GEOMETRY AS ADDRESS: ROUTING ATTENTION TO VISUAL MEMORY FOR LONG-HORIZON CAMERA-CONTROLLED VIDEO GENERATION

Anonymous authors
Paper under double-blind review



Figure 1: **Precise camera control and persistent scene memory for long-horizon exploration.** Given a single image and a text prompt, **GEAR** generates minute-long, photorealistic videos along arbitrary user-specified camera trajectories, while preserving spatial consistency and faithfully recovering previously observed content upon revisitation.

ABSTRACT

Long-horizon camera-controlled video generation requires recovering previously observed content from an ever-growing visual history. Existing approaches either search historical context implicitly or reconstruct it into persistent 3D memory, facing inefficient memory access or accumulated geometric errors. Our key insight is that geometry need not explain the scene – it only needs to determine *where visual memory should be read from*, while attention decides *what should be recovered*. Based on this insight, we introduce GEAR, a Geometry-Enabled Attention Routing framework that uses geometry as an explicit token-level address for visual memory. Rather than fusing historical observations into a persistent global 3D representation, GEAR retains them as frame latents and uses per-frame geometry only to establish token-level correspondences with target views, thereby avoiding persistent error accumulation from global fusion. Guided by these correspondences, a proposed Geometric Correspondence Attention (GCA) selectively injects geometrically matched historical features into target noisy patches during denoising. We further introduce an Invisible Octree to accumulate visibility evidence and reject geometrically plausible but occluded correspondences. Extensive experiments demonstrate that GEAR achieves state-of-the-art visual quality, precise camera control, and revisit consistency, enabling minute-long video generation along challenging trajectories. Additional videos are available at `https://anonymousgear46.github.io/geometry-as-address/`.

1 INTRODUCTION

Recent advances in video generation have enabled increasingly realistic and controllable visual synthesis (Team Wan et al., 2025; Team Seedance, 2026; Kling AI, 2026; Google DeepMind, 2025), opening the door to interactive world generation where camera trajectories or user actions control the synthesized observations (AlayaWorld Team et al., 2026; Zhu et al., 2026; Robbyant Team et al., 2026; Team HY-World, 2026; Alibaba Token Hub, 2026; Parker-Holder & Fruchter, 2025; Mao et al., 2026). Extending these models to long-horizon exploration, however, requires more than generating plausible short clips: the model must accurately follow the prescribed camera trajectory

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.

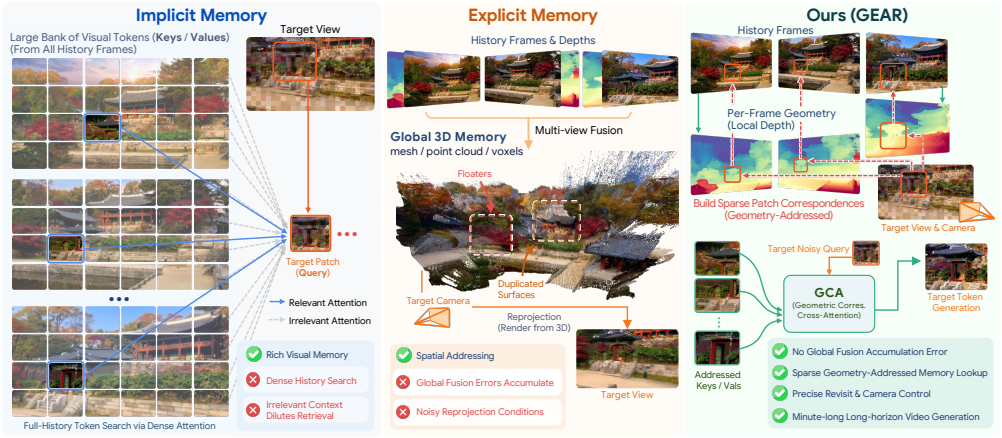

Figure 2: **Comparison of long-term memory paradigms.** Implicit memory preserves rich visual history but requires dense search over an increasingly large token set, while explicit 3D memory provides spatial addressing at the cost of accumulated reconstruction and fusion errors. GEAR instead uses per-frame geometry only to address historical latent patches, enabling sparse, spatially grounded memory retrieval without persistent global 3D fusion.

while preserving previously observed scene content over long temporal gaps. When a scene region is revisited after leaving the model's temporal context, its appearance must be recovered from past observations rather than inferred from the current context alone. Long-horizon generation therefore becomes a *memory access problem*: for each target region, the model must identify and retrieve the relevant visual evidence from history.

Existing approaches largely address this problem through two paradigms, illustrated in Fig. 2. History-based implicit methods retain previously generated observations as context memory and recover relevant information through attention (He et al., 2025a; Bai et al., 2025a; Xiao et al., 2025; Li et al., 2025b; Yu et al., 2025b; Xiao et al., 2025; Yu et al., 2025a; Sun et al., 2026). This preserves rich visual information, but as history grows, the model must search over an increasingly large token set to recover the few observations relevant to the current viewpoint, making memory access costly and vulnerable to irrelevant context. Frame retrieval or history compression (Xiao et al., 2025; Yu et al., 2025a; Zhang et al., 2025; Wu et al., 2026b) alleviates this burden, but operates at a coarser granularity or may discard information needed for precise revisitation. Reconstruction-based explicit methods instead fuse historical observations into a global 3D representation and reproject it to target views (Ren et al., 2025; Wu et al., 2025; Zhao et al., 2026; Chen et al., 2026). While this provides natural spatial addressing, local geometry errors can become persistent after global fusion and propagate into subsequent generations through noisy reprojection.

Recent methods have begun to bridge these two paradigms by exploiting geometry without relying on a single globally fused 3D memory. AnchorWeave (Wang et al., 2026) maintains multiple local geometric representations, reprojects retrieved memories as target-view anchor videos and adaptively fuses them through ControlNet (Zhang et al., 2023). UCM (Xu et al., 2026) instead warps positional encodings to geometrically align interactions between historical and target tokens. Most closely related, Lyra 2.0 (Shen et al., 2026) warps source coordinates and depth into target views, then injects their embeddings into DiT tokens. Despite these advances, geometry is still used to *select, align, or construct geometric conditioning signals from history*, while *access to the underlying visual features remains indirect*. This motivates a more explicit separation between geometry and visual memory: rather than using correspondence only to guide generation, we use it to directly determine which historical visual tokens each target token can access.

Rather than asking geometry to explain the scene, our key insight is to let geometry answer only **where memory should be read from**, while leaving **what should be recovered** to visual attention over the matched historical features. Based on this insight, we introduce **GEAR**, a Geometry-Enabled Attention Routing framework that **uses geometry as an explicit token-level address for visual memory**. For each target token, GEAR identifies a sparse set of geometrically matched historical tokens and gathers their visual features as keys and values for attention. In this way, geometry constrains the memory search space, while visual attention resolves which historical evidence is most useful for generation. We realize this mechanism through Geometric Correspondence Attention (GCA), which exposes each noisy target token only to its geometrically matched historical memory tokens as keys and values, and injects the aggregated features through a residual branch during denoising. Geometry thus serves as a transient address rather than persistent scene state: it explicitly determines where each target token can retrieve visual evidence, while correspondence er-

rors remain local to individual memory accesses instead of accumulating across views.

Cross-view projection alone, however, may produce false correspondences when a source-visible surface becomes occluded in the target view. We therefore introduce an Invisible Octree that accumulates visibility evidence over time and filters such correspondences without storing scene appearance. Together, these designs enable efficient patch-level memory access over long trajectories: rather than requiring the video model to search the entire history or the geometry estimator to reconstruct the entire world, GEAR uses geometry to identify which pieces of history are relevant to each piece of the future. Our contributions are summarized as follows:

- We propose GEAR, a Geometry-Enabled Attention Routing framework that decouples visual memory from geometric addressing. By converting per-frame 3D priors into token-level cross-view correspondences, GEAR enables fine-grained access to relevant historical visual features without error-prone global 3D fusion.
- We introduce Geometric Correspondence Attention for sparse token-level memory injection, together with an Invisible Octree for visibility-aware correspondence filtering.
- Extensive experiments demonstrate state-of-the-art visual quality, camera-control accuracy, and revisit consistency, enabling minute-long video generation along challenging user-specified trajectories with only lightweight adaptation of a pretrained video model.

2 Related Work

Implicit Camera-Controlled Video Generation. Early camera-controlled methods inject camera trajectories or user actions into video generation (He et al., 2025a; Bai et al., 2025a; Yu et al., 2025b; Li et al., 2025a). Subsequent streaming approaches extend generation to longer horizons (Huang et al., 2025b; Robbyant Team et al., 2026; He et al., 2025b), but finite temporal context limits scene persistence. To retain visual history, Xiao et al. (2025); Li et al. (2025c); Yu et al. (2025a); Sun et al. (2026) retrieve historical frames as additional context, while Hong et al. (2025); Wu et al. (2026b) compress history into compact latent representations. These approaches preserve rich visual information but still require dense attention with an increasing set of historical tokens.

Explicit Camera-Controlled Video Generation. Recent methods reconstruct historical observations into 3D memories and project them to target viewpoints as pixel-aligned conditions (Wu et al., 2025; Zhao et al., 2026; Zhang et al., 2026; Lee et al., 2026; Wang et al., 2026; Chen et al., 2026). While providing explicit spatial grounding, these methods are vulnerable to accumulated reconstruction and fusion errors. More closely related to our work, UCM (Xu et al., 2026) warps positional encodings to establish cross-view relationships, while Lyra 2.0 (Shen et al., 2026) injects warped correspondence coordinates as token embeddings. Our GEAR instead uses per-frame geometry to select visible historical tokens for sparse correspondence attention, retaining visual memory without persistent global 3D fusion.

3 Methods

3.1 Problem Formulation and Preliminaries

Camera-Conditioned Autoregressive Video Generation. Given an initial frame I_1, text prompts y and a long camera trajectory, our goal is to autoregressively synthesize visual observations with accurate camera control and long-term consistency with previously observed scene regions.

Following Wu et al. (2026a); Chen et al. (2026), we encode each frame independently using the VAE (Team Wan et al., 2025), $z_i = E(I_i)$, which preserves frame-level alignment between visual observations and camera poses, particularly under large camera motions, enabling cross-view correspondences to be constructed directly at the latent-token level.

After generating t frames, we maintain a streaming history bank with the historical latent observations and their camera parameters:

$$\mathcal{B}_t = \{(z_i, c_i)\}_{i=1}^{t}. \tag{1}$$

Given the next camera chunk $c_{t+1:t+k}$, a compact conditioning history $\mathcal{H}_t^{\text{ret}}$ is retrieved from $\mathcal{B}_t$ according to its geometric coverage of the target views, as detailed in Sec. 3.6. The video generator then samples the next latent chunk as:

$$z_{t+1:t+k} \sim p_\theta\big(\cdot \mid \mathcal{H}_t^{\text{ret}}, c_{t+1:t+k}, y\big)\,. \tag{2}$$

The generated latents and their camera parameters are appended to $\mathcal{B}_t$, and the same procedure is repeated over successive chunks of the prescribed trajectory.

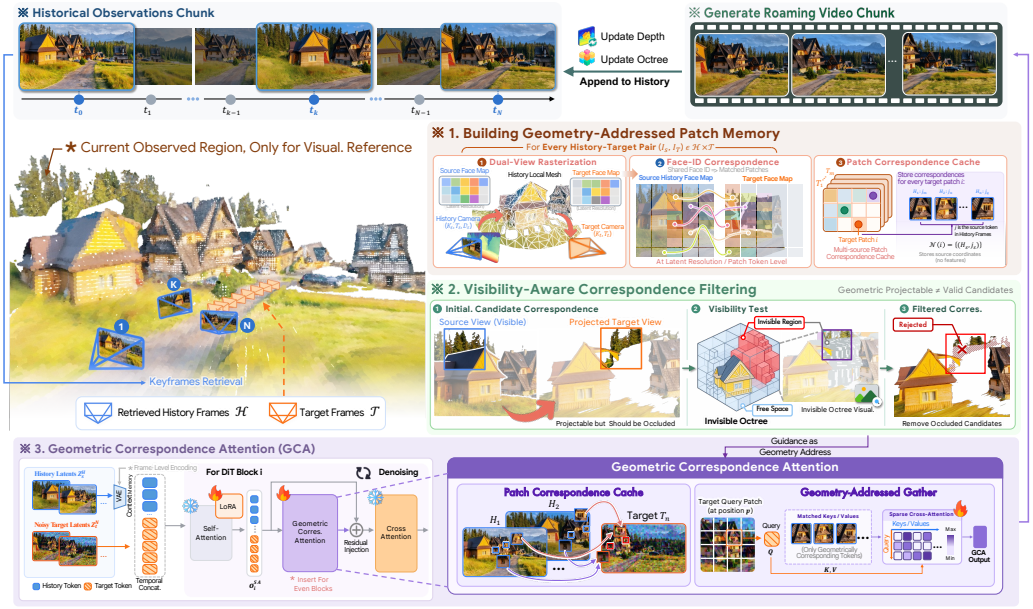

Figure 3: **System overview.** For each target chunk, GEAR constructs patch correspondences to retrieved history using per-frame geometry and filters occluded matches with the Invisible Octree. GCA then injects matched historical features into noisy target tokens during denoising, after which generated observations are appended to the history bank for continued rollout.

Flow-Matching Objective. We train the video generator with the standard flow-matching objective (Lipman et al., 2023). For a target latent block $\mathbf{z}_{t+1:t+k}$, we sample Gaussian noise $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and a flow timestep $\tau \sim \mathcal{U}(0, 1)$, and construct the linear interpolation:

$$\mathbf{z}^{\tau}_{t+1:t+k} = (1 - \tau)\mathbf{z}_{t+1:t+k} + \tau\boldsymbol{\epsilon}. \tag{3}$$

The conditional velocity field is optimized as:

$$\mathcal{L}_{\mathrm{FM}} = \mathbb{E}_{\mathbf{z},\boldsymbol{\epsilon},\tau}\left[\left\| v_\theta\left(\mathbf{z}^{\tau}_{t+1:t+k}, \tau \mid \mathcal{H}_t, \mathbf{c}_{t+1:t+k}, y\right) - \left(\boldsymbol{\epsilon} - \mathbf{z}_{t+1:t+k}\right)\right\|_2^2\right]. \tag{4}$$

Although history retrieval bounds the number of conditioning frames, only a sparse and view-dependent subset of their tokens is relevant to each target region. The key problem is therefore not merely which historical frames to retain, but how each target token should access the corresponding historical evidence. In the following, we introduce Geometry-Addressed Patch Memory to construct these fine-grained memory addresses from per-frame geometry.

3.2 Geometry-Addressed Patch Memory

Given the retrieved history, our goal is to establish patch-level correspondences between historical and target views, which subsequently serve as explicit addresses for memory retrieval. Rather than constructing a globally fused scene representation, we derive these correspondences independently from the local geometry associated with each historical frame, as illustrated in Fig. 3.

Local Geometric Anchor. For each historical frame s, we associate its latent z_s with an estimated depth map D_s, camera intrinsics K_s, and extrinsics T_s. We back-project D_s into 3D and connect neighboring pixels according to the image-grid topology, producing a local triangular mesh $\mathcal{G}_s = (\mathcal{V}_s, \mathcal{F}_s)$, where V_s and F_s denote the mesh vertices and faces, respectively. Since modern video VAEs aggressively compress the spatial resolution, e.g., by a factor of $16\times$, each latent token may cover pixels belonging to multiple surfaces. We construct the mesh at the original image resolution to preserve geometric discontinuities that would otherwise be blurred by directly downsampling depth to the latent grid.

Geometry-Guided Patch Correspondence Addressing. We then rasterize the same source mesh under both the source and target cameras at the latent spatial resolution $H_\ell \times W_\ell$, yielding two face-index maps:

$$F_s = \mathcal{R}_{H_\ell \times W_\ell}(\mathcal{G}_s; K_s, T_s), \quad F_{s\to t} = \mathcal{R}_{H_\ell \times W_\ell}(\mathcal{G}_s; K_t, T_t), \tag{5}$$

where each valid entry records the ID of the visible mesh face associated with a latent patch token. Since both maps are rasterized from the same local geometry, shared face IDs naturally establish a token-level correspondence:

$$\mathcal{C}_{s\to t} = \{(i, j) \mid F_{s\to t}(i) = F_s(j)\}. \tag{6}$$

Shared face identities thus provide a direct geometric bridge between source and target latent tokens, while preserving the fine spatial structure captured by the full-resolution source geometry.



Figure 4: **Streaming update of the Invisible Octree.** Historical observations incrementally accumulate coarse visibility evidence along the exploration trajectory. For a target view, the resulting invisible-space boundary is used to reject projectable but occluded source-target correspondences.

Multi-Source Patch Correspondence Cache. We construct $\mathcal{C}_{s\to t}$ for every selected history-target frame pair and organize them into a patch correspondence cache:

$$\mathcal{C} \in \mathbb{Z}^{N_t \times N_c \times H_\ell \times W_\ell \times 2}, \tag{7}$$

where N_t and N_c denote the numbers of target and retrieved historical condition frames. For each target patch and historical frame, $\mathcal{C}$ stores the coordinate (x_s, y_s) of its geometrically corresponding source patch, with unmatched entries marked as invalid. Since the same scene region may have been observed from multiple historical viewpoints, a target patch can naturally admit multiple source correspondences. We therefore retain all valid historical candidates. Crucially, since all correspondences are derived independently from each source observation, geometric errors remain local instead of accumulating into persistent artifacts through global 3D fusion.

3.3 Visibility-Aware Correspondence

Geometric projectability does not necessarily imply target-view visibility. As illustrated in Fig. 3, a surface visible in source frame s projects into target frame t while being occluded by geometry unobserved in s. Such candidates provide spatially incorrect historical evidence and should be removed before memory retrieval.

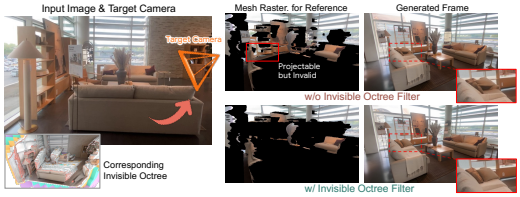

Figure 5: **Ablation of Invisible Octree.** The Invisible Octree removes projectable but occluded correspondences, preventing erroneous historical content from affecting target-view synthesis.

To validate these correspondences, we maintain an Invisible Octree as a sparse global visibility proxy. After each generated chunk, the estimated depth maps are used to incrementally update the octree with newly observed free-space and occlusion evidence, as illustrated in Fig. 4. The update is conservative: new observations only refine previously unknown or invisible regions, allowing the octree to expand with exploration without repeatedly overwriting established evidence. Its sparse and adaptive structure also supports long-horizon generation in unbounded environments. **Please refer to Appendix for details on the construction and streaming update of the Invisible Octree.**

For a target camera, we use the Invisible Octree to determine the visibility mask under the target viewpoint. Candidates lying behind the accumulated visibility boundary are rejected as occluded. Importantly, the octree stores no appearance and never serves as a rendering condition; it only provides a coarse binary filter over correspondences.

3.4 Geometric Correspondence Attention

The correspondence cache above specifies *where* each target patch should retrieve historical evidence from. Building on these memory addresses, we introduce **Geometric Correspondence Attention (GCA)** as an auxiliary memory pathway, which selectively aggregates only the geometrically matched historical tokens and injects them into target noisy tokens during denoising.

Sparse Correspondence Attention. Following context-memory-based approaches (Sun et al., 2026; Yu et al., 2025a), we concatenate the retrieved historical frames (Sec. 3.6) as context latents with the noisy target latents along the temporal dimension. Let h_i^T denote the post-self-attention feature of target token i, and h_j^H denote the corresponding feature of historical token j. From the patch correspondence cache, each target token obtains its geometrically matched historical candidates $\mathcal{N}(i)$. We then perform cross-attention:

$$q_i = W_Q h_i^T, \qquad k_j = W_K h_j^H, \qquad v_j = W_V h_j^H. \tag{8}$$

The attention weight assigned to candidate j is normalized only over the geometrically matched set, and the geometry-addressed memory feature is aggregated as:

$$\alpha_{ij} = \frac{\exp\left(q_i^\top k_j/\sqrt{d}\right)}{\sum_{m\in\mathcal{N}(i)} \exp\left(q_i^\top k_m/\sqrt{d}\right)}, \qquad o_i^{\mathrm{GCA}} = \sum_{j\in\mathcal{N}(i)} \alpha_{ij} v_j. \tag{9}$$

When multiple historical views observe the same target region, attention adaptively aggregates their complementary appearance information. If $\mathcal{N}(i)$ is empty, we set $o_i^{\mathrm{GCA}} = 0$, allowing the pretrained video model to synthesize unobserved content from its generative prior.



Figure 6: **Qualitative comparison on DL3DV-Eval.** Compared with explicit and implicit memory baselines, GEAR better preserves visual quality, camera adherence, and scene consistency throughout long-horizon generation. **See our supp. website for additional scenes and video comparisons.**

Lightweight Residual Injection. GCA is inserted after the original self-attention at selected DiT blocks, as shown in Fig. 3. Its output is projected back to the backbone feature space and injected through a residual connection:

$$\widetilde{h}_i^T = h_i^T + W_O o_i^{\mathrm{GCA}}, \tag{10}$$

where W_O projects the feature back to the DiT feature space. The backbone self-attention retains its pretrained spatiotemporal modeling, while GCA supplies a sparse geometry-addressed correction that anchors target features to relevant historical evidence.

3.5 Robust Training with Degraded History

During training, target chunks are conditioned on ground-truth history, whereas autoregressive inference relies on previously generated observations. This train–inference discrepancy causes generation errors to enter the history memory and progressively accumulate over long rollouts.

To expose the model to imperfect yet semantically consistent history, we introduce degraded-history augmentation. With probability p_{deg}, we corrupt the historical latents $\mathbf{z}_{\mathcal{H}}$ with a randomly sampled low noise level $\tau_h \sim \mathcal{U}(0, \tau_{\max})$:

$$\mathbf{z}_{\mathcal{H}}^{\tau_h} = (1-\tau_h)\mathbf{z}_{\mathcal{H}} + \tau_h \boldsymbol{\epsilon}_{\mathcal{H}}, \qquad \boldsymbol{\epsilon}_{\mathcal{H}} \sim \mathcal{N}(0, \mathbf{I}). \tag{11}$$

We then apply one reverse-flow step with the current model to obtain the degraded history:

$$\hat{\mathbf{z}}_{\mathcal{H}} = \mathrm{sg}[\mathbf{z}_{\mathcal{H}}^{\tau_h} - \tau_h\, v_\theta\,(\mathbf{z}_{\mathcal{H}}^{\tau_h}, \tau_h)]. \tag{12}$$

The resulting $\hat{\mathbf{z}}_{\mathcal{H}}$ replaces $\mathbf{z}_{\mathcal{H}}$ as the conditioning context, while keeping the flow-matching objective for the target chunk unchanged. This exposes the model to the mild distortions encountered during rollout and improves robustness to accumulated history errors in long-horizon inference.

3.6 Long-Horizon Inference

For long-horizon inference, we adopt a streaming strategy that bounds the historical context through keyframe retrieval and continuously updates the memory.

Keyframe History Retrieval. As the rollout progresses, retaining all historical frames introduces increasing computational cost and substantial view redundancy. We therefore retrieve a compact history according to its geometric coverage of the upcoming target chunk. Specifically, we project each historical frame's local geometry onto the target views, and greedily select frames that maximize newly covered regions. Meanwhile, the initial frame and the latest frame are retained to preserve scene identity and inter-chunk continuity. The retrieved history frames are thus $\mathcal{H}_t^{\mathrm{ret}} = \big[\mathcal{H}_{\mathrm{first}}, \mathcal{H}_{\mathrm{coverage}}, \mathcal{H}_{\mathrm{last}}\big]$.

Streaming Memory Update. After each chunk is generated, new frames are appended to the history bank with their camera parameters and depths estimated by the frozen 3D foundation model (Lin et al., 2026). The Invisible Octree is simultaneously updated with the newly observed geometry. This retrieve–generate–update procedure is repeated for subsequent chunks, enabling long-horizon autoregressive generation.

Table 1: **Quantitative comparison on DL3DV-Evaluation and WorldScore-Static.** GEAR performs favourably across the metrics. Best results are in **bold** and second are underlined.

Method	DL3DV-Evaluation						WorldScore-Static					
	SSIM ↑	LPIPS ↓	FVD ↓	TransErr ↓	RotErr ↓	ATE ↓	Content Align. ↑	Photo. Cons. ↑	Style Cons. ↑	Subjective Quality ↑	Revisit SSIM ↑	Revisit LPIPS ↓
Lyra2	0.3359	0.5097	975.89	0.0157	0.1721	0.2514	0.6319	0.9357	0.8600	0.5017	0.3941	0.3218
Spatia	0.3081	0.5422	1074.13	0.0617	0.6973	1.1204	0.6331	0.8588	0.8600	0.5012	0.4407	0.3541
WorldStereo	0.3061	0.5502	846.46	0.0239	0.1717	0.2212	0.7003	0.0837	0.8300	0.5013	0.6253	0.2193
UCM	0.3412	0.6007	1431.67	0.0377	0.5184	0.6410	0.6773	0.9692	0.7300	0.5005	0.3412	0.4890
HY-WorldPlay	0.2452	0.6213	1385.23	0.0508	0.6854	0.9634	0.5104	0.7939	0.1900	**0.5018**	0.2433	0.7288
Lingbot-World	0.2608	0.6168	1309.27	0.0417	0.5988	0.7449	0.5564	0.0889	0.6800	0.5006	0.2495	0.7677
Infinite-World	0.2532	0.6399	1582.43	0.1072	0.8473	2.3744	0.6302	0.9728	0.7000	0.5011	0.2536	0.7098
GEAR	**0.3645**	**0.4459**	**837.59**	**0.0116**	**0.1228**	**0.0436**	**0.7423**	**0.9732**	**0.8700**	**0.5018**	**0.6489**	**0.2019**

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

Dataset. We train GEAR on DL3DV-10K (Ling et al., 2024), a large-scale real-world dataset with diverse camera trajectories. Each sequence is divided into 55-frame clips at a resolution of 480×832. We employ Depth Anything 3 (Lin et al., 2026) to estimate camera poses and per-frame depths, and generate video captions with Qwen3-VL-8B-Instruct (Bai et al., 2025b).

We construct training samples under two conditioning modes. In image-to-video (I2V) mode, we train on the first 32 frames, with the initial frame providing the geometry for initializing the Invisible Octree and establishing correspondences with the target views. In history-to-video (H2V) mode, the first 32 frames constitute the history bank, from which nine keyframes are retrieved following Sec. 3.6; the remaining 23 frames serve as generation targets. The retrieved keyframes are used to construct the multi-source patch correspondence cache, while the complete history bank is used to build the global Invisible Octree for visibility-aware correspondence filtering.

Implementation Details. We adopt Wan2.1-I2V-14B (Team Wan et al., 2025) as the pretrained backbone and keep all original parameters frozen. To adapt the backbone to frame-aligned VAE latents, we introduce rank-32 LoRA adapters (Hu et al., 2022), and we insert GCA modules with a hidden dimension of 640 into every even-indexed DiT block (262M parameters for GCA, only 1.9% of the backbone). The LoRA adapters and GCA modules are jointly optimized for 10K iterations.

During training, I2V and H2V samples are drawn with probabilities of 30% and 70%, respectively. Starting from iteration 8K, degraded-history augmentation is further applied to H2V samples with a probability $p_{\text{deg}} = 40\%$ and $\tau_{\max} = 0.3$. We optimize the model using AdamW on 32 NVIDIA GPUs (96 GB each) with a learning rate of 1×10^{-4} and a linear warm-up over the first 1K iterations.

4.2 QUANTITATIVE EVALUATION

Baselines and Metrics. We compare GEAR with recent camera-controlled long-horizon video generation methods equipped with memory mechanisms. Explicit 3D baselines include Lyra2 (Shen et al., 2026), Spatia (Zhao et al., 2026), HY-WorldStereo (Zhang et al., 2026), and UCM (Xu et al., 2026), while implicit baselines include HY-WorldPlay (Sun et al., 2026), Lingbot-World (Robbyant Team et al., 2026), and Infinite-World (Wu et al., 2026b).

We first evaluate all methods on DL3DV-Evaluation (Ling et al., 2024). Given the same initial frame, each method autoregressively generates subsequent frames along the ground-truth camera trajectory. We report SSIM (Wang et al., 2004), LPIPS (Zhang et al., 2018), and FVD (Unterthiner et al., 2018) to evaluate frame-level fidelity and temporal quality. For camera-control accuracy, we recover camera poses from the generated videos using ViPE (Huang et al., 2025a) and align them with the ground-truth trajectories using the Umeyama transformation (Umeyama, 1991). We then report TransErr, RotErr, and Absolute Trajectory Error (ATE).

We further evaluate long-horizon revisitation on 50 randomly sampled scenes from the WorldScore Static set (Duan et al., 2025) using closed-loop camera trajectories. We report Content Alignment, Subjective Quality, Style Consistency, and Photometric Consistency, together with Revisit SSIM and Revisit LPIPS, which are computed between the generated revisit frame and the reference observation at the matched camera pose to assess the recovery of previously observed scene content.

Quantitative and Qualitative Comparison. As shown in Tab. 1, GEAR performs favorably across the evaluated metrics. Our method reduces ATE by 80.3% compared with the strongest baseline, demonstrating improved trajectory adherence, as further illustrated in Fig. 7. GEAR achieves the best content alignment, photometric consistency, style consistency, and revisit performance on WorldScore. These results demonstrate that GEAR faithfully recovers the previously observed content after long temporal gaps without compromising overall generation quality.

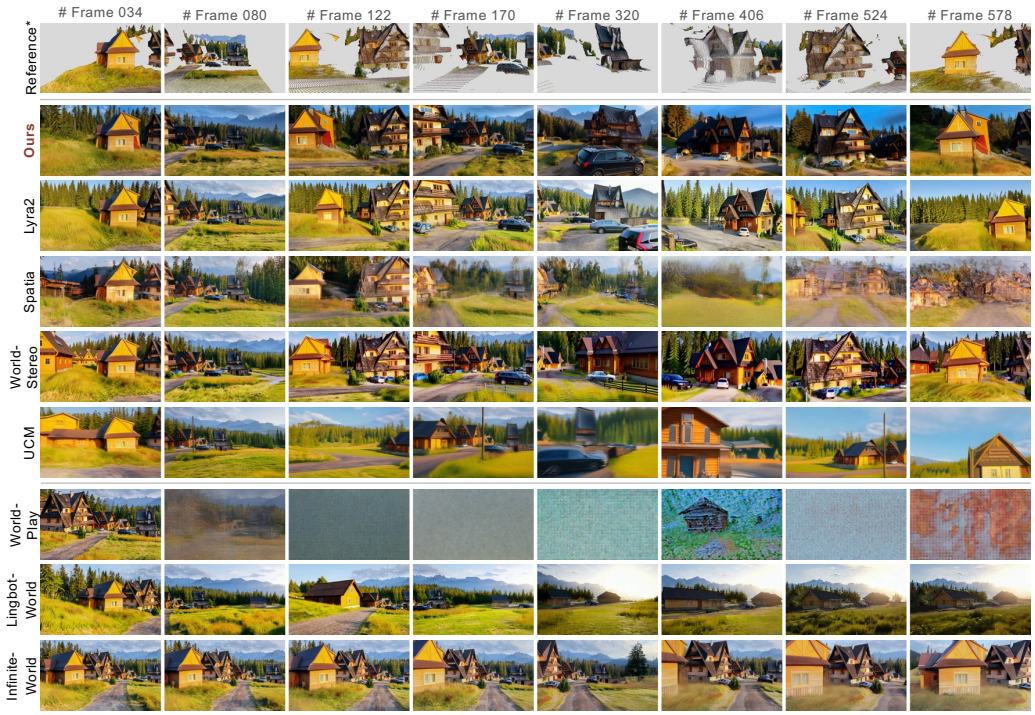

Figure 8: **Out-of-domain qualitative comparison for minute-long generation.** GEAR maintains visual fidelity and scene consistency over extended camera trajectories, while competing methods exhibit progressive drift and visual degradation. *First-frame geometry projected along the target trajectory for viewpoint reference only. **See our supp. website for additional video comparisons.**

The qualitative comparisons in Fig. 6 and 8 further demonstrate the advantages of GEAR over existing methods. Lyra2 provides competitive camera control but develops visual degradation and increasing trajectory drift under rapid or extended camera motion, while Spatia and WorldStereo exhibit progressive scene distortions, consistent with their vulnerability to accumulated reconstruction errors. UCM initially preserves coherent appearance but gradually deviates over extended rollouts. HY-WorldPlay and Lingbot-World struggle with both camera control and revisitation, whereas Infinite-World maintains comparatively stable appearance but insufficiently follows the precise trajectory. In particular, WorldStereo and Lingbot-World exhibit abrupt appearance changes and pronounced flicker during camera rotations, consistent with their low Photometric Consistency scores. These limitations become more pronounced during minute-long explorations with rapid camera motion (Fig. 8): most baselines exhibit severe visual degradation and develop increasing camera drift. GEAR maintains coherent appearance and accurate camera control, supporting the effectiveness of separating visual memory from geometric addressing for long-horizon generation. **Please refer to our anonymous supplementary website for additional scenes and more extensive video comparisons.**

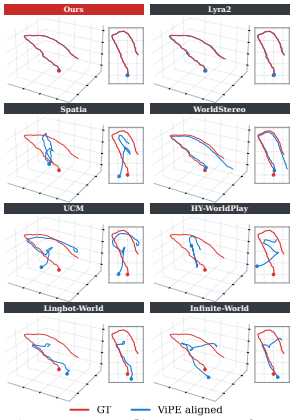

Figure 7: **Camera alignment.** GEAR achieves the best camera alignment on DL3DV-Evaluation.

4.3 Ablation Study

We evaluate the contribution of each key component of GEAR:

Without Geometric Correspondence Attention (GCA). We train an ablated variant without GCA while retaining the backbone's dense attention over the concatenated history and target tokens. As shown in Fig. 10, disabling GCA's residual injection increases temporal instability and deviation from the camera motion. The quantitative degradation in Tab. 2 further confirms the contribution of GCA to camera-control accuracy and long-horizon consistency.

Variant with Dense History Attention. To isolate geometric addressing from the additional capacity of GCA, we train a variant that retains the same GCA module and parameter count but replaces correspondence-restricted attention with dense attention over full retrieved historical context tokens. The resulting degradation in camera control and visual fidelity (Tab. 2) confirms the importance of our sparse, geometry-addressed memory retrieval.



Figure 9: **Ablation of Degraded-History Augmentation.** Training with degraded history mitigates error accumulation and improves visual stability during long-horizon autoregressive generation.



Figure 10: **Ablation of GCA.** GCA improves temporal stability and adherence to the prescribed camera trajectory.

Table 2: **Ablation study on DL3DV-Evaluation.** We ablate the contribution of each key component of GEAR.

Variant	SSIM ↑	LPIPS ↓	FVD ↓	TransErr ↓	RotErr ↓	ATE ↓
w/o GCA	0.1324	0.6549	1201.32	0.0406	0.4732	0.3471
w/ Dense Attention	0.2749	0.5995	913.64	0.0351	0.4347	0.2924
w/o Invisible Octree	0.3350	0.4869	878.28	0.0153	0.1551	0.0729
w/o Degraded History	0.3459	0.4763	1165.99	0.0127	0.1492	0.0664
w/ Corres. Disturbance	0.3597	0.4513	842.15	0.0121	0.1273	0.0464
GEAR	**0.3645**	**0.4459**	**837.59**	**0.0116**	**0.1228**	**0.0436**

Without Invisible Octree. Under large viewpoint changes as shown in Fig. 5, the projectable yet occluded correspondences not only introduce local appearance errors but also propagate structural inconsistencies through the autoregressive history. The Invisible Octree suppresses this error propagation by rejecting candidates that conflict with accumulated visibility evidence.

Without Degraded-History Augmentation. As shown in Fig. 9, training exclusively with ground-truth history leads to droplet-like floaters and visual degradation during long-horizon rollout. Degraded-history augmentation mitigates these artifacts by exposing the model to imperfect historical observations, substantially improving robustness and visual stability over extended generation.



Figure 11: **Ablation of Correspondence Disturbance.** We perturb candidates in multi-observed target regions by 64–128 pixels while retaining only 2–3 valid matches out of 9 conditions. The figure visualizes one such perturbation and the resulting error map; GCA recovers the target content with minimal deviation from the undisturbed result.

Robustness to Correspondence Disturbance. Depth errors accumulated during autoregressive rollout may introduce inaccurate history-to-target correspondences. For target regions observed in multiple historical frames, we retain only 2–3 valid observations out of 9 and perturb the rest by 64–128 pixels to simulate such errors. Despite these strong perturbations, GEAR maintains comparable performance as shown in Fig. 11 and Tab. 2, indicating that the learned attention of GCA suppresses feature-inconsistent distractors and preserves valid historical evidence.

5 Conclusion

We introduced GEAR, a geometry-enabled attention routing framework for long-horizon camera-controlled video generation. GEAR uses per-frame geometry to establish local history-to-target token correspondences, and leverages lightweight Geometric Correspondence Attention to retrieve and integrate relevant historical features during denoising. By treating geometry as an address rather than persistent memory, GEAR avoids error accumulation from global 3D fusion while enabling minute-long generation with precise camera control and consistent scene persistence. GEAR currently relies on an external 3D model for depth estimation, introducing additional computational overhead. A promising direction for future work is to jointly learn geometry and memory addressing within the generative model for more robust long-horizon generation.

AI USE STATEMENT

AI tools were used only to improve the clarity and readability of the manuscript. The authors take full responsibility for the final work.

ETHICS STATEMENT

This work studies camera-controlled long-horizon video generation using publicly available datasets and pretrained models, without collecting new personal data or involving human subjects. We assert that this research has been carried out in accordance with the code of ethics.

REPRODUCIBILITY STATEMENT

In order to support reproducibility and verification, we present implementation and evaluation details in our paper, and will release the associated source code upon acceptance.

REFERENCES

AlayaWorld Team, Kaipeng Zhang, Chuanhao Li, Yifan Zhan, Yongtao Ge, Yuanyang Yin, Jiaming Tan, Kang He, Liaoyuan Fan, Ruicong Liu, Mingliang Zhai, et al. AlayaWorld v1.1: Long-horizon and playable video world generation. *arXiv preprint arXiv:2608.13492*, 2026.

Alibaba Token Hub. Happy Oyster: An open-ended world model for real-time world creation and interaction. `https://www.happyoyster.com/`, April 2026. Accessed: 2026-08-28.

Jianhong Bai, Menghan Xia, Xiao Fu, Xintao Wang, Lianrui Mu, Jinwen Cao, Zuozhu Liu, Haoji Hu, Xiang Bai, Pengfei Wan, and Di Zhang. ReCamMaster: Camera-controlled generative rendering from a single video. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 14834–14844, 2025a.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-VL technical report. *arXiv preprint arXiv:2511.21631*, 2025b.

Yunpeng Bai, Haoxiang Li, and Qixing Huang. Positional encoding field. *arXiv preprint arXiv:2510.20385*, 2025c.

Yutian Chen, Shi Guo, Renbiao Jin, Tianshuo Yang, Xin Cai, Yawen Luo, Mingxin Yang, Mulin Yu, Linning Xu, and Tianfan Xue. AnyRecon: Arbitrary-view 3D reconstruction with video diffusion model. In *ACM SIGGRAPH Asia 2026 Conference Papers*. ACM, 2026.

Haoyi Duan, Hong-Xing Yu, Sirui Chen, Li Fei-Fei, and Jiajun Wu. WorldScore: A unified evaluation benchmark for world generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 27713–27724, 2025.

Google DeepMind. Veo 3 model card. `https://storage.googleapis.com/deepmind-media/Model-Cards/Veo-3-Model-Card.pdf`, May 2025. Published May 23, 2025; updated January 13, 2026.

Hao He, Yinghao Xu, Yuwei Guo, Gordon Wetzstein, Bo Dai, Hongsheng Li, and Ceyuan Yang. CameraCtrl: Enabling camera control for text-to-video generation. In *International Conference on Learning Representations*, 2025a.

Xianglong He, Chunli Peng, Zexiang Liu, Boyang Wang, Yifan Zhang, Qi Cui, Fei Kang, Biao Jiang, Mengyin An, Yangyang Ren, Baixin Xu, Hao-Xiang Guo, Kaixiong Gong, Cyrus Wu, Wei Li, Xuchen Song, Yang Liu, Eric Li, and Yahui Zhou. Matrix-Game 2.0: An open-source, real-time, and streaming interactive world model. *arXiv preprint arXiv:2508.13009*, 2025b.

Yicong Hong, Yiqun Mei, Chongjian Ge, Yiran Xu, Yang Zhou, Sai Bi, Yannick Hold-Geoffroy, Mike Roberts, Matthew Fisher, Eli Shechtman, Kalyan Sunkavalli, Feng Liu, Zhengqi Li, and Hao Tan. RELIC: Interactive video world model with long-horizon memory. *arXiv preprint arXiv:2512.04040*, 2025.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.

Jiahui Huang, Qunjie Zhou, Hesam Rabeti, Aleksandr Korovko, Huan Ling, Xuanchi Ren, Tianchang Shen, Jun Gao, Dmitry Slepichev, Chen-Hsuan Lin, et al. ViPE: Video pose engine for 3D geometric perception. *arXiv preprint arXiv:2508.10934*, 2025a.

Xun Huang, Zhengqi Li, Guande He, Mingyuan Zhou, and Eli Shechtman. Self Forcing: Bridging the train-test gap in autoregressive video diffusion. In *Advances in Neural Information Processing Systems*, 2025b.

Nikhil Keetha, Norman Müller, Johannes Schönberger, Lorenzo Porzi, Yuchen Zhang, Tobias Fischer, Arno Knapitsch, Duncan Zauss, Ethan Weber, Nelson Antunes, Jonathon Luiten, Manuel Lopez-Antequera, Samuel Rota Bulò, Christian Richardt, Deva Ramanan, Sebastian Scherer, and Peter Kontschieder. MapAnything: Universal feed-forward metric 3D reconstruction. In *International Conference on 3D Vision (3DV)*. IEEE, 2026.

Kling AI. Kling VIDEO 3.0 model user guide. `https://app.klingai.com/global/quickstart/klingai-video-3-model-user-guide`, February 2026. Official model documentation.

Samuli Laine, Janne Hellsten, Tero Karras, Yeongho Seol, Jaakko Lehtinen, and Timo Aila. Modular primitives for high-performance differentiable rendering. *ACM Transactions on Graphics*, 39(6), 2020.

JoungBin Lee, Jaewoo Jung, Jisang Han, Takuya Narihira, Kazumi Fukuda, Junyoung Seo, Sunghwan Hong, Yuki Mitsufuji, and Seungryong Kim. 3D scene prompting for scene-consistent camera-controllable video generation. In *International Conference on Learning Representations*, 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, 2025a. URL `https://arxiv.org/abs/2506.17201`.

Ruilong Li, Brent Yi, Junchen Liu, Hang Gao, Yi Ma, and Angjoo Kanazawa. Cameras as relative positional encoding. In *Advances in Neural Information Processing Systems*, pp. 15984–16009, 2025b.

Runjia Li, Philip Torr, Andrea Vedaldi, and Tomas Jakab. VMem: Consistent interactive video scene generation with surfel-indexed view memory. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 25690–25699, 2025c.

Haotong Lin, Sili Chen, Jun Hao Liew, Donny Y. Chen, Zhenyu Li, Guang Shi, Jiashi Feng, and Bingyi Kang. Depth Anything 3: Recovering the visual space from any views. In *International Conference on Learning Representations*, 2026.

Lu Ling, Yichen Sheng, Zhi Tu, Wentian Zhao, Cheng Xin, Kun Wan, Lantao Yu, Qianyu Guo, Zixun Yu, Yawen Lu, et al. DL3DV-10K: A large-scale scene dataset for deep learning-based 3D vision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22160–22169, 2024.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *International Conference on Learning Representations*, 2023.

Miles Macklin. Warp: A high-performance python framework for gpu simulation and graphics, March 2022. URL `https://github.com/NVIDIA/warp`. NVIDIA GPU Technology Conference (GTC).

Xiaofeng Mao, Zhen Li, Chuanhao Li, Xiaojie Xu, Kaining Ying, and Kaipeng Zhang. Yume1.5: A text-controlled interactive world generation model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026.

Jack Parker-Holder and Shlomi Fruchter. Genie 3: A new frontier for world models. https://deepmind.google/blog/genie-3-a-new-frontier-for-world-models/, August 2025. Google DeepMind; accessed 2026-08-28.

Xuanchi Ren, Tianchang Shen, Jiahui Huang, Huan Ling, Yifan Lu, Merlin Nimier-David, Thomas Müller, Alexander Keller, Sanja Fidler, and Jun Gao. GEN3C: 3D-informed world-consistent video generation with precise camera control. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.

Robbyant Team, Zelin Gao, Qiuyu Wang, Yanhong Zeng, Jiapeng Zhu, Ka Leong Cheng, Yixuan Li, Hanlin Wang, Yinghao Xu, Shuailei Ma, et al. Advancing open-source world models. *arXiv preprint arXiv:2601.20540*, 2026.

Tianchang Shen, Sherwin Bahmani, Kai He, Sangeetha Grama Srinivasan, Tianshi Cao, Jiawei Ren, Ruilong Li, Zian Wang, Nicholas Sharp, Zan Gojcic, et al. Lyra 2.0: Explorable generative 3D worlds. *arXiv preprint arXiv:2604.13036*, 2026.

Wenqiang Sun, Haiyu Zhang, Haoyuan Wang, Junta Wu, Zehan Wang, Zhenwei Wang, Yunhong Wang, Jun Zhang, Tengfei Wang, and Chunchao Guo. WorldPlay: Towards long-term geometric consistency for real-time interactive world modeling. In *International Conference on Machine Learning*, 2026.

Team HY-World. HY-World 2.0: A multi-modal world model for reconstructing, generating, and simulating 3D worlds. *arXiv preprint arXiv:2604.14268*, 2026. doi: 10.48550/arXiv.2604.14268. URL https://arxiv.org/abs/2604.14268.

Team Seedance. Seedance 2.0: Advancing video generation for world complexity. *arXiv preprint arXiv:2604.14148*, 2026. doi: 10.48550/arXiv.2604.14148. URL https://arxiv.org/abs/2604.14148.

Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, Jianyuan Zeng, et al. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025. doi: 10.48550/arXiv.2503.20314. URL https://arxiv.org/abs/2503.20314.

Shinji Umeyama. Least-squares estimation of transformation parameters between two point patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(4):376–380, 1991.

Thomas Unterthiner, Sjoerd Van Steenkiste, Karol Kurach, Raphael Marinier, Marcin Michalski, and Sylvain Gelly. Towards accurate generative models of video: A new metric & challenges. *arXiv preprint arXiv:1812.01717*, 2018.

Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004.

Zun Wang, Han Lin, Jaehong Yoon, Jaemin Cho, Yue Zhang, and Mohit Bansal. AnchorWeave: World-consistent video generation with retrieved local spatial memories. In *European Conference on Computer Vision*. Springer, 2026.

Qi Wu, Khiem Vuong, Minsik Jeon, Srinivasa Narasimhan, and Deva Ramanan. FrameCrafter: Novel view synthesis as video completion. In *European Conference on Computer Vision*, pp. 73–91. Springer, 2026a.

Ruiqi Wu, Xuanhua He, Meng Cheng, Tianyu Yang, Yong Zhang, Zhuoliang Kang, Xunliang Cai, Xiaoming Wei, Chunle Guo, Chongyi Li, and Ming-Ming Cheng. Infinite-World: Scaling interactive world models to 1000-frame horizons via pose-free hierarchical memory. In *International Conference on Machine Learning*, 2026b.

Tong Wu, Shuai Yang, Ryan Po, Yinghao Xu, Ziwei Liu, Dahua Lin, and Gordon Wetzstein. Video world models with long-term spatial memory. In *Advances in Neural Information Processing Systems*, 2025.

Zeqi Xiao, Yushi Lan, Yifan Zhou, Wenqi Ouyang, Shuai Yang, Yanhong Zeng, and Xingang Pan. WorldMem: Long-term consistent world simulation with memory. In *Advances in Neural Information Processing Systems*, pp. 49632–49652, 2025.

Tianxing Xu, Zixuan Wang, Guangyuan Wang, Li Hu, Zhongyi Zhang, Peng Zhang, Bang Zhang, and Songhai Zhang. UCM: Unified modeling of camera control and memory with time-aware positional encoding warping for world models. In *ACM SIGGRAPH 2026 Conference Papers*. ACM, 2026. doi: 10.1145/3799902.3811088.

Jiwen Yu, Jianhong Bai, Yiran Qin, Quande Liu, Xintao Wang, Pengfei Wan, Di Zhang, and Xihui Liu. Context as Memory: Scene-consistent interactive long video generation with memory retrieval. In *SIGGRAPH Asia 2025 Conference Papers*. ACM, 2025a. doi: 10.1145/3757377.3763833.

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

Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models, 2023.

Lvmin Zhang, Shengqu Cai, Muyang Li, Gordon Wetzstein, and Maneesh Agrawala. Frame context packing and drift prevention in next-frame-prediction video diffusion models. In *Advances in Neural Information Processing Systems*, 2025.

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 586–595, 2018.

Yisu Zhang, Chenjie Cao, Tengfei Wang, Xuhui Zuo, Junta Wu, Jianke Zhu, and Chunchao Guo. WorldStereo: Bridging camera-guided video generation and scene reconstruction via 3D geometric memories. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 40327–40339, 2026.

Jinjing Zhao, Fangyun Wei, Zhening Liu, Hongyang Zhang, Chang Xu, and Yan Lu. Spatia: Video generation with updatable spatial memory. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4245–4257, 2026.

Yixuan Zhu, Jiaqi Feng, Wenzhao Zheng, Yuan Gao, Xin Tao, Pengfei Wan, Jie Zhou, and Jiwen Lu. Astra: General interactive world model with autoregressive denoising. In *International Conference on Learning Representations*, 2026.

APPENDIX

This appendix provides additional details and results for GEAR. Appendix A describes the complete streaming generation procedure (A.1), including frame-aligned encoding (A.2), correspondence construction (A.3), visibility filtering (A.4), keyframe retrieval (A.5), and depth updates (A.6). Appendix B reports dataset preprocessing, training settings, and inference costs. Appendix C details the evaluation protocols and baseline configurations. Appendix D presents additional qualitative comparisons and long-horizon generation results. **Additional video results and visualizations of our method details are available on our anonymous project page: `https://anonymousgear46.github.io/geometry-as-address/`.**

A ADDITIONAL METHOD DETAILS

A.1 END-TO-END STREAMING GENERATION

Algorithm 1 End-to-end streaming generation with GEAR

Require: History bank $\mathcal{B}_t = \{(I_s, z_s, c_s, D_s)\}_{s=1}^{t}$; future cameras $c_{t+1:T}$; text condition y; Invisible Octree $\mathcal{O}_t$; chunk length k

Ensure: Generated frames $\hat{I}_{t+1:T}$ and updated history bank $\mathcal{B}_T$

while $t < T$ **do**
  $\mathcal{T} \leftarrow \{t+1, \dots, \min(t+k, T)\}$
  $\mathcal{H} \leftarrow \textsc{RetrieveKeyframes}(\mathcal{B}_t, \{c_u\}_{u\in\mathcal{T}})$ ▷ Greedy target-view coverage; retain first and latest frames
  for all $s \in \mathcal{H}$ **do**
    $G_s \leftarrow \textsc{BuildLocalMesh}(D_s, c_s)$ ▷ Back-project depth and connect neighboring pixels
    $F_s \leftarrow \textsc{RasterizeFaceIDs}(G_s, c_s, H_\ell, W_\ell)$
  end for
  Initialize $\mathcal{C}[u, s, i] \leftarrow \perp$ for $u \in \mathcal{T}$, $s \in \mathcal{H}$, and target patches i
  for all $u \in \mathcal{T}$ **do**
    $M_u^{\text{inv}} \leftarrow \textsc{ProjectInvisibleOctree}(\mathcal{O}_t, c_u)$ ▷ Target-view invisible mask
    for all $s \in \mathcal{H}$ **do**
      $F_{s\to u} \leftarrow \textsc{RasterizeFaceIDs}(G_s, c_u, H_\ell, W_\ell)$
      $\mathcal{C}_{s\to u} \leftarrow \textsc{MatchFaceIDs}(F_s, F_{s\to u})$ ▷ Store matched source-patch coordinates
      $\mathcal{C}_{s\to u} \leftarrow \textsc{FilterOccluded}(\mathcal{C}_{s\to u}, M_u^{\text{inv}})$
      Write valid matches from $\mathcal{C}_{s\to u}$ into $\mathcal{C}[u, s, :]$
    end for
  end for
  $z_{\mathcal{H}} \leftarrow \{z_s\}_{s\in\mathcal{H}}$; $x^{(0)} \sim \mathcal{N}(0, I)$
  Choose a denoising schedule $1 = \tau_0 > \tau_1 > \cdots > \tau_{25} = 0$
  for $n = 0, \dots, 24$ **do**
    $v^{(n)} \leftarrow v_\theta(x^{(n)}, \tau_n \mid z_{\mathcal{H}}, \{c_u\}_{u\in\mathcal{T}}, y, \mathcal{C})$ ▷ GCA uses the filtered cache
    $x^{(n+1)} \leftarrow \textsc{SolverStep}(x^{(n)}, v^{(n)}, \tau_n, \tau_{n+1})$
  end for
  $\hat{z}_{\mathcal{T}} \leftarrow x^{(25)}$; $\hat{I}_{\mathcal{T}} \leftarrow \textsc{DecodeFramewise}(\hat{z}_{\mathcal{T}})$
  $\hat{D}_{\mathcal{T}} \leftarrow \textsc{DepthAnything3}(\hat{I}_{\mathcal{T}})$
  for all $u \in \mathcal{T}$ **do**
    $\mathcal{B}_u \leftarrow \mathcal{B}_{u-1} \cup \{(\hat{I}_u, \hat{z}_u, c_u, \hat{D}_u)\}$
  end for
  $\mathcal{O}_{\max\mathcal{T}} \leftarrow \textsc{UpdateInvisibleOctree}(\mathcal{O}_t, \{(\hat{D}_u, c_u)\}_{u\in\mathcal{T}})$
  $t \leftarrow \max\mathcal{T}$
end while
return $\hat{I}_{t_0+1:T}, \mathcal{B}_T$

We summarize the complete inference procedure in Algorithm 1. At generation step t, the history bank contains the observed frames, their frame-aligned VAE latents, camera parameters, and estimated depths: $\mathcal{B}_t = \{(I_s, z_s, c_s, D_s)\}_{s=1}^{t}$, where $c_s = (K_s, T_s)$ denotes the camera intrinsics and pose. The Invisible Octree $\mathcal{O}_t$ accumulates visibility evidence from the *complete* history bank.

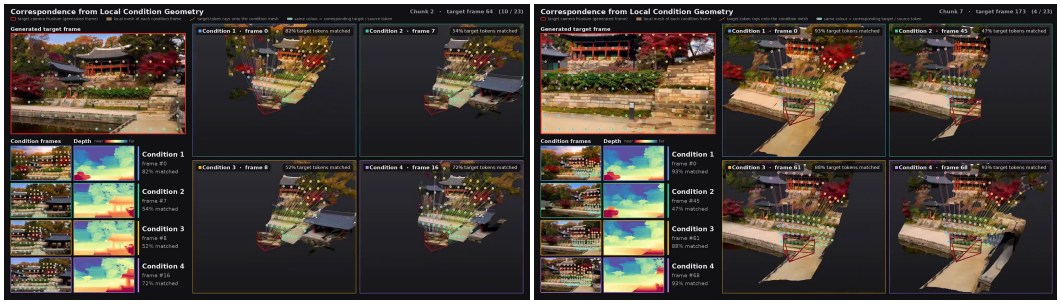

Figure 12: **Visualization of Geometry-Guided Patch Correspondence Construction.** Local geometry from each historical observation is used to build a history-to-target patch correspondence cache at latent resolution. Guided by this cache, GCA allows each noisy target token to attend only to its geometrically matched historical memory tokens as keys and values.

For each upcoming camera chunk, we retrieve a compact set of historical frames according to their geometric coverage of the target views, while retaining the first and latest frames for scene identity and inter-chunk continuity. Each retrieved frame independently provides a local mesh constructed from its depth map. Rasterizing this mesh under the source and target cameras yields face-ID maps, from which we build a multi-source patch correspondence cache. We then project the Invisible Octree into each target view and remove candidates marked as occluded by the resulting visibility mask. The retrieved history latents, noisy target latents, and filtered correspondence cache jointly condition the DiT model throughout $N_{\text{denoise}} = 25$ denoising steps. Finally, we decode the generated latents, estimate their depths with Depth Anything 3 (Lin et al., 2026), append the new observations to the history bank, and update the Invisible Octree before processing the next chunk.

A.2 Frame-Aligned VAE Encoding

GCA requires the geometric correspondences constructed in Sec. 3.2 to address latent tokens associated with specific camera views. The original Wan VAE (Team Wan et al., 2025) encodes the first video frame separately but temporally compresses subsequent frames by a factor of 4. Consequently, a latent frame after the first generally aggregates observations captured at different camera poses. Assigning a single pose to that latent frame cannot provide an exact geometric interpretation for all observations. The ambiguity becomes more pronounced under rapid camera motion, when the aggregated frames may depict substantially different scene regions. Thus, correspondences computed from frame-level depth and camera parameters cannot be unambiguously transferred to temporally compressed latent tokens.

Following (Wu et al., 2026a; Chen et al., 2026), we instead apply the pretrained Wan VAE's single-frame encoding path independently to every video frame. Each frame is treated as a separate one-frame input, bypassing temporal compression while retaining the VAE's spatial encoding:

$$z_f = \mathcal{E}_{\text{Wan}}(I_f), \qquad f = 1, \ldots, F, \tag{13}$$

where $\mathcal{E}_{\text{Wan}}$ denotes the VAE applied to an individual frame. This produces F latent frames for F video frames, so each latent frame has a unique associated image, depth map, and camera pose. Within that frame, each spatial latent token also has a well-defined location on the image grid. We can therefore rasterize per-frame geometry at the latent resolution and use the resulting source–target patch correspondences to index historical tokens directly during GCA. The generated latents are likewise decoded frame by frame to preserve the same alignment at inference.

A.3 Geometry-Addressed Correspondence Construction

Depth back-projection and mesh connectivity. For a historical frame s, let D_s be its estimated depth map and $c_s = (K_s, T_s)$ its camera parameters. We construct a separate local mesh $G_s = (V_s, F_s)$ from this observation. For each pixel $p = (u, v)$ with a finite, positive depth, we back-project its pixel center into world coordinates:

$$\mathbf{x}_s(p) = T_s^{-1} \left(D_s(p) K_s^{-1} \begin{bmatrix} u + \frac{1}{2} \\ v + \frac{1}{2} \\ 1 \end{bmatrix} \right), \tag{14}$$

where T_s denotes the world-to-camera transform and homogeneous coordinates are understood. Each valid pixel contributes one vertex. We then split each 2×2 image-grid cell along a fixed

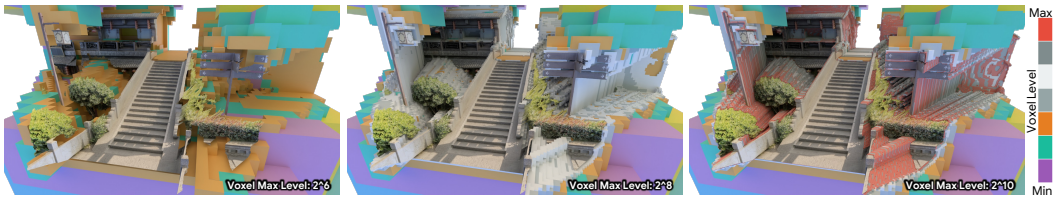

Figure 13: **Visualization of the Invisible Octree update process.** Partially visible voxels are recursively subdivided until each node becomes fully visible or fully invisible, or the maximum voxel resolution is reached. With increasing resolution, the Invisible Octree progressively approximates the true invisible regions in the target view.

diagonal to form two candidate triangles. This retains the spatial resolution of the depth map during mesh construction rather than smoothing depth discontinuities by first downsampling into the latent grid.

Invalid-depth and discontinuity filtering. A candidate triangle is discarded if any of its vertices has an invalid depth. We also remove triangles that would connect surfaces across a sharp depth discontinuity. Specifically, for a triangle f with pixel vertices p_1, p_2, p_3, we retain it only if

$$\max_{(p_i,p_j)\in E(f)} \frac{|D_s(p_i) - D_s(p_j)|}{\min(D_s(p_i), D_s(p_j))} \leq \tau_{\text{disc}}, \tag{15}$$

where $E(f)$ contains its three edges and τ_{disc} is the relative depth-discontinuity threshold. This filtering prevents triangles from spanning foreground–background boundaries and producing correspondences through unsupported geometry. Meshes are constructed independently for each historical frame; their vertices and faces are not fused across observations.

Dual-view face-ID rasterization. As shown in Fig. 12, for each selected source frame s and target frame t, we transform the vertices of the *same* local mesh G_s into the respective camera clip spaces and rasterize them with `nvdiffrast` (Laine et al., 2020) at the latent resolution $H_\ell \times W_\ell$:

$$F_s = \mathcal{R}(G_s; c_s), \qquad F_{s\to t} = \mathcal{R}(G_s; c_t). \tag{16}$$

The rasterizer's depth test assigns each covered latent-grid location the ID of its nearest visible triangle; a zero ID denotes background. We use these discrete IDs without interpolating them and express both maps in a common image-coordinate convention before matching.

Because the two maps refer to the same source mesh, a valid shared face ID defines a source–target patch correspondence:

$$\mathcal{C}_{s\to t} = \big\{(i,j) \;\big|\; F_{s\to t}(i) = F_s(j) > 0\big\}, \tag{17}$$

where i and j index target and source latent-grid locations, respectively. Face IDs are matched only *within* each source mesh; the source-frame index is retained when combining matches from multiple historical views. The resulting cache stores the matched source-patch coordinates for each target patch and source frame, with unmatched entries marked invalid. These per-source candidates are subsequently filtered using the target-view Invisible Octree mask before they are accessed by GCA.

A.4 Invisible Octree Construction and Streaming Update

Octree states. We maintain an Invisible Octree $\mathcal{O}$ as a sparse proxy for accumulated visibility evidence along the camera trajectory. Each allocated node v is assigned one of three states: *free*, *invisible*, or *partially visible*. A free node lies entirely in observed free space along the relevant camera rays, whereas an invisible node lies entirely behind the observed depth surface and has not yet been resolved by an observation. A partially visible node intersects the boundary between these regions, or contains a mixture of visible and invisible space.

Given a camera with depth map D, let $[z_v^-, z_v^+]$ be the depth range of v in camera space, and let $d_v^{\min}$ and $d_v^{\max}$ be the minimum and maximum valid depths over its projected footprint. A node containing no observed surface points is classified as

$$s(v) = \begin{cases} \text{free}, & z_v^+ < d_v^{\min} \text{ or no valid depth in the footprint}, \\ \text{invisible}, & z_v^- > d_v^{\max}, \\ \text{partially visible}, & \text{otherwise}. \end{cases} \tag{18}$$

Free and invisible nodes are terminal for the current visibility update. Partially visible nodes are recursively subdivided until their children can be classified or the maximum resolution is reached. Only invisible leaf nodes can be revisited using subsequent camera observations.

Initialization from the first view. Given the first camera c_1 and depth map D_1, we set the octree's world-space bounds according to the scene scale and initialize a coarse grid with $N_{\min} = 2^4$ cells per axis. A one-cell-thick *background shell* of invisible cells encloses the full scene to account for regions with invalid depth estimates.

Both the global invisible octree $\mathcal{O}$ and a global visible mesh $\mathcal{G}^{\text{vis}}$, triangulated from back-projected depth, are initialized from the first frame. Octree nodes are classified against depths rendered from $\mathcal{G}^{\text{vis}}$, ensuring consistency between the two global proxies. For the first chunk, classification starts from the coarsest grid and recursively refines the octree up to a resolution equivalent to 2^{10} cells per axis. The two structures serve complementary purposes: $\mathcal{O}$ represents currently unresolved invisible space, whereas $\mathcal{G}^{\text{vis}}$ represents surfaces that have already been observed. **The global visible mesh is used only for visibility comparison;** the source-specific local meshes used to construct GCA correspondences remain independent. We visualize the octree subdivision and update process in Fig. 13.

Identifying regions to update. For a newly generated frame with camera c_t and estimated depth D_t, we first render both global proxies into its view. We build a BVH over the invisible octree leaves and ray-cast it to obtain the depth $d_t^{\text{inv}}(p)$ of the first invisible voxel along each camera ray. Separately, we rasterize $\mathcal{G}^{\text{vis}}$ to obtain its visible-surface depth $d_t^{\text{vis}}(p)$. Their depth ordering identifies image regions where previously invisible space appears in front of the surface already represented by the global visible mesh. With a small comparison tolerance ϵ, the corresponding update mask can be written as

$$M_t^{\text{upd}}(p) = \mathbb{1}\left[d_t^{\text{inv}}(p) + \epsilon < d_t^{\text{vis}}(p)\right]. \tag{19}$$

Only pixels with a valid current depth and a valid invisible-voxel intersection are considered for this comparison. We use D_t within M_t^{upd} to reclassify the corresponding octree region by the same depth-based procedure used during initialization. Previously invisible nodes can therefore be refined as new observations reveal their contents. We further back-project and triangulate the masked depth observations and incorporate the resulting surfaces into $\mathcal{G}^{\text{vis}}$. Restricting both updates to newly exposed regions prevents established visibility evidence from being repeatedly overwritten by depth estimates from later generated frames.

Visibility-aware correspondence filtering. The global Invisible Octree and visible mesh are updated after each generated chunk as the camera progresses along its trajectory. For each target camera, we project the accumulated global proxies into the target view and compare their depths to determine the target-view invisible mask. This mask is then applied directly to the source–target correspondence cache, removing candidates that fall within regions determined to be occluded before GCA accesses the corresponding historical tokens. Importantly, neither global proxy provides appearance features to the generator. Appearance information remains entirely within the historical frame latents, while $\mathcal{O}$ and $\mathcal{G}^{\text{vis}}$ provide only visibility information for filtering their geometrically addressed correspondences.

A.5 Keyframe History Retrieval

For each target chunk, we retrieve historical frames based on their coverage of the views to be generated. Let $\mathcal{T}$ denote the target frames in the chunk, and let $\mathcal{Q} = \{(u, i) \mid u \in \mathcal{T}, i \in \Omega_u\}$ be the set of their spatial locations. For each historical frame s, we project its depth-derived local geometry into every target view. After filtering occluded projections with the target-view Invisible Octree mask, we obtain a coverage set $\mathcal{V}_s \subseteq \mathcal{Q}$.

We retain the first and latest historical frames to provide scene identity and continuity across chunks. Starting from these frames, we greedily add the candidate with the largest marginal contribution to target-view coverage. To limit redundant selection from densely observed regions, we track how many selected frames cover each target location q:

$$n_q(\mathcal{S}) = \sum_{s \in \mathcal{S}} \mathbb{1}[q \in \mathcal{V}_s], \qquad \Delta(s \mid \mathcal{S}) = \sum_{q \in \mathcal{V}_s} \mathbb{1}[n_q(\mathcal{S}) < N_{\text{covered}}], \tag{20}$$

where $\mathcal{S}$ is the current set of selected frames and $N_{\text{covered}} = 3$. At each iteration, we select the remaining frame with the largest $\Delta(s \mid \mathcal{S})$ until the history budget is reached. Once a target location has been covered three times, further coverage of that location contributes no additional score.

Table 3: **Training configuration.** Rank-32 LoRA adapters and GCA modules are trained jointly on the frozen Wan2.1-I2V-14B backbone.

Setting	Value
Backbone	Wan2.1-I2V-14B (Team Wan et al., 2025)
Training resolution	480×832
LoRA rank	32
GCA hidden dimension	640
GCA placement	Even-indexed DiT blocks
Numerical precision	BF16 mixed precision
I2V / H2V sampling ratio	30% / 70%
Retrieved history keyframes	9
Optimizer	AdamW
Peak learning rate	10^{-4}
Learning-rate warm-up	1K iterations
Training iterations	10K
Global batch size	32
Hardware	32 NVIDIA GPUs (96 GB each)

Table 4: **Inference cost.** Denoising time is measured per step under matched inputs.

Operation	Scope	Cost
Wan2.1-14B	Per step, One 96GB GPU	33.8 s / 39.9 GB
Wan2.1-14B with GCA	Per step, One 96GB GPU	34.5 s (+2.1%) / 40.9 GB
Depth Anything 3	48-frame call	14 s
Invisible Octree update	23-frame chunk	3.4 s

This encourages the selected conditions to span the upcoming target views while retaining multiple observations where available.

A.6 Depth Update for Streaming Generation

After generating each video chunk, we estimate the depths of its decoded frames using Depth Anything 3 (Lin et al., 2026). Estimating the new frames in isolation could introduce inconsistencies with the geometry stored in the history bank. We therefore include uniformly sampled historical frames as anchors. For a history bank containing N_{hist} frames, we use the sampling interval $N_{\text{interval}} = \max\big(1, \lfloor \frac{N_{\text{hist}}}{25} \rfloor\big)$ and select historical frames at this interval in temporal order. We jointly feed the anchor frames and newly generated frames to DA3, together with their corresponding camera intrinsics and extrinsics. We use its pose-conditioned mode so that depth estimation is informed by the prescribed camera trajectory. We retain only the depths of the newly generated frames; the depths already stored in the history bank are left unchanged. These new depth maps are then appended to the bank and used to update the visibility proxies for subsequent chunks.

B Dataset and Implementation Details

Dataset preprocessing. We train on DL3DV-10K (Ling et al., 2024) after filtering scenes with pronounced motion blur or insufficient illumination, retaining approximately 6.5K scenes. Each retained sequence is divided into consecutive, non-overlapping 55-frame clips at a resolution of 480×832. We use Depth Anything 3 (Lin et al., 2026) to obtain per-frame camera poses and depths, and Qwen3-VL-8B-Instruct (Bai et al., 2025b) to generate video captions. For history-to-video (H2V) training, the first 32 frames form the history bank, from which nine keyframes are retrieved; the remaining 23 frames serve as generation targets. After filtering and processing, the resulting dataset comprises approximately 30K high-quality video clips.

Training configuration. We use Wan2.1-I2V-14B (Team Wan et al., 2025) as the pretrained backbone and freeze its original parameters. Rank-32 LoRA adapters and GCA modules are jointly trained, with GCA inserted after self-attention in every even-indexed DiT block. Each GCA module has a hidden dimension of 640. We zero-initialize its output projection W_O, so that the GCA residual is initially zero and does not perturb the pretrained backbone features at the start of training. The complete training settings are summarized in Table 3.

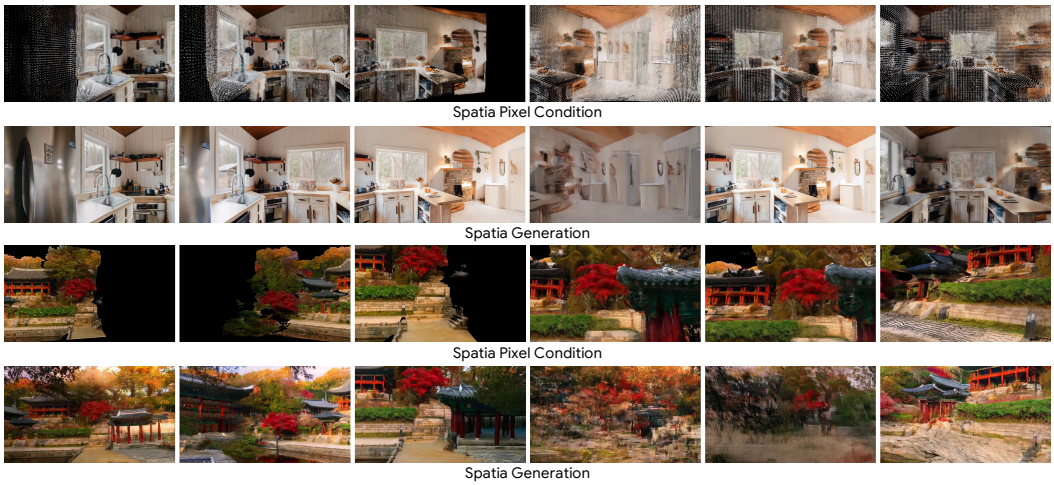

Figure 14: **Visualization of the 3D pixel condition and generated frames of Spatia.** Even with relatively clean 3D pixel-aligned conditioning, Spatia exhibits visible temporal instability and image degradation. Under more complex camera trajectories and noisier 3D pixel-aligned conditioning, frame jitter emerges in the first generated chunk, followed by a complete breakdown of visual content in the second.

Inference configuration and computation cost. We use 25 denoising steps with a classifier-free guidance scale of 5. Table 4 reports the denoising time per step, depth-estimation latency, and Invisible Octree update time. We measure backbone-only and GCA-enhanced denoising on a single NVIDIA 96GB GPU using identical input dimensions and history configurations. Depth-estimation latency is measured for a 48-frame inference call, while Invisible Octree update time is reported per 23-frame generated chunk. GCA adds only 1.9% to the backbone parameter count with minimal denoising overhead. We implement Invisible Octree management and updates in Warp (Macklin, 2022) to enable parallel execution on the GPU. Both coverage-based history retrieval and rasterization-based correspondence construction can be efficiently parallelized on the GPU, introducing negligible computational overhead during inference.

C Evaluation Protocols

C.1 Baseline Configuration and Method Comparison

Baseline availability. We discuss AnchorWeave (Wang et al., 2026) as a related method, but exclude it from quantitative comparisons since its inference weights were not publicly available at the time of evaluation.

Depth and camera-scale alignment. DL3DV-Evaluation (Ling et al., 2024) provides camera trajectories whose scale may differ from that of the depth predicted by baselines. Such a mismatch changes the effective magnitude of the prescribed camera motion and can confound camera-control comparisons. To establish a common geometric scale, we estimate multi-view depths for each evaluation scene using Depth Anything 3 (Lin et al., 2026). For the explicit memory methods requiring an initial-frame depth map, we align their predicted depth to the DA3 estimate of the first frame by least-squares scale fitting over valid pixels. We also provide the same aligned initial-frame depth to the geometry-based baselines. For UCM (Xu et al., 2026), we replace the depth estimator used in its original implementation with pose-conditioned DA3, as used by Lyra 2.0 (Shen et al., 2026) and GEAR in our evaluation, because depth estimated without conditioning on the supplied camera poses may be inconsistent with the evaluation trajectory.

Implicit-memory baselines. For HY-WorldPlay (Sun et al., 2026), Lingbot-World (Robbyant Team et al., 2026), and Infinite-World (Wu et al., 2026b), we normalize camera translations according to scene scale before inference while preserving the prescribed camera rotations and relative motion. Infinite-World (Wu et al., 2026b) accepts discrete action inputs rather than continuous camera poses; following its processing protocol, we convert consecutive relative camera transformations into the corresponding action sequence. Its trajectory-adherence results should therefore be interpreted in light of this action discretization.

Geometry-based correspondence methods. Similar to our method, UCM (Xu et al., 2026) and Lyra 2.0 (Shen et al., 2026) leverage estimated geometry to establish correspondences between historical observations and target viewpoints, without fusing these observations into a persistent global

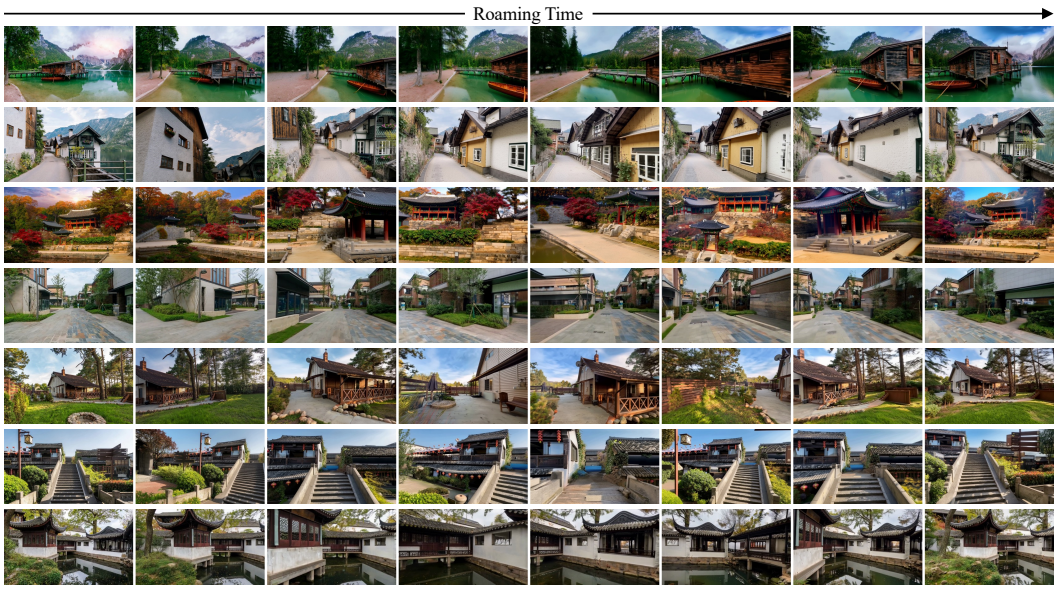

Figure 15: **More results across a broader range of data.**

3D representation. The key distinction lies in how the resulting correspondences are incorporated into the generation process. Following PE-Field (Bai et al., 2025c), UCM projects historical observations into relevant target views and warps their positional encodings, allowing historical and target tokens to interact through geometry-aware attention. To limit computation, each historical frame is assigned a single relevant target viewpoint for this warping. Lyra 2.0 instead forward-warps canonical source coordinates and depth from multiple retrieved frames, encodes the resulting correspondence maps, and adds their embeddings to DiT tokens. These designs provide geometric cues for memory access, but do not explicitly restrict each target patch to attend to its set of matched historical patches.

GEAR stores the source coordinates of valid history-to-target patch correspondences and uses them to gather historical features as keys and values for Geometric Correspondence Attention. A target patch can draw on multiple historical observations, while the Invisible Octree filters candidates that are projectable but occluded in the target view. The resulting attention output is injected through a residual branch during denoising. This provides direct access to the visual content of geometrically matched memory patches while keeping depth errors local to individual source views.

In the comparisons in Figs. 6 and 8, UCM preserves coherent views early in the rollout but exhibits increasing camera drift and visual degradation under longer or faster camera motion. Lyra 2.0 is the strongest geometry-correspondence baseline in these examples, yet also degrades under substantial viewpoint changes. These observations are consistent with the quantitative results in Table 1.

Globally fused 3D memory methods. Spatia (Zhao et al., 2026) reconstructs historical observations with MapAnything (Keetha et al., 2026), updates a persistent scene point cloud, and renders it from target viewpoints to produce spatial guidance for subsequent video generation. As shown in Fig. 14, this can recover convincing observations on some relatively simple rotational trajectories in WorldScore-Static. In the more challenging DL3DV-Evaluation examples and long-horizon trajectories, however, we observe scene distortion and loss of previously visible content, in some cases beginning within the first generation window and becoming more severe in subsequent rollouts. These failures are consistent with errors in the accumulated point cloud being repeatedly rendered into the conditioning signal. GEAR avoids this source of persistent geometric error by retaining visual observations as frame latents and using their independently estimated geometry only to address memory.

D Additional Qualitative and Video Results

We present additional qualitative results on DL3DV-Evaluation (Fig. 17, 18, 19), WorldScore-Static (Fig. 20, 21, 22), and the Long-Horizon dataset (Fig. 16), together with further results demonstrating the performance of our method across a broader range of data (Fig. 15). **Please refer to our supplementary project page: `https://anonymousgear46.github.io/geometry-as-address/` for richer and more dynamic visualizations.**



Figure 16: **Qualitative comparison of minute-long challenging camera trajectories results**

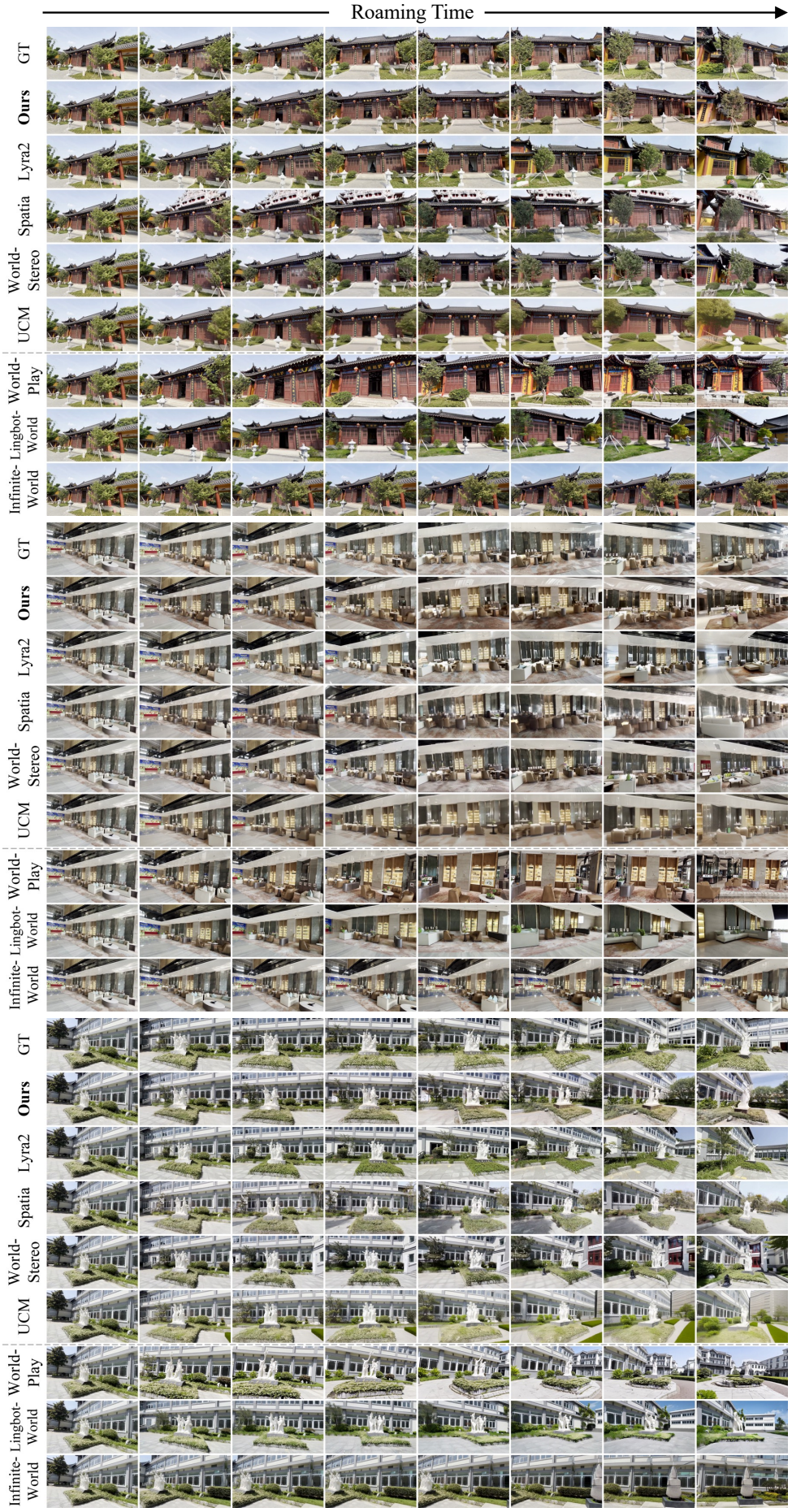

Figure 17: **Qualitative comparison of DL3DV-Evaluation results.**

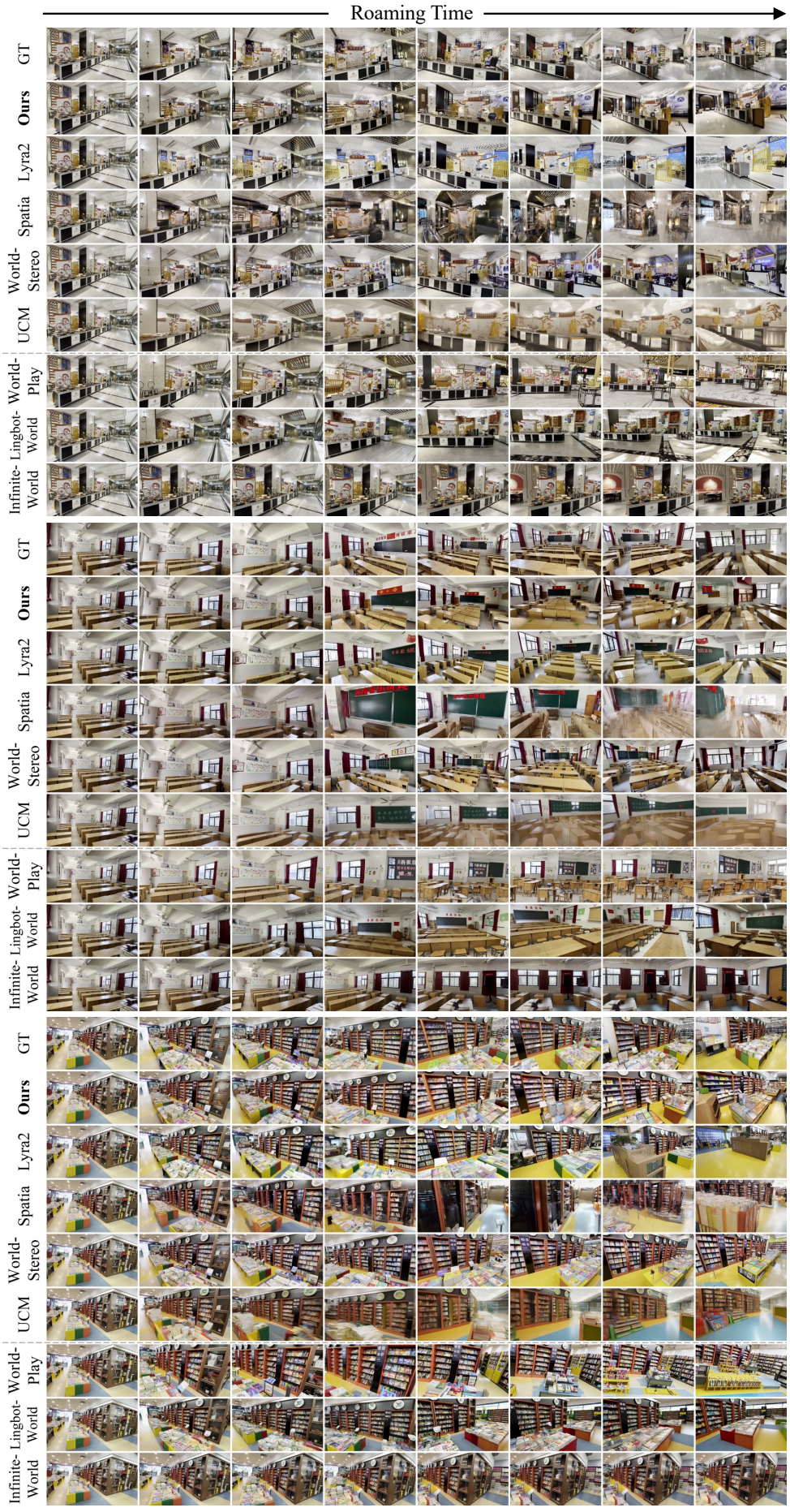

Figure 18: **Qualitative comparison of DL3DV-Evaluation results.**

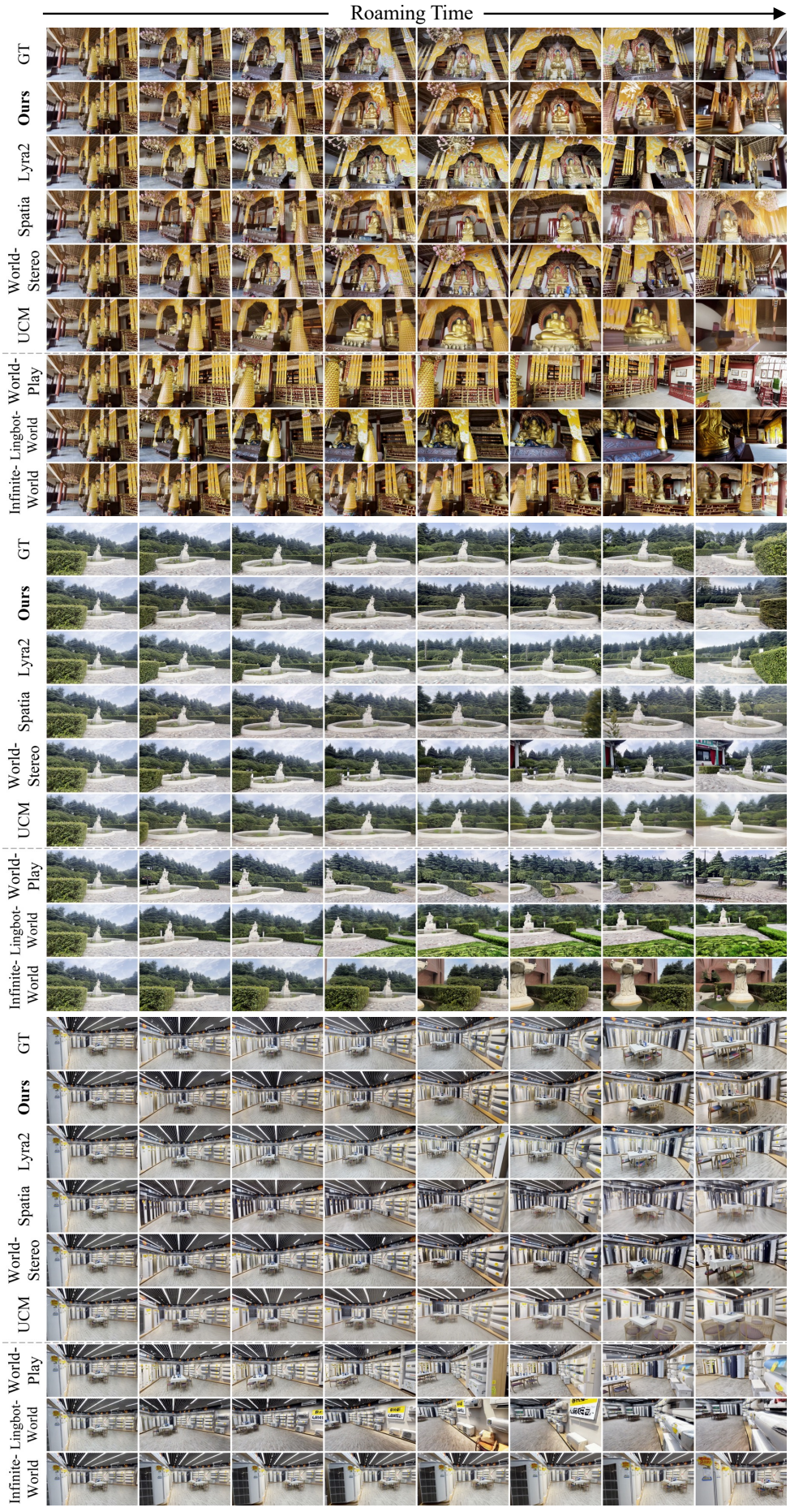

Figure 19: **Qualitative comparison of DL3DV-Evaluation results.**

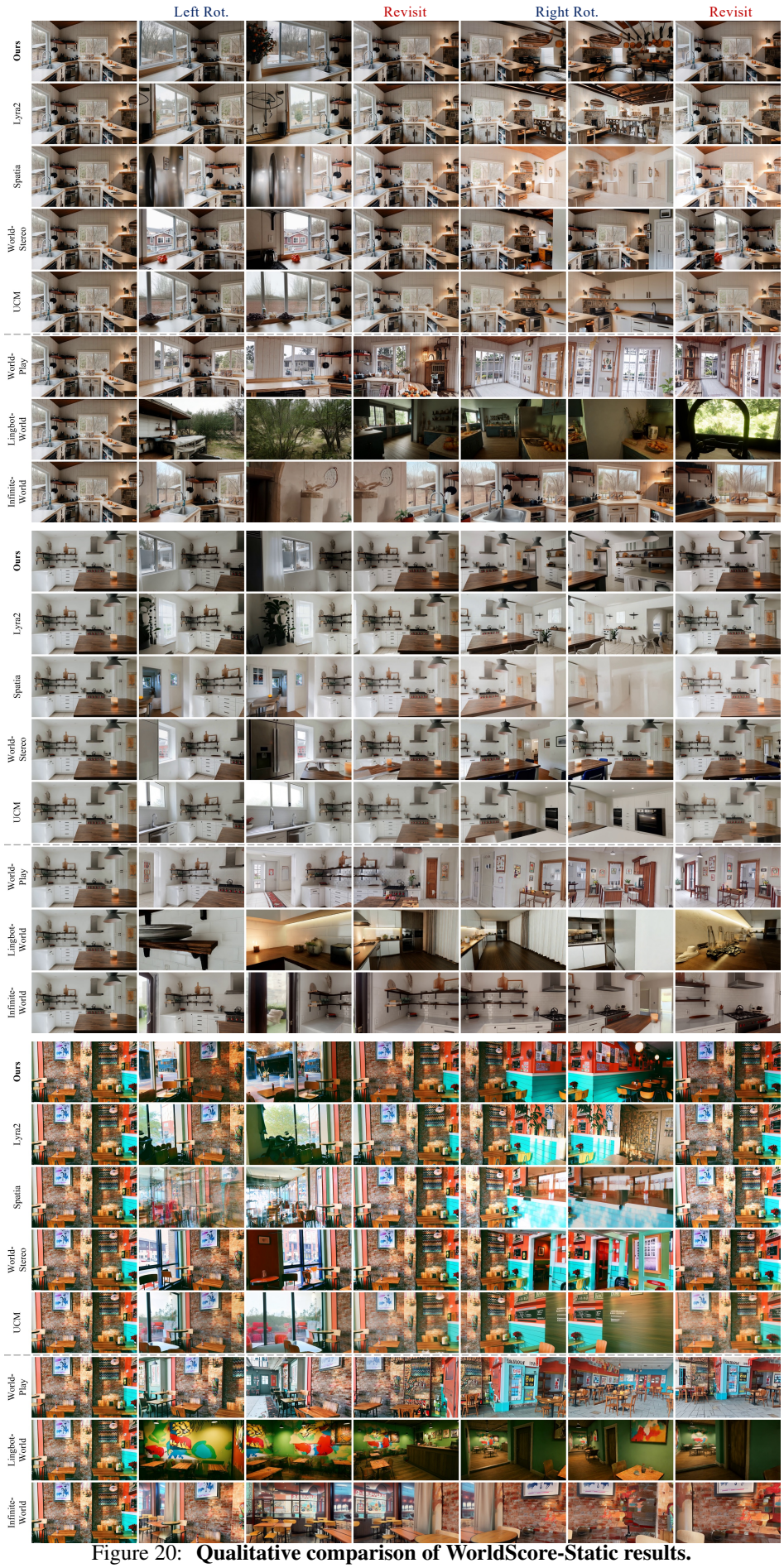

Figure 20: **Qualitative comparison of WorldScore-Static results.**

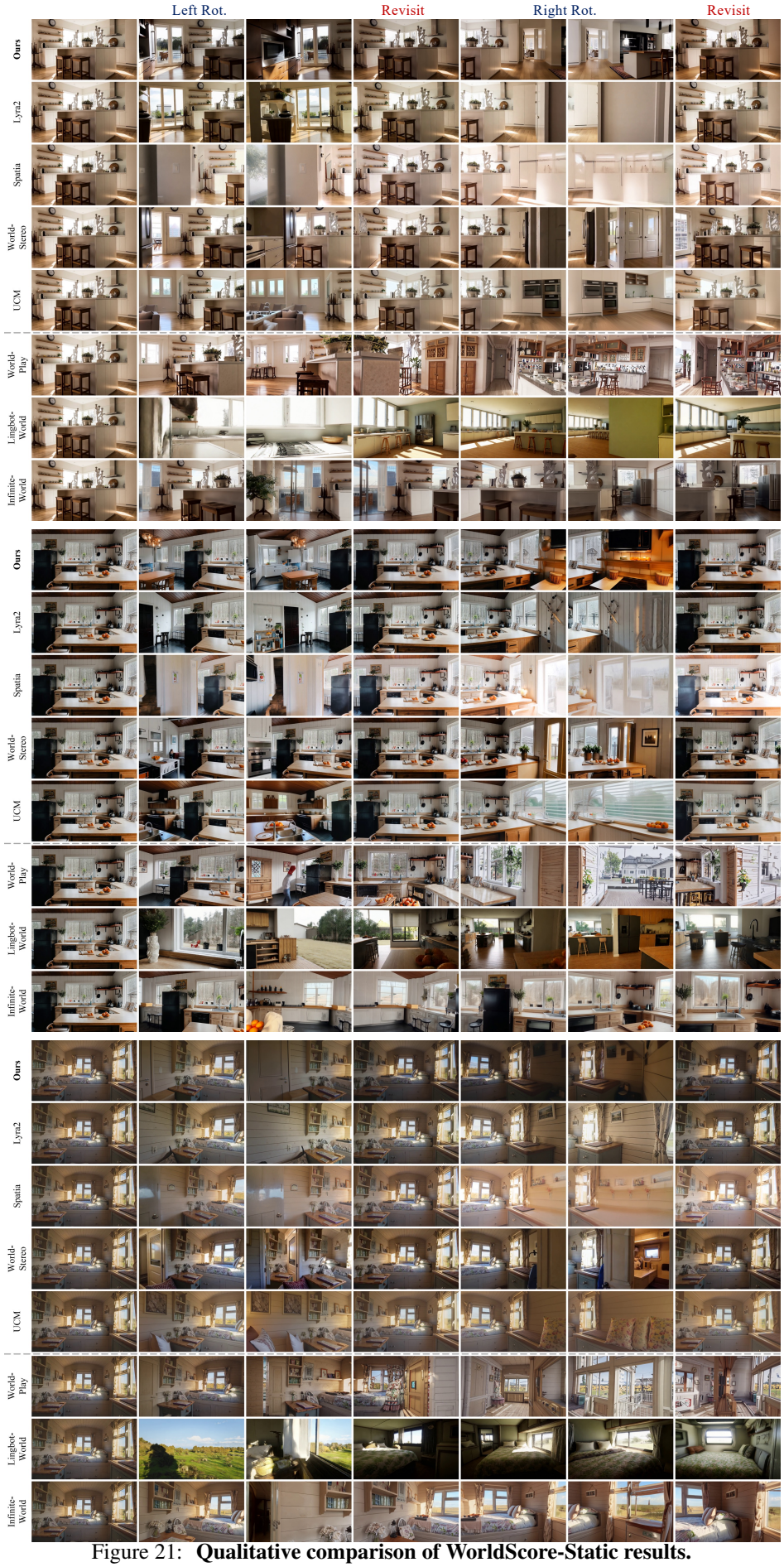

Figure 21: **Qualitative comparison of WorldScore-Static results.**

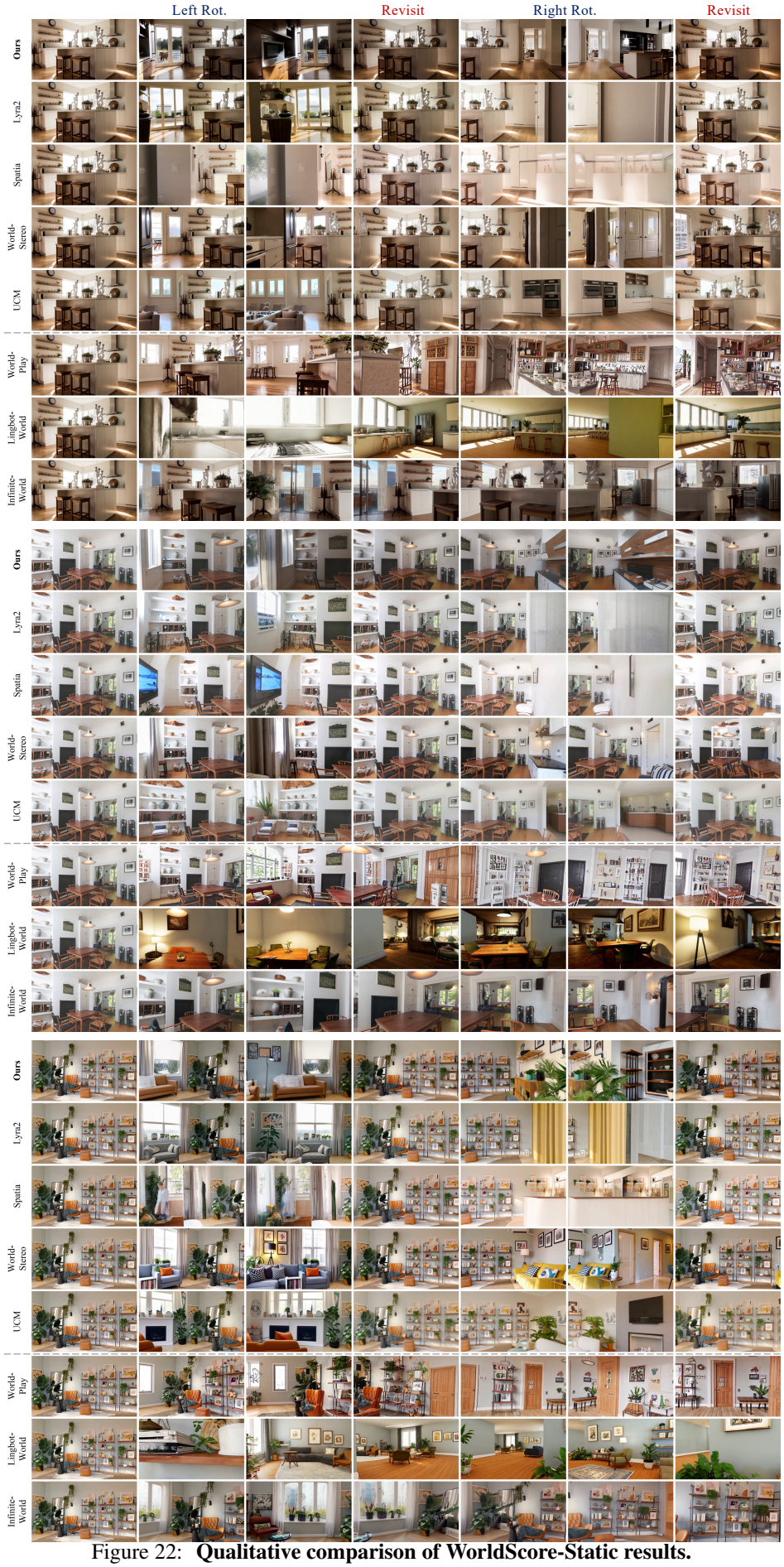

Figure 22: **Qualitative comparison of WorldScore-Static results.**