

1. Supplementary Materials

1.1. More Qualitative Results

It is highly recommended to refer to the webpage for visualization.

1.2. Implementation Details.

Data Curation. We sample frames from ScanNet++ [9] and ARKitScenes [1] at an interval of three to reduce temporal redundancy. To curate datasets for object grounding, for each sampled frame, we use Qwen3-VL [5] to perform object grounding with the following prompt template:

Object Grounding Prompt Template

You are a helpful assistant. Identify ALL objects that belong to these categories: YOUR_CATEGORY. Requirements:

1. Return ALL instances of objects from these categories (can be multiple).
2. Each object must be CLEARLY VISIBLE with SHARP, DISTINCT boundaries (not blurry or pixelated).
3. If the image is blurry, low quality, or no clear objects exist, return an empty array: [].
4. Format: ["label": "DETECTED_CATEGORY", "box": [x1, y1, x2, y2], ...]

We then filter detections using a center-region threshold (center ratio) and select the instance closest to the image center as the grounding target. For each clip, we take a frame where the target is centered as the final frame and extract the preceding 161 frames (inclusive) to form the target video clip. We also use Qwen3-VL to caption the clip with the following prompt:

Video Caption Prompt Template

You are a helpful assistant. Describe a video sequence in which the camera moves through the environment and, at the end, a {centered_object_label} appears centered in the frame. Use simple sentences. Do NOT use complex grammar. Good example: "The camera moves through a room. It pans left and right. At the end, a {centered_object_label} is centered in the frame." Bad example: "The camera pans around, revealing various objects and eventually discovering a {centered_object_label} positioned in the center, where it comes to rest in the frame." Provide a concise description in 1–2 simple sentences.

Training with context. During training, we sample both context frames and target generation frames from source

videos. Target clips are 161 frames long. Context frames are randomly drawn but always in contiguous groups of four to satisfy the Wan2.2 [6] VAE encoder. For efficiency, we pre-encode frames into latents during preprocessing and sample latents directly, yielding roughly 2× faster training. A key consideration is ensuring sufficient overlap between context and target frames so the model can learn geometric consistency. ScanNet++ sequences are long with ample overlap, so we sample context and targets from the same clip. ARKitScenes sequences are shorter with less view overlap but include multiple videos per environment; therefore, we sample context and targets from different videos within the same environment.

Spatial Distance. To measure the distance between the point cloud reconstructed by VGGT [7] and the ground-truth point cloud, we first register the two coordinate frames. ScanNet++ [9] provides frames with calibrated camera poses and depth maps. When reconstructing point clouds from generated videos, we append 40 ground-truth frames as anchors and use the estimated point clouds of these anchors to perform registration. After alignment, we compute the point-cloud distance.

1.3. Benchmark setting.

Since it is not feasible to make the setting for different methods perfectly fair, we list the detailed setting for each methods for reference.

Wan2.2-5B [6]. We generate 121-frame videos at a resolution of 1280×704 . The model takes the first frame and a text instruction as input, utilizing 50 inference steps and a classifier-free guidance scale of 5.0.

Veo 3 [2]. We utilize the Veo 3 image to video model via its official API, generating videos at 1280×720 with 192 frames.

FramePack [11]. We employ FramePack to generate videos at a resolution of 704×544 . The inference process utilizes 50 sampling steps with a classifier-free guidance scale of 3.0. The model is conditioned on a context window of 105 frames, which are encoded into a multi-scale latent representation comprising 16 latent frames at 4x compression, 2 latent frames at 2x compression, and 9 latent frames at 1x compression.

Anysplat [3]. We employ Anysplat for 3D reconstruction and rendering for video outputs. The model utilizes 84 context frames along with the first frame serving as an anchor for reconstruction, which correspond to 84 latent frames plus the first frame for the video generation setting. We estimate camera poses using VGGT [7] and use them for coordinate regularization.

Gen3C [4]. We follow the official implementation of Gen3C and its instruction on Multiview Images Input. First, we run VGGT [7] to get the depth information, camera intrinsics, and extrinsics of the context frame. We also include



Figure 1. Failure and success cases.

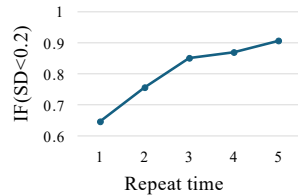


Figure 2. Ablation on grounding success rate over repeat time.

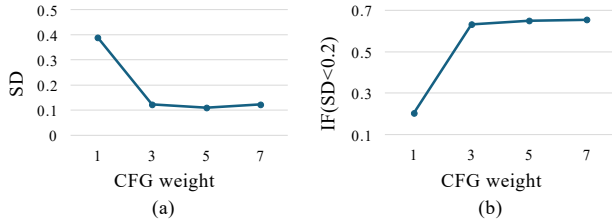


Figure 3. Ablation on different CFG weights .

Table 1. Ablation on context sampling.

	SD↓	IF↑	IF(SD< 0.2)↑
Non-continuous	0.1099	0.7327	0.6486
Continuous	0.3246	0.8497	0.4600

model is trained to generate 49 frames, we run three infer-
ences to get the complete output.

1.4. Experiments.

Repeat time. Consistent with prior findings that Veo3 [2] improves with more repeats [8], we observe the same trend. For object grounding, we compute IF (SD < 0.2) across repeated runs and deem a case successful if any repeat succeeds. As shown in Fig. 2, increasing the number of repeats from 1 to 5 raises IF (SD < 0.2), indicating that repeated sampling substantially boosts performance.

CFG strength. We investigate the impact of different CFG scales in Fig. 3. While a weight of 1 (no CFG) yields suboptimal results, performance improves significantly once the scale reaches a sufficient level (e.g., 3) and remains robust across a wide range (e.g., 3–7). This suggests that the use of CFG itself is more critical than fine-tuning the exact weight.

Non-continuous context sampling. We investigate the effectiveness of non-continuous context sampling during training, as shown in Table 1. Our results demonstrate that non-continuous sampling significantly improves geometric consistency with the context and achieves highly faithful grounding. We hypothesize that this improvement stems from the model developing a more robust spatial understanding through such diverse and non-continuous context modeling.

the first frame of the generation as the first context frame. During the VGGT inference, we include context frames and ground truth frames as a single run to get the camera poses of the ground truth. During the Gen3C inference, we use the camera poses of ground truth as camera control. We input 85 context frames and output 121 frames, all at resolution 576x320.

TrajectoryCrafter [10]. We use the same preprocessing algorithm as Gen3C, *i.e.* VGGT, to get the point cloud from 85 context frames and camera pose control from 121 ground truth frames, all in one pass to make sure they are in the same corodinate system. Then we project the point cloud based on ground truth camera and get masks. Since the

Random zero camera pose. Existing camera control methods typically normalize the camera pose of the first frame to zero during both training and inference. We find that, particularly with limited datasets, fixing the first frame to the origin restricts spatial exploration, thereby limiting generalization in complex scenarios such as 360° rotations. To address this, we propose randomly selecting a reference frame to serve as the zero pose, which effectively alleviates this issue and improves generalization.

Computational analysis. Although our model processes relatively long contexts (default 337 frames) and generates extended sequences (default 181 frames), we maintain computational costs within an acceptable range. Inference requires approximately 34GB of VRAM and takes 2 minutes on a single A100 GPU using CFG and 50 denoising steps, without optimizations such as VAE slicing, tiling, or dynamic model loading [11]. This efficiency stems primarily from two factors: (1) the use of a moderate resolution (416×256), which we find sufficient for high visual quality and experimental validation; and (2) the Wan2.2 VAE, which achieves high spatial and temporal compression ratios while preserving visual fidelity.

Failure cases. Our method still suffers from artifacts such as temporal discontinuities (Fig. 1(a)) and incorrect grounding (Fig. 1(b)) for some cases.

References

- [1] Gilad Baruch, Zhuoyuan Chen, Afshin Dehghan, Tal Dimry, Yuri Feigin, Peter Fu, Thomas Gebauer, Brandon Joffe, Daniel Kurz, Arik Schwartz, et al. Arkitscenes: A diverse real-world dataset for 3D indoor scene understanding using mobile RGB-D data. In *Advances in Neural Information Processing Systems Datasets and Benchmarks Track*, 2021. 1
- [2] Google. Veo 3. <https://aistudio.google.com/models/veo-3>, 2025. 1, 2
- [3] Lihan Jiang, Yucheng Mao, Linning Xu, Tao Lu, Kerui Ren, Yichen Jin, Xudong Xu, Mulin Yu, Jiangmiao Pang, Feng Zhao, et al. AnySplat: Feed-forward 3d gaussian splatting from unconstrained views. *SIGGRAPH Asia*, 2025. 1
- [4] 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 *CVPR*, 2025. 1
- [5] Qwen team. Qwen3-vl. <https://github.com/QwenLM/Qwen3-VL>, 2025. 1
- [6] Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Fei Wu Yu, Haiming Zhao, Jianxiao Yang, et al. WAN: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025. 1
- [7] Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. VGGT: Visual geometry grounded transformer. In *CVPR*, 2025. 1
- [8] Thaddäus Wiedemer, Yuxuan Li, Paul Vicol, Shixiang Shane Gu, Nick Matarese, Kevin Swersky, Been Kim, Priyank Jaini, and Robert Geirhos. Video models are zero-shot learners and reasoners. *arXiv preprint arXiv:2509.20328*, 2025. 2
- [9] Chandan Yeshwanth, Yueh-Cheng Liu, Matthias Nießner, and Angela Dai. ScanNet++: A high-fidelity dataset of 3d indoor scenes. In *ICCV*, 2023. 1
- [10] Mark YU, Wenbo Hu, Jinbo Xing, and Ying Shan. Trajectorycrafter: Redirecting camera trajectory for monocular videos via diffusion models. *ICCV*, 2025. 2
- [11] 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 *NeurIPS*, 2025. 1, 3