Grounding World Simulation Models in a Real-World Metropolis

Junyoung Seo*[1], Hyunwook Choi*[1], Minkyung Kwon[1], Jinhyeok Choi[1],
Siyoon Jin[1], Gayoung Lee[2], Junho Kim[2], JoungBin Lee[1], Geonmo Gu[2],
Dongyoon Han[2,1], Sangdoo Yun[2,3], Seungryong Kim†[1], and Jin-Hwa Kim†[2,3]

[1]KAIST AI [2]NAVER AI Lab [3]SNU AIIS
https://seoul-world-model.github.io

Abstract. What if a world simulation model could render not an imagined environment but a city that actually exists? Prior generative world models synthesize visually plausible yet artificial environments by imagining all content. We present **Seoul World Model (SWM)**, a city-scale world model grounded in the real city of Seoul. SWM anchors autoregressive video generation through retrieval-augmented conditioning on nearby street-view images. However, this design introduces several challenges, including temporal misalignment between retrieved references and the dynamic target scene, limited trajectory diversity and data sparsity from vehicle-mounted captures at sparse intervals. We address these challenges through cross-temporal pairing, a large-scale synthetic dataset enabling diverse camera trajectories, and a view interpolation pipeline that synthesizes coherent training videos from sparse street-view images. We further introduce a Virtual Lookahead Sink to stabilize long-horizon generation by continuously re-grounding each chunk to a retrieved image at a future location. We evaluate SWM against recent video world models across three cities: Seoul, Busan, and Ann Arbor. SWM outperforms existing methods in generating spatially faithful, temporally consistent, long-horizon videos grounded in actual urban environments over trajectories reaching hundreds of meters, while supporting diverse camera movements and text-prompted scenario variations.

1 Introduction

World models aim to learn internal representations of environments and predict their future states [12]. With recent advances in video generation, such models have rapidly evolved toward video world simulation, where sequences of frames are generated conditioned on images, text prompts, and user actions, treating each frame as a predicted state of a simulated world [1,7,8,13,20,24,32,46,63,64]. These models can generate dynamic and interactive environments, including object motion, weather changes, and physical interactions. Yet they operate entirely within imagined worlds: given a starting image, everything beyond it, *e.g.*, the geometry of unseen streets, distant buildings, is imagined by the model.

*Equal contribution †Co-corresponding authors

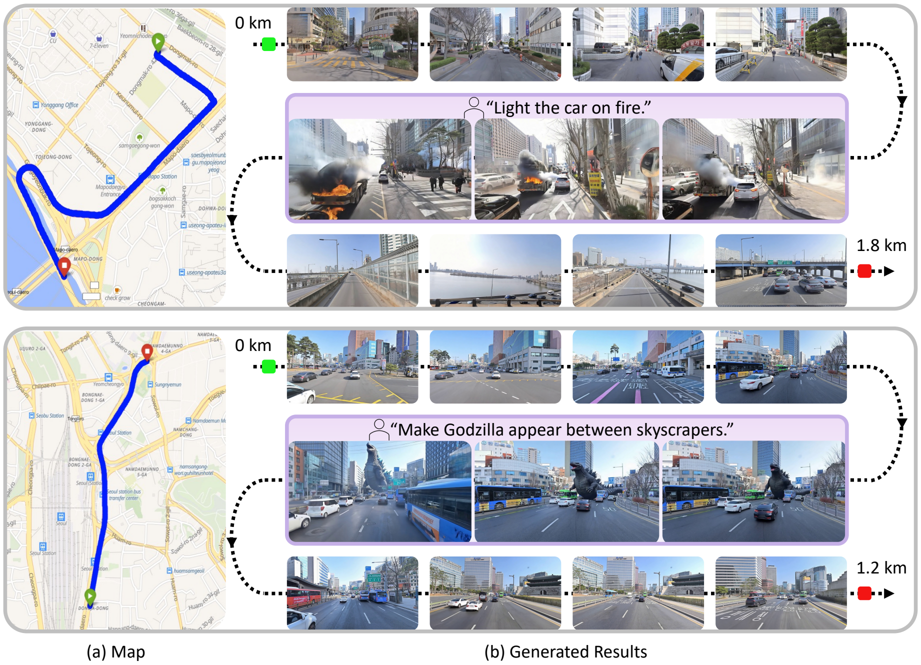

Fig. 1: Seoul World Model (SWM) generates videos over a kilometer grounded in a real city. A camera trajectory placed on a map produces continuous dynamic video depicting actual surroundings along the route. Users can further reshape the scene through text prompts, enabling imaginative scenarios.

What if a world model could operate on a world that physically exists? Users could navigate familiar city streets and experience hypothetical scenarios, such as a massive wave engulfing one's own city, or exploring familiar streets under a golden sunset. In addition, such a real-world grounded simulation would enable urban planning visualization, autonomous driving scenario generation, and location-based exploration [9, 17, 39]. Yet this direction remains unexplored: while large-scale 3D reconstruction systems model real cities [29, 44], they are fundamentally static and lack generative simulation capabilities, and no world simulation model has been grounded in a specific real-world location.

We formalize this goal as **real-world grounded video world simulation** and instantiate it in Seoul, a large and densely structured metropolis, introducing **Seoul World Model (SWM)**. Our key observation is that widely available street-view photographs provide a scalable source of location-specific visual references. SWM fine-tunes a pretrained video world simulation model [1] on 440k Seoul street-view images, real-world driving videos [43], and synthetic urban data. During generation, SWM performs retrieval-augmented generation: given geographic coordinates, camera actions, and text prompts, it retrieves nearby street-view images and conditions generation on complementary geometric and appearance references. This anchors each generated chunk to the real geometric layout and appearance of the location. Fig. 1 shows an example trajectory generated in Seoul with frames mapped to their corresponding city locations.

While retrieval-augmented grounding provides a natural way to anchor generation to real-world locations, it introduces three key challenges, each addressed by a corresponding design choice:

(1) Temporal misalignment. Street-view images capture a specific moment, while the simulated world should remain dynamic. Retrieved references may therefore contain transient elements inconsistent with the generated scene. We address this with **cross-temporal pairing**, which pairs references and targets from different timestamps during training, encouraging the model to disentangle persistent structure from transient content.

(2) Limited trajectory coverage and temporal sparsity. Real street-view data is captured by vehicle-mounted cameras at sparse intervals, restricting both trajectory types and temporal continuity. We construct a synthetic urban dataset using an Unreal-Engine-based simulator [10] that provides paired street-view references and target videos with diverse camera trajectories, including pedestrian paths. We additionally develop a view interpolation pipeline, namely an **intermittent freeze-frame** strategy, that synthesizes temporally coherent video between sparse street-view keyframes.

(3) Long-horizon error accumulation. Over long trajectories, autoregressive generation accumulates drift that weakens spatial grounding. Prior methods mitigate this with an attention sink, a fixed global context frame, typically the first frame, that persists throughout generation [28, 40]. However, this static anchor becomes less informative as the camera moves away from the starting locations. We instead propose a **virtual lookahead sink**: at each generation chunk, we retrieve a nearby street-view image and insert it at a future temporal position, acting as a virtual destination that re-anchors generation to upcoming locations, inspired by recent talking-head methods [21, 38].

SWM demonstrates that world simulation can be faithfully grounded in real, physically existing environments at city scale. We evaluate SWM across three cities: Seoul, Busan, and Ann Arbor, where the latter two cities are entirely absent from training, testing cross-city generalization without any fine-tuning. SWM outperforms recent video world models in visual quality, camera adherence, temporal coherence, and structural fidelity to real locations, and maintains stable generation over trajectories reaching hundreds of meters, demonstrating text-prompted scenarios and diverse camera trajectories.

2 Related Work

2.1 Video Generative Models

Video generation has advanced rapidly with diffusion models [16,35,41], enabling high-fidelity video synthesis. Early video diffusion models typically used UNet backbones with temporal modules [3,4,15], while more recent work [22,30,49,56] has shifted toward Diffusion Transformers [11, 33] for improved scalability and quality. Recently, long-horizon video generation has emerged as a central target, motivating autoregressive and streaming formulations, where the model rolls out videos chunk-by-chunk while conditioning each new chunk on the generated context [6, 18, 28, 40, 57, 58]. As generation extends, however, these models increasingly suffer from exposure bias and error accumulation. To address this, several methods [28, 40, 55, 57] preserve long-range information with persistent global anchors such as attention sinks [54], which keep a fixed set of tokens and improve long-range temporal consistency without attending to the full history.

2.2 Video World Models

Building on the progress in video generation, video world simulation models [1, 7, 8, 13, 20, 23–25, 32, 36, 45, 46, 48, 53, 59, 63, 64] use a generative model as a dynamic model. Conditioned on past observations and actions, they predict future observations to simulate how the environment evolves [1, 20]. Recent models generate interactive visual observations conditioned on user actions across diverse settings, including game environments [13, 45, 48], autonomous driving [23], and open-domain scenarios [20, 32, 46, 64]. Action representations vary from discrete keyboard and mouse inputs [13, 45] to continuous camera trajectories [8, 23, 53, 59, 63, 64] and natural language instructions [32]. To maintain coherent world states over extended interactions, recent methods incorporate persistent memory beyond the local context window [7, 25, 53, 59]. Despite these advances, existing world models operate entirely within imagined or synthetic environments, generating futures without grounding in external real-world observations. This becomes a key limitation when the simulated environment must stay faithful to a specific physical location.

2.3 Geometry-Aware Video Generation

A separate line of work incorporates 3D geometric reasoning into video generation to improve spatial consistency. In novel view synthesis, recent methods render point clouds from predicted depth to achieve geometric consistency for single-scene reconstruction [34, 60]. Recent world models integrate geometry into autoregressive generation through joint video-3D prediction [7, 8, 63], 3D scene representations maintained across generation [24, 50], and memory or spatial retrieval mechanisms that reuse previously generated context [7, 25, 36, 53, 59]. These approaches build geometric representations from the model's predictions or generation history, and are typically focused on nearly static settings [25, 34, 50, 60].

3 Data Construction

For SWM training, we build aligned pairs between street-view references and target video sequences. Each reference is associated with its camera pose and depth map, providing geometric conditions that ground the generated video to real-world geometric structure. We construct these pairs from two primary sources:

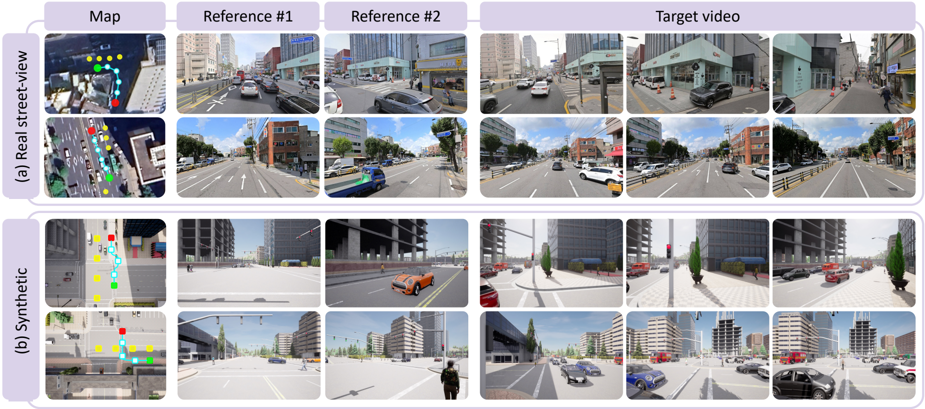

Fig. 2: Data overview: (a) real street-view and (b) synthetic datasets. The satellite-view map (leftmost column) shows the target video trajectory, with green and red indicating the start and end points, and yellow indicating reference view locations selected via cross-temporal pairing.

real street-view images captured in Seoul (Sec. 3.1) and synthetic urban data from an Unreal-Engine-based simulator (Sec. 3.2). We additionally incorporate a publicly available driving video dataset [43] to increase scenario diversity. Fig. 2 shows examples from the real and synthetic datasets.

3.1 Street-View Dataset

Collection. We collect 1.2M panoramic images covering major urban areas of Seoul. Each image is associated with GPS coordinates and capture timestamps as metadata, obtained from NAVER Map. License plates and pedestrians are blurred for de-identification. After the processing steps below, 440K images are used for training.

Cross-temporal pairing. We define a training sequence as N consecutive street-view images along a route, which serves as the target sequence for supervision, and assign K spatially nearby panoramas as references that condition generation (Sec. 4.1). Each panorama is rendered into a pinhole view: training sequences are rendered facing the forward driving direction with a random yaw rotation within ±90°, while references are rendered to match the viewing direction of the paired training frame.

A key design choice is **cross-temporal pairing**: references must be captured at a different timestamp from the target sequence. This mirrors inference, where retrieved street-view images come from locations near the target but often differ in transient content such as vehicles or pedestrians. Without this constraint, co-captured references share identical transient content with the target, making it difficult to distinguish persistent structures from transient objects; the model has no incentive to separate them and learns to reproduce both. Cross-temporal pairing removes this ambiguity during training: because transient content differs between reference and target, the model must learn to rely on persistent spatial structure that remains consistent across timestamps. Fig. 2(a) shows representative cross-temporal pairs; Fig. 6 visualizes the resulting attention pattern.

View interpolation. City-scale street-view databases provide panoramic images at sparse spatial intervals (typically 5–20 m between views) rather than continuous video. Training a video generation model directly on such sparse sequences is challenging, as pretrained video diffusion models learn to produce temporally smooth, continuous motion; abrupt jumps between distant viewpoints break this temporal continuity. We therefore develop an interpolation pipeline that synthesizes T-frame videos from N sparse keyframes ($T \gg N$), leveraging a pretrained latent video generative model [1].

A straightforward way to enable keyframe interpolation is to concatenate the keyframe latents along the channel dimension of the latents at their corresponding timestamps, while zero-padding the conditioning channels at non-keyframe timestamps, as shown in Fig. 3(a) (*e.g.* Wan2.1-FLF2V [49]). However, we observe that this approach yields weak adherence to the keyframes, with generated frames deviating from the inputs. We attribute this to a mismatch with the video 3D VAE's temporal compression: the encoder compresses every 4 consecutive frames into a single latent, whereas an isolated keyframe does not form a valid 4-frame group.

To address this, we propose an **intermittent freeze-frame** strategy that ensures each keyframe forms a complete 4-frame group matching the 3D VAE's temporal stride. During training, the pixel frame at each keyframe position is repeated 4 consecutive times, so the 3D VAE encodes it into exactly one latent; the resulting training videos alternate between smooth motion and brief freeze segments. At inference, each given keyframe is similarly repeated 4 times and encoded into a single latent, which then replaces the latent at the corresponding position in the noisy input latent of the diffusion model. After generation and decoding, the three repeated frames per keyframe are discarded to recover the intended video, as illustrated in Fig. 3(b). Quantitative results are in Appendix C.1.

Annotation. We generate text captions for all videos using Qwen2.5-VL-72B [2] and augment them with predefined camera actions (straight, stop, left turn, right turn). While GPS metadata provides approximate positions, it lacks sufficient accuracy and does not include camera pose information. We use Depth Anything V3 [27] to estimate per-keyframe depth maps and camera poses, and align them to real-world scale using GPS metadata. Details are provided in Appendix A.2.

3.2 Synthetic Dataset

To complement the driving-like trajectories of real street-view data with diverse camera paths, we construct a synthetic dataset from CARLA [10], an Unreal Engine-based urban simulator. We render 12.7K videos from 6 urban maps spanning approximately 431,500 m^2 of city area across three trajectory types:

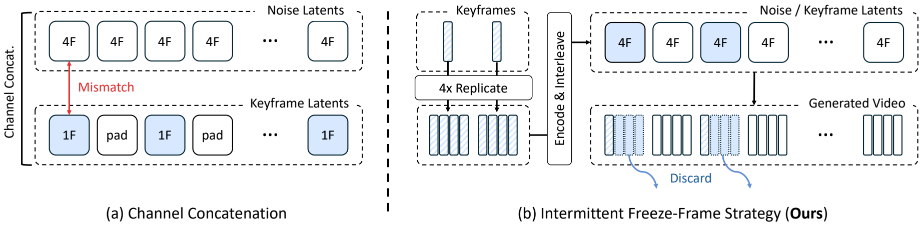

Fig. 3: View interpolation pipeline: (a) Keyframe conditioning via channel concatenation, and (b) Keyframe conditioning with intermittent freeze-frame strategy (**Ours**). 4F and 1F denote latents derived from four frames and one frame, respectively, prior to the 4× temporal compression of the 3D VAE.

1. **Pedestrian trajectories**: first-person videos rendered from autonomous pedestrian agents, covering sidewalk movement, street crossing, and similar on-foot paths.
2. **Vehicle trajectories**: driving-perspective videos captured across diverse road types, including highways, urban streets, and elevated roads. The trajectories cover lane changes, turns, and straight driving.
3. **Free-camera trajectories**: random paths that freely navigate the scene while avoiding collisions with buildings, terrain, and other scene geometry.

Street-view reference. For each map, we render street-view reference images at regular intervals of 10 m along all roads, with eight directional views (uniformly covering 360° horizontal view) for each location. Following the same cross-temporal pairing principle as the real data, reference images and target video sequences are rendered at different simulated timestamps. Examples are shown in Fig. 2(b); additional details are in Appendix A.3.

4 Model

SWM generates videos grounded in a real city through retrieval-augmented conditioning from a user-specified starting location, camera motion, and a text prompt. We build on a pretrained Diffusion Transformer (DiT) [1,33] that operates in a latent space compressed from pixel-space T frames $\mathbf{X} = \{x_t\}_{t=0}^{T-1}$ via a 3D VAE. Generation proceeds autoregressively in chunks with frame length T. For the i-th chunk, the model receives a camera trajectory $\mathbf{C}^{(i)} = \{c_t\}_{t=0}^{T-1}$, a text prompt $P^{(i)}$, and noisy latents $\mathbf{Z}^{(i)} = \{z_l^{(i)}\}_{l=0}^{L-1}$ to produce target latents $\hat{\mathbf{Z}}^{(i)} = \{\hat{z}_l^{(i)}\}_{l=0}^{L-1}$, where L is the number of compressed latents per chunk. Each subsequent chunk additionally conditions on H history latents $\mathbf{Z}_{\text{hist}}^{(i)} = \{\hat{z}_l^{(i-1)}\}_{l=L-H}^{L-1}$ from the tail of the preceding chunk's output, providing temporal continuity.

For each chunk, nearby street-view images are retrieved from a geo-indexed database (Sec. 4.1). These retrieved images serve two roles: as a virtual lookahead sink (Sec. 4.2) that prevents error accumulation in city-scale long-horizon generation, and as conditioning for geometric and semantic referencing (Sec. 4.3) that grounds the generated video to the geometry and appearance of real locations.

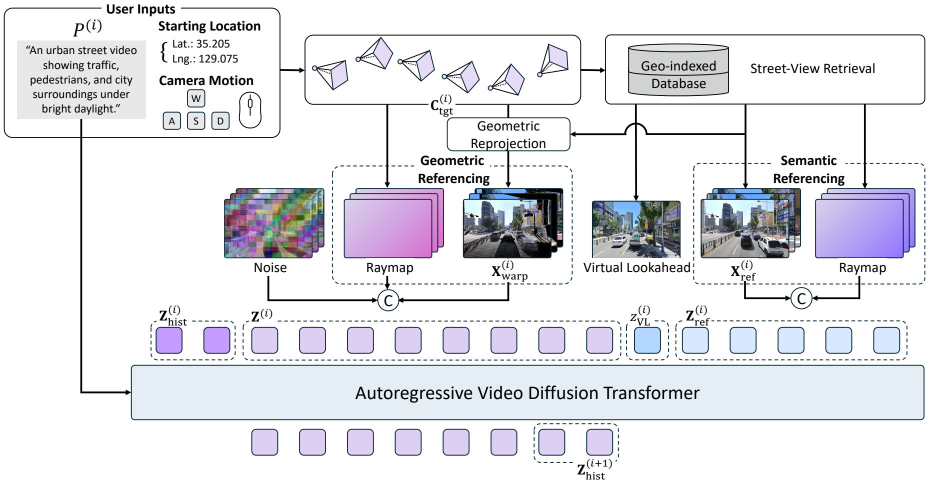
User Inputs
$P^{(i)}$
"An urban street video showing traffic, pedestrians, and city surroundings under bright daylight."
Starting Location
Lat.: 35.205
Lng.: 129.075
Camera Motion
W
A
S
D
$\mathbf{C}_{\text{tgt}}^{(i)}$
Geo-indexed Database
Street-View Retrieval
Geometric Reprojection
Geometric Referencing
Semantic Referencing
Noise
Raymap
$\mathbf{X}_{\text{warp}}^{(i)}$
Virtual Lookahead
$\mathbf{X}_{\text{ref}}^{(i)}$
Raymap
C
C
$\mathbf{Z}_{\text{hist}}^{(i)}$
$\mathbf{Z}^{(i)}$
$z_{\text{VL}}^{(i)}$
$\mathbf{Z}_{\text{ref}}^{(i)}$
Autoregressive Video Diffusion Transformer
$\mathbf{Z}_{\text{hist}}^{(i+1)}$

Fig. 4: Model overview. Given a start location, SWM autoregressively generates video grounded in a real city based on text prompt $P^{(i)}$, and target camera trajectory $\mathbf{C}^{(i)}$, retrieving the relevant street-view images from a geo-indexed database. These retrieved images provide a Virtual Lookahead Sink for long-horizon stability and serve as conditioning for geometric referencing and semantic referencing to ground the generation in real-world geometry and appearance. The model generates each chunk autoregressively conditioned on self-generated history.

Since our retrieval-augmented framework is orthogonal to the training strategy for autoregressive generation, we evaluate it under Teacher Forcing [51] and Self-Forcing [18] as two separate configurations. Fig. 4 provides an architectural overview.

4.1 Street-View Retrieval

The retrieval database consists of 1.2M panoramic images covering Seoul. Each panorama is rendered into 8 equi-angular pinhole views, with metric-scale depth maps and 6-DoF camera poses estimated via Depth Anything V3 [27], and aligned to real-world scale using GPS metadata, following the same preprocessing as the training data (Sec. 3.1). For each i-th target chunk, given target camera trajectory $\mathbf{C}^{(i)}$, we retrieve reference images in two stages: (1) nearest-neighbor search identifies candidate street-view locations along the target trajectory, and (2) depth-based reprojection filtering retains only those whose projected pixels exceed a coverage threshold in the nearest target view. This yields up to K pinhole references $\mathbf{X}_{\text{ref}}^{(i)} = \{x_{\text{ref},k}^{(i)}\}_{k=0}^{K-1}$ with their camera poses $\mathbf{C}_{\text{ref}}^{(i)} = \{c_{\text{ref},k}^{(i)}\}_{k=0}^{K-1}$ and depth estimates $\mathbf{D}_{\text{ref}}^{(i)} = \{d_{\text{ref},k}^{(i)}\}_{k=0}^{K-1}$, each aligned to the viewing direction of the matched target viewpoint.

4.2 Virtual Lookahead Sink

Autoregressive generation accumulates errors across chunks, as each step feeds the last few output latents as history latents to generate the next chunk. At the

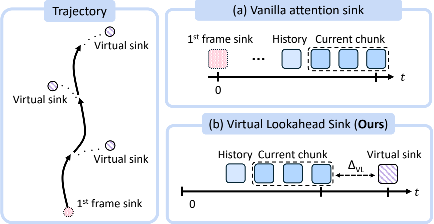

Fig. 5: Virtual Lookahead Sink: (a) Vanilla attention sink, (b) virtual lookahead sink (**Ours**). (a) anchors to the initial frame, whose guidance weakens as the camera moves farther away. (b) dynamically retrieves the nearest street-view as a virtual future destination.

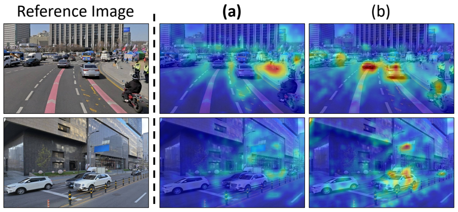

Fig. 6: Attention scores on references. (a) Ours with cross-temporal pairing, and (b) Ours without cross-temporal pairing. Cross-temporal pairing makes the model less attentive to dynamic objects (*e.g.*, cars) at the reference.

city scale, where the camera may travel hundreds of meters, per-chunk drift compounds into misalignment between retrieved references and the generated scene. We observe that world models trained with forcing-based distillation [6,18] still degrade under these conditions. Prior work mitigates long-horizon degradation by maintaining an attention sink (Fig. 5(a)), typically the initial frame, as a fixed global context throughout generation [28,40]. However, this static anchor becomes increasingly irrelevant as the camera moves farther from the starting point in our scenario.

To address this, we propose a **virtual lookahead sink**, tailored for retrieval-augmented long-horizon generation, dynamically updating the sink with a retrieved street-view image. Specifically, given the target trajectory end point $c^{(i)}_{T-1}$ of each chunk, we retrieve the nearest street-view image to this endpoint and treat it as a virtual future destination, placed with a sufficient temporal gap from the current chunk. By placing a clean, error-free frame ahead of the chunk being generated, the model has a stable anchor to converge toward; retrieving this anchor from a spatially nearby location further ensures that the grounding remains relevant to the region being generated. Because the anchor is not a reconstruction target, it need not coincide with the exact future trajectory; each chunk refreshes it during generation. Fig. 5(b) illustrates this mechanism.

We encode the retrieved image into a single latent $z^{(i)}_{\mathrm{VL}}$ and assign it a RoPE [42] temporal position embedding beyond the current generation chunk. The latent sequence fed to the model $\mathbf{Z}^{(i)}_{\mathrm{seq}}$ and its RoPE temporal positions $\mathbf{p}^{(i)}_{\mathrm{seq}}$ are given by:

$$\begin{aligned} \mathbf{Z}^{(i)}_{\mathrm{seq}} &= \left[\mathbf{Z}^{(i)}_{\mathrm{hist}};\ \mathbf{Z}^{(i)};\ z^{(i)}_{\mathrm{VL}}\right], \quad \text{and} \\ \mathbf{p}^{(i)}_{\mathrm{seq}} &= \Big[\underbrace{1,\dots,H}_{\text{history}};\ \underbrace{H{+}1,\dots,H{+}L}_{\text{target}};\ \underbrace{H{+}L{+}\Delta_{\mathrm{VL}}}_{\text{sink}}\Big], \end{aligned} \tag{1}$$

where Δ_{VL} is a temporal offset hyperparameter. During training, a ground-truth future frame is sampled at a random temporal offset, exposing the model to varying lookahead distances so that it learns how the anchor's proximity affects generation [38]; at inference, Δ_{VL} is fixed and the ground-truth frame is replaced by a retrieved street-view image.

4.3 Geometric and Semantic Referencing

Since each retrieved reference and the target observe the same underlying scene from known camera poses, their geometric relationship enables two forms of conditioning. Geometric warping reprojects a reference into the target viewpoint, providing dense spatial layout cues, but loses fine appearance detail due to depth errors and occlusion [24, 34, 37, 53]. Conversely, injecting the original reference preserves appearance but lacks explicit spatial alignment. We therefore condition generation through two complementary pathways: geometric referencing for spatial layout and semantic referencing for appearance detail. Given K retrieved street-view images $\mathbf{X}_{\mathrm{ref}}^{(i)}$ with poses $\mathbf{C}_{\mathrm{ref}}^{(i)}$ and depth estimates $\mathbf{D}_{\mathrm{ref}}^{(i)}$, the two pathways operate as follows.

Geometric referencing. For each target frame $x_t^{(i)}$ to be generated, we reproject the spatially nearest reference $x_{\mathrm{ref},j}^{(i)}$ into the target viewpoint via depth-based forward splatting [34], to get warped target image $x_{\mathrm{warp},t}^{(i)}$:

$$x_{\mathrm{warp},t}^{(i)} = \mathrm{Render}\big(\mathrm{Unproj}(x_{\mathrm{ref},j}^{(i)},\ d_{\mathrm{ref},j}^{(i)}),\ c_{\mathrm{ref},j\to t}^{(i)}\big), \tag{2}$$

where $c_{\mathrm{ref},j\to t}^{(i)}$ is the relative camera transformation, $\mathrm{Unproj}(\cdot)$ lifts the reference image to 3D using depth, and $\mathrm{Render}(\cdot)$ projects the 3D points into the target view. Each target frame uses only its single nearest reference to avoid the noisy artifacts that arise when multiple images are simultaneously splatted into the same view. The warped video $\mathbf{X}_{\mathrm{warp}}^{(i)} = \{x_{\mathrm{warp},t}^{(i)}\}_{t=0}^{T-1}$ is encoded by the 3D VAE and channel-wise concatenated with the noisy target latent at the DiT input.

Semantic referencing. To preserve appearance detail, the original references are injected into the transformer's latent sequence. Each $x_{\mathrm{ref},k}^{(i)}$ is encoded into a single latent $z_{\mathrm{ref},k}^{(i)}$, patch-embedded, and concatenated with the target latents along the temporal axis at RoPE position $p_{\mathrm{ref},k}^{(i)} = H{+}L{+}G + k\Delta_{\mathrm{ref}}$, where G is a large temporal gap separating references from the generation window and Δ_{ref} is the inter-reference spacing. Unlike geometric referencing, which uses only the nearest reference per frame, semantic referencing allows each target latent to attend to all K references, enabling the model to gather complementary appearance cues. Camera poses for all latents, including target, reference, and sink, are encoded via Plücker ray embeddings, projected into the latent space through a convolutional encoder, and concatenated with the latent channels.

Since references and targets are captured at different times, dynamic objects in the references need not match those in the generated frame. Instead of introducing explicit mechanisms for this, we leverage the cross-temporal pairing strategy (Sec. 3.1), which encourages the model to focus on persistent scene structure while ignoring transient objects, as shown in Fig. 6.

5 Experiments

5.1 Implementation Details

Model and training setup. SWM fine-tunes Cosmos-Predict2.5-2B [1]. We train with AdamW [31] with a learning rate of 4.8e−5 for 10K iterations at a total batch size of 48 across 24 NVIDIA H100 GPUs. For the Self-Forcing (SF) variant [18], we first perform ODE initialization with 1K pairs for 6K steps from the Teacher-Forcing (TF) model checkpoint, followed by 10K iterations of fine-tuning. Under TF, each chunk (T=77 frames) conditions on H=5 ground-truth history latents during training with K=5 references and G=50 gap; at inference, the history is replaced by self-generated output. Under SF, the model generates from self-produced history (H=3) in KV cache, with shorter chunks (12 frames) and K=1, achieving 15.2 fps with a single H100 GPU. Both configurations apply the Virtual Lookahead Sink with $\varDelta_{\mathrm{VL}}$=5. Further details are in Appendix A.1.

Evaluation benchmarks. For evaluation, we construct two benchmark datasets, Busan-City-Bench and Ann-Arbor-City-Bench (from the MARS dataset [26]), since our model is trained on Seoul data. Each benchmark contains 30 test sequences, each consisting of 365 frames (approximately 100 meters each). References are retrieved from nearby locations but exclude any street-view image belonging to the test sequence itself, ensuring that the model cannot access the ground-truth viewpoint during generation.

Baselines. We introduce a new task, real-world grounded world simulation, a setting that requires inputs not fully supported by existing world models. We therefore evaluate recent video world models capable of generating dynamic environments by providing each baseline with its supported inputs; per-baseline adaptation details are in Appendix B.2. The baselines include Aether [63], DeepVerse [7], Yume1.5 [32], HY-World1.5 [20], FantasyWorld [8], and Lingbot [46]. We report qualitative and quantitative comparisons in Fig. 8 and Tab. 1.

Metrics. We evaluate generation quality from three aspects. **Visual and temporal quality** is measured with FID [14] and FVD [47], and we report Image Quality from VBench [19]. **Camera-following accuracy** is measured with Rotation Error (RotErr) and Translation Error (TransErr), which quantify how accurately the generated camera motion follows the target trajectory. **3D adherence** is evaluated with masked PSNR and LPIPS [61] computed only on static regions, since the generated dynamics do not need to match the ground-truth, by applying SAM3 [5] to segment moving objects.

5.2 Results

Generation results. Fig. 7 demonstrates the capabilities of SWM across diverse scenarios, trajectories, and long-horizon generation in Seoul city. Fig. 7(a) shows that, despite grounding generation in real-world references, SWM remains

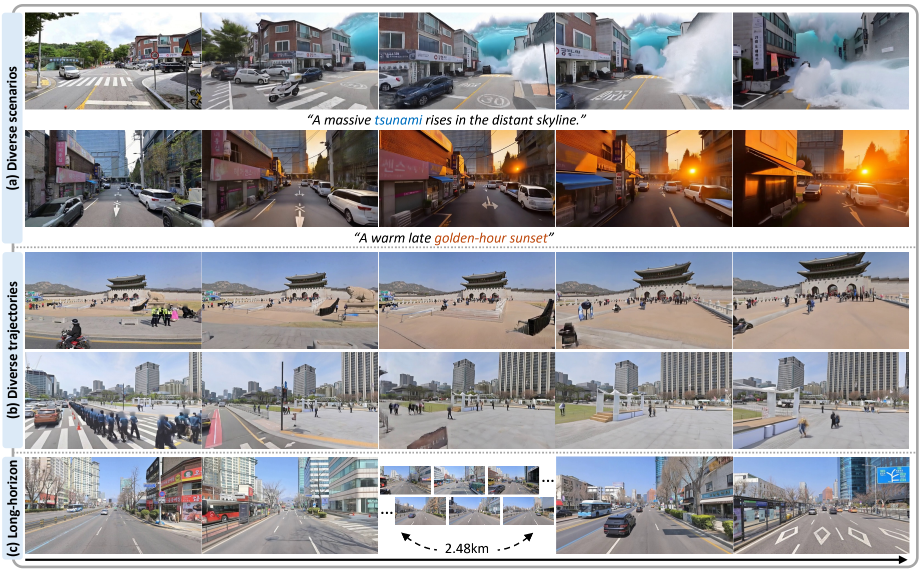
(a) Diverse scenarios
“A massive tsunami rises in the distant skyline.”
“A warm late golden-hour sunset”
(b) Diverse trajectories
(c) Long-horizon
...
...
2.48km

Fig. 7: Qualitative results. Grounded in a real-world city, SWM can (a) generate high-fidelity videos across diverse scenarios from user-guided text prompts, (b) remain robust to diverse camera trajectories, and (c) reduce error accumulation, enabling long-horizon generation over several kilometers.

controllable via text prompts and can produce diverse scene conditions while preserving the underlying city layout. Fig. 7(b) demonstrates that training with synthetic urban data enables SWM to follow trajectories beyond standard driving paths, including pedestrian-style camera motions. Fig. 7(c) shows stable long-horizon generation, maintaining spatial consistency without noticeable error accumulation.

Comparison with other models. As shown in Tab. 1, SWM achieves the best performance on both Busan-City-Bench and Ann-Arbor-City-Bench [26] across visual and temporal fidelity, camera-following accuracy, and 3D adherence to real locations. In contrast, existing world models often drift over long trajectories, leading to misalignment in both camera motion and scene structure and resulting in blurred videos, reduced motion, or complete collapse. By leveraging retrieved images, SWM remains anchored to real-world scene layout and preserves alignment with the target trajectory, demonstrating stronger real-world grounding, as illustrated in Fig. 8.

5.3 Ablation Study

Tab. 2 summarizes ablations on three components of SWM, including dataset construction, the referencing strategy and the attention sink design. Representative qualitative ablations of the conditioning strategies are shown in Fig. 9, with additional examples provided in Appendix C.2.

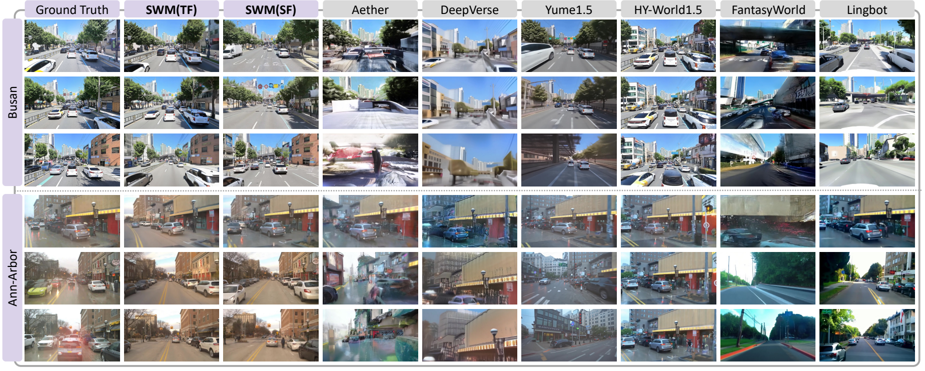

Fig. 8: Qualitative comparisons with other methods. Across Busan- and Ann-Arbor-City-Bench, SWM produces physically grounded videos consistent with real urban structure and camera motion. Other world models, not originally designed for real-world grounding, tend to struggle with scene coherence, trajectory alignment, or long-horizon stability in this setting.

Table 1: Quantitative comparison with other methods. We evaluate visual and temporal fidelity, camera-following accuracy, and 3D adherence. Values are reported as Busan-City-Bench / Ann-Arbor-City-Bench.

Method	FID↓	FVD↓	Img.Q.↑	RotErr↓	TransErr↓	mPSNR↑	mLPIPS↓
Aether [63]	141.24/132.77	1096.50/1214.84	0.55/0.51	0.030/0.078	0.083/0.192	11.10/13.03	0.671/0.635
DeepVerse [7]	130.32/182.95	892.63/1524.97	0.53/0.46	0.062/0.251	0.103/0.469	12.20/13.43	0.679/0.727
Yume1.5 [32]	54.82/85.62	425.24/993.62	0.73/0.61	0.153/0.326	0.104/0.271	12.09/14.15	0.667/0.623
HY-World1.5 [20]	49.63/67.02	544.04/864.76	**0.78**/0.54	0.044/0.193	0.079/0.221	11.87/14.26	0.588/0.575
FantasyWorld [8]	83.51/67.72	783.11/917.57	0.63/0.49	0.056/0.215	0.141/0.302	10.01/11.97	0.654/0.592
Lingbot [46]	62.14/57.99	717.44/1039.50	0.75/0.60	0.081/0.269	0.073/0.239	10.48/12.51	0.645/0.641
SWM (TF)	**28.43**/56.61	**301.76**/**640.17**	**0.78**/**0.66**	**0.020**/**0.055**	**0.015**/**0.154**	**14.56**/**15.18**	**0.392**/**0.481**
SWM (SF)	32.50/**43.97**	325.87/779.94	0.77/0.57	0.028/0.217	0.033/0.208	13.52/14.20	0.478/0.573

Effect of dataset construction. Among all variants, removing cross-temporal pairing leads to the largest degradation across metrics, indicating that the model fails to disregard dynamic objects that are mismatched between retrieved references and generated frames. Removing synthetic data slightly improves FID, but substantially harms camera-following accuracy and 3D adherence since the model no longer learns diverse trajectories during training.

Effect of referencing. Ablations on the conditioning pathways confirm that geometric and semantic referencing play complementary roles. Geometric referencing supports camera alignment and static structural consistency, while semantic referencing improves appearance fidelity by injecting visual details. Removing either pathway degrades overall quality, yielding suboptimal results.

Effect of attention sink. We compare four configurations: (1) our full model with the Virtual Lookahead (VL) Sink, (2) without any attention sink, (3) with a conventional First-Frame (FF) attention sink using the first frame as an attention sink, and (4) with a First-Position (FP) Sink that places a retrieved image at the

Table 2: Ablation on data strategy, conditioning strategy, and attention sink design. We evaluate visual and temporal fidelity, camera-following accuracy, and 3D adherence. Values are reported as Busan-City-Bench.

Variant	FID↓	FVD↓	Img.Q.↑	RotErr↓	TransErr↓	mPSNR↑	mLPIPS↓
Full model	28.43	**301.76**	0.78	0.020	**0.015**	**14.56**	0.392
w/o cross-temporal pairing	44.74	487.87	0.77	0.057	0.123	12.54	0.519
w/o synthetic data	**27.74**	365.24	0.78	0.021	0.020	13.52	0.427
w/o real street-view data	29.82	467.58	0.77	0.059	0.050	13.99	0.411
w/o geometric referencing	33.01	398.74	**0.79**	0.036	0.051	12.33	0.525
w/o semantic referencing	30.27	326.18	0.78	0.032	0.022	14.08	0.442
w/o any attention sink	33.06	342.81	0.78	0.021	0.016	14.16	0.406
w/ first frame attention sink	32.71	378.92	0.78	**0.018**	0.018	14.25	0.388
w/ first position attention sink	32.41	354.61	0.78	0.026	0.027	14.35	**0.379**

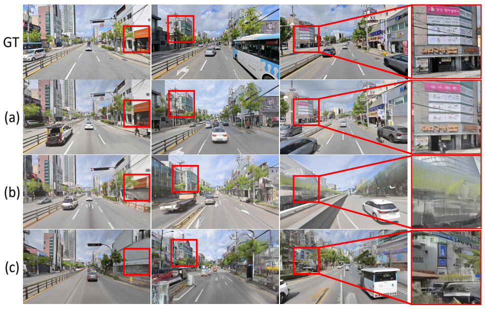

Fig. 9: Qualitative ablation results. (a) Our full model, (b) ours without VL sink, and (c) ours without semantic referencing. VL Sink prevents error accumulation over long trajectories, while semantic referencing preserves appearance details.

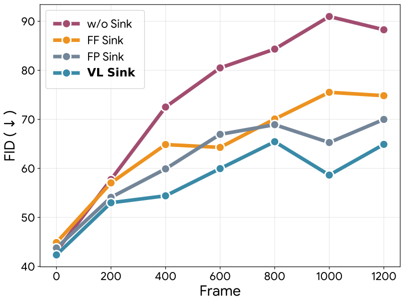

Fig. 10: Performance over time. We present sliding window FID with a 200-frame window size for different attention sink strategies. Our VL Sink achieves the lowest FID.

first-frame position instead of the first frame. As shown in Tab. 2 and Fig. 10, removing the sink causes drift in camera motion and scene structure, while FF and FP sinks reduce this drift but remain limited as the camera moves far from the anchor. The VL Sink achieves the lowest sliding-window FID and the slowest degradation over time.

6 Conclusion

We presented **Seoul World Model**, a video world model that grounds generation in a real city through retrieval-augmented conditioning on street-view images. Cross-temporal pairing, synthetic urban data, and a Virtual Lookahead Sink collectively address the temporal, spatial, and long-horizon challenges of city-scale grounding. We hope this work encourages further exploration of world simulation that operates in the physical world beyond imagined environments.

Acknowledgements

We would like to sincerely thank everyone at NAVER and NAVER Cloud who contributed their time, expertise, and feedback throughout this work. We are grateful to Jinbae Im, Moonbin Yim, and Bado Lee for their help with data preprocessing. We also thank Jongchae Na, Hyunjoon Cho, Hochul Hwang, Myoung-Suk Chae, and Kwangkean Kim for their support with data processing and guidance on the use of map data and metadata. Finally, we thank Jonghak Kim, Jieun Shin, and Hyeeun Shin for valuable discussions and thoughtful feedback. Their support and collaboration are greatly appreciated.

References

1. Agarwal, N., Ali, A., Bala, M., Balaji, Y., Barker, E., Cai, T., Chattopadhyay, P., Chen, Y., Cui, Y., Ding, Y., et al.: Cosmos world foundation model platform for physical ai. arXiv preprint arXiv:2501.03575 (2025)
2. Bai, S., Chen, K., Liu, X., Wang, J., Ge, W., Song, S., Dang, K., Wang, P., Wang, S., Tang, J., et al.: Qwen2. 5-vl technical report. arXiv e-prints (2025)
3. Blattmann, A., Dockhorn, T., Kulal, S., Mendelevitch, D., Kilian, M., Lorenz, D., Levi, Y., English, Z., Voleti, V., Letts, A., et al.: Stable video diffusion: Scaling latent video diffusion models to large datasets. arXiv preprint arXiv:2311.15127 (2023)
4. Blattmann, A., Rombach, R., Ling, H., Dockhorn, T., Kim, S.W., Fidler, S., Kreis, K.: Align your latents: High-resolution video synthesis with latent diffusion models. In: CVPR (2023)
5. Carion, N., Gustafson, L., Hu, Y.T., Debnath, S., Hu, R., Suris, D., Ryali, C., Alwala, K.V., Khedr, H., Huang, A., et al.: Sam 3: Segment anything with concepts. arXiv preprint arXiv:2511.16719 (2025)
6. Chen, B., Martí Monsó, D., Du, Y., Simchowitz, M., Tedrake, R., Sitzmann, V.: Diffusion forcing: Next-token prediction meets full-sequence diffusion. NeurIPS (2024)
7. Chen, J., Zhu, H., He, X., Wang, Y., Zhou, J., Chang, W., Zhou, Y., Li, Z., Fu, Z., Pang, J., et al.: Deepverse: 4d autoregressive video generation as a world model. arXiv preprint arXiv:2506.01103 (2025)
8. Dai, Y., Jiang, F., Wang, C., Xu, M., Qi, Y.: Fantasyworld: Geometry-consistent world modeling via unified video and 3d prediction. arXiv preprint arXiv:2509.21657 (2025)
9. Deng, B., Tucker, R., Li, Z., Guibas, L., Snavely, N., Wetzstein, G.: Streetscapes: Large-scale consistent street view generation using autoregressive video diffusion. In: SIGGRAPH (2024)
10. Dosovitskiy, A., Ros, G., Codevilla, F., Lopez, A., Koltun, V.: Carla: An open urban driving simulator. In: CoRL (2017)
11. Esser, P., Kulal, S., Blattmann, A., Entezari, R., Müller, J., Saini, H., Levi, Y., Lorenz, D., Sauer, A., Boesel, F., et al.: Scaling rectified flow transformers for high-resolution image synthesis. In: ICML (2024)
12. Ha, D., Schmidhuber, J.: World models. arXiv preprint arXiv:1803.10122 (2018)
13. He, X., Peng, C., Liu, Z., Wang, B., Zhang, Y., Cui, Q., Kang, F., Jiang, B., An, M., Ren, Y., et al.: Matrix-game 2.0: An open-source real-time and streaming interactive world model. arXiv preprint arXiv:2508.13009 (2025)

14. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S.: Gans trained by a two time-scale update rule converge to a local nash equilibrium. NeurIPS (2017)
15. Ho, J., Chan, W., Saharia, C., Whang, J., Gao, R., Gritsenko, A., Kingma, D.P., Poole, B., Norouzi, M., Fleet, D.J., et al.: Imagen video: High definition video generation with diffusion models. arXiv preprint arXiv:2210.02303 (2022)
16. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. NeurIPS (2020)
17. Hu, A., Russell, L., Yeo, H., Murez, Z., Fedoseev, G., Kendall, A., Shotton, J., Corrado, G.: Gaia-1: A generative world model for autonomous driving. arXiv preprint arXiv:2309.17080 (2023)
18. Huang, X., Li, Z., He, G., Zhou, M., Shechtman, E.: Self forcing: Bridging the train-test gap in autoregressive video diffusion. arXiv preprint arXiv:2506.08009 (2025)
19. Huang, Z., He, Y., Yu, J., Zhang, F., Si, C., Jiang, Y., Zhang, Y., Wu, T., Jin, Q., Chanpaisit, N., Wang, Y., Chen, X., Wang, L., Lin, D., Qiao, Y., Liu, Z.: Vbench: Comprehensive benchmark suite for video generative models. In: CVPR (2024)
20. HunyuanWorld, T.: Hy-world 1.5: A systematic framework for interactive world modeling with real-time latency and geometric consistency. arXiv preprint (2025)
21. Jiang, J., Zeng, W., Zheng, Z., Yang, J., Liang, C., Liao, W., Liang, H., Zhang, Y., Gao, M.: Omnihuman-1.5: Instilling an active mind in avatars via cognitive simulation. arXiv preprint arXiv:2508.19209 (2025)
22. Kong, W., Tian, Q., Zhang, Z., Min, R., Dai, Z., Zhou, J., Xiong, J., Li, X., Wu, B., Zhang, J., et al.: Hunyuanvideo: A systematic framework for large video generative models. arXiv preprint arXiv:2412.03603 (2024)
23. Li, B., Ma, Z., Du, D., Peng, B., Liang, Z., Liu, Z., Ma, C., Jin, Y., Zhao, H., Zeng, W., et al.: Omninwm: Omniscient driving navigation world models. arXiv preprint arXiv:2510.18313 (2025)
24. Li, G., Zheng, S., Xu, S., Chen, J., Li, B., Hu, X., Zhao, L., Jiang, P.T.: Magicworld: Interactive geometry-driven video world exploration. arXiv preprint arXiv:2511.18886 (2025)
25. Li, R., Torr, P., Vedaldi, A., Jakab, T.: Vmem: Consistent interactive video scene generation with surfel-indexed view memory. In: ICCV (2025)
26. Li, Y., Li, Z., Chen, N., Gong, M., Lyu, Z., Wang, Z., Jiang, P., Feng, C.: Multiagent multitraversal multimodal self-driving: Open mars dataset. In: CVPR (2024)
27. Lin, H., Chen, S., Liew, J., Chen, D.Y., Li, Z., Shi, G., Feng, J., Kang, B.: Depth anything 3: Recovering the visual space from any views. arXiv preprint arXiv:2511.10647 (2025)
28. Liu, K., Hu, W., Xu, J., Shan, Y., Lu, S.: Rolling forcing: Autoregressive long video diffusion in real time. arXiv preprint arXiv:2509.25161 (2025)
29. Liu, Y., Luo, C., Fan, L., Wang, N., Peng, J., Zhang, Z.: Citygaussian: Real-time high-quality large-scale scene rendering with gaussians. In: ECCV (2024)
30. Liu, Y., Zhang, K., Li, Y., Yan, Z., Gao, C., Chen, R., Yuan, Z., Huang, Y., Sun, H., Gao, J., et al.: Sora: A review on background, technology, limitations, and opportunities of large vision models. arXiv preprint arXiv:2402.17177 (2024)
31. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. In: ICLR (2019)
32. Mao, X., Li, Z., Li, C., Xu, X., Ying, K., He, T., Pang, J., Qiao, Y., Zhang, K.: Yume-1.5: A text-controlled interactive world generation model. arXiv preprint arXiv:2512.22096 (2025)
33. Peebles, W., Xie, S.: Scalable diffusion models with transformers. In: ICCV (2023)

34. Ren, X., Shen, T., Huang, J., Ling, H., Lu, Y., Nimier-David, M., Müller, T., Keller, A., Fidler, S., Gao, J.: Gen3c: 3d-informed world-consistent video generation with precise camera control. In: CVPR (2025)
35. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: CVPR (2022)
36. Schneider, M.A., Höllein, L., Nießner, M.: Worldexplorer: Towards generating fully navigable 3d scenes. In: SIGGRAPH Asia (2025)
37. Seo, J., Fukuda, K., Shibuya, T., Narihira, T., Murata, N., Hu, S., Lai, C.H., Kim, S., Mitsufuji, Y.: Genwarp: Single image to novel views with semantic-preserving generative warping. NeurIPS (2024)
38. Seo, J., Mira, R., Haliassos, A., Bounareli, S., Chen, H., Tran, L., Kim, S., Landgraf, Z., Shen, J.: Lookahead anchoring: Preserving character identity in audio-driven human animation. arXiv preprint arXiv:2510.23581 (2025)
39. Shang, Y., Lin, Y., Zheng, Y., Fan, H., Ding, J., Feng, J., Chen, J., Tian, L., Li, Y.: Urbanworld: An urban world model for 3d city generation. arXiv preprint arXiv:2407.11965 (2024)
40. Shin, J., Li, Z., Zhang, R., Zhu, J.Y., Park, J., Shechtman, E., Huang, X.: Motionstream: Real-time video generation with interactive motion controls. arXiv preprint arXiv:2511.01266 (2025)
41. Song, J., Meng, C., Ermon, S.: Denoising diffusion implicit models. In: ICLR (2021)
42. Su, J., Ahmed, M., Lu, Y., Pan, S., Bo, W., Liu, Y.: Roformer: Enhanced transformer with rotary position embedding. Neurocomputing (2024)
43. Sun, P., Kretzschmar, H., Dotiwalla, X., Chouard, A., Patnaik, V., Tsui, P., Guo, J., Zhou, Y., Chai, Y., Caine, B., Vasudevan, V., Han, W., Ngiam, J., Zhao, H., Timofeev, A., Ettinger, S., Krivokon, M., Gao, A., Joshi, A., Zhang, Y., Shlens, J., Chen, Z., Anguelov, D.: Scalability in perception for autonomous driving: Waymo open dataset. In: CVPR (2020)
44. Tancik, M., Casser, V., Yan, X., Pradhan, S., Mildenhall, B., Srinivasan, P.P., Barron, J.T., Kretzschmar, H.: Block-nerf: Scalable large scene neural view synthesis. In: CVPR (2022)
45. Tang, J., Liu, J., Li, J., Wu, L., Yang, H., Zhao, P., Gong, S., Yuan, X., Shao, S., Lu, Q.: Hunyuan-gamecraft-2: Instruction-following interactive game world model. arXiv preprint arXiv:2511.23429 (2025)
46. Team, R., Gao, Z., Wang, Q., Zeng, Y., Zhu, J., Cheng, K.L., Li, Y., Wang, H., Xu, Y., Ma, S., et al.: Advancing open-source world models. arXiv preprint arXiv:2601.20540 (2026)
47. Unterthiner, T., Van Steenkiste, S., Kurach, K., Marinier, R., Michalski, M., Gelly, S.: Towards accurate generative models of video: A new metric & challenges. arXiv preprint arXiv:1812.01717 (2018)
48. Valevski, D., Leviathan, Y., Arar, M., Fruchter, S.: Diffusion models are real-time game engines. In: ICLR (2025)
49. Wan, T., Wang, A., Ai, B., Wen, B., Mao, C., Xie, C.W., Chen, D., Yu, F., Zhao, H., Yang, J., et al.: Wan: Open and advanced large-scale video generative models. arXiv preprint arXiv:2503.20314 (2025)
50. Wang, Z., Lin, H., Yoon, J., Cho, J., Zhang, Y., Bansal, M.: Anchorweave: World-consistent video generation with retrieved local spatial memories. arXiv preprint arXiv:2602.14941 (2026)
51. Williams, R.J., Zipser, D.: A learning algorithm for continually running fully recurrent neural networks. Neural Computation (1989)
52. Wu, C., Li, J., Zhou, J., Lin, J., Gao, K., Yan, K., Yin, S.m., Bai, S., Xu, X., Chen, Y., et al.: Qwen-image technical report. arXiv preprint arXiv:2508.02324 (2025)

53. Wu, T., Yang, S., Po, R., Xu, Y., Liu, Z., Lin, D., Wetzstein, G.: Video world models with long-term spatial memory. arXiv preprint arXiv:2506.05284 (2025)
54. Xiao, G., Tian, Y., Chen, B., Han, S., Lewis, M.: Efficient streaming language models with attention sinks. In: ICLR (2024)
55. Yang, S., Huang, W., Chu, R., Xiao, Y., Zhao, Y., Wang, X., Li, M., Xie, E., Chen, Y., Lu, Y., et al.: Longlive: Real-time interactive long video generation. arXiv preprint arXiv:2509.22622 (2025)
56. Yang, Z., Teng, J., Zheng, W., Ding, M., Huang, S., Xu, J., Yang, Y., Hong, W., Zhang, X., Feng, G., et al.: Cogvideox: Text-to-video diffusion models with an expert transformer. In: ICLR (2025)
57. Yi, J., Jang, W., Cho, P.H., Nam, J., Yoon, H., Kim, S.: Deep forcing: Training-free long video generation with deep sink and participative compression. arXiv preprint arXiv:2512.05081 (2025)
58. Yin, T., Zhang, Q., Zhang, R., Freeman, W.T., Durand, F., Shechtman, E., Huang, X.: From slow bidirectional to fast autoregressive video diffusion models. In: CVPR (2025)
59. Yu, J., Bai, J., Qin, Y., Liu, Q., Wang, X., Wan, P., Zhang, D., Liu, X.: Context as memory: Scene-consistent interactive long video generation with memory retrieval. In: SIGGRAPH Asia (2025)
60. Yu, W., Xing, J., Yuan, L., Hu, W., Li, X., Huang, Z., Gao, X., Wong, T.T., Shan, Y., Tian, Y.: Viewcrafter: Taming video diffusion models for high-fidelity novel view synthesis. In: TPAMI (2025)
61. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 586–595 (2018)
62. Zhang, S., Yang, X., Zi, B., Huang, H., Zhang, C., Li, X.: Telestyle: Content-preserving style transfer in images and videos. arXiv preprint arXiv:2601.20175 (2026)
63. Zhu, H., Wang, Y., Zhou, J., Chang, W., Zhou, Y., Li, Z., Chen, J., Shen, C., Pang, J., He, T.: Aether: Geometric-aware unified world modeling. In: ICCV (2025)
64. Zhu, Y., Feng, J., Zheng, W., Gao, Y., Tao, X., Wan, P., Zhou, J., Lu, J.: Astra: General interactive world model with autoregressive denoising. arXiv preprint arXiv:2512.08931 (2025)

Appendix

A Implementation Details

A.1 Model and Training Details

Additional training details. During Teacher-Forcing (TF) training, we inject small Gaussian noise (μ=0, σ=0.1) to the conditioning history frames with 50% probability, reducing the gap between clean training inputs and self-generated inference history. The three training datasets are mixed via ratio-based interleaved sampling: Waymo [43] (20%), Seoul street-view (40%), and synthetic (40%), roughly following the relative dataset sizes. All camera extrinsics are expressed in a unified Right-Down-Forward (RDF) coordinate system and are defined relative to the first frame of each chunk.

Under Self-Forcing (SF) [18], the model generates latents from self-produced history latents cached in KV memory with causal attention, where each token can attend only to tokens at earlier or equal positions. The Virtual Lookahead (VL) Sink and semantic reference tokens carry RoPE [42] positions outside the current generation window (beyond the last generated frame for the VL Sink and at a large temporal offset for the references), yet all generated tokens must attend to them. We resolve this by separating RoPE temporal positions from token ordering. Specifically, the VL Sink and reference tokens are assigned RoPE positions corresponding to their intended temporal locations (beyond the current generation window for the VL Sink and at a large temporal offset for the references), while being prepended to the beginning of each chunk's token sequence. Because RoPE encodes temporal information through positional embeddings rather than token ordering, and because these tokens are placed before the generated tokens in the sequence, they remain visible to all generated tokens under the causal mask.

To enable classifier-free guidance (CFG) and ensure graceful degradation when certain conditions are unavailable, we apply the following dropout schedule during training: text captions are replaced with empty-string T5 embeddings with a probability of 20%; reference conditions are zeroed out with a probability of 20% for street-view and synthetic data, while Waymo samples never include references (effectively corresponding to 100% reference dropout); warped video inputs are replaced with zeros with a probability of 20%.

Additional model details. SWM fine-tunes Cosmos-Predict2.5-2B-I2W [1], a 2B-parameter Diffusion Transformer (DiT) with 28 blocks, 16 attention heads, and a hidden dimension of 2048. The model operates in a 16-channel latent space produced by a 3D VAE with 4× temporal and 8× spatial compression. The DiT backbone processes the noisy target latent, channel-concatenated with the encoded warped video retrieved from reference images. Three separate patch embedding modules handle different token types: the main embedder processes the target latent concatenated with warped conditioning channels, while dedicated reference and lookahead embedders each process 16-channel encoded latents with an additional 1-channel padding mask. Both the reference and lookahead embedders are initialized by copying the weights of the main embedder. Camera poses are first converted into 6-channel Plücker ray maps from the camera extrinsics and intrinsics, and then encoded using a shallow encoder. The resulting camera embeddings are added as residuals to both the main video tokens and the reference tokens.

Starting from arbitrary coordinates. By default, the user selects a starting coordinate that corresponds to an existing street-view location, and the corresponding pinhole image is directly used as the first frame. When the user specifies an arbitrary coordinate that does not correspond to any street-view location, we use the nearest available street-view image as the first frame and generate a buffer chunk that navigates toward the target starting point. The generation then continues from the next chunk onward at the specified coordinate, and the buffer chunk is discarded from the final output.

A.2 Street-View Data Processing

Depth and camera poses. We estimate per-keyframe depth maps and camera poses using Depth Anything 3 (DA3) [27]. For each driving sequence, we collect target pinhole keyframes rendered from equirectangular panoramas, along with the corresponding reference panorama images rendered into eight directional views uniformly covering 360°. All target and reference images within a subsequence are jointly fed into DA3, which estimates scale-consistent depth maps and relative camera poses across all input images in a single forward pass.

For longer sequences that exceed the capacity of a single DA3 forward pass, we partition the sequence into non-overlapping chunks and run DA3 independently on each chunk. To recover real-world metric scale, we align each chunk's camera poses to real-world coordinates using GPS metadata. Specifically, we estimate a similarity transformation by matching the camera displacement from the first frame to the last frame of each chunk in the DA3 coordinate frame with the corresponding displacement derived from GPS coordinates. Metric depth is then obtained by scaling the affine-invariant DA3 depth with the estimated scale factor.

Because all street-view images in the database are processed through DA3 with GPS-based metric alignment, they share a globally consistent coordinate system. This ensures that both semantic references and the virtual lookahead sink, which are retrieved from the same database, have compatible camera poses regardless of when or where they were captured.

Text captioning. We generate text captions for all training videos using Qwen2.5-VL-72B [2]. Each video is captioned with a structured prompt that instructs the model to produce both a long caption (up to 280 words) and a short caption (up to 30 words), covering urban scenery, dynamic actors, environmental conditions, specific events, and the camera trajectory. During training, we randomly select between the long and short caption variants. In addition to the VLM-generated captions, we prepend a predefined camera-action sentence describing the trajectory direction (e.g., straight, left turn, right turn, stop), derived from the camera pose sequence.

Stylized video augmentation. We observe that text prompts describing events occurring within a scene (e.g., flooding, fire, monster appearance) generalize well to our fine-tuned model, as these capabilities are largely inherited from the pretrained world simulation model [1]. However, prompts involving global style changes such as day-to-night transitions or weather variations tend to be less faithfully followed after fine-tuning on street-view data. To address this, we construct a video stylization pipeline combining Qwen-Image-Edit [52] and TeleStyle [62]. Given a style-related text prompt, Qwen-Image-Edit first edits the starting frame to reflect the target style, and TeleStyle then propagates this style consistently across the entire video. Using this pipeline, we augment a subset of the interpolated street-view videos with diverse style prompts, producing 10K additional stylized training videos.

Coverage area. Fig. 11 shows the spatial distribution of our street-view data collection. We focus on densely populated districts within the Seoul Metropolitan Area, where diverse urban structures, road types, and streetscapes provide rich training signal for the model. The covered region extends approximately 44.8 km in the east–west direction and 31.0 km in the north–south direction.

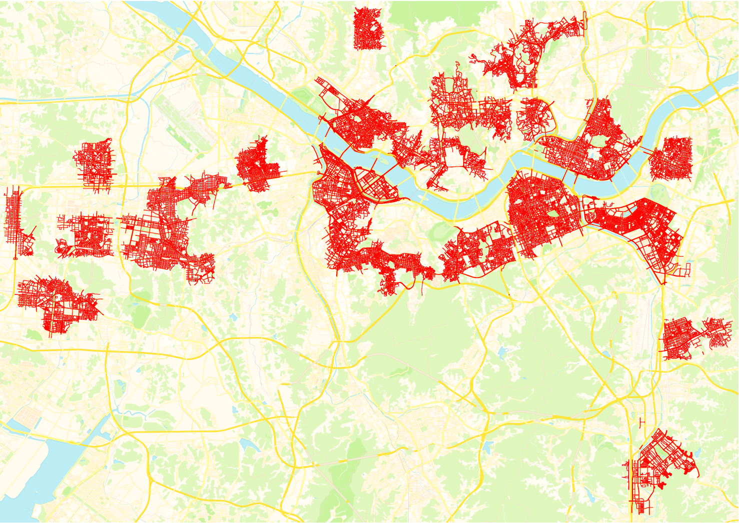

Fig. 11: Coverage area. Spatial distribution of collected street-view data within the Seoul Metropolitan Area.

A.3 Synthetic Dataset

We use CARLA [10] (v0.9.15), an Unreal Engine-based simulator, to construct our synthetic video dataset across five predefined maps: `Town01`, `Town02`, `Town03`, `Town04`, `Town05`, and `Town06`. We randomly spawn vehicles and pedestrians before rendering to simulate realistic traffic. For cross-temporal pairing, we divide target videos and street-view reference images into multiple subsets and render each subset under distinct traffic, lighting, and weather conditions, ensuring the model consistently encounters a temporal gap between references and targets across all data sources. We additionally render depth maps and extract camera parameters per frame to support geometric and semantic referencing. Since CARLA natively operates in real-world metric scale, the rendered depth maps and camera poses are directly compatible with the GPS-aligned metric geometry used for our street-view data.

Target video. For pedestrian and vehicle trajectories, we attach RGBD sensors to randomly spawned self-driving pedestrians and vehicles, and capture target videos along their natural motion paths. For free-camera trajectories, we sample a random initial position and viewing direction, then move the camera along diverse paths by continuously randomizing the viewing direction and movement speed with collision detection.

Street-view reference. For each map, we render street-view reference images at regular 10 m intervals along all roads, with horizontal eight directional views uniformly covering 360° per location. To reduce the sim-to-real gap, we apply slight positional jitter to the sampling interval and vary the lane position on multi-lane roads. In total, this yields 4K street-view positions and 32K reference frames across all maps.

B Evaluation Details

B.1 Evaluation Metrics

We evaluate generation quality from three aspects: visual and temporal fidelity using FID [14] and FVD [47], camera-following accuracy using RotErr and TransErr, and 3D adherence using masked PSNR and LPIPS [61]. Below we provide additional details on how each metric is computed.

For Busan-City-Bench, ground-truth video is unavailable because the benchmark is constructed from sparsely captured street-view images. We therefore use our view interpolation pipeline (Sec. 3.1) to synthesize temporally continuous videos from the ground-truth street-view keyframes and treat these interpolated videos as the ground truth for FVD computation. For Ann-Arbor-City-Bench, we directly use the ground-truth video sequences from the MARS dataset [26].

PSNR and LPIPS are computed between the generated video and the ground-truth video (or the ground-truth image sequence). Because our model targets dynamic video generation grounded in real-world references rather than exact 4D reconstruction, dynamic objects (*e.g.*, vehicles and pedestrians) in the generated video do not necessarily match those in the ground truth. To focus the reconstruction metrics on how faithfully the model preserves static scene structure from the references, we segment dynamic objects in both generated and ground-truth frames. Specifically, we use SAM3 [5] with text prompts for dynamic-object categories (*e.g.*, pedestrian, vehicle) to extract per-frame dynamic masks, and compute PSNR and LPIPS only over the static regions.

For camera-following accuracy, we extract per-frame camera extrinsics from both the generated and ground-truth frames using DA3 [27], process frames in non-overlapping chunks, and compute relative poses by setting the first frame as the identity. To ensure a fair comparison regardless of scale differences, we independently normalize each translation trajectory by its maximum translation norm. RotErr measures the mean geodesic distance on SO(3) between the predicted and ground-truth relative rotations, and TransErr measures the mean ℓ_2 distance between the scale-normalized relative translations, similarly to Vmem [25].

B.2 Baseline Adaptation

Since real-world grounded world simulation is a new task, existing world models do not natively support the full set of inputs required by our benchmark (start frame, camera trajectory, text prompt, and retrieved street-view references). We therefore adapt each baseline to our evaluation setup by providing the subset of inputs supported by each model. Below we describe how camera trajectories are supplied to each model. All baselines receive the first frame of the target sequence as the starting image and generate autoregressively to cover the full benchmark length.

Aether [63]. Aether accepts camera input as a 6-channel Plücker ray map that is channel-concatenated with the image latent. We convert the benchmark camera extrinsics to Aether's format by computing relative poses with respect to the first frame of each chunk, generating per-pixel ray origins and directions, and applying the model's signed-log normalization to translations. Autoregressive generation uses 41-frame chunks with a 1-frame overlap, recomputing relative poses for each chunk.

DeepVerse [7]. DeepVerse does not accept continuous camera parameters. Instead, it discretizes camera motion into 27 action classes (9 translation directions × 3 rotation states) and maps each action to a natural-language sentence, which is encoded as a text embedding. We convert benchmark trajectories by computing per-chunk relative transforms, quantizing the translation direction, and retrieving the corresponding pre-computed text embedding.

Yume1.5 [32]. In Yume1.5, camera control is expressed entirely through text. Camera extrinsics are converted into WASD-style keyboard commands (8 translation directions) and mouse-style rotation descriptions (4 rotation directions) through majority voting within each chunk. These action descriptions are prepended to the scene caption and encoded using the T5 text encoder.

HY-World1.5 [20]. HY-World1.5 uses dual-path camera conditioning. Continuous camera parameters are injected through Projective Rotary Position Embedding (PRoPE), which applies the camera projection matrix to attention queries and keys, while discretized motion states (9 translation and 9 rotation classes) are added to the timestep conditioning. We convert benchmark trajectories by computing first-frame-relative camera poses, rescaling translations to a median per-step magnitude of 0.08 (matching the model's expected scale), normalizing intrinsics by the image dimensions, and discretizing relative motions into the model's action space.

FantasyWorld [8]. FantasyWorld accepts camera input as per-pixel 6-channel Plücker ray embeddings injected into the diffusion transformer via Adaptive Layer Normalization. We convert benchmark camera trajectories by constructing camera-to-world matrices from the extrinsics and computing per-pixel ray origins and directions. Camera translations are rescaled using the average scene depth estimated via monocular depth prediction on the first frame, following the model's default inference configuration.

LingBot [46]. LingBot-World conditions on camera trajectories via 6-channel Plücker ray maps injected into each transformer block through scale-and-shift modulation. We convert benchmark camera trajectories by computing poses relative to the first frame of each chunk and constructing per-pixel ray origins and directions from the resulting cameras.

C Additional Results and Analyses

C.1 View Interpolation

As described in Sec. 3.1, our view interpolation pipeline synthesizes temporally continuous video from sparse street-view keyframes. We compare two conditioning strategies for injecting keyframe information into the pretrained video diffusion model (Fig. 3).

The **channel concatenation** baseline encodes each keyframe latent and concatenates it along the channel dimension at the corresponding timestamp, zero-padding non-keyframe positions. This requires widening the input projection from 18 to 34 channels. As discussed in the main paper, this approach suffers from weak keyframe adherence because an isolated keyframe does not form a valid 4-frame group for the 3D VAE's temporal compression.

Our **intermittent freeze-frame** strategy ensures that each keyframe forms a complete 4-frame group matching the 3D VAE's temporal stride, without modifying the network architecture. Each keyframe is repeated 4 consecutive times at its corresponding pixel position, so the 3D VAE compresses it into exactly one latent. During training, the resulting videos alternate between smooth motion segments and brief freeze segments at keyframe positions. At inference, each input keyframe is encoded into a single clean latent, which then replaces the corresponding position in the noisy input latent at every diffusion step, ensuring exact keyframe conditioning. After generation and decoding, three out of four repeated frames at each keyframe position are discarded to recover the intended video.

Tab. 3 compares the two strategies quantitatively on the Waymo [43] test set, measuring PSNR, SSIM, and LPIPS between the interpolated video and the ground-truth video. Fig. 12 shows a qualitative comparison on Seoul street-view sequences.

Table 3: Quantitative comparison of view interpolation strategies. We compare the channel concatenation baseline with our Intermittent Freeze-Frame approach.

Method	PSNR↑	SSIM↑	LPIPS↓
Channel Concatenation	22.52	0.628	0.245
Intermittent Freeze-Frame (Ours)	**25.03**	**0.703**	**0.162**

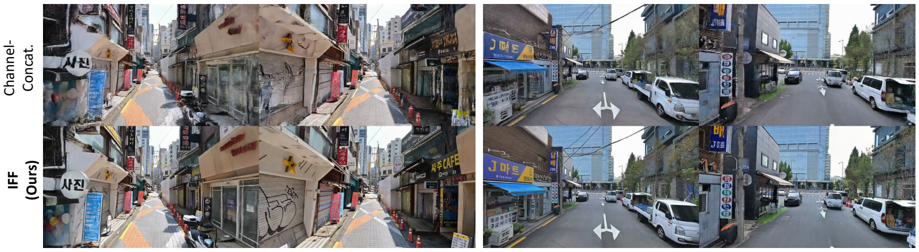

Fig. 12: Qualitative comparison of view interpolation strategies. Channel concatenation and our Intermittent Freeze-Frame (IFF).

C.2 Additional Qualitative Results

Fig. 16 presents qualitative ablation results showing the effect of each component on generation quality. Additional video results are provided in **the project page:** `https://seoul-world-model.github.io`.

C.3 Effect of Reference Sparsity

In practice, the density of available street-view images varies across locations. To evaluate how SWM behaves under sparse retrieval, we reduce the number of retrieved references K per chunk from the default K=5 down to K=1 and report results on Busan-City-Bench in Fig. 13.

As K decreases, mPSNR drops, since fewer references provide less coverage of the target scene. In contrast, FID and FVD show no clear degradation; K=1 achieves the best FID and the second-best FVD. We speculate that the underlying video diffusion model retains its generative capability even with fewer reference constraints, producing visually plausible frames that score well on distributional metrics despite being less grounded to the specific location. These results suggest that reference conditioning primarily improves geometric and appearance grounding rather than overall visual realism.

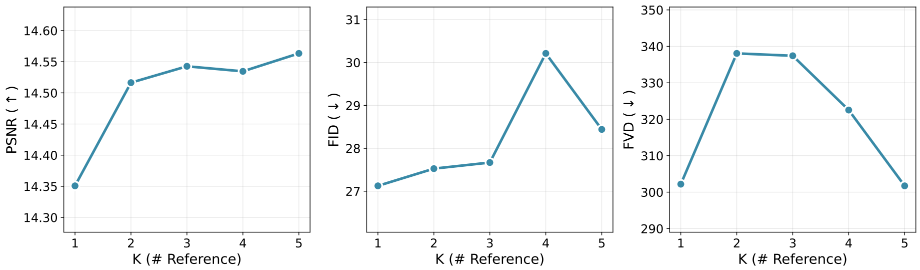

Fig. 13: Effect of the number of retrieved references K.

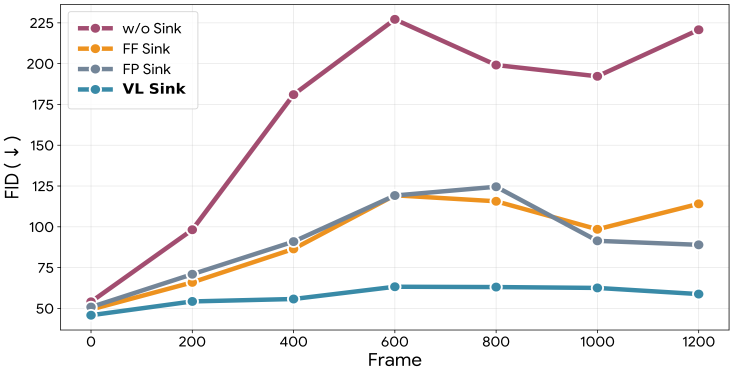

Fig. 14: Attention sink comparison under Self-Forcing. Sliding-window FID over time for different sink strategies in the SF configuration.

C.4 Additional Ablation on Attention Sink with SF Variant

In addition to the attention sink comparison in the main paper (Tab. 2), Fig. 14 compares the SF variant's generation stability across different attention sink types over extended trajectories.

C.5 Comparison with Video Generative Models for Static Scenes

While the primary baselines in Tab. 1 are dynamic world models, we additionally compare SWM with two representative models designed for static 3D scene video generation: GEN3C [34] and VMem [25]. Although these models do not generate dynamic environments, they share a relevant capability with SWM: conditioning on multi-view reference images to produce geometrically consistent video from novel viewpoints.

GEN3C [34] accepts multi-view images as input and generates video from specified camera trajectories. However, its design assumes that the multi-view images capture a static scene at the same time instant, and the generated videos depict a static scene with only camera motion. We adapt GEN3C to our setup by providing the retrieved street-view images as multi-view inputs.

VMem [25] uses a geometric-aware memory that stores previously generated frames and retrieves them based on camera-pose similarity to ensure multi-view consistency. Rather than relying on its self-generated frames, we populate VMem's memory with our retrieved street-view pinhole images, enabling it to leverage real-world observations during generation. Similar to GEN3C, VMem focuses on static scene generation.

Tab. 4 compares these models with SWM on both benchmarks. Both GEN3C and VMem struggle to handle the temporal inconsistency inherent in street-view references captured at different times: dynamic objects in the references appear frozen in the generated videos, and the models cannot synthesize plausible motion. This results in worse FID and FVD compared to SWM. However, for static-region metrics (mPSNR, mLPIPS), GEN3C shows competitive performance with SWM, reflecting its strength in static scene reconstruction. These results highlight that while static 3D models can partially address the geometric grounding aspect of our task, generating dynamic, temporally coherent video grounded in real locations requires explicit modeling of scene dynamics, as SWM provides.

Table 4: Quantitative comparison with video generative models for static scenes on Busan-City-Bench.

Method	FID↓	FVD↓	Img.Q.↑	RotErr↓	TransErr↓	mPSNR↑	mLPIPS↓
VMem [25]	100.75	913.26	0.748	0.105	0.212	12.74	0.550
GEN3C [34]	45.72	416.94	0.732	0.030	0.082	14.16	0.524
SWM (TF)	**28.43**	**301.76**	**0.781**	**0.020**	**0.015**	**14.56**	**0.392**

C.6 Extended Long-Horizon Evaluation

To further examine the attention sink's role under extended generation, we construct a long-horizon version of Busan-City-Bench where each sequence is 1,460 frames long (4× the standard 365-frame benchmark, covering approximately 500 m per sequence). Tab. 5 compares the attention sink variants on this extended benchmark.

The performance gaps between sink variants become more pronounced over the longer horizon. Removing the sink entirely causes notable FID degradation (37.37 vs. 25.13), confirming that an attention sink is important for maintaining visual quality over long distances. The first-position sink achieves the better camera-following accuracy (RotErr 0.019, TransErr 0.021), while the full model with the VL Sink achieves the best mPSNR and mLPIPS, indicating the strongest grounding to real-world appearance under extended generation.

Table 5: Long-horizon attention sink ablation on Busan-City-Bench (1,460 frames). Sequences are 4× longer than the standard benchmark.

Variant	FID↓	FVD↓	Img.Q.↑	RotErr↓	TransErr↓	mPSNR↑	mLPIPS↓
Full model (VL Sink)	**25.13**	**394.58**	0.764	0.027	0.029	**13.70**	**0.480**
w/o any attention sink	37.37	550.81	0.751	0.026	0.041	12.94	0.575
w/ first-frame attention sink	30.85	440.65	0.767	0.045	0.044	13.08	0.534
w/ first-position attention sink	28.57	439.69	**0.769**	**0.019**	**0.021**	13.34	0.507

D Discussions

D.1 Limitations and Failure Cases

The quality of SWM's generation is closely tied to the quality of its training data. Because city-wide video data is not readily available, we synthesize temporally continuous video from sparse street-view *image* sequences through our view interpolation pipeline. While these interpolated videos provide effective training supervision, they remain lower in quality than real captured video, and incorporating real video data as it becomes available would further improve generation quality.

A related limitation stems from the capture pattern of street-view imagery. Street-view images are typically captured at equal *distance* intervals rather than equal *time* intervals. When the capture vehicle slows down or stops, consecutive street-view frames can span a large temporal gap. Although we filter sequences based on capture-time metadata, noisy metadata causes some temporally inconsistent sequences to pass the filter. When these sequences are converted into interpolated training video, dynamic objects such as vehicles may abruptly appear or disappear between frames. This artifact propagates into the trained model, which occasionally generates sudden appearance or disappearance of vehicles, as shown in Fig. 15.

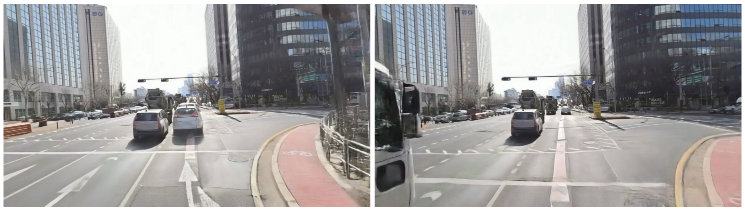

Fig. 15: Failure cases. Vehicles occasionally appear or disappear abruptly due to temporally inconsistent street-view sequences in the training data.

D.2 Other Discussions

Relationship to street-view interpolation. Although SWM uses a view interpolation pipeline to construct training data (Sec. 3.1), the task SWM addresses is fundamentally different from street-view interpolation.

Street-view interpolation takes a sequence of street-view images captured along a trajectory and synthesizes smooth video between them. The camera path is fixed to the original capture route, and any dynamic objects visible in the input street-view images (*e.g.*, parked cars, pedestrians) are interpolated as they appear, preserving a temporally consistent scene within each sequence.

SWM operates under a different setting. Given a starting location within the coverage area of a panoramic street-view database, the user specifies a free-form camera trajectory and a text prompt. The model retrieves nearby street-view images as visual references, but these references are captured independently at different times, so the dynamic objects they contain (vehicles, pedestrians, signage) are mutually inconsistent. Rather than interpolating between these inconsistent snapshots, SWM generates a coherent dynamic scene by learning to disentangle persistent structure from transient content through cross-temporal pairing (Sec. 3). The model freely navigates the covered region along arbitrary camera paths, synthesizes plausible object motion, and responds to text prompts that alter scene conditions such as weather, time of day, or hypothetical events.

Why is cross-temporal pairing important? Cross-temporal pairing prevents the model from learning a spurious temporal correlation between retrieved references and the target video, which would cause abrupt transitions at inference time.

Street-view images are typically captured sequentially by a vehicle or pedestrian moving along a route. Consecutive street-view locations are therefore not only spatially close but also likely to be temporally close: a street-view image

5 m ahead may have been captured just seconds before or after the current one. Without cross-temporal pairing, the references retrieved for a target training sequence would come from nearby locations captured at nearly the same time, meaning the dynamic objects in the references (vehicles, pedestrians) are temporally consistent with those in the target. During training, the model would learn to copy or interpolate these dynamic objects from the references into the generated frames.

This becomes problematic at inference. The model generates video autoregressively, so each chunk must be temporally coherent with its self-generated history while remaining spatially grounded in the retrieved references. If the model has learned to rely on temporal consistency between references and the target, it will attempt to reflect dynamic objects from the retrieved street-view images into the generated video. However, at inference the retrieved references are captured at arbitrary past times unrelated to the generated scene, so these dynamic objects are inconsistent with the ongoing generation. This leads to abrupt appearance or disappearance of objects and visual discontinuities. Cross-temporal pairing eliminates this issue by ensuring that references and targets always come from different timestamps during training, teaching the model to attend to persistent scene structure while ignoring transient content (Fig. 6). Ablation results demonstrating this effect are included in the project page.

D.3 Societal Impact

Real-world grounded video world simulation enables several beneficial applications: urban planners can preview proposed streetscape, autonomous driving systems can be tested against diverse scenarios grounded in real city layouts, and location-based applications can let users explore familiar places under novel conditions.

Training on street-view imagery requires careful handling of personal data. All street-view data used in this work was collected in compliance with local regulations and is publicly served by the map provider, with pedestrian faces, license plates, and other sensitive regions blurred prior to release. As the training data contains no unblurred personal identifiers, the model does not learn to reproduce identifiable faces or license plates.

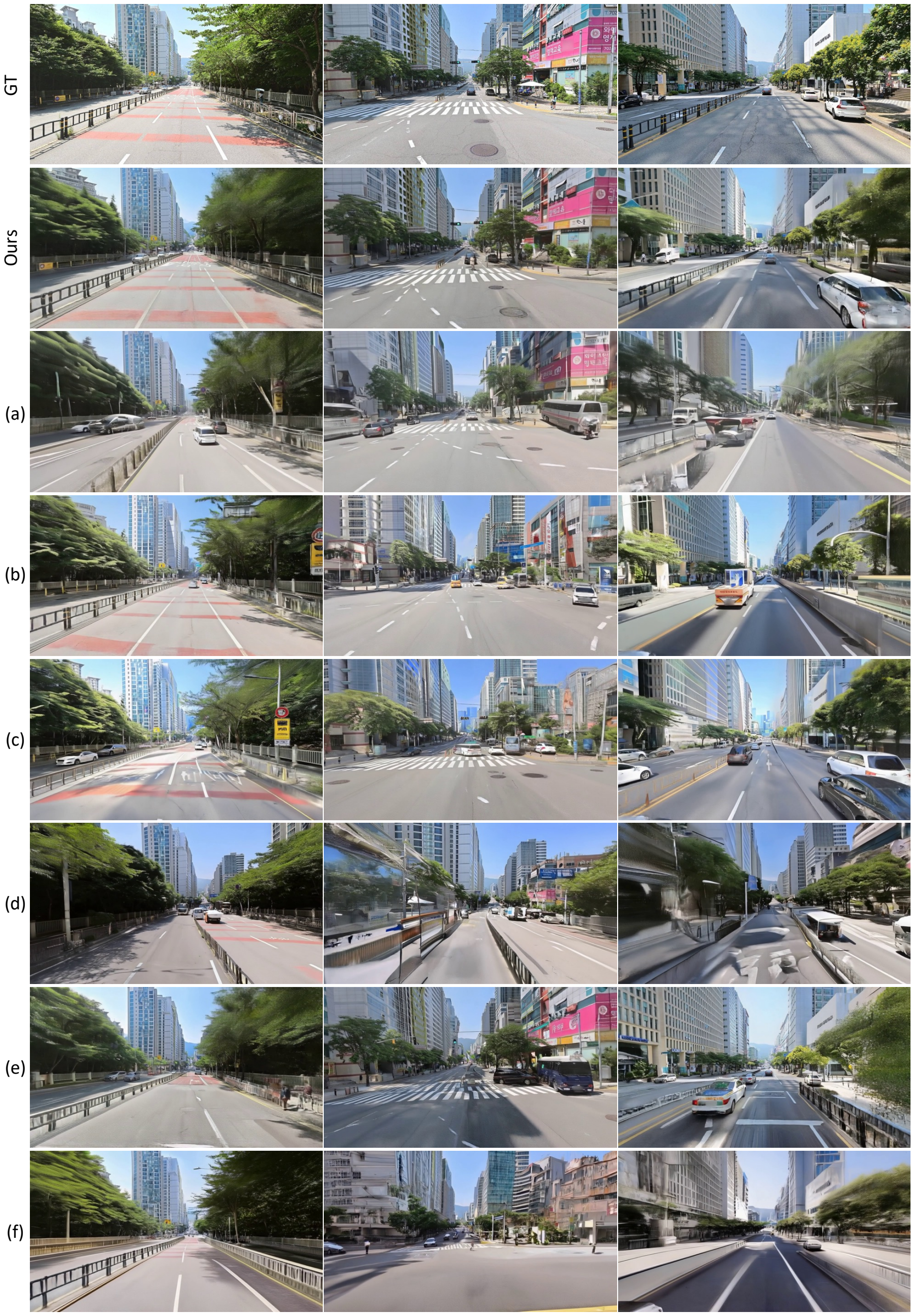

Fig. 16: Additional qualitative ablation results. We show the effect of each component on generation quality on the Busan long-horizon benchmark. Specifically, (a) ours without VL sink, (b) ours with first-frame attention sink, (c) ours with first-position attention sink, (d) ours without geometric referencing, (e) ours without semantic referencing, and (f) ours without cross-temporal pairing.