WARPHAMMER: DENSIFYING SCENE WARPS WITH 3D OBJECT PRIORS FOR EXTREME VIEW SYNTHESIS

Anonymous authors
Paper under double-blind review

ABSTRACT

Projection-conditioned novel view synthesis (NVS) warps an explicit 3D reconstruction of the input view into the target camera and conditions a generator on the warped rendering. This works well for small viewpoint changes but degrades sharply under large orbital motion: the warp becomes sparse around the orbited object, where hidden surfaces dominate the new view and mirror-like artifacts emerge, causing the generator to lose both pixel content and the implicit camera cue carried by the warp. We introduce WarpHammer, a training-free framework that resolves this failure mode by augmenting the warped scene with an explicit 3D reconstruction of the object obtained from a native 3D generative prior (e.g., SAM3D). The reconstructed object adds missing foreground surfaces and occludes background points that should no longer be visible, restoring both appearance and camera cues without fine-tuning the base model. The same explicit object representation further unlocks a capability current NVS pipelines do not support: incorporating auxiliary views of the object from sources outside the target scene, for example, a casual snapshot of a car paired with a manufacturer studio shot of the same model. We process the reference and auxiliary images jointly with a pretrained multi-view geometry foundation model, which predicts a unified point cloud that we fuse into the 3D object reconstruction. This yields substantially more faithful geometry than single-image reconstruction, without requiring user-provided camera poses for the auxiliary views. On five benchmarks, WarpHammer produces stable novel views at viewpoint deviations where strong baselines collapse, and is the first scene-level NVS method that can naturally fuse auxiliary, unposed object views from an external source, recovering their relative pose rather than requiring it as input.

1 INTRODUCTION

Novel view synthesis (NVS) from a single image or a sparse set of views is fundamentally underconstrained: the input does not contain sufficient evidence to recover unobserved regions of the scene. Recent work has therefore shifted from purely per-scene reconstruction to learned visual priors. These include pretrained image generators (Rombach et al., 2022; Podell et al., 2024), multi-view generative models (Liu et al., 2023; 2024b; Long et al., 2024; Zhou et al., 2025), and video generators (Blattmann et al., 2023; Chen et al., 2023; Liang et al., 2026). Trained on large-scale visual data, these models learn statistical regularities that enable plausible completion of missing scene content.

There are two dominant paradigms in NVS. Camera-conditioned methods encode the target camera directly, to infer scene structure implicitly (Jin et al., 2025; Gao et al., 2024; Wu et al., 2025; Zhou et al., 2025; Jiang et al., 2025a). Projection-conditioned methods instead construct an explicit 3D proxy from the input, render it from the target viewpoint, and condition generation on the resulting warp (Ren et al., 2025; Yu et al., 2024; Hu et al., 2025; Chen et al., 2025b; Ma et al., 2025; Yu et al., 2025; Yan et al., 2025; Cao et al., 2025; Liu et al., 2024a; Zhang et al., 2025b). This explicit geometric conditioning enables strong camera control and state-of-the-art NVS performance (Ren et al., 2025; Hu et al., 2025). We build on this paradigm.

However, both paradigms remain brittle under large viewpoint extrapolation. Camera-conditioned models can fail due to out-of-distribution motion (Wu et al., 2026), while projection-conditioned

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.

methods introduce artifacts since the warped point cloud develops holes and sparse regions as the view angle increases (Zhang et al., 2025c; Vuong et al., 2026).

We focus primarily on object-anchored scenes, where a foreground object constitutes the main subject of the image and therefore more clearly determines the perceptual fidelity of the synthesized view. Such scenes are ubiquitous in product visualization, object-centric capture, and content creation. They are also where large viewpoint changes matter most, and such changes are particularly challenging here because substantial portions of the foreground object are unobserved in the input.

Importantly, the projected warp also acts as a geometric scaffold that communicates the prescribed camera motion to the generator (Vuong et al., 2026). This provides an advantage over camera-conditioned image/video generative models, which often produce a plausible or likely camera motion rather than faithfully following the user-specified trajectory (Vuong et al., 2026). Consequently, when the warp becomes severely sparse at extreme viewpoints, the generator loses both appearance evidence and reliable geometric guidance, and pose and reconstruction errors increase rapidly under large orbital motion (Fig. 1). Yet this failure is not directly measured by standard camera-control metrics: rotation error, for example, can remain low even when the object is rendered in an incorrect orientation. To quantify it, we introduce OrbErr, an orbital-fidelity metric for object-centric NVS (Appendix D.1).

We introduce WarpHammer, a training-free framework that mitigates this failure mode particularly on object-anchored scenes, by densifying the warped conditioning signal with an explicit object-centric 3D prior. Our key observation is that the scene warp is most unreliable exactly where object geometry matters most: on the newly visible surfaces of the central object. We augment the standard scene cache with a reconstructed 3D object from a native 3D generative model. The reconstructed mesh plays a dual role: it adds geometry for newly visible object surfaces, and it occludes background points that should no longer be visible once the camera has orbited past the object, thus eliminating the background leakage that scene-only warps suffer from at large angles.

A key aspect of our design is the separation between geometry and appearance. Native 3D generators (e.g., Hunyuan3D, TRELLIS) provide reliable shape but their texture and illumination may deviate from the reference scene, and they often lack photorealism (Sobol et al., 2026). We therefore use the reconstruction as geometric guidance only, leaving the diffusion model's scene-identity conditioning responsible for appearance.

Beyond improving single-image extreme-view synthesis, the explicit object representation enables a second capability that standard warp-based pipelines do not naturally support: incorporating an auxiliary object view from outside the target scene. For example, a dashcam image may contain a car from one viewpoint, while a catalog or manufacturer image may show the same or a related car from another viewpoint. Such external views can contain object-specific information that a generative prior cannot infer from the reference alone, such as the shape of unobserved surfaces, a logo, or a license plate. These auxiliary images are typically unposed and may be visually mismatched with the reference scene. We jointly process the reference and auxiliary object views with a pretrained multi-view geometry foundation model, which aligns the auxiliary geometry to the reference object without camera calibration from the user. The generated video remains anchored to the appearance and context of the reference scene; when the auxiliary appearance is mismatched, only its geometry is used (§3.3).

Our contributions are:

- We substantially strengthen extreme-view synthesis in existing projection-conditioned scene NVS methods without additional training, by densifying sparse target-view warps with an explicit object-centric 3D reconstruction.
- We define and address an auxiliary-view NVS setting, where an unposed object image from an external source is fused into the reference scene as structural evidence rather than as a calibrated scene view.
- We propose an appearance-aware fusion strategy that separates object geometry from appearance, enabling auxiliary views to improve structure without forcing inconsistent color, texture, or illumination into the output. We also introduce **SynView-X**, a benchmark for object-centric NVS under large viewpoint changes.

- We introduce OrbErr, an orbital-fidelity metric for object-centric NVS, together with a benchmark for unposed object-view fusion, and evaluate across single-view and auxiliary-view regimes on five datasets.

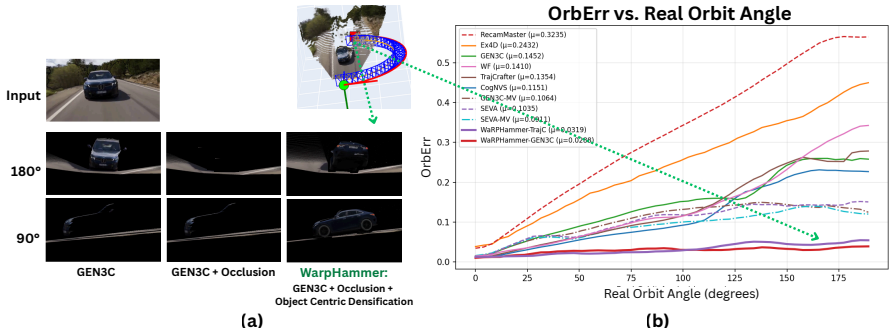

Figure 1: Object-centric densification improves extreme-view synthesis. **(a) Qualitative:** GEN3C shows sparsity and artifacts; WarpHammer remains geometry-consistent. **(b) Quantitative:** OrbErr vs. angle: baselines degrade beyond $\sim 110^\circ$, while WarpHammer maintains low error.

2 Related Work

Sparse-view and multi-view NVS. Classical NVS methods such as NeRFs (Mildenhall et al., 2020; Barron et al., 2022) and 3D Gaussian Splatting (Kerbl et al., 2023) achieve high-quality rendering from dense posed views, while generalizable methods reduce the number of inputs using learned priors (Yu et al., 2021; Wang et al., 2021; Zhou & Tulsiani, 2023). However, these methods still assume views from the same calibrated scene. WarpHammer instead allows an auxiliary object image to be unposed, out-of-scene, and appearance-mismatched, treating it as structural evidence rather than as another calibrated scene view.

Object-centric generative view synthesis. A parallel line of work uses generative priors for single-image object view synthesis. Zero-1-to-3 (Liu et al., 2023) learns camera-conditioned novel-view generation from a single object image. SyncDreamer (Liu et al., 2024b) and Wonder3D (Long et al., 2024) improve multi-view consistency by jointly generating synchronized views, normals, or other geometry-aware intermediates. SV3D (Voleti et al., 2024) extends this idea with video diffusion and explicit orbital camera control, while SV4D (Xie et al., 2025) produces multi-frame, multi-view consistent videos for dynamic objects. Zero-to-Hero (Sobol et al., 2024) further improves single-image NVS at challenging viewpoints through training-free attention-map manipulation at inference. These methods demonstrate that generative models encode strong object-level 3D priors, but they typically synthesize isolated objects on clean backgrounds and stop short of integrating objects back into a real reference scene with its original lighting and surroundings. WarpHammer instead uses object-centric 3D generation as structural guidance inside a scene-level NVS pipeline, while appearance, lighting, and background context remain anchored to the reference image.

Camera-controlled and projection-conditioned generative NVS. Recent video diffusion models enable camera-controllable generation from images or monocular videos by injecting camera motion through learned pose embeddings, Plücker rays, relative position encodings, or dedicated control branches (Wang et al., 2024; He et al., 2024; Bahmani et al., 2025b;a; Li et al., 2025; Luo et al., 2025; He et al., 2025). These approaches provide flexible camera control, but the camera signal is mostly implicit, and object geometry may drift under large viewpoint changes.

Projection-conditioned methods make the camera conditioning more explicit. They lift the source image or video into a 3D proxy, render it from the target camera, and use the resulting warp to guide a diffusion model. ViewCrafter (Yu et al., 2024) and TrajectoryCrafter (Yu et al., 2025) follow this strategy for single-image and monocular-video inputs, respectively, while GEN3C (Ren et al., 2025) maintains a persistent 3D cache for world-consistent generation. SEVA (Zhou et al., 2025) accepts posed input views and target cameras, and related methods explore learned camera encoders

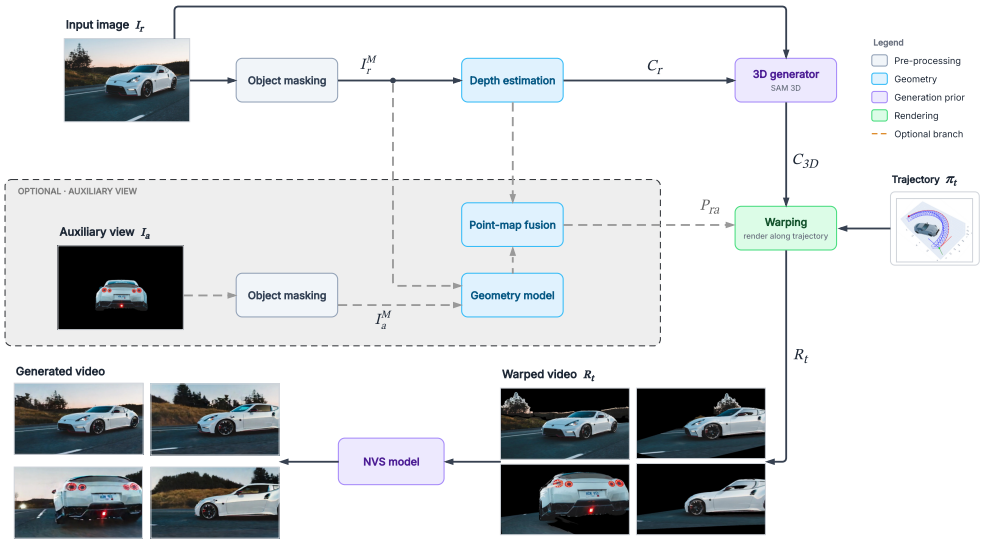

Figure 2: **Overview of WarpHammer.** From a reference image I_r and target trajectory $\{\pi_t\}_{t=1}^{T}$ (and an optional auxiliary object image I_a), WarpHammer builds a densified point-cloud cache by augmenting the scene warp with an explicit 3D object reconstruction, optionally fused with I_a via a multi-view geometry foundation model, and renders it to condition a video diffusion prior.

or inference-time motion guidance (Zhang et al., 2025a; Song et al., 2025). However, these methods ultimately condition on geometry visible in the input view. As the target camera moves to large angles, hidden object surfaces must be hallucinated by the diffusion prior. WarpHammer addresses this limitation by augmenting the scene warp with an explicit object-centric 3D reconstruction, optionally strengthened by an auxiliary object view.

Geometry-completed proxies for large-viewpoint synthesis. Several recent methods also improve large-viewpoint synthesis by constructing more complete geometric proxies. CogNVS (Chen et al., 2025a) reconstructs co-visible regions from monocular video, inpaints hidden regions, and applies test-time finetuning for temporal coherence. EX-4D (Hu et al., 2025) uses a depth-watertight mesh as guidance for extreme-viewpoint 4D synthesis. Concurrent work FreeOrbit4D (Cao et al., 2026) builds on VACE (Jiang et al., 2025b) depth conditioning to complete foreground and background geometry for arbitrary camera redirection in monocular videos. These methods share a similar motivation, but primarily derive the proxy from monocular video or depth and target dynamic or 4D content. WarpHammer targets single-image static-scene NVS, using a native 3D generative prior for object geometry and optionally refining it with an unposed external object image.

3 METHOD

Figure 2 gives an overview of the pipeline. We first describe the single-image pipeline (§3.1–§3.2), then show how the same machinery extends to an auxiliary, unposed object view from a different scene (§3.3).

Problem setup. We assume the reference scene image $I_r \in \mathbb{R}^{H\times W\times 3}$ contains a dominant foreground object, the *central object*, around which the target camera trajectory orbits. We further assume that the central object is rigid, that an off-the-shelf segmenter isolates it, and that enough of it is visible in I_r for the 3D prior to reconstruct it. Given I_r and a 6-DoF camera trajectory $\{\pi_t\}_{t=1}^{T}$ specified *relative* to the input view, our goal is to synthesize a novel-view video V_g of T RGB frames at resolution $H \times W$ that follows the trajectory while preserving scene appearance and object identity. §3.3 extends this setup with an additional, uncalibrated auxiliary image I_a of the same object, possibly captured in a different scene and under different illumination.

3.1 PRELIMINARIES: PROJECTION-CONDITIONED NVS VIA SCENE CACHES

We build WarpHammer on top of two representative backbones: GEN3C (Ren et al., 2025) and TrajectoryCrafter (Yu et al., 2025), both of which lift the reference image into a point cloud via monocular depth estimation, render it along the target trajectory, and condition a video diffusion prior on the resulting warped buffers. We describe the lifted geometry abstractly as a *cache*

$$\mathcal{C} = \{P, A\}, \qquad P \subset \mathbb{R}^3, \;\; A \in \mathbb{R}^{|P|\times 3}, \tag{1}$$

where P are the lifted scene points and A their RGB attributes; trajectory poses $\{\pi_t\}$ are expressed relative to the reference camera. Rendering $\mathcal{C}$ along $\{\pi_t\}_{t=1}^T$ produces warped renderings $\{R_t\}_{t=1}^T$ and per-pixel confidence masks $\{M_t\}_{t=1}^T$, $M_t \in [0,1]^{H\times W}$, which are binary validity masks in the base pipelines, that anchor synthesized frames to the input scene while leaving the diffusion prior to fill newly visible regions. The video diffusion prior $\mathcal{G}$ maps these buffers, together with I_r, to the output video, $V_g = \mathcal{G}(I_r, \{R_t, M_t\}_{t=1}^T)$.

3.2 DENSIFYING THE OBJECT WITH A 3D OBJECT PRIOR

The object is where sparsity hurts most, and where it matters most: background regions typically remain supported as the camera translates, but the object's hidden surfaces become dominant the moment the camera orbits, and these are precisely the surfaces that define its identity. We address this directly by densifying the foreground cache with an explicit 3D reconstruction from a point-map-conditioned native object generator (Chen et al., 2025c) which recovers a complete object representation from a single image.

Object reconstruction and placement. We isolate the central foreground object in I_r using an off-the-shelf segmenter, producing masked image I_r^M. We then pass I_r^M together with the masked object points P_r of the reference cache $\mathcal{C}_r$ (Equation (1)) to a point-cloud-conditioned 3D generator f_{3D}, instantiated with SAM3D (Chen et al., 2025c), an image-to-3D object generator that accepts a point map as conditioning and returns a textured mesh aligned to it; we denote the resulting object geometry by $P_{3D} = f_{3D}(I_r^M, P_r)$. SAM3D predicts the object's pose and scale in the frame of its conditioning point map, so the mesh is returned directly in the reference camera frame at the scale of P_r, with no separate registration step. Other image-to-3D generators can also serve as f_{3D} (Appendix B.1).

Cache fusion and visibility. We form the fused cache by inserting the densified object into the reference cache,

$$\mathcal{C}_f \;=\; \mathcal{C}_r \;\cup\; \mathcal{C}_{3D}, \qquad \mathcal{C}_{3D} = \{P_{3D}\}, \tag{2}$$

where $\mathcal{C}_{3D}$ carries only the reconstructed geometry and no color attributes, and resolve visibility during rendering. Because SAM3D natively returns a mesh, we render the densified object portion of $\mathcal{C}_f$ from that mesh instead of splatting the sampled P_{3D} as points; its conditioning values are defined below. We splat $\mathcal{C}_r$ and rasterize the mesh into a shared depth buffer, and discard a reference point when its depth exceeds the mesh depth along the same ray by more than a small tolerance. This removes stale reference points that should be hidden once the camera has orbited past the object.

Because of this occlusion, reconstruction errors do more than add incorrect geometry: a misplaced surface can also remove correct reference points behind it. The warped video $\{R_t\}$ and mask $\{M_t\}$ are produced jointly from $\mathcal{C}_f$ and handed to the diffusion prior as in the base pipeline, with the foreground region now densely supported across the full trajectory.

Geometry-focused conditioning. While SAM3D yields reliable shape estimates, its texture and color can deviate from the reference scene's illumination and appearance and generally fall short of the reference image's photorealism. We therefore use the reconstructed mesh without its color. Let Ω_t be the pixels of target frame t that the mesh covers but no warped reference point does, and let $\hat{D}_t$ be the depth rendered from the mesh. Within Ω_t,

$$R_t(u) = g\big(\hat{D}_t(u)\big), \tag{3}$$

where g maps the rendered depth to a normalized gray value replicated across the three color channels (Appendix B.3). The depth code is thus written into the rendered RGB conditioning frames $\{R_t\}$,

not into the cache attributes A, so that the densified region carries geometric structure but no donor-specific color identity. The diffusion prior, conditioned on I_r, then fills in the appearance of these regions, keeping it coherent with the reference scene. When no foreground object is available for augmentation, $\mathcal{C}_{3D} = \emptyset$ and $\mathcal{C}_f = \mathcal{C}_r$, so our formulation reduces to the underlying projection-conditioned pipeline.

3.3 Fusing an Auxiliary Object View

Beyond single-image extreme-view synthesis, the object representation from §3.2 also lets WarpHammer incorporate an uncalibrated auxiliary image I_a of the same object, even when it is captured outside the reference scene, e.g., from a catalog, studio image, or web retrieval.

Since I_a has no known pose relative to I_r and may differ in background, lighting, color, or texture, WarpHammer uses it as additional *geometric evidence* rather than as a calibrated second view; whether its RGB is retained depends on the appearance match, as described below.

Joint geometry from an unposed auxiliary view. We mask the auxiliary image to the object, I_a^M, and pass it, together with the masked reference I_r^M, to a feed-forward multi-view geometry model f_g, instantiated with VGGT (Wang et al., 2025):

$$[P_r^g,\, P_a^g] \;=\; f_g(I_r^M,\, I_a^M). \tag{4}$$

The model jointly predicts dense object pointmaps P_r^g, P_a^g in a shared coordinate frame, so the auxiliary pointmap arrives already registered to the reference object, but in VGGT's own frame and scale. Masking the inputs to f_g is important in practice: it focuses the model on object surfaces and stabilizes the predicted relative pose for wide baselines, including hard front-to-back configurations.

Cross-view point-map fusion. We keep confidence-filtered points and fuse the auxiliary geometry with the reference cache by mapping it into the reference cache's frame. The anchor is the reference object, which is observed twice: as P_r, lifted by the model's depth, and as P_r^g, predicted by VGGT. Both are indexed by the same reference pixels, so they provide direct 3D correspondences, from which we estimate a similarity transform (scale s, rotation $\mathbf{R}$, translation $\mathbf{t}$) in closed form (Appendix B.2):

$$\hat{P}_a \;=\; s\,\mathbf{R}\,P_a^g + \mathbf{t}, \qquad P_{ra} \;=\; P_r \,\cup\, \hat{P}_a. \tag{5}$$

We call P_{ra} the *fused object point map*: it combines the surfaces visible in the reference view (P_r) with those contributed by the auxiliary view ($\hat{P}_a$), in a single frame and scale.

Aligned 3D generation with auxiliary support. The fused object point map replaces P_r as the conditioning of f_{3D} in the densification step,

$$P_{3D} \;=\; f_{3D}(I_r^M,\, P_{ra}), \tag{6}$$

where $\hat{P}_a$ supplies real evidence for surfaces that are weakly observed or absent from the reference view. The fused cache becomes $\mathcal{C}_f = \mathcal{C}_r \cup \mathcal{C}_{3D} \cup \mathcal{C}_a$, with the auxiliary cache $\mathcal{C}_a = \{\hat{P}_a, A_a\}$ also expressed in the reference frame; rendering and conditioning then proceed exactly as in §3.2.

Appearance-neutral auxiliary conditioning. The auxiliary view reuses the depth code g of Equation (3), with Ω_t given by the footprint of its aligned geometry projected into each target view. Unlike the mesh texture, which is never used, auxiliary RGB is replaced by this depth code only when a configuration flag enables neutralization, i.e., when the donor color is not to be trusted; in that case, the mask inside the footprint is set to a constant $\alpha < 1$ instead of full validity, so that the prior uses the depth code as a structural hint rather than reproducing it as gray appearance. Otherwise, auxiliary RGB and its validity mask are retained. Pixels outside the footprint fall back to the reference scene cache with zero auxiliary confidence. Appendix B.3 and B.4 define these buffers and measure the suppression of donor color."

4 Experiments

We compare WarpHammer against TrajectoryCrafter (Yu et al., 2025), GEN3C (Ren et al., 2025), ReCamMaster (Bai et al., 2025), EX-4D (Hu et al., 2025), CogNVS (Chen et al., 2025a), SEVA (Zhou

et al., 2025), and WorldForge (Song et al., 2025), spanning projection-conditioned and camera-conditioned paradigms. FreeOrbit4D (Cao et al., 2026) shares a similar idea of completing the geometric proxy, but accepts only video as input; for completeness, we therefore compare with it qualitatively only (Appendix G). For comparison with posed multi-view conditioning, we use the multi-view variants of GEN3C and SEVA, which accept a second input view *with known pose from the same scene*, denoted GEN3C-MV and SEVA-MV, and report them as stronger-supervision references with GT camera poses.

Benchmarks. We evaluate on five benchmarks spanning synthetic, scanned, and real-world captured data. We introduce SynView-X, a benchmark for object-centric NVS under large orbital viewpoint changes, constructed from CAD assets drawn from ShapeNet (Chang et al., 2015) and Objaverse (Deitke et al., 2023), augmented with scanned objects from GSO (Downs et al., 2022). Existing benchmarks do not specifically target extreme orbital deviations around a dominant foreground object. SynView-X fills this gap by providing ground-truth multi-view sequences with controlled large-angle coverage. For scanned and real-world evaluation we use OmniObject3D (Wu et al., 2023), Co3D (Reizenstein et al., 2021), Objectron (Ahmadyan et al., 2021), and Mip-NeRF 360 (Barron et al., 2022). For each dataset, evaluation pairs are constructed by selecting images with large orbital viewpoint differences; one serves as the reference input and the other as the target view. Dataset details and construction protocol are provided in Appendix C.

Metrics. Our evaluation measures three complementary aspects of performance: image fidelity, geometric accuracy, and perceptual video quality. For image fidelity, we evaluate PSNR, SSIM, and LPIPS on the masked object region at the target viewpoint, and CLIP-F (Bai et al., 2025) for semantic alignment with the target view, focusing the evaluation on the synthesized object rather than potentially unrelated background, where applicable (details in Appendix D and F).

For geometric accuracy, we propose OrbErr, a metric tailored to object-centric orbital trajectories. A faithful orbit must satisfy two conditions simultaneously: the camera must follow the intended orbital viewing direction, and the rendered object must place its visual features at spatial locations consistent with that viewpoint. OrbErr therefore combines two complementary terms:

$$E_{\mathrm{orb}}(t) = \tfrac{1}{2} E_{\mathrm{euler}}(t) \; + \; \tfrac{1}{2} E_{\mathrm{disp}}(t), \tag{7}$$

where E_{euler} measures normalized wrapped angular deviation in azimuth and elevation from monocular pose estimates (Lin et al., 2025), and E_{disp} measures object-region feature displacement using mutual nearest-neighbor matches between DINOv2 (Oquab et al., 2023) patch features of the rendered and ground-truth frames. The displacement of these mutual matches is combined with a coverage penalty, preventing the metric from rewarding methods that match only a small subset of easy patches. Reporting the orbital-angle and feature-displacement terms jointly penalizes partial failures that rotation error or PSNR alone misses, including incorrect object orientation, mirror-like duplications, and incomplete object reconstruction.

4.1 Results

Quantitative. Table 1 reports single-view results averaged across all five benchmarks. Both WarpHammer-GEN3C and WarpHammer-TC improve PSNR, SSIM, and LPIPS over their backbones, and achieve the lowest SynView-X OrbErr. For completeness, Table 7 in the Appendix reports VBench (Huang et al., 2024) video-quality metrics.

Table 2 reports results in the auxiliary-view setting. Without any ground-truth camera poses or a full-scene second image, WarpHammer-GEN3C outperforms all baselines without GT poses and approaches the image similarity of the GT-pose variants, which use in-scene views. This supports our key insight: an unposed, background-free object image, registered rather than supplied as a calibrated view, can recover much of the benefit of a ground-truth second view with known pose. Appearance-mismatched auxiliaries and a multi-object example are shown in Appendix B.4 and A.

Qualitative. Fig. 3 shows a qualitative comparison along a wide orbital trajectory. Warp-based baselines preserve the input view at small offsets but progressively lose the object as the warp collapses: mirror-like duplications and background leakage dominate at wide angles. Camera-conditioned generation (ReCamMaster) avoids warping artifacts but fails to follow the prescribed

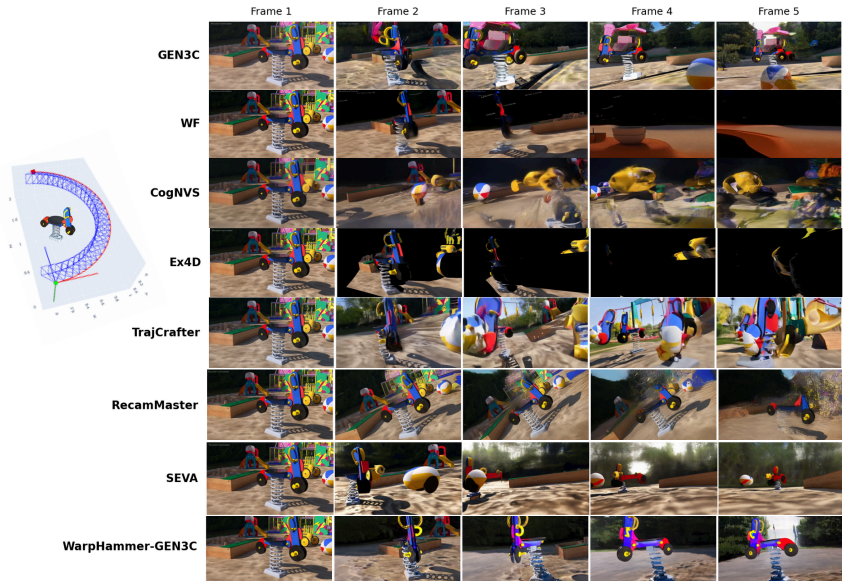

Figure 3: **Qualitative comparison along an extreme orbital trajectory.** Single view NVS. Left: the target camera follows a wide arc around the central object (red curve). Right: five frames sampled from small to extreme orbital offsets.

Table 1: **Single-view** mean NVS results across 5 datasets (CO3D, Objectron, 360°, OmniObject3D, SynView-X). WarpHammer variants are shaded. Best results in **bold**. Green superscripts give differences from each backbone. †OrbErr is reported only on SynView-X.

		Image Similarity			Semantic	Camera Pose		
Group	Method	PSNR↑	SSIM↑	LPIPS↓	CLIP-F↑	RotErr↓	TransErr↓	OrbErr† ↓
Camera-cond.	ReCamM.	7.91	0.481	0.851	93.42	163.49	0.102	0.3460
Camera-cond.	SEVA	7.68	0.480	0.736	95.10	167.28	0.047	0.2855
Warp-based	EX-4D	7.88	0.464	0.790	95.18	190.15	0.083	0.3192
Warp-based	CogNVS	8.08	0.513	0.768	94.26	176.51	0.063	0.3063
Warp-based	WF	7.64	0.524	0.568	**95.73**	119.85	0.085	0.3067
Warp-based	TrajCr.	7.58	0.460	0.791	93.62	177.85	0.069	0.3022
Warp-based	GEN3C	7.45	0.416	0.767	94.40	184.10	0.093	0.3058
Ours	WarpHammer-TC	10.79$^{+3.21}$	0.522	0.510	93.25	125.52$^{-52.33}$	0.077	0.2288
Ours	WarpHammer-GEN3C	**11.67**$^{+4.22}$	**0.589**	**0.444**	94.34	**91.21**$^{-92.89}$	**0.024**	**0.2045**

Table 2: **Auxiliary-view** NVS results averaged across 5 datasets (CO3D, Objectron, 360°, OmniObject3D, SynView-X). WarpHammer is the shaded row. Gray rows receive GT poses and in-scene second views and are stronger-supervision references, not equal-input baselines. **Bold**: best among non-gray rows.

		Image Similarity			Semantic	Camera Pose	
Setting	Method	PSNR↑	SSIM↑	LPIPS↓	CLIP-F↑	RotErr↓	TransErr↓
GT poses	SEVA-MV	14.60	0.705	0.389	94.61	193.04	0.040
GT poses	GEN3C-MV	14.31	0.696	0.446	93.06	176.96	0.069
No GT poses	SEVA-MV	10.56	0.513	0.519	92.89	97.13	0.050
No GT poses	GEN3C-MV	9.97	0.502	0.546	93.30	81.38	0.088
No GT poses	WarpHammer-GEN3C	**14.09**	**0.635**	**0.389**	**94.84**	**80.03**	**0.035**

orbital trajectory, producing a near-static sequence that barely changes viewpoint. WarpHammer-GEN3C maintains a coherent, identity-preserving rendering across the full orbital range, follows the

prescribed camera trajectory, and produces plausible background completion in newly visible regions. More qualitative results and animated comparisons are provided in Appendix H.

4.2 Ablation

We ablate the two core components of WarpHammer on 50 SynView-X samples: the object-centric 3D prior and the VGGT-based multi-view fusion. We study four variants across two settings: *Single-view Full* vs. *w/o 3D prior*, and *Auxiliary-view Full* vs. *w/o VGGT*. Results are reported in Tab. 3.

Effect of the 3D prior (single-view). Removing the explicit 3D object reconstruction (*Single-view w/o 3D prior*) sharply degrades both image fidelity and camera accuracy simultaneously, confirming that the 3D prior serves a dual role: it supplies the missing surface geometry for newly visible object regions, and restores the implicit camera cue that the warp normally carries but loses when it goes sparse at extreme angles.

Effect of auxiliary-view fusion (VGGT). Comparing *Aux-view Full* against *Aux-view w/o VGGT* isolates the contribution of VGGT-based geometric alignment between the auxiliary view and the reference scene. Even without VGGT, SAM3D still produces a plausible object shape; what VGGT provides is the joint alignment that places the auxiliary geometry consistently inside the scene coordinate frame. Without it, the orbital trajectory drifts relative to the object, confirming that aligned conditioning, not 3D content alone, is the critical ingredient for orbital fidelity.

Table 3: Ablation study on SynView-X (n=50). Best results per setting in **bold**.

		Image Similarity			Semantic	Camera Pose		
Setting	Variant	PSNR↑	SSIM↑	LPIPS↓	CLIP-F↑	RotErr↓	TransErr↓	OrbErr↓
Aux-view	Full	**16.69**	**0.560**	**0.368**	**94.94**	**25.36**	**0.134**	**0.132**
Aux-view	w/o VGGT	9.80	0.360	0.609	94.83	49.37	0.154	0.192
Single-view	Full	**12.06**	**0.468**	**0.482**	94.50	**31.05**	**0.132**	**0.1338**
Single-view	w/o 3D prior	10.05	0.360	0.600	**94.84**	49.45	0.154	0.1772

Original-RGB and appearance-neutralized auxiliary conditioning are compared in Figure 5 in the Appendix.

5 Summary and Conclusions

WarpHammer addresses a key limitation of projection-conditioned NVS: under large viewpoint changes, the scene warp becomes sparse around the foreground object, weakening both the geometric support and the implicit camera cue provided to the generator. We overcome this by densifying the warped conditioning signal with an explicit object-centric 3D reconstruction. The resulting representation also opens a practical new use case for NVS: incorporating an unposed, out-of-scene auxiliary image of the object. Across synthetic, scanned, and real-world benchmarks, our results show that object-centric geometry completion is an effective path toward more robust, flexible, and practically usable NVS systems.

Limitations. WarpHammer relies on accurate object segmentation and the quality of the external 3D reconstruction; segmentation failures or incorrect geometry propagate into the synthesized views (Appendix B.5). The method is most effective for scenes with a dominant foreground object and may be less reliable for cluttered scenes, thin structures, or transparent materials. The auxiliary-view setting is sensitive to geometric ambiguities when the auxiliary and reference views are either nearly identical or highly dissimilar. Finally, adding SAM3D to the pipeline increases inference time relative to the base backbone, which may be a consideration in latency-sensitive applications.

AI USE STATEMENT

Generative AI assistants were used during the manuscript rephrasing and editing, code debugging, figure preparation. The authors retain responsibility for the final text, methods, figures, code, experimental results, and scientific conclusions.

ETHICS STATEMENT

Our work advances novel view synthesis for generating images and videos of objects and scenes from novel camera viewpoints. As with other generative visual technologies, the proposed method could potentially be misused to create manipulated visual content. We therefore encourage responsible use and appropriate disclosure of synthetically generated imagery. The method may also inherit biases or limitations from the pretrained generative models on which it relies. Our research does not involve human-subject studies or the collection of new personally identifiable information.

REPRODUCIBILITY STATEMENT

Implementation details for both backbones, the appearance-neutral auxiliary conditioning scheme, and the confidence masking procedure are given in Appendix B and B.3. Dataset construction and the pair-selection protocol for all five benchmarks are described in Appendix C, and the definition of our OrbErr metric, together with its motivation relative to RotErr, is given in Appendix D.1. WarpHammer is training-free and builds on publicly available pretrained models.

REFERENCES

Adel Ahmadyan, Liangkai Zhang, Artsiom Ablavatski, Jianing Wei, and Matthias Grundmann. Objectron: A large scale dataset of object-centric videos in the wild with pose annotations. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 7822–7831, 2021.

Sherwin Bahmani, Ivan Skorokhodov, Guocheng Qian, Aliaksandr Siarohin, Willi Menapace, Andrea Tagliasacchi, David B. Lindell, and Sergey Tulyakov. AC3D: Analyzing and improving 3d camera control in video diffusion transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025a.

Sherwin Bahmani, Ivan Skorokhodov, Aliaksandr Siarohin, Willi Menapace, Guocheng Qian, Michael Vasilkovsky, Hsin-Ying Lee, Chaoyang Wang, Jiaxu Zou, Andrea Tagliasacchi, David B. Lindell, and Sergey Tulyakov. VD3D: Taming large video diffusion transformers for 3d camera control. In *International Conference on Learning Representations*, 2025b.

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

Jonathan T. Barron, Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5470–5479, 2022.

Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, Dominik Lorenz, Yam Levi, Zion English, Vikram Voleti, Adam Letts, et al. Stable video diffusion: Scaling latent video diffusion models to large datasets. *arXiv preprint arXiv:2311.15127*, 2023.

Chenjie Cao, Chaohui Yu, Shang Liu, Fan Wang, Xiangyang Xue, and Yanwei Fu. MVGen-Master: Scaling multi-view generation from any image via 3D priors enhanced diffusion model. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025. arXiv:2411.16157.

Wei Cao, Hao Zhang, Fengrui Tian, Yulun Wu, Yingying Li, Shenlong Wang, Ning Yu, and Yaoyao Liu. Freeorbit4d: Training-free arbitrary camera redirection for monocular videos via geometry-complete 4d reconstruction. 2026.

Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015.

Haoxin Chen, Menghan Xia, Yingqing He, Yong Zhang, Xiaodong Cun, Shaoshu Yang, Jinbo Xing, Yaofang Liu, Qifeng Chen, Xintao Wang, Chao Weng, and Ying Shan. VideoCrafter1: Open diffusion models for high-quality video generation. *arXiv preprint arXiv:2310.19512*, 2023.

Kaihua Chen, Tarasha Khurana, and Deva Ramanan. Reconstruct, inpaint, test-time finetune: Dynamic novel-view synthesis from monocular videos. In *Advances in Neural Information Processing Systems*, 2025a.

Luxi Chen, Zihan Zhou, Min Zhao, Yikai Wang, Ge Zhang, Wenhao Huang, Hao Sun, Ji-Rong Wen, and Chongxuan Li. FlexWorld: Progressively expanding 3D scenes for flexible-view synthesis. *arXiv preprint arXiv:2503.13265*, 2025b.

Xingyu Chen, Fu-Jen Chu, Pierre Gleize, Kevin J Liang, Alexander Sax, Hao Tang, Weiyao Wang, Michelle Guo, Thibaut Hardin, Xiang Li, et al. Sam 3d: 3dfy anything in images. *arXiv preprint arXiv:2511.16624*, 2025c.

Matt Deitke, Dustin Schwenk, Jordi Salvador, Luca Weihs, Oscar Michel, Eli VanderBilt, Ludwig Schmidt, Kiana Ehsani, Aniruddha Kembhavi, and Ali Farhadi. Objaverse: A universe of annotated 3d objects. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 13142–13153, 2023.

Laura Downs, Anthony Francis, Nate Koenig, Brandon Kinman, Ryan Hickman, Krista Reymann, Thomas B McHugh, and Vincent Vanhoucke. Google scanned objects: A high-quality dataset of 3d scanned household items. In *2022 International Conference on Robotics and Automation (ICRA)*, pp. 2553–2560. Ieee, 2022.

Ruiqi Gao, Aleksander Holynski, Philipp Henzler, Arthur Brussee, Ricardo Martin-Brualla, Pratul Srinivasan, Jonathan T. Barron, and Ben Poole. CAT3D: Create anything in 3D with multi-view diffusion models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. arXiv:2405.10314.

Hao He, Yinghao Xu, Yuwei Guo, Gordon Wetzstein, Bo Dai, Hongsheng Li, and Ceyuan Yang. CameraCtrl: Enabling camera control for text-to-video generation. *arXiv preprint arXiv:2404.02101*, 2024.

Hao He, Ceyuan Yang, Shanchuan Lin, Yinghao Xu, Meng Wei, Liangke Gui, Qi Zhao, Gordon Wetzstein, Lu Jiang, and Hongsheng Li. CameraCtrl II: Dynamic scene exploration via camera-controlled video diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2025.

Tao Hu, Haoyang Peng, Xiao Liu, and Yuewen Ma. EX-4D: Extreme viewpoint 4D video synthesis via depth watertight mesh. *arXiv preprint arXiv:2506.05554*, 2025.

Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, Nattapol Chanpaisit, et al. Vbench: Comprehensive benchmark suite for video generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 21807–21818, 2024.

Hanwen Jiang, Hao Tan, Peng Wang, Haian Jin, Yue Zhao, Sai Bi, Kai Zhang, Fujun Luan, Kalyan Sunkavalli, Qixing Huang, and Georgios Pavlakos. RayZer: A self-supervised large view synthesis model. *arXiv preprint arXiv:2505.00702*, 2025a.

Zeyinzi Jiang, Zhen Han, Chaojie Mao, Jingfeng Zhang, Yulin Pan, and Yu Liu. Vace: All-in-one video creation and editing. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 17191–17202, 2025b.

Haian Jin, Hanwen Jiang, Hao Tan, Kai Zhang, Sai Bi, Tianyuan Zhang, Fujun Luan, Noah Snavely, and Zexiang Xu. LVSM: A large view synthesis model with minimal 3D inductive bias. In *International Conference on Learning Representations (ICLR)*, 2025. arXiv:2410.17242.

Wolfgang Kabsch. A solution for the best rotation to relate two sets of vectors. *Acta Crystallographica Section A*, 32(5):922–923, 1976.

Bernhard Kerbl, Georgios Kopanas, Thomas Leimk"uhler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4):1–14, 2023.

Teng Li, Guangcong Zheng, Rui Jiang, Shuigen Zhan, Tao Wu, Yehao Lu, Yining Lin, and Xi Li. RealCam-I2V: Real-world image-to-video generation with interactive complex camera control. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 28785–28796, 2025.

Jinglin Liang, Zijian Zhou, Rui Huang, Shuangping Huang, and Yichen Gong. Orbitnvs: Harnessing video diffusion priors for novel view synthesis. *arXiv preprint arXiv:2603.19613*, 2026.

Haotong Lin, Sili Chen, Junhao Liew, Donny Y Chen, Zhenyu Li, Guang Shi, Jiashi Feng, and Bingyi Kang. Depth anything 3: Recovering the visual space from any views. *arXiv preprint arXiv:2511.10647*, 2025.

Ruoshi Liu, Rundi Wu, Basile Van Hoorick, Pavel Tokmakov, Sergey Zakharov, and Carl Vondrick. Zero-1-to-3: Zero-shot one image to 3d object. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9298–9309, 2023.

Xi Liu, Chaoyi Zhou, and Siyu Huang. 3DGS-enhancer: Enhancing unbounded 3D gaussian splatting with view-consistent 2D diffusion priors. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024a. arXiv:2410.16266.

Yuan Liu, Cheng Lin, Zijiao Zeng, Xiaoxiao Long, Lingjie Liu, Taku Komura, and Wenping Wang. SyncDreamer: Generating multiview-consistent images from a single-view image. In *International Conference on Learning Representations*, 2024b.

Xiaoxiao Long, Yuan-Chen Guo, Cheng Lin, Yuan Liu, Zhiyang Dou, Lingjie Liu, Yuexin Ma, Song-Hai Zhang, Marc Habermann, Christian Theobalt, and Wenping Wang. Wonder3D: Single image to 3d using cross-domain diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

Yawen Luo, Jianhong Bai, Xiaoyu Shi, Menghan Xia, Xintao Wang, Pengfei Wan, Di Zhang, Kun Gai, and Tianfan Xue. CamCloneMaster: Enabling reference-based camera control for video generation. *arXiv preprint arXiv:2506.03140*, 2025.

Baorui Ma, Huachen Gao, Haoge Deng, Zhengxiong Luo, Tiejun Huang, Lulu Tang, and Xinlong Wang. You see it, you got it: Learning 3D creation on pose-free videos at scale. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025. arXiv:2412.06699.

Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. NeRF: Representing scenes as neural radiance fields for view synthesis. In *European Conference on Computer Vision*, pp. 405–421. Springer, 2020.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. SDXL: Improving latent diffusion models for high-resolution image synthesis. In *International Conference on Learning Representations (ICLR)*, 2024.

Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, et al. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024.

Jeremy Reizenstein, Roman Shapovalov, Philipp Henzler, Luca Sbordone, Patrick Labatut, and David Novotny. Common objects in 3d: Large-scale learning and evaluation of real-life 3d category reconstruction. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 10901–10911, 2021.

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*, pp. 6121–6132, 2025.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

Ido Sobol, Chenfeng Xu, and Or Litany. Zero-to-hero: Enhancing zero-shot novel view synthesis via attention map filtering. *Advances in Neural Information Processing Systems*, 37:30522–30553, 2024.

Ido Sobol, Kihyuk Sohn, Yoav Blum, Egor Zakharov, Max Bluvstein, Andrea Vedaldi, and Or Litany. Realiz3D: 3D generation made photorealistic via domain-aware learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026.

Chenxi Song, Yanming Yang, Tong Zhao, Ruibo Li, and Chi Zhang. WorldForge: Unlocking emergent 3d/4d generation in video diffusion model via training-free guidance. *arXiv preprint arXiv:2509.15130*, 2025.

Vikram Voleti, Chun-Han Yao, Mark Boss, Adam Letts, David Pankratz, Dmitry Tochilkin, Christian Laforte, Robin Rombach, and Varun Jampani. SV3D: Novel multi-view synthesis and 3d generation from a single image using latent video diffusion. In *European Conference on Computer Vision*, 2024.

Khiem Vuong, Deva Ramanan, and Srinivasa Narasimhan. Fixanything: 3d-consistent rendering refinement via video generative priors. 2026.

Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 5294–5306, 2025.

Qianqian Wang, Zhicheng Wang, Kyle Genova, Pratul P. Srinivasan, Howard Zhou, Jonathan T. Barron, Ricardo Martin-Brualla, Noah Snavely, and Thomas Funkhouser. IBRNet: Learning multi-view image-based rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4690–4699, 2021.

Zhouxia Wang, Ziyang Yuan, Xintao Wang, Tianshui Chen, Menghan Xia, Ping Luo, and Ying Shan. MotionCtrl: A unified and flexible motion controller for video generation. In *ACM SIGGRAPH Conference Papers*, 2024.

Rundi Wu, Ruiqi Gao, Ben Poole, Alex Trevithick, Changxi Zheng, Jonathan T. Barron, and Aleksander Holynski. CAT4D: Create anything in 4D with multi-view video diffusion models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025. arXiv:2411.18613.

Tong Wu, Jiarui Zhang, Xiao Fu, Yuxin Wang, Jiawei Ren, Liang Pan, Wayne Wu, Lei Yang, Jiaqi Wang, Chen Qian, et al. Omniobject3d: Large-vocabulary 3d object dataset for realistic perception, reconstruction and generation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 803–814, 2023.

Zirui Wu, Zeren Jiang, Martin R. Oswald, and Jie Song. From rays to projections: Better inputs for feed-forward view synthesis. *arXiv preprint arXiv:2601.05116*, 2026.

Jianfeng Xiang, Zelong Lv, Sicheng Xu, Yu Deng, Ruicheng Wang, Bowen Zhang, Dong Chen, Xin Tong, and Jiaolong Yang. Structured 3d latents for scalable and versatile 3d generation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 21469–21480, 2025.

Yiming Xie, Chun-Han Yao, Vikram Voleti, Huaizu Jiang, and Varun Jampani. SV4D: Dynamic 3d content generation with multi-frame and multi-view consistency. In *International Conference on Learning Representations*, 2025.

Yunzhi Yan, Zhen Xu, Haotong Lin, Haian Jin, Haoyu Guo, Yida Wang, Kun Zhan, Xianpeng Lang, Hujun Bao, Xiaowei Zhou, and Sida Peng. StreetCrafter: Street view synthesis with controllable video diffusion models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025. arXiv:2412.13188.

Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelNeRF: Neural radiance fields from one or few images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4578–4587, 2021.

Mark Yu, Wenbo Hu, Jinbo Xing, and Ying Shan. Trajectorycrafter: Redirecting camera trajectory for monocular videos via diffusion models. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 100–111, 2025.

Wangbo Yu, Jinbo Xing, Li Yuan, Wenbo Hu, Xiaoyu Li, Zhipeng Huang, Xiangjun Gao, Tien-Tsin Wong, Ying Shan, and Yonghong Tian. ViewCrafter: Taming video diffusion models for high-fidelity novel view synthesis. *arXiv preprint arXiv:2409.02048*, 2024.

Songchun Zhang, Huiyao Xu, Sitong Guo, Zhongwei Xie, Pengwei Liu, Hujun Bao, Weiwei Xu, and Changqing Zou. SpatialCrafter: Unleashing the imagination of video diffusion models for scene reconstruction from limited observations. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 27794–27805, 2025a.

Xiang Zhang, Yang Zhang, Lukas Mehl, Markus Gross, and Christopher Schroers. High-fidelity novel view synthesis via splatting-guided diffusion. *arXiv preprint arXiv:2502.12752*, 2025b.

Yuqi Zhang, Guanying Chen, Jiaxing Chen, Chuanyu Fu, Chuan Huang, and Shuguang Cui. CloseUpShot: Close-up novel view synthesis from sparse-views via point-conditioned diffusion model. *arXiv preprint arXiv:2511.13121*, 2025c.

Zibo Zhao, Zeqiang Lai, Qingxiang Lin, Yunfei Zhao, Haolin Liu, Shuhui Yang, Yifei Feng, Mingxin Yang, Sheng Zhang, Xianghui Yang, et al. Hunyuan3d 2.0: Scaling diffusion models for high resolution textured 3d assets generation. *arXiv preprint arXiv:2501.12202*, 2025.

Jensen Zhou, Hang Gao, Vikram Voleti, Aaryaman Vasishta, Chun-Han Yao, Mark Boss, Philip Torr, Christian Rupprecht, and Varun Jampani. Stable virtual camera: Generative view synthesis with diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 12405–12414, 2025.

Zhizhuo Zhou and Shubham Tulsiani. SparseFusion: Distilling view-conditioned diffusion for 3d reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

Supplementary Material: WarpHammer: Densifying Scene Warps with 3D Object Priors for Extreme View Synthesis

In this supplementary material, we provide a multi-object example, implementation and dataset details, and additional quantitative and qualitative results.

A Multi-Object View Synthesis

The object-centric representation extends naturally to scenes with multiple foreground objects. We reconstruct each object from its reference and auxiliary crops, place its two-view geometry into the scene with one similarity transform, and render the composed cache to condition video generation. The auxiliary image is not a view of the target scene: it shows the same objects in a different arrangement, each seen from a different viewpoint. Since each object is registered with its own transform, the auxiliary does not correspond to any single camera of the target scene. Note that the auxiliary image is not used as a whole: only the segmented crops of the individual objects are used, so its background, table and object arrangement provide no information about the target scene. Figure 4 illustrates a 200° orbit around a picnic scene containing an apple, mug, book and basket. The full video is provided in the supplementary HTML gallery.

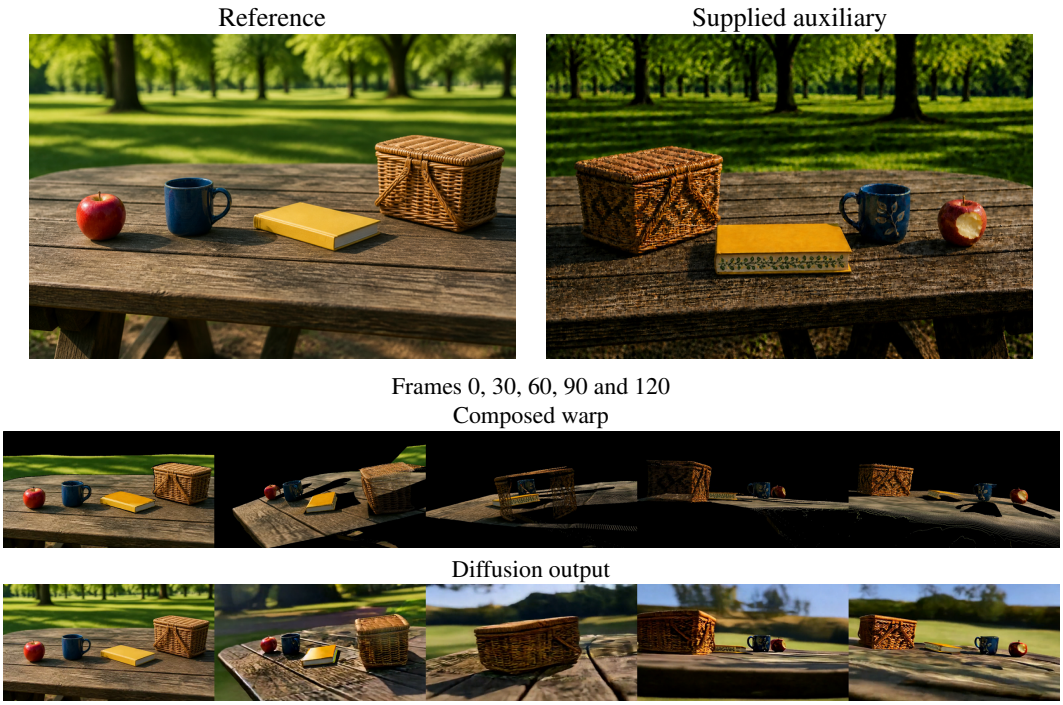

Figure 4: Multi-object synthesis over a 200° orbit. The regenerated auxiliary supplies additional object views.

B Implementation Details

B.1 Choice of 3D Generator

We instantiate the 3D generator f_{3D} with SAM3D (Chen et al., 2025c), which accepts the masked reference point cloud P_r as conditioning and returns the object mesh directly in the reference camera frame and at the scale of P_r. Other image-to-3D generators, such as TRELLIS (Xiang et al., 2025) and Hunyuan3D-2 (Zhao et al., 2025), can also be used as f_{3D}. Since these generators reconstruct

objects in a canonical frame, their output requires an additional registration step that aligns it with P_r before it is fused into the cache.

B.2 Cross-View Point-Map Alignment

The similarity transform in Equation (5) maps the VGGT prediction into the frame and scale of the reference cache. Because P_r and P_r^g are indexed by the same reference pixels, they provide direct 3D correspondences on the set $\mathcal{U}$ of object pixels valid in both. The transform is computed in closed form. We first center both point sets at their medians over $\mathcal{U}$, $\bar{P}_r$ and $\bar{P}_r^g$; s is the ratio of their median distances to these centers, $\mathbf{R}$ is the least-squares rotation between the centered, rescaled correspondences, obtained from a singular value decomposition with the Kabsch algorithm (Kabsch, 1976), and $\mathbf{t} = \bar{P}_r - s\mathbf{R}\bar{P}_r^g$ aligns the centers. Medians make the fit robust to outlier points. Before alignment, confidence filtering discards VGGT points below the 25th percentile of the per-view confidence.

B.3 Appearance-Neutral Auxiliary Conditioning

We use depth-coded auxiliary conditioning to incorporate geometric support without copying mismatched donor color. We refer to it as appearance-neutral conditioning, and to replacing auxiliary RGB with it as neutralization. For appearance-matched captures, as in the main results, we retain auxiliary RGB: $\tilde{R}_t^a$ is the auxiliary rendering itself. The auxiliary footprint is obtained automatically by projecting the aligned auxiliary geometry into each target view; it determines where the auxiliary conditioning is applied. A configuration flag enables neutralization when needed; auxiliary RGB is retained otherwise.

Motivation. An external object image can provide useful geometry despite differences in color, texture or illumination. Neutralization separates this geometric information from the donor's RGB appearance, allowing the reference image to guide the synthesized appearance. Section B.4 compares RGB and depth-coded conditioning under matched and mismatched donor colors.

Depth-coded buffer and confidence mask. Let R_t^a denote the auxiliary RGB rendering at target frame t, D_t^a the projected auxiliary depth cue, R_t^s the rendering from the reference scene cache, and Ω_t^a the rendered auxiliary object footprint. We define the depth-coded auxiliary buffer $\tilde{R}_t^a$ and mask M_t^a jointly as

$$\big(\tilde{R}_t^a(u,v),\ M_t^a(u,v)\big) = \begin{cases} \big(g(D_t^a(u,v)),\ \alpha\big), & (u,v) \in \Omega_t^a, \\ \big(R_t^s(u,v),\ 0\big), & \text{otherwise.} \end{cases} \tag{8}$$

Here, $g(\cdot)$ converts the auxiliary depth rendering into an appearance-neutral RGB conditioning signal, and $\alpha \in (0,1)$ is a constant confidence inside the auxiliary footprint, used in place of the validity mask when the donor color is not to be trusted. Setting $\alpha < 1$ lets the prior use the depth code as a structural hint rather than reproduce it as gray appearance. When auxiliary RGB is retained, the renderer's validity mask is used instead.

Implementation. We normalize auxiliary depth using the valid depth range at cue construction:

$$\bar{D}_t^a(u,v) = \frac{D_t^a(u,v) - d_{\min}}{d_{\max} - d_{\min} + \epsilon}, \tag{9}$$

where $d_{\min}$ and $d_{\max}$ are the minimum and maximum valid depths and ϵ ensures numerical stability. In the cache-based car experiments, normalization is applied before projection; the normalized depth attributes are then rendered along the trajectory. The depth-coding function g repeats the normalized scalar depth value across the RGB channels:

$$g(D_t^a(u,v)) = \big(\bar{D}_t^a(u,v), \bar{D}_t^a(u,v), \bar{D}_t^a(u,v)\big). \tag{10}$$

Within the auxiliary footprint Ω_t^a, the depth-coded rendering replaces the auxiliary RGB contribution in GEN3C's warped conditioning, with M_t^a providing the corresponding confidence mask.

Depth coding and luminance conversion are distinct: the former replaces RGB with a geometric signal, whereas the latter repeats the grayscale luminance of R_t^a across RGB channels, removing chromatic color while retaining intensity variation. The grayscale-conditioned example in Figure 5 uses the latter.

B.4 Ablation of Appearance-Neutral Conditioning

Suppressing donor-color transfer. Figure 15 shows two cross-instance examples pairing an orange reference car with a pink auxiliary car. These qualitative results illustrate appearance preservation with a different donor instance; the supplementary HTML gallery provides the full videos.

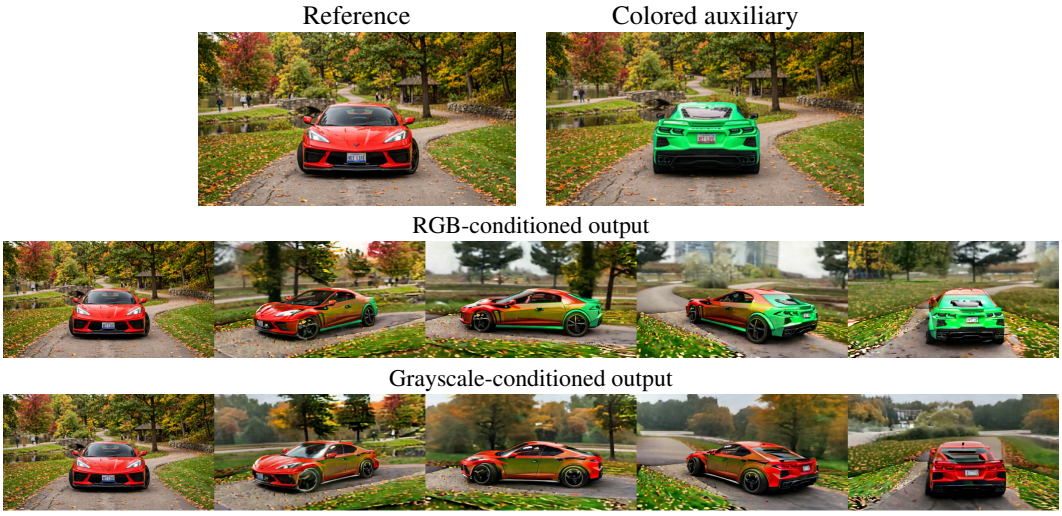

Figure 5: Grayscale conditioning in auxiliary-view synthesis. Above: reference and hue-rotated auxiliary. Rows show RGB-conditioned synthesis with the colored auxiliary and the grayscale-conditioned result, obtained from the luminance of the original auxiliary.

B.5 Segmentation Sensitivity

We ablate the sensitivity of our method to errors in the segmentation masks. We evaluate 6 conditions on one front/rear pair with fixed perturbation and generation seeds: clean masks, erosion at radii of $1\%, 4\%, 8\%$ and interior dropout at radii of $1\%, 4\%$ of the object's equivalent radius $\sqrt{A/\pi}$. Reconstruction and synthesis are rerun for each condition with the same preprocessing. Erosion and dropout up to 4% retain a recognizable single vehicle throughout the trajectory, despite changes in appearance and proportions, while 8% erosion strongly alters the body (Figure 6).



Figure 6: Segmentation sensitivity on a fixed-seed local example. Left: reference and auxiliary masks, with IoU relative to their clean counterparts. Right: corresponding output frames. Percentages specify perturbation radius, not mask-area change. Erosion and dropout up to 4% preserve a recognizable vehicle, while 8% erosion alters the body.

C Dataset Details

C.1 Dataset Creation

For each benchmark, we construct evaluation pairs by selecting two views of the same object (or scene) separated by a large orbital offset. The first view serves as the *reference* input, the second as the auxiliary target, and a sequence of intermediate frames provides the ground-truth orbital video. All camera poses are converted to a common OpenCV world-to-camera convention. The dataset-specific selection and pre-processing protocols are described below.

CO3D (Reizenstein et al., 2021) (160 pairs). For each sequence, we greedily select up to 20 pairs that maximize frame separation while satisfying $|\text{yaw}| \in [100°, 190°]$ and $|\text{pitch}| \leq 40°$. Pairs are further filtered by foreground-mask coverage in $[5\%, 95\%]$ using the segmentation masks provided with CO3D. The ground-truth orbital video consists of the ordered real frames lying between the two selected views.

Mip-NeRF 360 (Barron et al., 2022) (57 pairs). Camera poses are taken from the COLMAP reconstructions provided with the dataset, and the dense scene point cloud is used to estimate the 3D scene centre, which serves as the per-view crop anchor. Foreground masks are extracted by an off-the-shelf saliency segmentation model and made cross-view consistent by projecting the foreground 3D points from one view into the other and retaining only mask components that overlap with these projections. Intermediate ground-truth frames are obtained by traversing the shorter azimuthal arc between the two selected cameras.

OmniObject3D (Wu et al., 2023) (500 pairs). We use the rendered images and camera poses provided by OmniObject3D. For each object, a pair satisfying our orbital-rotation criteria is sampled at random from all qualifying combinations. The reference view is composited onto a randomly sampled perspective crop from an HDRI panorama to simulate a realistic background, while the auxiliary

view is placed on a white background, yielding the unposed object input used by WarpHammer. Ground-truth video frames are taken as the rendered frames closest, in spherical distance, to a linearly interpolated arc between the two selected views.

Objectron (Ahmadyan et al., 2021) (9 pairs). We decode the original sequences and convert their camera transforms to our common OpenCV convention. Per sequence, we select the pair with the largest in-range |yaw|, prioritizing pairs whose annotated 3D bounding box is well-framed. Foreground masks are obtained from an off-the-shelf saliency model, and intermediate frames are taken directly from the source monocular videos.

SynView-X (900 pairs). SynView-X aggregates synthetic and scanned assets rendered to support extreme orbital viewpoint changes. It is composed of 777 pairs from ShapeNetCore (Chang et al., 2015), 20 pairs from GSO (Downs et al., 2022), 103 pairs from Objaverse (Deitke et al., 2023). The ShapeNet, GSO, and Objaverse assets are rendered offline; for ShapeNet and GSO, yaw and pitch are sampled deterministically per asset to ensure reproducibility, with the orbit direction matching the conditioning convention used by GEN3C so that the stored angles can be passed directly as conditioning signals. Together, these subsets cover man-made objects, real-scanned assets, and procedurally generated scenes, giving SynView-X the diversity needed to stress-test extreme-viewpoint synthesis under controlled conditions.

Pair-selection summary. Across all five benchmarks, evaluation pairs lie on an orbital arc around the object of interest, with yaw separation in $[100^\circ, 190^\circ]$ and bounded pitch. The reference view is provided as input to all methods, the auxiliary view is used as the unposed conditioning input for WarpHammer auxiliary-view variants. In the auxiliary-view setting, endpoint image similarity is therefore measured at the view whose masked object is also supplied, without pose, as input; its pose, background and all intermediate frames are held out. Intermediate frames define the ground-truth orbital video for video-quality and OrbErr computation.

D Evaluation Metrics Details

D.1 Orbital Evaluation: Limitations of RotErr and Definition of OrbErr

The standard relative rotation error

$$E_{\mathrm{rot}}(t) = \cos^{-1}\left(\frac{\mathrm{tr}(\hat{R}_t R_t^\top) - 1}{2}\right) \tag{11}$$

measures the geodesic distance on $\mathrm{SO}(3)$ between the predicted and ground-truth rotation matrices for frame t. Following CameraCtrl (He et al., 2024) and ReCamMaster (Bai et al., 2025), we compute this per-frame error and sum it over all frames of a sequence,

$$\mathrm{RotErr} = \sum_{t=1}^{T} E_{\mathrm{rot}}(t), \tag{12}$$

and report the mean sequence-level RotErr across evaluation samples, in degrees. Because Equation (12) accumulates over T frames rather than averaging, the reported values are sequence totals and are not bounded by the 180° maximum of a single-frame geodesic distance; the same accumulation convention is applied to every method, so the comparison remains like-for-like. This is a useful metric for general camera-control evaluation, but it is not fully aligned with object-centric orbital novel view synthesis. In our setting, the desired motion is primarily defined by the orbital viewpoint around the object, i.e., its azimuth and elevation, rather than by an arbitrary 3D rotation.

TransErr. Following CameraCtrl (He et al., 2024), we measure the translation error as the summed squared ℓ_2 distance between the ground-truth and estimated camera translations,

$$\mathrm{TransErr} = \sum_{t=1}^{T} \left\| T_t - \hat{T}_t \right\|_2^2, \tag{13}$$

where T_t and $\hat{T}_t$ are the ground-truth and estimated translation vectors at frame t. Because monocular pose estimates are defined only up to scale, both trajectories are first expressed relative to their first frame, and the estimated trajectory is rescaled so that its translation between the first two frames matches that of the ground truth.

Conflation of orbital and non-orbital errors. RotErr collapses all rotational degrees of freedom into a single scalar and does not distinguish whether the error originates in azimuth, elevation, or in-plane roll. Azimuth and elevation determine the object-centric viewpoint, while roll changes the image-plane orientation without necessarily changing the orbital position around the object. Concretely, consider two predictions with the same geodesic rotation error:

1. $\hat{R}_t^{(1)}$: azimuth error of $30°$, with no elevation or roll error. 2. $\hat{R}_t^{(2)}$: roll error of $30°$, with no azimuth or elevation error.

Both yield $E_{\text{rot}}(t) = 30°$, yet they represent different failure modes: the first observes the object from the wrong orbital viewpoint, while the second preserves the orbital direction but introduces an in-plane rotation. For object-centric orbital synthesis, these errors should not be treated as equivalent. In Figure 7, we illustrate such a failure case: GEN3C achieves a low rotation error under large viewpoint changes, yet the car orientation is incorrect, indicating that the metric fails to capture this error.

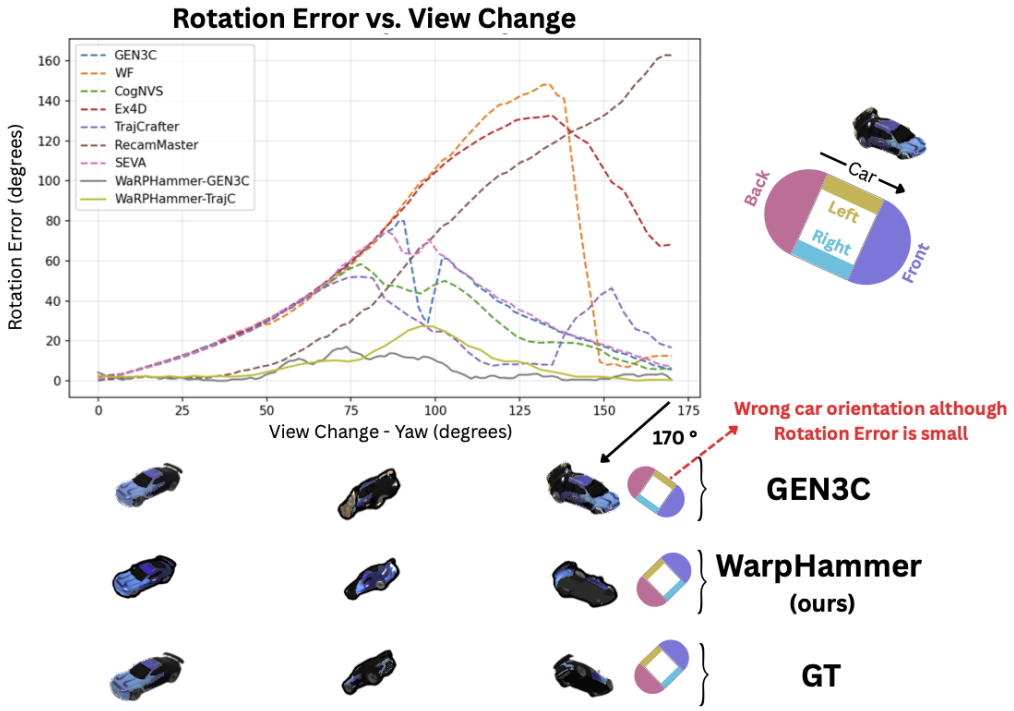

Figure 7: **Rotation error vs. view change.** Despite decreasing rotation error at large viewpoint changes, the predicted object orientation can still be incorrect (see comparison with GT), indicating that rotation error alone is insufficient for evaluating object-centric consistency in orbital settings.

Insensitivity to orbital position. RotErr depends only on rotation matrices and ignores camera translation entirely. Even if rotation is perfectly recovered, a camera centre that lies off the intended orbital arc is not penalized. Conversely, under monocular pose estimation, absolute translation is unrecoverable up to scale, so naively reporting translation error is unreliable. A metric tailored to orbital fidelity must therefore evaluate the orbital viewpoint without depending on metric translation.

OrbErr. We introduce OrbErr, a metric designed for object-centric orbital evaluation. OrbErr combines two complementary terms: (i) a rotation-derived Euler-angle error that measures azimuth

and elevation agreement, and (ii) a feature-displacement error that checks whether object features appear at their expected spatial positions in the generated frame. This allows the metric to penalize cases where the camera estimate appears plausible but the object is rendered from an incorrect orbital viewpoint.

Given monocular pose estimates $\{\hat{R}_t\}_{t=0}^{T}$ and target rotations $\{R_t\}_{t=0}^{T}$, where $t = 0$ denotes the reference view and $t = 1, \ldots, T$ the generated frames of Section 3.1, we first compute rotations relative to the initial frame:

$$\hat{R}_t^{\mathrm{rel}} = \hat{R}_t \hat{R}_0^{\top}, \qquad R_t^{\mathrm{rel}} = R_t R_0^{\top}. \tag{14}$$

We extract the camera forward direction as the third row of the relative rotation,

$$\hat{f}_t = e_3^{\top} \hat{R}_t^{\mathrm{rel}}, \qquad f_t = e_3^{\top} R_t^{\mathrm{rel}}, \qquad \hat{f}_t, f_t \in \mathbb{R}^3, \tag{15}$$

and decompose it into azimuth and elevation:

$$\hat{\theta}_t = \mathrm{atan2}(\hat{f}_{t,x}, \hat{f}_{t,z}), \qquad \theta_t = \mathrm{atan2}(f_{t,x}, f_{t,z}), \tag{16}$$

$$\hat{\phi}_t = \mathrm{atan2}\Big(-\hat{f}_{t,y}, \sqrt{\hat{f}_{t,x}^2 + \hat{f}_{t,z}^2}\Big), \qquad \phi_t = \mathrm{atan2}\Big(-f_{t,y}, \sqrt{f_{t,x}^2 + f_{t,z}^2}\Big). \tag{17}$$

The wrapped angular distance is

$$d_{\angle}(a, b) = \left|\mathrm{atan2}(\sin(a - b), \cos(a - b))\right|. \tag{18}$$

We define the normalized Euler-angle orbital error as

$$E_{\mathrm{euler}}(t) = \frac{1}{2} \frac{d_{\angle}(\hat{\theta}_t, \theta_t)}{\pi} + \frac{1}{2} \frac{d_{\angle}(\hat{\phi}_t, \phi_t)}{\pi}. \tag{19}$$

This term measures whether the generated frame corresponds to the intended orbital viewing direction, focusing on azimuth and elevation rather than arbitrary 3D rotation error.

To additionally verify that the rendered object appearance agrees with the intended viewpoint, we compute a feature-displacement error using DINOv2 (Oquab et al., 2023) patch features on a $G \times G$ grid, restricted to object masks obtained from SAM 2 (Ravi et al., 2024). Unlike a one-sided nearest-neighbor distance, which can match many ground-truth patches to the same generated patch, we use mutual nearest-neighbor matches between the ground-truth and generated object regions. Specifically, nearest neighbors are computed in both directions between the generated and ground-truth object patches, and only reciprocal matches are retained.

Let Ω_t^{gt} and Ω_t^{gen} denote the valid object patches in the ground-truth and generated frames, respectively. We ℓ_2-normalize all DINOv2 patch features and compute nearest neighbors by cosine similarity. For each ground-truth patch $i \in \Omega_t^{\mathrm{gt}}$, let $\mathrm{NN}_{\mathrm{gen}}(i)$ be its nearest generated patch. Similarly, for each generated patch $j \in \Omega_t^{\mathrm{gen}}$, let $\mathrm{NN}_{\mathrm{gt}}(j)$ be its nearest ground-truth patch. The set of mutual nearest-neighbor matches is

$$\mathcal{M}_t = \{(i, j) : j = \mathrm{NN}_{\mathrm{gen}}(i),\ i = \mathrm{NN}_{\mathrm{gt}}(j)\}. \tag{20}$$

The mutual feature-displacement error is

$$E_{\mathrm{disp}}^{\mathrm{mnn}}(t) = \frac{1}{|\mathcal{M}_t|} \sum_{(i,j) \in \mathcal{M}_t} \frac{\|p_i - p_j\|_2}{\sqrt{2}(G - 1)}, \tag{21}$$

where p_i and p_j are the two-dimensional grid coordinates of the matched patches. The normalization by $\sqrt{2}(G - 1)$ converts the displacement to units of the grid diagonal.

Because mutual matching may produce a small set of confident matches even when the generated object is incomplete, we also include a coverage penalty:

$$E_{\mathrm{cov}}(t) = 1 - \frac{2|\mathcal{M}_t|}{|\Omega_t^{\mathrm{gt}}| + |\Omega_t^{\mathrm{gen}}|}. \tag{22}$$

The final feature-displacement term is

$$E_{\mathrm{disp}}(t) = \frac{1}{2} E_{\mathrm{disp}}^{\mathrm{mnn}}(t) + \frac{1}{2} E_{\mathrm{cov}}(t). \tag{23}$$

If no mutual matches are found, we set $E_{\text{disp}}(t) = 1$.

The final orbital error averages both components:

$$E_{\text{orb}}(t) = \frac{1}{2}E_{\text{euler}}(t) + \frac{1}{2}E_{\text{disp}}(t). \tag{24}$$

OrbErr therefore evaluates both the estimated orbital viewing direction and the spatial consistency of object features. The mutual nearest-neighbor constraint makes the feature-displacement term more robust to many-to-one matches, repeated textures, and mirror-like duplications, while the coverage penalty discourages incomplete or weakly supported object reconstructions from receiving artificially low error. We use OrbErr as the primary measure of orbital fidelity, while still reporting RotErr for compatibility with prior camera-controlled video generation work.

Sanity check with known cameras. To verify that OrbErr responds to orbital and structural errors rather than to appearance changes, we implemented the full metric above and evaluated it on controlled inputs with exact cameras. We procedurally render three asymmetric, textured box assemblies (independent of our reconstructions) along a five-frame orbit, azimuth 0° to 120° in 30° steps. Keeping the first frame fixed, we modify frames 2–5 in three ways: (i) re-rendering them with a yaw, pitch, or roll offset of 5°, 15°, or 30°; (ii) appearance edits (hue $+10^\circ$, brightness $+10\%$, Gaussian blur σ=2 px); and (iii) structural edits with unchanged cameras (removing a central band covering 25% of the object width, or pasting a copy of the object shifted by 30% of the image width). Table 4 reports means over the 12 edited frames. OrbErr increases monotonically with the yaw and pitch offset on every object. Hue and brightness changes leave it near zero, below every geometric and structural edit. Missing and duplicated parts keep the camera exact, so RotErr is zero, whereas OrbErr assigns them errors comparable to a 5° orbital offset; this is the correct-camera, wrong-object failure illustrated in Figure 7. RotErr also assigns identical errors to equal roll and yaw offsets, while the Euler term is exactly zero under roll and the full OrbErr is lower for roll than for yaw at 15° and 30°. Finally, replacing the exact cameras with VGGT pose estimates, a different estimator from the Depth Anything 3 model used for the benchmark, yields 0.011 on the unedited sequence, an estimator floor well below the differences in Table 4.

Table 4: OrbErr sanity check on procedurally rendered objects with exact cameras. Means over three objects and four edited frames. RotErr is the per-frame geodesic error in degrees; the last column uses VGGT-estimated poses instead of exact cameras (unedited sequence: 0.011).

Edit	Amount	RotErr (°)	E_{euler}	OrbErr	OrbErr (VGGT)
Yaw	5°	5	0.014	0.114	0.124
	15°	15	0.042	0.165	0.170
	30°	30	0.083	0.219	0.229
Pitch	5°	5	0.014	0.122	0.130
	15°	15	0.042	0.164	0.169
	30°	30	0.083	0.223	0.225
Roll	5°	5	0.000	0.121	0.132
	15°	15	0.000	0.150	0.159
	30°	30	0.000	0.166	0.179
Hue	$+10^\circ$	0	0.000	0.003	0.015
Brightness	$+10\%$	0	0.000	0.002	0.013
Blur	σ=2 px	0	0.000	0.066	0.081
Missing part	25%	0	0.000	0.108	0.122
Duplicated object	30% shift	0	0.000	0.122	0.139

D.2 Pixel-Metric Sensitivity to Image Alignment

Image fidelity and orbital fidelity capture complementary aspects of NVS. Figure 8 illustrates the sensitivity of pixel-aligned scores using two transformations of the same photograph: a translation of 3% of the image width yields 14.3 dB PSNR, while horizontal reflection yields 13.3 dB. These controls involve no synthesis and show how alignment and orientation both affect the score. We therefore report target-view PSNR, SSIM and LPIPS together with orbital and video-quality metrics.



Figure 8: Pixel-metric sensitivity to alignment and orientation. Panels 1–3: reference, real near-endpoint photograph, and our $180°$ output. Panels 4–5: the photograph translated by 3% of its width (14.3 dB) and horizontally reflected (13.3 dB), scored against itself. The output-to-photograph score is illustrative: the real capture is not a pixel-aligned ground-truth endpoint.

E Conditioning Sparsity and Orbital Range

These analyses test the mechanism motivating WarpHammer: whether object completion restores projected foreground support, and whether that gain increases with orbital range. They measure conditioning coverage rather than generated-image quality, complementing the synthesis results.

E.1 Foreground Conditioning Support

Figure 9 measures conditioning support along a $180°$ orbit in one scene. Coverage is the fraction of target pixels receiving a projected point, measured over the full frame and within the object-only rendered footprint. We compare the reference cache with the densified cache used by WarpHammer.

At the sparsest foreground view, near $109°$, densification increases object coverage from 20.3% to 84.5%, a factor of 4.2. At the endpoint, coverage rises from 45.7% to 98.9%. The full-frame minimum remains approximately 18.9% near $106°$, showing that the added support is concentrated on the foreground object. The largest gap occurs before the endpoint, as the originally visible surfaces rotate out of view. These measurements quantify how object-centric completion strengthens the foreground conditioning available to the generator.

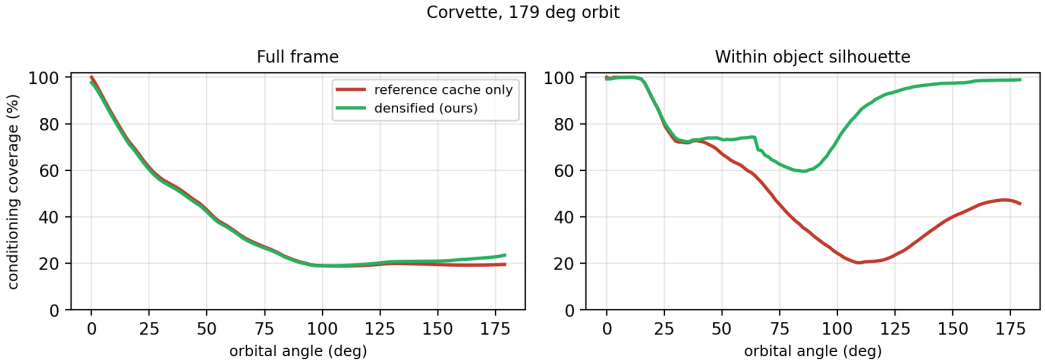

Figure 9: Conditioning coverage along the orbit for the reference cache (red) and densified cache (green). **Left:** full-frame coverage. **Right:** coverage within the rendered object footprint, increasing from 20.3% to 84.5% at the sparsest reference view near $109°$. Densification supplies foreground support where the scene warp is sparse.

E.2 Effect of Orbital Range

We render the same scene over $45°$, $90°$ and $180°$ orbits to measure how the benefit of densification changes with viewpoint range (Table 5 and Fig. 10). At each orbit's sparsest reference-only frame, object coverage increases from 70.4% to 74.2%, from 44.7% to 72.2%, and from 20.3% to 84.5%, respectively. The widening coverage gap shows that densification provides its largest support gain in the extreme-view regime targeted by WarpHammer.

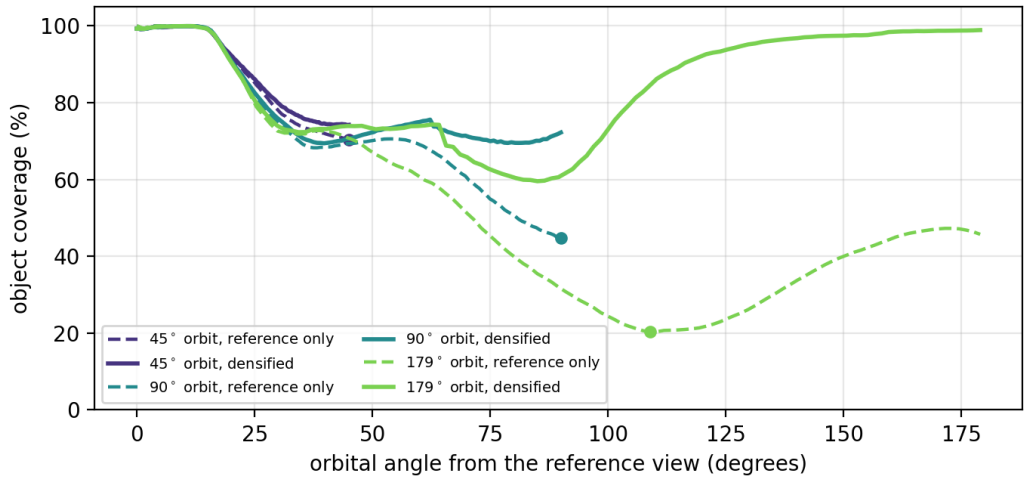

Figure 10: Effect of orbital range on foreground support. Dashed curves: reference-only cache; solid curves: densified cache. Markers identify the sparsest reference-only frame of each orbit. The coverage gain grows with orbital range. The later rise in reference-only coverage reflects reprojection of existing points, not newly reconstructed surfaces.

Orbital range	Worst angle	Reference-only	Densified
45°	45°	70.4%	74.2% (×1.05)
90°	90°	44.7%	72.2% (×1.62)
179°	109°	20.3%	84.5% (×4.17)

Table 5: Conditioning coverage within the rendered object footprint at the sparsest reference-only frame of each orbit, for the scene of Figure 9. The relative support gain from densification increases from $1.05\times$ at $45°$ to $4.17\times$ at $179°$.

F Additional Quantitative Results

We provide a per-dataset breakdown of the single-view novel view synthesis benchmark in Table 6, complementing the aggregated results reported in the main paper. The table reports image similarity (PSNR, SSIM, LPIPS), semantic consistency (CLIP-F), camera pose accuracy (RotErr, TransErr), feature matching (MatchPx), and VBench video-quality metrics (TempFlick, SubjCons, BGCons, ImagQual, AesQual, MotSmooth) across the five evaluation datasets: CO3D, Objectron, 360°, OmniObject3D, and SynView-X.

Across image-similarity metrics, WarpHammer achieves the best LPIPS on all five datasets and the best PSNR on four of five. On SynView-X, WarpHammer also achieves the best SSIM.

Video quality. Tables 7 and 8 report full-frame VBench metrics (Huang et al., 2024) for the single-view and auxiliary-view settings, averaged across the five datasets. These assess temporal, subject and background consistency, per-frame image quality, and motion smoothness, complementing the object-region and camera-motion metrics.

Table 6: Novel view synthesis benchmark results (single-view setting, no multi-view conditioning). ↑ higher is better, ↓ lower is better.

Dataset	Metric	TrajCr.	GEN3C	ReCamM.	EX-4D	CogNVS	SEVA	WarpHammer
CO3D	PSNR↑	5.86	6.52	3.92	5.25	5.70	6.42	**10.45**
	SSIM↑	0.416	0.394	0.407	0.412	0.472	0.439	**0.581**
	LPIPS↓	0.819	0.798	0.942	0.834	0.810	0.798	**0.429**
	CLIP-F↑	93.53	93.80	93.62	95.52	94.10	**95.90**	94.97
	RotErr↓	173.0	172.5	156.0	178.1	160.9	155.8	**140.27**
	TransErr↓	0.035	0.029	0.049	0.022	0.023	**0.016**	0.022
	MatchPx↑	0.004	**0.009**	0.001	0.006	0.004	0.008	0.008
	TempFlick↑	0.931	0.948	**0.971**	0.952	0.959	0.958	0.955
	SubjCons↑	0.806	0.753	0.710	0.789	0.807	**0.869**	0.829
	BGCons↑	0.908	0.889	0.848	0.887	0.871	**0.934**	0.909
	ImagQual↑	0.565	0.519	0.357	**0.579**	0.565	0.560	0.496
	AesQual↑	0.953	0.784	0.453	0.978	0.884	**0.986**	0.445
	MotSmooth↑	0.384	0.691	**0.916**	0.619	0.624	0.659	0.775
Objectron	PSNR↑	10.43	9.19	14.59	11.50	11.62	9.51	**15.14**
	SSIM↑	0.675	0.649	**0.783**	0.698	0.712	0.656	0.714
	LPIPS↓	0.765	0.741	0.774	0.750	0.738	0.731	**0.377**
	CLIP-F↑	94.14	94.19	93.62	**95.53**	94.92	95.06	94.36
	RotErr↓	313.4	285.8	261.5	300.0	300.3	311.0	**11.76**
	TransErr↓	0.076	0.087	0.103	0.046	0.058	0.028	**0.018**
	MatchPx↑	0.001	**0.004**	0.000	0.001	0.001	0.002	0.003
	TempFlick↑	0.949	**0.968**	0.928	0.964	0.965	0.962	0.966
	SubjCons↑	0.801	0.770	0.651	0.787	0.804	**0.839**	0.759
	BGCons↑	**0.916**	0.913	0.863	0.900	0.888	0.828	0.908
	ImagQual↑	0.502	0.462	0.276	0.467	0.471	**0.523**	0.498
	AesQual↑	0.456	0.228	0.111	**0.484**	0.300	0.469	0.343
	MotSmooth↑	0.532	0.854	**0.976**	0.752	0.797	0.663	0.784
360°	PSNR↑	4.71	4.31	3.39	4.62	5.25	5.40	**11.26**
	SSIM↑	0.228	0.208	0.239	0.182	0.332	0.278	**0.582**
	LPIPS↓	0.894	0.878	0.964	0.927	0.846	0.850	**0.421**
	CLIP-F↑	92.66	94.31	93.95	94.13	93.32	**94.92**	93.74
	RotErr↓	178.9	249.2	169.9	250.4	221.8	158.9	**133.31**
	TransErr↓	0.038	0.061	0.071	0.041	**0.030**	**0.030**	0.050
	MatchPx↑	0.005	0.007	0.001	0.005	0.005	**0.009**	0.007
	TempFlick↑	0.896	0.923	**0.956**	0.916	0.930	0.912	0.924
	SubjCons↑	0.801	0.763	0.724	0.762	0.768	**0.826**	0.748
	BGCons↑	0.894	0.913	0.885	0.881	0.877	**0.921**	0.895
	ImagQual↑	0.504	0.476	0.364	**0.654**	0.501	0.584	0.490
	AesQual↑	**0.999**	0.841	0.741	0.908	0.758	0.996	0.403
	MotSmooth↑	0.144	0.445	**0.641**	0.162	0.264	0.312	0.496
Omni Object3D	PSNR↑	5.70	4.75	4.49	5.43	5.90	7.07	**9.36**
	SSIM↑	0.511	0.412	0.440	0.475	0.511	**0.601**	0.482
	LPIPS↓	0.801	0.794	0.839	0.775	0.779	0.656	**0.537**
	CLIP-F↑	94.51	95.05	93.95	94.87	**95.29**	95.06	94.78
	RotErr↓	169.6	171.5	160.3	155.6	148.9	152.2	**141.08**
	TransErr↓	0.109	0.150	0.113	0.141	0.091	0.052	**0.013**
	MatchPx↑	0.002	0.004	0.000	0.002	0.002	**0.008**	0.004
	TempFlick↑	0.953	0.972	**0.981**	0.967	0.975	0.962	0.971
	SubjCons↑	0.810	0.768	0.741	0.707	0.830	**0.840**	0.793
	BGCons↑	0.908	0.908	0.882	0.877	0.908	**0.919**	0.909
	ImagQual↑	0.536	0.489	0.369	0.403	0.522	**0.624**	0.530
	AesQual↑	**0.932**	0.846	0.609	0.823	0.860	0.930	0.450
	MotSmooth↑	0.549	0.857	**0.896**	0.576	0.791	0.677	0.881
SynView-X	PSNR↑	11.19	12.50	**13.18**	12.61	11.93	10.00	12.14
	SSIM↑	0.470	0.417	0.537	0.555	0.538	0.428	**0.586**
	LPIPS↓	0.676	0.624	0.734	0.666	0.665	0.644	**0.456**
	CLIP-F↑	93.25	94.63	91.95	**95.85**	93.66	94.57	93.85
	RotErr↓	54.37	41.51	69.73	66.64	50.66	58.50	**29.63**
	TransErr↓	0.086	0.137	0.173	0.165	0.113	0.107	**0.017**
	MatchPx↑	0.003	0.007	0.000	0.002	0.003	**0.009**	0.008
	TempFlick↑	0.951	0.972	**0.979**	0.974	0.974	0.953	0.964
	SubjCons↑	0.773	0.797	0.648	0.736	0.788	**0.803**	0.781
	BGCons↑	0.901	**0.920**	0.875	0.901	0.884	0.915	0.899
	ImagQual↑	0.421	0.440	0.245	0.346	0.385	**0.547**	0.421
	AesQual↑	0.723	0.735	0.083	0.773	0.635	**0.883**	0.392
	MotSmooth↑	0.463	0.849	**0.883**	0.721	0.761	0.610	0.789

Table 7: **Single-view** VBench video-quality metrics averaged across five datasets. OrbErr is reported on SynView-X only, matching Table 1; WarpHammer-GEN3C has the lowest reported OrbErr.

		Consistency			Quality	Motion	Camera Pose
	Method	TempFlick↑	SubjCons↑	BGCons↑	ImagQual↑	MotSmooth↑	OrbErr↓
Warp-based	TrajCr.	0.936	0.798	0.905	0.506	0.414	0.3022
	GEN3C	0.957	0.770	**0.909**	0.477	0.739	0.3058
	EX-4D	0.955	0.756	0.889	0.490	0.566	0.3192
	CogNVS	0.961	0.799	0.886	0.489	0.647	0.3063
	WF	**0.965**	0.779	0.888	0.484	0.577	0.3067
Camera-cond.	ReCamM.	0.963	0.695	0.871	0.322	**0.862**	0.3460
	SEVA	0.949	**0.835**	0.903	0.568	0.584	0.2855
	WarpHammer-TC	0.931	0.786	0.891	**0.569**	0.464	0.2288
	WarpHammer-GEN3C	0.956	0.782	0.904	0.487	0.745	**0.2045**

Table 8: **Auxiliary-view** VBench video-quality metrics across 5 datasets (CO3D, Objectron, 360°, OmniObject3D, SynView-X). Gray methods use GT camera poses and in-scene second views and are shown for reference only.

		Consistency			Quality	Motion
	Method	TempFlick↑	SubjCons↑	BGCons↑	ImagQual↑	MotSmooth↑
w/ GT	GEN3C-MV	0.951	0.755	0.894	0.433	0.642
	SEVA-MV	0.954	0.812	0.904	0.576	0.501
w/o GT	GEN3C-MV	0.955	0.769	0.899	0.426	0.684
	SEVA-MV	0.938	0.769	0.886	**0.581**	0.442
	WarpHammer-GEN3C	**0.966**	**0.785**	**0.905**	0.471	**0.835**

G SINGLE-IMAGE COMPARISON WITH FREEORBIT4D

FreeOrbit4D (Cao et al., 2026) and WarpHammer both render completed geometry to guide video diffusion, but obtain that geometry differently. FreeOrbit4D generates multiview frames with SV4D from an input video and fuses them into a 4D proxy; WarpHammer uses a native 3D generative prior applied to a single image. We compare the methods on a static reference image to examine geometry completion without inter-frame parallax.

Protocol. We repeat the reference image into a 45-frame 854×480 clip and supply the same foreground mask used by WarpHammer, retaining FreeOrbit4D's released configuration, checkpoints and seed. Both trajectories use a $180°$ yaw.

Native-video reference. We also run the authors' bundled `car-turn` video with their released configuration (Figure 11). Its proxy remains dense, with projected coverage changing from 49.0% to 39.1% along the orbit. The generated sequence follows the proxy, including vegetation occlusions.

Single-image results. On the static-input example in Figure 12, FreeOrbit4D's proxy coverage falls from 96.6% to 32.6%. The far-side reconstruction becomes sparse, and the generated car develops a shorter wheelbase and an altered roofline. WarpHammer retains the car's silhouette and proportions across the orbit. This example illustrates the benefit of a native 3D object prior: it supplies unobserved surface geometry without relying on parallax between input frames.

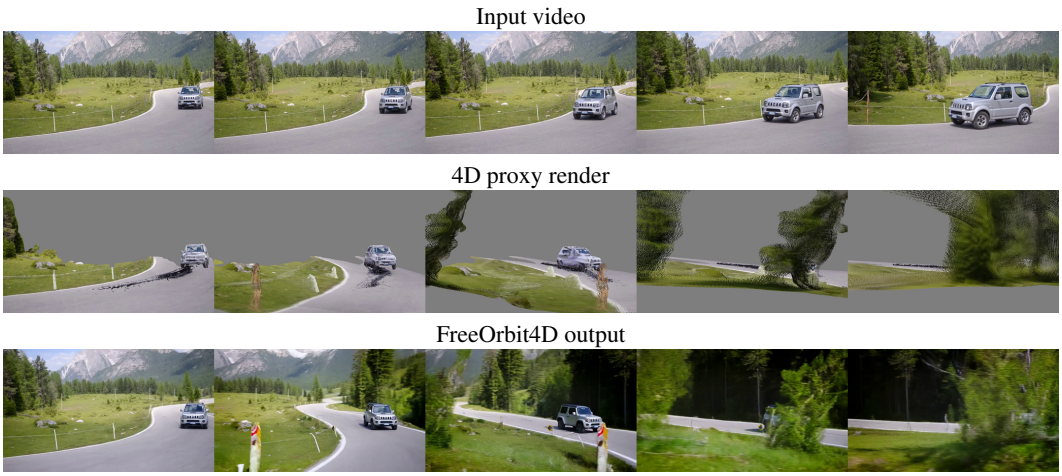

Figure 11: FreeOrbit4D on the authors' bundled `car-turn` video using the released configuration. Rows: input video, 4D proxy render and generated output. Motion-rich input supports a dense proxy; foreground vegetation occludes the vehicle in some views.

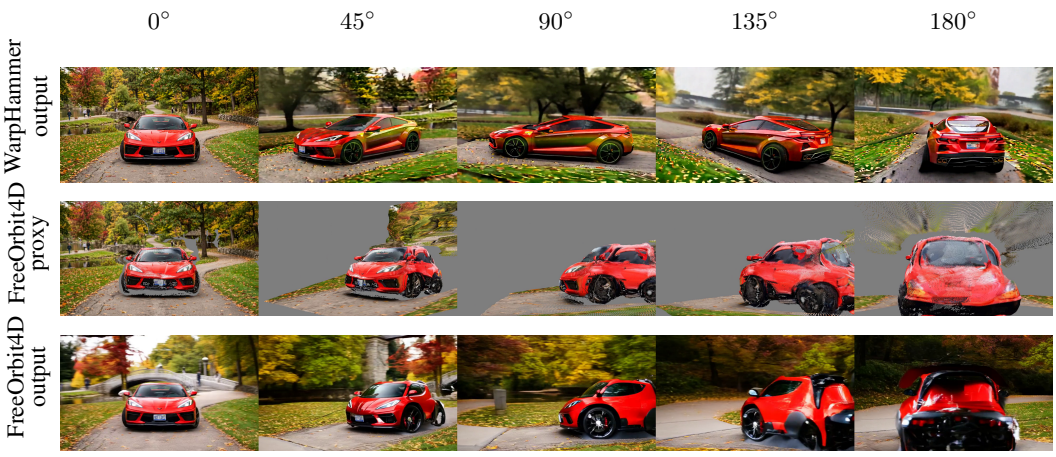

Figure 12: Single-image comparison at corresponding azimuths. Rows: WarpHammer output, FreeOrbit4D proxy and FreeOrbit4D output. FreeOrbit4D receives a repeated reference image; WarpHammer uses the image directly. In this example, object-prior completion preserves the car's proportions as the opposite side comes into view.

H Additional Qualitative Results

For additional visual examples, including cars, furniture, auxiliary-view fusion and multi-object scenes, please see the accompanying supplementary HTML gallery.

We present additional qualitative comparisons that complement the main paper. Figure 13 shows novel view synthesis results with auxiliary view, where applicable, from the back of the car showing a full orbital trajectory around a car and comparing all baselines side-by-side with WarpHammer. The example is selected to span a wide range of azimuthal viewpoints, including angles far from the input view, where the differences between methods become most apparent.

Warp-based baselines (TrajectoryCrafter, GEN3C, EX-4D, CogNVS) reproduce the input view faithfully at small offsets but progressively degrade as the orbital angle grows: occluded regions of the car are filled with warped ghost geometry, and large-angle frames frequently exhibit mirror-like duplications of the visible side rather than the genuine far side of the object. Camera-conditioned baselines (ReCamMaster, SEVA) avoid such warping artifacts but instead produce near-static syntheses that fail to follow the prescribed orbital trajectory, so the object viewpoint barely changes across frames. In contrast, WarpHammer produces a coherent orbital sequence in which the car is rendered from the correct viewpoint at each frame, with consistent geometry, plausible far-side appearance, and stable background, demonstrating the benefit of object-centric 3D densification under large viewpoint changes.



Figure 13: Novel view synthesis comparison along a full orbital trajectory around a car. Warp-based baselines exhibit ghosting and mirror-like duplication at large angles, camera-conditioned baselines fail to follow the orbital motion, while WarpHammer produces a coherent, geometry-faithful orbital sequence.

Further auxiliary-view examples. Figure 14 adds sofa and office-chair sequences from the project-page gallery, showing the reference, auxiliary and five uniformly spaced output frames. Figure 15 shows two cross-instance car-fusion examples using the same auxiliary rear view with different reference scenes. The generated cars retain the orange reference appearance despite the pink auxiliary. Full videos accompany these qualitative examples.

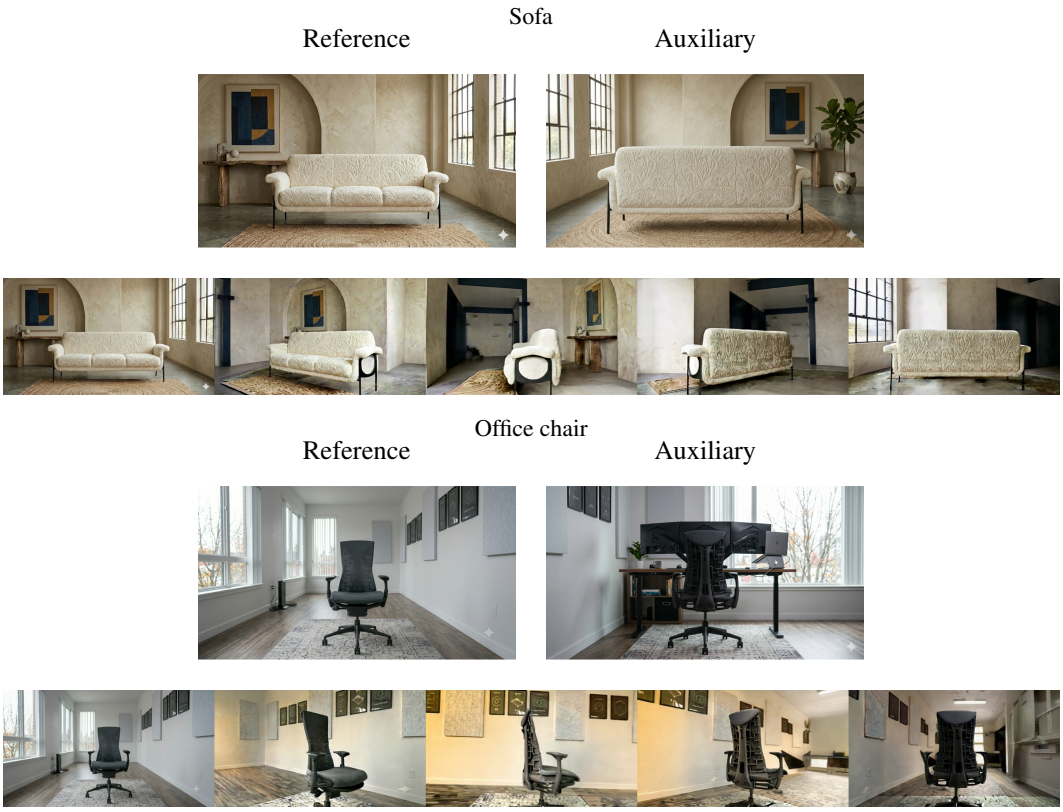

Figure 14: Additional auxiliary-view synthesis examples. Each output row samples the corresponding project-page video at five uniformly spaced times, from the initial view to the final view. Inputs and output frames are shown without cropping.

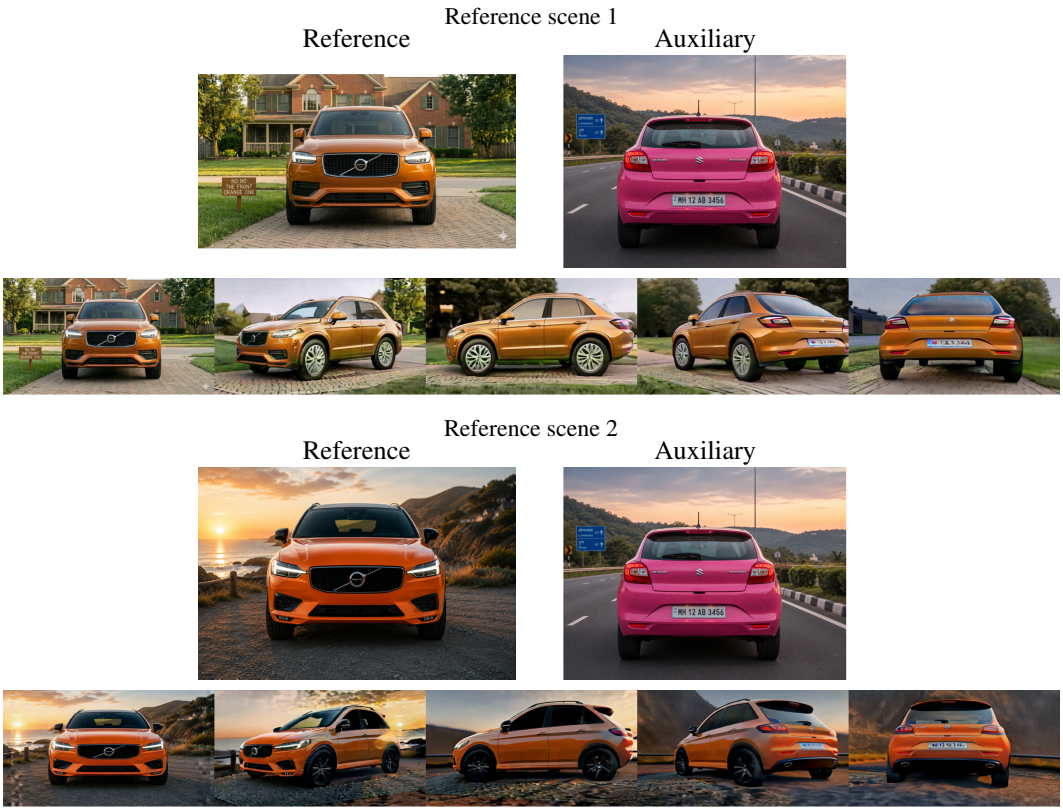

Figure 15: Cross-instance auxiliary fusion in two reference scenes. An orange reference car is paired with the rear view of a different, pink car. The output rows show five uniformly spaced frames from the project-page fusion videos, illustrating reference appearance retained during the synthesized orbit.