	80.00	83.33

producing appearance drift and geometric discontinuities in the final 3DGS reconstruction. In contrast, VALERANT generates action-conditioned video rollouts from the currently committed context and incrementally integrates selected observations, which better preserves local continuity along the exploration trajectory.

The quantitative results in Table 1 further support these observations. VALERANT achieves an AS of 80.00%, compared with 20.00% for HY-World 2.0, indicating that the VLM evaluator more frequently favors its overall visual quality, 3D consistency, trajectory coherence, and game-scene realism. Human evaluation exhibits a similar trend: VALERANT receives 83.33% of user preferences, while HY-World 2.0 receives 16.67%. The close agreement between the VLM-based assessment and human preferences suggests that the qualitative advantages of VALERANT are consistently perceived under both evaluation protocols.