ESC: ESCAPING 2D DIFFUSION AMBIGUITY WITH GEOMETRIC CONSTRAINTS FOR CONSISTENT NOVEL VIEW SYNTHESIS

Anonymous authors
Paper under double-blind review

ABSTRACT

Generating 3D-consistent novel views from 2D diffusion models remains a fundamental challenge in computer vision. In this work, we introduce a framework that enhances geometric alignment for monocular view synthesis under drastic camera changes by jointly generating a target view and its corresponding depth map. By conditioning the diffusion pipeline on the single source image, source depth, and target camera rays, our approach forces the model to inherently reason in 3D space even under extreme viewpoint shifts. To guarantee spatial fidelity, we propose a novel per-pixel depth consistency constraint, which verifies and enforces 3D consistency across generated views. Our framework establishes a robust pipeline for geometrically consistent view synthesis, yielding superior visual accuracy and spatial alignment compared to current state-of-the-art methods. With this work, we pave the road towards robust 3D world models capable of understanding complex environments from a single view image.

1 INTRODUCTION

Monocular novel view synthesis (NVS) is a fundamental yet highly challenging problem in computer vision, requiring the generation of unseen views of a 3D scene from only a single 2D image. This setting is inherently ambiguous: a single observation provides limited evidence about scene depth, object geometry, and occluded regions, making accurate 3D reasoning difficult without additional geometric constraints. This difficulty increases with extreme viewpoint changes, where local pixel interpolation is insufficient. Because large portions of the target view may be completely unobserved in the source image, prediction becomes an ill-posed extrapolation problem that requires the model to synthesize both scene structure and appearance. To bridge this gap, recent approaches have increasingly leveraged diffusion models (Ho et al., 2020; Rombach et al., 2021; Lipman et al., 2023), whose strong generative priors provide plausible visual completions when direct image evidence is missing.

However, relying primarily on generative priors introduces a fundamental tension between photorealistic synthesis and geometric validity. A prevalent strategy for addressing this tension follows a "geometry-warping and 2D-completion" pipeline: pixels or features from the source view are first projected into the target view using estimated scene geometry, and a 2D diffusion model is then employed to complete the newly exposed regions (Ren et al., 2025; Lin et al., 2026; Huang et al., 2025; Chen et al., 2026; Rahary et al., 2026; Cao et al., 2024; Yu et al., 2024). Although depth-aware warping provides a geometrically grounded initialization, the central challenge of NVS remains in the regions it cannot resolve: projection transfers only visible surfaces and cannot recover content occluded in the source view. A 2D completion model must therefore infer target-view visibility, depth discontinuities, and severe distortions under large camera shifts without explicit 3D awareness. Treating these ambiguous regions as ordinary missing pixels provides no spatial constraints, often yielding locally plausible but geometrically invalid structures with broken occlusion boundaries, inconsistent depth, and severe floating artifacts.

Escaping these geometric ambiguities requires reasoning beyond the 2D canvas. Rather than relying on purely image-plane priors, we propose to ground synthesis in underlying scene geometry. We

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

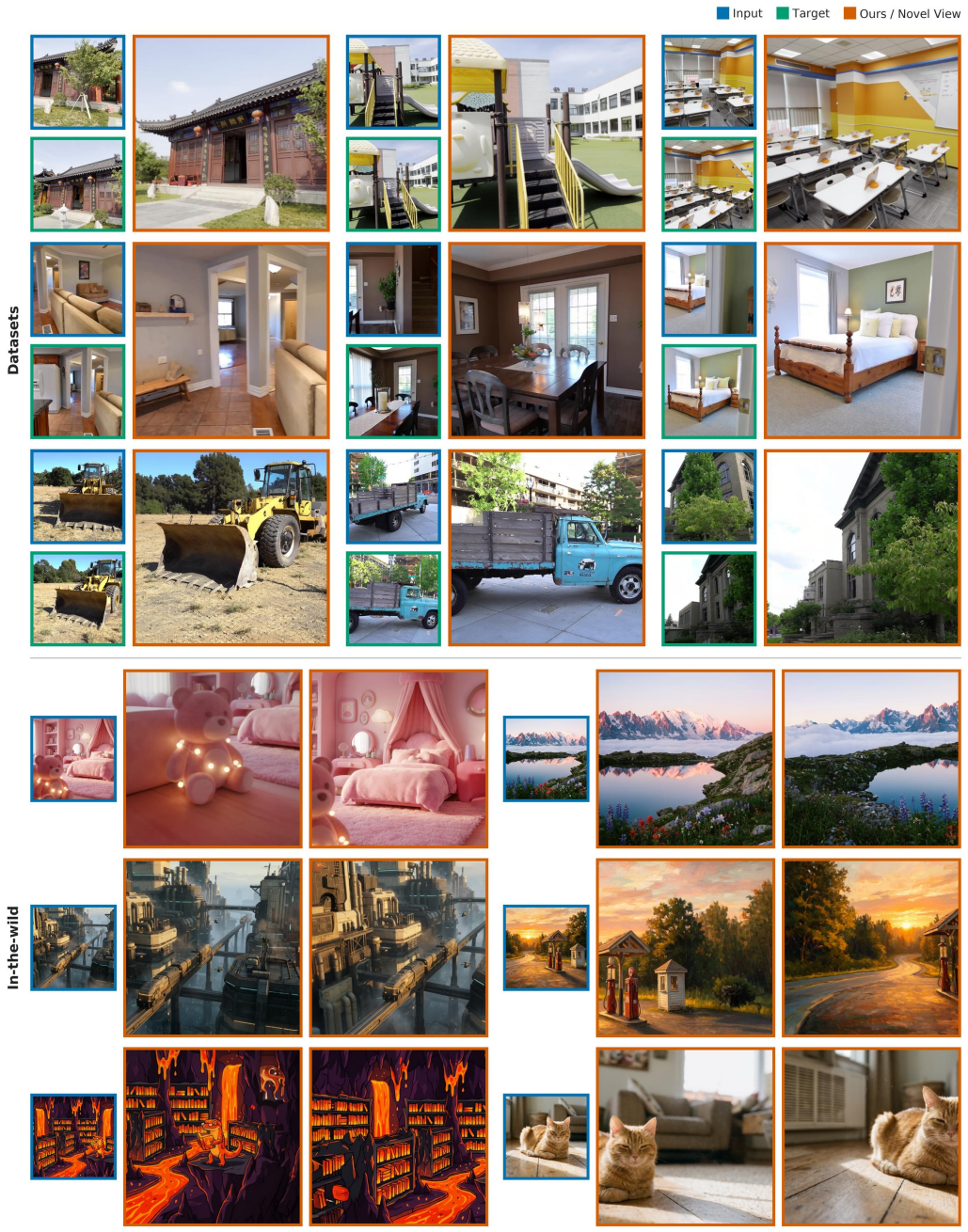

Figure 1: **Novel view synthesis highlights.** Given a single input view, its corresponding depth map, and Plücker ray conditioning, our method generates novel views that remain 3D-consistent with the target across challenging datasets (DL3DV, RealEstate10K, T&T) and selective in-the-wild scenes. For the datasets, we illustrate a single novel view generation alongside the ground-truth target, whereas for the in-the-wild setting, we demonstrate two distinct novel view generations.

introduce **ESC**[1], a geometry-aware diffusion framework that jointly predicts target-view depth and appearance conditioned on source view information and Plücker rays. However, simply exposing

[1]Our framework name, **ESC**ape the Allegory of the Cave (ESC), draws an analogy to Plato's allegory: standard 2D models are trapped observing only 2D "shadows" on the image plane without understanding the 3D shapes casting them. ESC guides the model outside the cave by conditioning directly on 3D geometry.

the diffusion model to depth data during training does not guarantee 3D consistency. Due to the probabilistic nature of diffusion, the model can still introduce global geometric violations. To rigorously bind the generative process to the underlying scene geometry, we introduce **P**rojection **C**onsistency **C**onstraint **(PCC)**, an explicit geometric constraint that enforces 3D consistency across synthesized views by penalizing geometric contradictions when reprojecting synthesized pixels into 3D space. By applying PCC during inference, our proposed framework ensures completed regions remain spatially consistent, enabling photorealistic novel-view synthesis that preserves coherent visibility and 3D structure.

We comprehensively evaluate our framework across challenging large-scale datasets, including DL3DV (Ling et al., 2023), RealEstate10K (Zhou et al., 2018), and Tanks & Temples (Knapitsch et al., 2017), as well as on diverse in-the-wild captures. As shown in Figure 1, our framework mitigates geometric ambiguity and produces reliable novel views even from out-of-distribution images sampled from a state-of-the-art text-to-image model (Google DeepMind, 2026).

In summary, our primary contributions are as follows:

- **Geometry-Grounded Generation Framework:** We introduce ESC, a multi-modal diffusion pipeline that conditions generation on Plücker rays via a Ray ControlNet and jointly predicts RGB and depth, grounding novel view synthesis directly in explicit 3D scene geometry.
- **Explicit Geometric Consistency Constraint:** We propose the PCC, a novel depth-reprojection constraint that penalizes 3D contradictions to enforce spatial consistency across views. Experiments show that PCC captures 3D spatial consistency where 2D and pose-based metrics fail.
- **State-of-the-Art Performance:** Our full framework surpasses existing baselines, delivering superior visual quality and robust geometric consistency under extreme viewpoint changes.

2 Related Work

Approaches to Novel View Synthesis. Recent NVS approaches broadly fall into feed-forward regressors and diffusion-based generative architectures. Feed-forward models, including OVIE (Rahary et al., 2026) and LagerNVS (Szymanowicz et al., 2026), predict a target view in a single pass, prioritizing inference speed at the cost of photorealism and fine-grained texture. In contrast, diffusion methods achieve superior visual fidelity through iterative denoising. PhotoNVS (Yu et al., 2023) is a pixel-space DDPM requiring extensive denoising, whereas VIVID (Elata et al., 2024) uses latent diffusion with classifier-free guidance. GLD (Jang et al., 2026) extends this paradigm to higher resolutions and multi-view consistency through a multi-stage cascade diffusion design. Our work builds on this generative paradigm.

Depth-Aware and Ray-Conditioned Synthesis with Geometric Foundation Models. Recent NVS methods, including Sharp (Mescheder et al., 2026), Flash3D (Szymanowicz et al., 2025), and Voyager (Huang et al., 2025), highlight the importance of explicit depth priors and precise camera conditioning for geometric fidelity. In 3D Gaussian Splatting frameworks (Kerbl et al., 2023; Mescheder et al., 2026; Szymanowicz et al., 2025), depth estimates are used to place or initialize Gaussian primitives along target camera rays, thereby anchoring the rendering process in physical space. We adopt this principle within a diffusion backbone, treating NVS as a ray-conditioned generative process anchored at explicit depths. This formulation allows constraints such as PCC to enforce consistency between synthesized textures and the underlying 3D structure. Building on Qwen-Edit (Wu et al., 2025), we replace the ambiguous natural-language pose prompts used by recent LoRA-based multi-view adaptations (fal.ai, 2025; dx8152, 2025; Hu et al., 2021) with direct Plücker-ray conditioning (Hartley and Zisserman, 2004) and explicit depth priors, providing precise, pixel-aligned viewpoint control.

In parallel, we leverage 3D reconstruction models to recover 3D information from one or more images of a scene. This area has shifted from specialized, task-specific pipelines (Schonberger and Frahm, 2016a; Bhat et al., 2023; Bochkovskii et al., 2025) toward unified, feed-forward geometric foundation models. DUSt3R (Wang et al., 2024b) and MASt3R (Leroy et al., 2024) infer 3D scene properties from arbitrary image collections without prior camera calibration, while VGGT (Wang et al., 2025) and DA3 (Lin et al., 2025) enable single-pass joint inference of dense depth and camera parameters from uncalibrated views. Rather than relying on classical multi-view stereo, we use DA3

to extract dense depth maps and target camera poses, which supply the geometric priors needed to ground our generative NVS process.

Novel View Synthesis Evaluation. Standard novel-view synthesis (NVS) is evaluated using image-quality metrics (Rahary et al., 2026; Jang et al., 2026; Szymanowicz et al., 2026; Ren et al., 2025; Elata et al., 2024; Szymanowicz et al., 2025; Yu et al., 2023) such as PSNR, SSIM (Wang et al., 2004), LPIPS (Zhang et al., 2018) and FID (Heusel et al., 2017), together with SfM (Schonberger and Frahm, 2016b)-derived pose accuracy (He et al., 2024; Song et al., 2026; Yu et al., 2024). Image metrics require reference views and do not measure cross-view geometry, while pose metrics assess camera alignment rather than scene structure. Consequently, photorealistic but geometrically inconsistent views may score well, whereas plausible geometry in occluded regions may be penalized.

3 METHOD

We argue that both generating novel views with 3D alignment to the input and evaluating that alignment require explicit reasoning in 3D space, beyond 2D image-quality and camera-pose metrics. We address this through two complementary components. In Section 3.1, we introduce the *Projection Consistency Constraint (PCC)*, a 3D consistency objective that measures geometric integrity across synthesized views. In Section 3.2, we present ESC, a geometry-aware diffusion framework that jointly synthesizes RGB and depth conditioned on Plücker rays and uses PCC for inference-time Best-of-N selection to enforce 3D coherence.

3.1 PROJECTION CONSISTENCY CONSTRAINT: A 3D CONSISTENCY MEASURE FOR NOVEL-VIEW SYNTHESIS

We propose *Projection Consistency Constraint* (PCC), a bidirectional depth-reprojection metric that measures 3D geometric agreement between views (Figure 2). Given a source view with depth map $(\mathbf{I}_s, \mathbf{D}_s)$ and a target view with depth map $(\mathbf{I}_t, \mathbf{D}_t)$, with known camera intrinsics $\mathbf{K}$ and relative pose $[\mathbf{R} \mid \mathbf{t}]$—where $\mathbf{R}$ is the rotation matrix and $\mathbf{t}$ is the translation vector mapping the source camera coordinate frame to the target camera coordinate frame—PCC proceeds in three steps.

Step 1: Unproject. Each pixel (u, v) in the source view is lifted to a 3D point in the source camera frame using the depth value and the pinhole model:

$$\mathbf{p}_{u,v}^{(s)} = \mathbf{D}_s(u, v) \cdot \mathbf{K}^{-1} \begin{pmatrix} u \\ v \\ 1 \end{pmatrix}. \tag{1}$$

where $\mathbf{D}_s(u, v)$ represents the depth at pixel (u, v). An identical unprojection procedure is applied to the target view using $\mathbf{D}_t$ to obtain the target 3D points, $\mathbf{p}_{u,v}^{(t)}$. When evaluating synthesized views where ground-truth geometry is unavailable, we estimate the depth and camera intrinsics using a pre-trained monocular depth predictor.

Step 2: Reproject. The source 3D points $\mathbf{p}^{(s)}$ are transformed into the target camera frame via the relative pose and projected onto the target image plane. Z-buffer splatting (Zwicker et al., 2001) is used to handle occlusion: points are sorted by depth (farthest first) so that closer points overwrite farther ones, producing a reprojected depth map $\hat{\mathbf{D}}_{s\to t}$ (Figure 2(c)). The same procedure is applied symmetrically to the target 3D points $\mathbf{p}^{(t)}$, which are reprojected into the source camera frame to produce $\hat{\mathbf{D}}_{t\to s}$ (Figure 2(h)).

Step 3: Compare. For each pixel covered by the reprojection, we compute the depth difference. We first define a per-pixel validity mask $\mathcal{V}$ that discards three classes of unreliable correspondences: (i) points whose reprojected depth $\hat{z}$ is non-positive, meaning they fall *behind* the target camera; (ii) points that project outside the target image domain $\Omega_t = \{(u, v) : 0 \le u < W,\ 0 \le v < H\}$; and (iii) points originating from invalid source depths (below a small threshold $\epsilon = 0.01\,\text{m}$). The masked depth difference is:

$$\mathcal{V}_{s\to t}(u, v) = \mathbf{1}[\hat{z}_{s\to t}(u, v) > 0] \cdot \mathbf{1}[(u, v) \in \Omega_t] \cdot \mathbf{1}[\mathbf{D}_s > \epsilon], \tag{2}$$

$$\Delta_t(u, v) = \mathcal{V}_{s\to t}(u, v) \cdot \Big(\mathbf{D}_t(u, v) - \hat{\mathbf{D}}_{s\to t}(u, v)\Big). \tag{3}$$

The backward difference Δ_s is defined symmetrically by swapping the roles of source and target. Together, these two terms capture complementary types of geometric inconsistency.

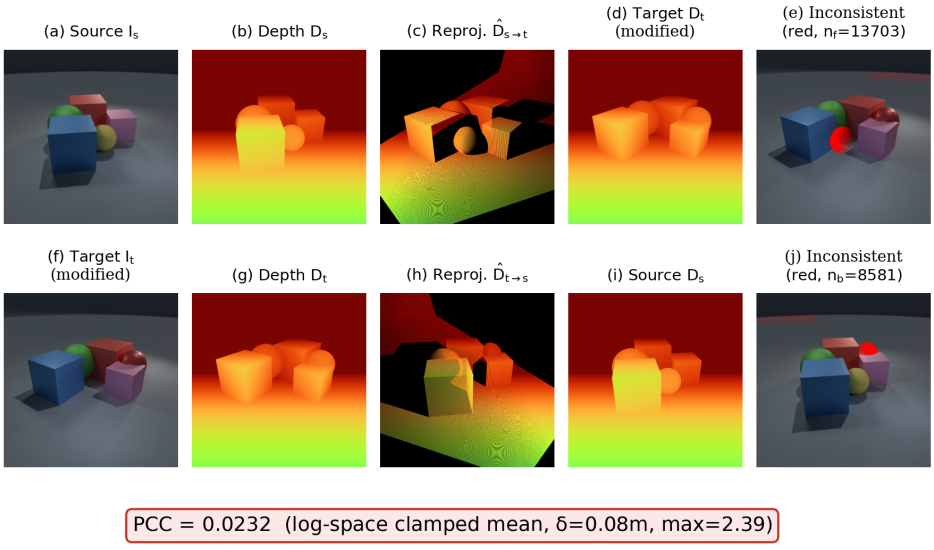

Figure 2: **PCC computation pipeline.** The source view (a) is from the original scene, while the target view (f) has been modified: a red sphere is added and a yellow sphere is removed. **Top row (Δ_t, detecting missing geometry):** the source depth (b) is unprojected to 3D and reprojected into the target camera frame (c); black regions indicate pixels outside the reprojection coverage. Where the yellow sphere was removed, target camera rays pass through and hit the cubes in the back instead of striking the sphere surface, producing large positive Δ_t values. These inconsistent pixels are highlighted in red (e), with $n_f = 13703$ pixels exceeding the error threshold δ. **Bottom row (Δ_s, detecting hallucinated geometry):** The modified target depth is reprojected back into the source frame (h). The added red sphere reprojects into a region with no corresponding geometry in the source perspective, yielding large positive Δ_s values and $n_b = 8581$ inconsistent pixels (j). The final PCC score of 0.0232 quantifies the overall geometric disagreement between the two views.

A positive Δ_t at pixel (u, v) indicates that the reprojected source geometry lies *in front of* the target surface: a source-view object occludes a point that the target view claims to see. This detects *missing geometry*—content expected in the target view but absent. In Figure 2, removing the yellow sphere causes rays to reach the background instead, producing a large positive Δ_t (panel (e), red pixels). Conversely, a positive Δ_s indicates that the reprojected target geometry lies *in front of* the source surface, so a target-view object appears where the source has no corresponding geometry. This detects *hallucinated geometry*—target-view content without a source counterpart. The added red sphere illustrates this case: its reprojection intrudes in front of the source background, producing a large positive Δ_s (panel (j), red pixels). Together, Δ_t and Δ_s detect geometry that disappears or moves behind surfaces from the target view, and appears or moves in front from the source view, respectively.

Rather than count inconsistent pixels, we use a *distance-based* score that captures inconsistency magnitude. Differences below the noise threshold δ are suppressed, while a clamped log transform compresses large outliers and preserves the ordering of moderate errors. Specifically, the forward contribution is:

$$L_f = \sum_{(u,v)\in\mathcal{M}_{s\to t}} \min\Big(\log\big(1+\max(\Delta(u,v)-\delta,\ 0)\big),\ \tau\Big), \tag{4}$$

where $\mathcal{M}_{s\to t}$ denotes valid reprojection pixels and τ is the per-pixel clamping threshold. The process is applied symmetrically in the backward direction ($t \to s$) to obtain L_b. The final PCC loss is the total log-space depth error normalised by the number of valid coverage pixels across both directions:

$$\text{PCC} = \frac{L_f + L_b}{|\mathcal{M}_{s\to t}| + |\mathcal{M}_{t\to s}|}. \tag{5}$$

Lower PCC values indicate better multi-view geometric consistency.

3.2 ESC: Escape the Allegory of the Cave

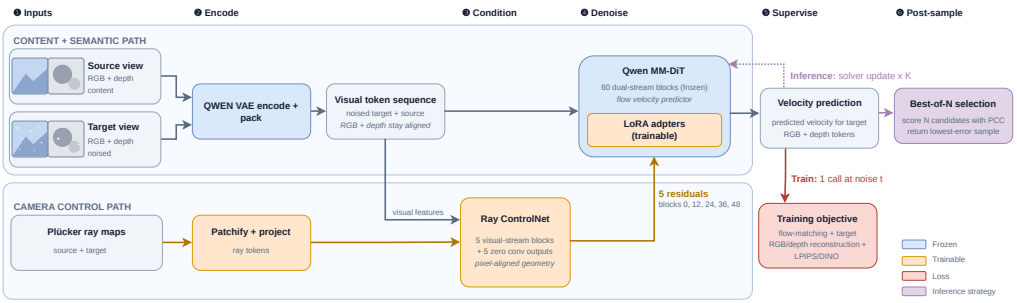

Figure 3: **ESC Framework.** The framework extends Qwen MM-DiT with depth information and Plücker rays conditioning by leveraging LoRA adapters and a Ray ControlNet. At inference time, we leverage PCC-based sampling to enforce improved generated scene geometry.

We present ESC, a geometry-aware diffusion framework for novel-view synthesis that simultaneously predicts target RGB images and depth maps (Figure 3). We first describe our input representation, followed by our parameter-efficient adaptation strategy and the overall training and inference pipeline.

Input Representation and Denoising Pipeline. To perform geometry-aware view synthesis, our framework ingests three primary conditioning inputs: a reference source image, its corresponding rendered depth map, and the geometry of the source and target camera parameterized by Plücker ray maps. The framework also ingests a noise sample representing the target image and depth map. Note that at training time, the noise sample corresponds to a noisy version of the target groundtruth and depth. The source image and the rendered depth map are processed and tokenized using Qwen VAE tokenizer (Wu et al., 2025; Kingma and Welling, 2022). Similarly, the noise sample corresponding to the target image and depth is fed to Qwen VAE. The resulting (source or target) depth tokens are concatenated directly to the end of the (source or target) image tokens, creating a visual token sequence. The Plücker rays (Hartley and Zisserman, 2004), which represent the camera poses of both source and target views, are tokenized and fed to a Ray ControlNet module (Zhang et al., 2023) along with the visual token sequence to guide the subsequent synthesis process with explicit spatial awareness. The synthesis process is performed through the Qwen Multimodal-Diffusion Transformer (Qwen MM-DiT) backbone (Peebles and Xie, 2023; Wu et al., 2025), which learns to perform iterative token denoising under the combined conditioning of the visual token sequence and Ray ControlNet output.

Parameter-Efficient Adaptation for View Synthesis. Optimization of the framework is performed through a Low-Rank Adaptation (LoRA) (Hu et al., 2021) of the Qwen MM-DiT network and the Ray ControlNet network. Specifically, parameter-efficient LoRA adapters are injected into the dual-stream image branch to align the newly introduced depth tokens with native visual features, while the Ray ControlNet is utilized for precise Plücker ray conditioning. This targeted configuration provides the necessary expressive capacity to incorporate geometric and spatial controls without destabilizing the foundation model.

The training loss $\mathcal{L}_{\text{base}}$ is formulated as a weighted sum of multi-modal objectives, where pixel-space losses are gated by the flow-matching timestep t:

$$\mathcal{L}_{\text{base}} = w_{\text{fm,rgb}}\mathcal{L}_{\text{fm,rgb}} + w_{\text{fm},d}\mathcal{L}_{\text{fm},d} + \mathbf{1}_{\{t \le t_{\text{trunc}}\}}\big(w_{\text{recon}}\mathcal{L}_{\text{recon}} + w_{\text{lpips}}\mathcal{L}_{\text{lpips}} + w_{\text{dino}}\mathcal{L}_{\text{dino}}\big), \quad (6)$$

where the $w_{(\cdot)}$ coefficients denote scalar weights controlling the relative contribution of each respective term. $\mathcal{L}_{\text{fm,rgb}}$ and $\mathcal{L}_{\text{fm},d}$ represent the latent flow-matching objectives for the generated target RGB and target inverse depth fields, respectively, with inverse depth targets derived by inverting the pseudo-ground-truth depth estimates provided by DA3[2]. To ensure high visual fidelity, the remaining terms are computed in pixel space on the decoded target views: $\mathcal{L}_{\text{recon}}$ denotes a pixel reconstruction loss, while $\mathcal{L}_{\text{lpips}}$ (Zhang et al., 2018) and $\mathcal{L}_{\text{dino}}$ (Oquab et al., 2024) serve as perceptual

[2]Inverse depth d is computed from the depth z as $d = \frac{1}{z}$.



Figure 4: **Qualitative novel-view synthesis.** Source views (blue), baselines (gray), ESC variants (orange), and ground truth (green) on DL3DV (top), RE10K (middle), and Tanks & Temples/Barn (bottom). ESC-PCC-BoN best preserves geometry and detail across datasets.



Figure 5: **Best-of-N synthesis.** Source (blue), ground truth (green), the PCC-selected best sample, and five alternatives from the same candidate pool (orange).

loss terms. The indicator function $\mathbf{1}_{\{t \le t_{\text{trunc}}\}}$ ensures these three pixel-space objectives are evaluated exclusively within a truncated timestep interval, where t_{trunc} is the chosen timestep threshold.

Inference-time Best-of-N (ESC-PCC-BoN). We leverage the stochastic nature of the diffusion sampling process to optimize geometric fidelity. For each source-to-target view synthesis request, we generate N independent samples $\{\hat{y}_i\}_{i=1}^{N}$, by varying the initial random noise seeds. The predicted target inverse depth of each candidate sample is evaluated via the PCC metric established in Section 3.1, based on the depth estimates generated by the ESC. Notably, we disable scale alignment (mentioned in Appendix C.3) for PCC during Best-of-N selection. The pipeline then selects and returns the optimal sample $\hat{y}^\star$ that minimizes the geometric alignment error relative to the source reference view x_{src}:

$$\hat{y}^\star = \arg \min_{i \in \{1,\dots,N\}} \text{PCC}\left(\hat{y}_i,\, x_{\text{src}}\right). \tag{7}$$

4 Experiments

This section evaluates ESC on selected datasets, demonstrating consistent improvements over established baselines in perceptual realism and 3D consistency (Section 4.1). We defer the systematic analysis of PCC to Appendix C.12, where controlled geometric perturbation and occlusion experiments show that PCC captures multi-view geometric integrity overlooked by standard 2D perceptual and camera-pose metrics, which often fail to penalize structural hallucinations.

4.1 MAIN RESULTS

Datasets and Preprocessing. We evaluate our method on three datasets: DL3DV (Ling et al., 2023), RealEstate10K (Re10K)(Zhou et al., 2018), and Tanks & Temples (T&T) (Knapitsch et al., 2017). We leverage Depth-Anything 3 (DA3) (Lin et al., 2025) as a preprocessing step to extract dense depth maps. A comprehensive discussion of the datasets and their specific preprocessing pipelines may be found in Appendix A.

Training details. We optimize our model parameters end-to-end using the AdamW optimizer (Loshchilov and Hutter, 2017) with weight decay, coupled with a cosine learning-rate schedule and an initial warm-up phase. To stabilize training and enhance generation quality, we maintain an exponential moving average (EMA) (Ho et al., 2020; Polyak and Juditsky, 1992) of the trainable parameters, which is subsequently used for all evaluations. Throughout both stages, diffusion timesteps are sampled from a logit-normal noise distribution. For real image datasets, our model is trained on a joint mixture of the DL3DV and RE10k training sets and evaluated on DL3DV, Re10K and T&T. All the training configurations are presented in Table 4.

Evaluation. We assess four metric categories: (i) *Synthesis Quality* using PSNR and SSIM (Wang et al., 2004); (ii) *Perceptual Quality* using LPIPS (Zhang et al., 2018), DINO-Sim (Oquab et al., 2024), and DreamSim (Fu et al., 2023); (iii) *Pose/View Prompt Consistency* using pose-AUC, RPE-R, and RPE-T (He et al., 2024; Song et al., 2026); and (iv) *Geometric Consistency* using our proposed PCC. We benchmark against five baselines: PhotoNVS (Yu et al., 2023), VIVID (Elata et al., 2024), OVIE (Rahary et al., 2026), GLD (Jang et al., 2026), and LagerNVS (Szymanowicz et al., 2026).

Models. To systematically evaluate the performance of our architecture and its ability to scale to complex, unconstrained real-life scenes, we train and evaluate two core model configurations. Specifically, *ESC-Base* represents our framework, utilizing depth conditioning to generate a *single* sample per target view. Finally, *ESC-PCC-BoN* employs our complete framework with inference-time Best-of-N strategy, which generates *multiple* candidate samples and selects the optimal output according to our PCC metric.

Discussions. Quantitative results are compiled in Table 1 and Table 9. ESC-Base achieves the strongest LPIPS, DINO-Sim, and DreamSim results across the three evaluated datasets. PCC-based Best-of-N selection provides additional improvements in these perceptual metrics and lowers PCC.

While GLD and LagerNVS score higher on conventional pixel and pose metrics in Table 9, these criteria reflect different priorities rather than true geometric fidelity. Metrics like PSNR and SSIM strictly measure pixel alignment, which inherently favors blurry, averaged imagery and is highly sensitive to minor pose estimation errors. Because neither these nor pose metrics capture bidirectional depth-reprojection consistency, we highlight this disconnect between conventional scores and actual visual fidelity in our qualitative comparisons (Figure 4, Figure 20, and Figure 21).

We attribute ESC's performance to two complementary components: *geometry-aware RGB–depth generation* and *PCC-guided inference-time selection*. First, unlike methods that predict only RGB, ESC jointly predicts RGB and depth while conditioning on Plücker rays through the Ray ControlNet. This design provides explicit geometric signals during generation and encourages synthesized appearance to remain coupled to the predicted scene structure. Second, PCC-based Best-of-N selection provides an inference-time mechanism for converting additional sampling compute into improved depth-reprojection consistency. ESC-PCC-BoN generates N independent diffusion samples and selects the candidate with the lowest PCC. Relative to the single-sample ESC-Base setting, this requires approximately N times the denoising compute, plus the comparatively small cost of PCC evaluation. ESC-PCC-BoN should therefore be interpreted as a test-time scaling result rather than a compute-matched comparison with single-sample baselines; ESC-Base provides the corresponding one-sample comparison. As shown in Table 1, this selection improves both perceptual and geometric metrics. On DL3DV, moving from ESC-Base to ESC-PCC-BoN reduces LPIPS from 0.4448 to 0.4340, DreamSim from 0.1420 to 0.1402, and PCC from 0.2448 to 0.2257, while increasing DINO-Sim from 0.8366 to 0.8386. Thus, PCC selection provides a measurable quality–compute trade-off, although total sampling compute grows approximately linearly with N.

Our model generates diverse, highly plausible completions for occluded scene regions. Rather than producing a deterministic or blurred average, it generates multiple distinct, high-quality variations

Model	Perceptual Quality			Geometry
	LPIPS ↓	DINO-Sim ↑	DreamSim ↓	PCC ↓
DL3DV Dataset				
PhotoNVS	0.574 ± 0.123	0.565 ± 0.090	0.388 ± 0.129	0.263 ± 0.168
VIVID	0.641 ± 0.105	0.691 ± 0.208	0.295 ± 0.129	0.247 ± 0.162
OVIE	0.710 ± 0.149	0.276 ± 0.351	0.648 ± 0.273	0.312 ± 0.160
GLD†	0.567 ± 0.101	0.702 ± 0.150	0.255 ± 0.117	0.249 ± 0.164
LagerNVS†	0.471 ± 0.176	0.673 ± 0.233	0.323 ± 0.203	0.249 ± 0.175
ESC-Base† (*Ours*)	0.445 ± 0.157	0.837 ± 0.139	0.142 ± 0.125	0.245 ± 0.165
ESC-PCC-BoN† (*Ours*)	$\mathbf{0.434 \pm 0.156}$	$\mathbf{0.839 \pm 0.139}$	$\mathbf{0.140 \pm 0.123}$	$\mathbf{0.226 \pm 0.156}$
RE10K Dataset				
PhotoNVS†	0.366 ± 0.123	0.767 ± 0.103	0.218 ± 0.067	0.090 ± 0.163
VIVID†	0.374 ± 0.141	0.884 ± 0.094	0.109 ± 0.066	0.088 ± 0.148
OVIE	0.501 ± 0.181	0.717 ± 0.325	0.788 ± 0.138	0.134 ± 0.152
GLD†	0.441 ± 0.105	0.813 ± 0.141	0.139 ± 0.105	0.090 ± 0.143
LagerNVS†	0.324 ± 0.125	0.884 ± 0.120	0.126 ± 0.103	0.078 ± 0.149
ESC-Base‡(*Ours*)	0.261 ± 0.125	0.916 ± 0.065	0.067 ± 0.050	0.077 ± 0.143
ESC-PCC-BoN‡(*Ours*)	$\mathbf{0.215 \pm 0.122}$	$\mathbf{0.922 \pm 0.067}$	$\mathbf{0.066 \pm 0.051}$	$\mathbf{0.066 \pm 0.136}$
Tanks & Temples Dataset				
PhotoNVS	0.769 ± 0.078	0.234 ± 0.167	0.590 ± 0.129	0.447 ± 0.411
VIVID	0.753 ± 0.082	0.379 ± 0.235	0.504 ± 0.170	0.442 ± 0.423
OVIE	0.822 ± 0.098	0.064 ± 0.151	0.788 ± 0.138	0.510 ± 0.479
GLD	0.636 ± 0.073	0.510 ± 0.207	0.403 ± 0.145	0.418 ± 0.409
LagerNVS	0.642 ± 0.099	0.274 ± 0.267	0.663 ± 0.196	0.490 ± 0.486
ESC-Base (*Ours*)	0.584 ± 0.102	0.724 ± 0.187	0.236 ± 0.137	0.383 ± 0.388
ESC-PCC-BoN (*Ours*)	$\mathbf{0.560 \pm 0.116}$	$\mathbf{0.734 \pm 0.185}$	$\mathbf{0.223 \pm 0.134}$	$\mathbf{0.285 \pm 0.314}$

Table 1: **Quantitative comparison on novel view synthesis across three datasets.** We evaluate perceptual quality using LPIPS, DINO similarity, and DreamSim, and geometric consistency using PCC, on the DL3DV, RE10K, and T&T datasets. Arrows indicate whether higher (↑) or lower (↓) is better. We compare our method (ESC) and its variants against prior baselines (PhotoNVS, VIVID, OVIE, GLD, LagerNVS), with results reported as mean ± standard deviation. The best, second-best, and third-best results in each column are highlighted. ESC consistently outperforms all baselines across perceptual metrics, and the geometry-guided ESC-PCC-BoN variant further improves geometric consistency. Models marked with † were trained on the training split of the dataset. Models marked with ‡ were trained on a restricted subset comprising less than half of the full training split.

from identical source inputs and conditioning states (see multi-sample variations in Figures 5, 22 and 23). As shown, ESC-PCC-BoN produces improved predictions closer to the ground-truth view. Additional results on a custom synthetic Blender simulation dataset with ground-truth depth maps are provided in Appendix F.2.

5 Conclusion

In this paper, we presented ESC, a geometry-aware diffusion framework designed to overcome the inherent 3D ambiguities of monocular novel view synthesis. By moving beyond pure 2D image-plane inpainting, ESC explicitly grounds the generative process in 3D space through joint RGB and depth prediction, guided by Plücker ray conditioning. To strictly enforce geometric fidelity, we introduced the PCC, a novel bidirectional depth-reprojection mechanism that evaluates the 3D spatial validity of every generated pixel to guarantee structural agreement between the source and target views. Our extensive evaluations demonstrate that PCC serves as both a highly robust metric—capturing true multi-view 3D consistency where standard 2D perceptual and camera-pose metrics fail—and an effective objective for inference-time Best-of-N selection. Ultimately, by coupling the strong visual priors of modern diffusion models with explicit geometric verification, our framework successfully escapes the constraints of 2D ambiguity, yielding novel views that are both photorealistic and rigorously structurally coherent.

AI USE STATEMENT

In this work, we used generative AI tools for generate synthetic data sets, formulate mathematical claims, assist in the writing of proofs, implement methods, formulate mathematical claims and interpret results.

We have not used generative AI tools for help develop theoretical models or conceptual frameworks, provide critical ingredients for proving mathematical claims, propose or refine hypotheses, design or provide feedback on research methodology or experiments, clean and reformat dataset and interpret results, assist with translation or support qualitative and thematic data analysis and create or modify scientific figures or images are not applicable to this work. Additionally, we used generative AI tools for sourcing/searching for information, edit the paper to improve readability, format the paper and draft parts of the research paper. We reviewed all AI-assisted work. We manually inspected and tested all LLM-generated code for correctness and consistency with the intended implementation. All authors also inspected the LLM-generated mathematical claims and verified their correctness and consistency with the paper's assumptions, derivations, and empirical findings. We reviewed all LLM-edited text to ensure that it accurately reflects our methods, findings, and conclusions, preserves the intended meaning, and introduces no unsupported claims. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We took several measures to ensure the reproducibility of our results. The model architecture, training objectives, and inference procedure are described in Section 3.2, while the datasets, training setup, and evaluation protocol are presented in Section 4.1. Complete hyperparameter and checkpoint configurations are provided in Table 4, with dataset and preprocessing details in Appendix A. The three benchmark datasets used in our experiments—DL3DV, RealEstate10K, and Tanks & Temples—are publicly available. Our synthetic Blender dataset is also straightforward to recreate: it uses simple geometric primitives available through the open-source Kubric toolkit (Greff et al., 2022), and its complete generation procedure is given in Algorithm 1. The appendix further provides pseudocode for PCC in Algorithm 2. Together, these resources provide the information needed to reproduce our training and evaluation pipeline.

REFERENCES

Shariq Farooq Bhat, Reiner Birkl, Diana Wofk, Peter Wonka, and Matthias Müller. Zoedepth: Zero-shot transfer by combining relative and metric depth, 2023. https://arxiv.org/abs/2302.12288.

Blender Online Community. Blender - the free and open source 3d creation software, 2026. http://www.blender.org.

Aleksei Bochkovskii, Amaël Delaunoy, Hugo Germain, Marcel Santos, Yichao Zhou, Stephan R. Richter, and Vladlen Koltun. Depth pro: Sharp monocular metric depth in less than a second, 2025. https://arxiv.org/abs/2410.02073.

Chenjie Cao, Chaohui Yu, Fan Wang, Xiangyang Xue, and Yanwei Fu. Mvinpainter: Learning multi-view consistent inpainting to bridge 2d and 3d editing, 2024. https://arxiv.org/abs/2408.08000.

Kaihua Chen, Tarasha Khurana, and Deva Ramanan. Reconstruct, inpaint, test-time finetune: Dynamic novel-view synthesis from monocular videos, 2026. https://arxiv.org/abs/2507.12646.

dx8152. Qwen-edit-2509-multiple-angles. https://huggingface.co/dx8152/Qwen-Edit-2509-Multiple-angles, 2025.

Noam Elata, Bahjat Kawar, Yaron Ostrovsky-Berman, Miriam Farber, and Ron Sokolovsky. Novel view synthesis with pixel-space diffusion models, 2024. https://arxiv.org/abs/2411.07765.

fal.ai. Qwen-image-edit-2511-multiple-angles-lora. https://huggingface.co/fal/Qwen-Image-Edit-2511-Multiple-Angles-LoRA, 2025.

Stephanie Fu, Netanel Tamir, Shobhita Sundaram, Lucy Chai, Richard Zhang, Tali Dekel, and Phillip Isola. Dreamsim: Learning new dimensions of human visual similarity using synthetic data. In *Advances in Neural Information Processing Systems*, volume 36, pages 50742–50768, 2023.

Google DeepMind. Imagen — google deepmind. https://deepmind.google/models/imagen/, 2026. Accessed: 2026-08-27.

Klaus Greff, Francois Belletti, Lucas Beyer, Carl Doersch, Yilun Du, Daniel Duckworth, David J Fleet, Dan Gnanapragasam, Florian Golemo, Charles Herrmann, Thomas Kipf, Abhijit Kundu, Dmitry Lagun, Issam Laradji, Hsueh-Ti (Derek) Liu, Henning Meyer, Yishu Miao, Derek Nowrouzezahrai, Cengiz Oztireli, Etienne Pot, Noha Radwan, Daniel Rebain, Sara Sabour, Mehdi S. M. Sajjadi, Matan Sela, Vincent Sitzmann, Austin Stone, Deqing Sun, Suhani Vora, Ziyu Wang, Tianhao Wu, Kwang Moo Yi, Fangcheng Zhong, and Andrea Tagliasacchi. Kubric: a scalable dataset generator. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2 edition, 2004.

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

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. https://proceedings.neurips.cc/paper_files/paper/2017/file/8a1d694707eb0fefe65871369074926d-Paper.pdf.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *CoRR*, abs/2006.11239, 2020. https://arxiv.org/abs/2006.11239.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models, 2021. https://arxiv.org/abs/2106.09685.

Tianyu Huang, Wangguandong Zheng, Tengfei Wang, Yuhao Liu, Zhenwei Wang, Junta Wu, Jie Jiang, Hui Li, Rynson W. H. Lau, Wangmeng Zuo, and Chunchao Guo. Voyager: Long-range and world-consistent video diffusion for explorable 3d scene generation, 2025. https://arxiv.org/abs/2506.04225.

Wooseok Jang, Seonghu Jeon, Jisang Han, Jinhyeok Choi, Minkyung Kwon, Seungryong Kim, Saining Xie, and Sainan Liu. Repurposing geometric foundation models for multi-view diffusion. *arXiv preprint arXiv:2603.22275*, 2026.

Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *CVPR*, 2017.

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering, 2023. https://arxiv.org/abs/2308.04079.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes, 2022. https://arxiv.org/abs/1312.6114.

Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Tanks and temples: Benchmarking large-scale scene reconstruction. *ACM Transactions on Graphics*, 36(4), 2017.

Vincent Leroy, Yohann Cabon, and Jérôme Revaud. Grounding image matching in 3d with mast3r, 2024. https://arxiv.org/abs/2406.09756.

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, 2025. https://arxiv.org/abs/2511.10647.

Kuan Heng Lin, Zhizheng Liu, Pablo Salamanca, Yash Kant, Ryan Burgert, Yuancheng Xu, Koichi Namekata, Yiwei Zhao, Bolei Zhou, Micah Goldblum, Paul Debevec, and Ning Yu. Vista4d: Video reshooting with 4d point clouds, 2026. https://arxiv.org/abs/2604.21915.

Lu Ling, Yichen Sheng, Zhi Tu, Wentian Zhao, Cheng Xin, Kun Wan, Lantao Yu, Qianyu Guo, Zixun Yu, Yawen Lu, Xuanmao Li, Xingpeng Sun, Rohan Ashok, Aniruddha Mukherjee, Hao Kang, Xiangrui Kong, Gang Hua, Tianyi Zhang, Bedrich Benes, and Aniket Bera. Dl3dv-10k: A large-scale scene dataset for deep learning-based 3d vision, 2023. https://arxiv.org/abs/2312.16256.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling, 2023. https://arxiv.org/abs/2210.02747.

Ilya Loshchilov and Frank Hutter. Fixing weight decay regularization in adam. *CoRR*, abs/1711.05101, 2017. http://arxiv.org/abs/1711.05101.

Lars Mescheder, Wei Dong, Shiwei Li, Xuyang Bai, Marcel Santos, Peiyun Hu, Bruno Lecouat, Mingmin Zhen, Amaël Delaunoy, Tian Fang, Yanghai Tsin, Stephan R. Richter, and Vladlen Koltun. Sharp monocular view synthesis in less than a second, 2026. https://arxiv.org/abs/2512.10685.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision, 2024. https://arxiv.org/abs/2304.07193.

Nobuyuki Otsu. A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 9(1):62–66, 1979. doi: 10.1109/TSMC.1979.4310076.

William Peebles and Saining Xie. Scalable diffusion models with transformers, 2023. https://arxiv.org/abs/2212.09748.

B. T. Polyak and A. B. Juditsky. Acceleration of stochastic approximation by averaging. *SIAM Journal on Control and Optimization*, 30(4):838–855, 1992. doi: 10.1137/0330046. https://doi.org/10.1137/0330046.

Adrien Ramanana Rahary, Nicolas Dufour, Patrick Perez, and David Picard. One view is enough! monocular training for in-the-wild novel view generation, 2026. https://arxiv.org/abs/2603.23488.

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, 2025. https://arxiv.org/abs/2503.03751.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. *CoRR*, abs/2112.10752, 2021. https://arxiv.org/abs/2112.10752.

Johannes L. Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016a.

Johannes L. Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016b.

Chenxi Song, Yanming Yang, Tong Zhao, Ruibo Li, and Chi Zhang. Taming video models for 3d and 4d generation via zero-shot camera control, 2026. https://arxiv.org/abs/2509.15130.

Stanislaw Szymanowicz, Eldar Insafutdinov, Chuanxia Zheng, Dylan Campbell, João F. Henriques, Christian Rupprecht, and Andrea Vedaldi. Flash3d: Feed-forward generalisable 3d scene reconstruction from a single image, 2025. https://arxiv.org/abs/2406.04343.

Stanislaw Szymanowicz, Minghao Chen, Jianyuan Wang, Christian Rupprecht, and Andrea Vedaldi. Lagernvs: Latent geometry for fully neural real-time novel view synthesis, 2026. https://arxiv.org/abs/2603.20176.

Jianyuan Wang, Christian Rupprecht, and David Novotny. Posediffusion: Solving pose estimation via diffusion-aided bundle adjustment, 2024a. https://arxiv.org/abs/2306.15667.

Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer, 2025. https://arxiv.org/abs/2503.11651.

Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric 3d vision made easy, 2024b. https://arxiv.org/abs/2312.14132.

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

Chenfei Wu, Jiahao Li, Jingren Zhou, Junyang Lin, Kaiyuan Gao, Kun Yan, Sheng ming Yin, Shuai Bai, Xiao Xu, Yilei Chen, Yuxiang Chen, Zecheng Tang, Zekai Zhang, Zhengyi Wang, An Yang, Bowen Yu, Chen Cheng, Dayiheng Liu, Deqing Li, Hang Zhang, Hao Meng, Hu Wei, Jingyuan Ni, Kai Chen, Kuan Cao, Liang Peng, Lin Qu, Minggang Wu, Peng Wang, Shuting Yu, Tingkun Wen, Wensen Feng, Xiaoxiao Xu, Yi Wang, Yichang Zhang, Yongqiang Zhu, Yujia Wu, Yuxuan Cai, and Zenan Liu. Qwen-image technical report, 2025. https://arxiv.org/abs/2508.02324.

Jason J. Yu, Fereshteh Forghani, Konstantinos G. Derpanis, and Marcus A. Brubaker. Long-term photometric consistent novel view synthesis with diffusion models, 2023. https://arxiv.org/abs/2304.10700.

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, 2024. https://arxiv.org/abs/2409.02048.

Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models, 2023. https://arxiv.org/abs/2302.05543.

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 586–595, 2018. doi: 10.1109/CVPR.2018.00068.

Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: Learning view synthesis using multiplane images. *ACM Trans. Graph. (Proc. SIGGRAPH)*, 37, 2018. https://arxiv.org/abs/1805.09817.

Matthias Zwicker, Hanspeter Pfister, Jeroen Van Baar, and Markus Gross. Surface splatting. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 371–378, 2001.

A Datasets **15**
A.1 Overview . . . 15
A.2 The Blender + Kubric Synthetic Dataset . . . 15
A.3 Mixed Real-World Training Dataset . . . 16
A.4 Evaluation Pair Selection . . . 18

B Training Configurations **18**
B.1 Plücker Ray Map Conditioning . . . 18

C PCC: Supplementary Document **19**
C.1 Scale Jitter Ablation . . . 19
C.2 Scene Visualizations of the Occlusion Ablation . . . 20
C.3 Scale Alignment for PCC . . . 21
C.4 Far Depth Pruning . . . 21
C.5 Camera-Pose Accuracy . . . 21
C.6 Visualization of Scale-Alignment for DA3 . . . 22
C.7 PCC Evaluation in Practice . . . 23
C.8 PCC Pseudo Code . . . 25
C.9 Differentiable PCC Loss via Inverse Rendering . . . 25
C.10 Quantitative Results of the Jitter and Occlusion Experiments . . . 26
C.11 Multi-View PCC . . . 27
C.12 PCC vs Standard Evaluation Metrics . . . 30
C.12.1 Evaluating PCC Against 2D Perception Metrics . . . 30
C.12.2 Evaluating PCC Against Camera Pose Metrics . . . 31
C.12.3 Evaluation on a Real-World Scene . . . 32

D Sensitivity Analyses **32**
D.1 How Do Frame Gap and Camera Displacement Affect Performance? . . . 32
D.2 Escaping the 2D Cave: Does Depth Conditioning Truly Benefit Novel View Synthesis? 34

E Baseline Models **35**
E.1 Inference Efficiency . . . 35

F Additional Results **36**
F.1 Compute Time . . . 36
F.2 How does ESC perform in simulation with ground-truth geometry? . . . 44

A DATASETS

A.1 OVERVIEW

We train and evaluate ESC and baseline models on a mix of one synthetic and three real-world novel-view synthesis benchmarks.

- **Blender** (Blender Online Community, 2026) **+ Kubric** (Greff et al., 2022) **(ours, synthetic).** A procedurally generated multi-view dataset of ∼2,000 training scenes and 100 validation scenes. Each scene contains a small object-centered scene of solid-coloured `Cube`/`Sphere` primitives spawned with Kubric and rendered with Blender from a fixed 15-view yaw×pitch grid.
- **DL3DV-10K** (Ling et al., 2023). A large in-the-wild capture collection of ∼10K trajectories covering indoor and outdoor scenes, with COLMAP-derived intrinsics and extrinsics. We use 8,559 scenes of the training split for finetuning and the dedicated evaluation split for testing.
- **RealEstate10K (RE10K)** (Zhou et al., 2018). A real-world collection of ∼72K indoor real-estate walk-through clips with per-frame camera poses. We use a subset of the official train split for finetuning and the held-out test videos for evaluation.
- **Tanks & Temples (T&T)** (Knapitsch et al., 2017). An evaluation benchmark of high-resolution outdoor and indoor captures with COLMAP-aligned poses; we never train on T&T and only use it to probe out-of-distribution generalization.

The synthetic Blender + Kubric dataset is used for controlled pre-training and ablation studies; the DL3DV/RE10K mix is used for the main finetuning runs; and all three real-world benchmarks (DL3DV, RE10K, T&T) are used for quantitative evaluation.

A.2 THE BLENDER + KUBRIC SYNTHETIC DATASET

Engine stack. Each scene is procedurally composed by Kubric (Greff et al., 2022) — which spawns the primitive objects, assigns their materials, and orchestrates the multi-view camera grid and then handed off to Blender (Blender Online Community, 2026) as the underlying renderer to produce RGB, depth and camera poses.

Assets. We deliberately keep the scene content minimal. Each scene contains a small set of solid-coloured **Kubric primitives only**: `kb.Cube` and `kb.Sphere` instances spawned on a flat ground plane. The number of objects per scene is small and varies slightly: empirically every scene contains 5 or 6 primitives (mean $N_{\text{obj}} \approx 5.2$ across a random sample of 50 scenes). Each primitive is assigned:

- a class drawn uniformly at random from {`Cube`, `Sphere`};
- a scale $s \sim \mathcal{U}(0.8, 1.6)$;
- a colour cycled through a fixed 8-colour palette (red, green, blue, yellow, pink, cyan, orange, purple);
- a Principled BSDF material with metallic $\sim \mathcal{U}(0, 1)$ and roughness $\sim \mathcal{U}(0.2, 0.8)$.

By restricting the visible content to low frequencies and providing *robust geometric cues*, this primitive representation is ideally suited for diffusion-based NVS training, where every pixel can be directly mapped to a well-defined analytic surface for depth supervision.

Placement. Objects are arranged in a ring around the scene origin at azimuth $\theta_i = 2\pi i / N_{\text{obj}}$ with radius $r_i \sim \mathcal{U}(1.0, 2.5)$ m, and sit directly on the ground plane at $z = s_i$ (so the object base touches $z = 0$).

Lighting. Because the dataset uses no HDRI environment, all illumination comes from a single deterministic light rig: the CLEVR (Johnson et al., 2017) three-point setup, instantiated via `kb.assets.utils.get_clevr_lights(light_jitter=0.5)`, plus a uniform ambient term $\mathbf{c}_{\text{amb}} = 0.05$ on every channel.

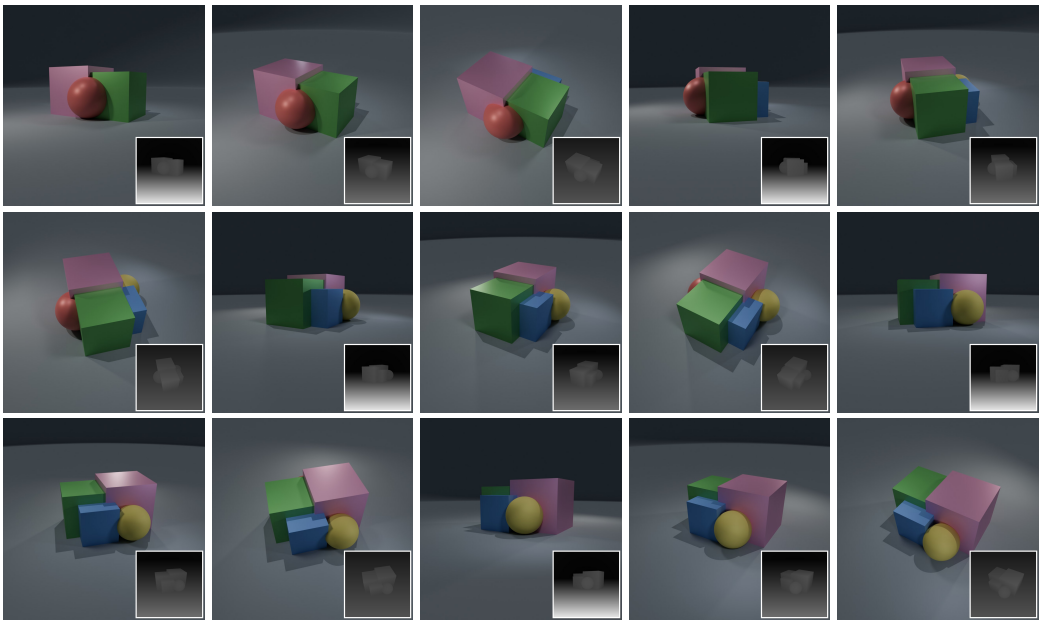

Figure 6: **One Blender + Kubric scene rendered from all 15 grid viewpoints, with ground-truth disparity overlaid.** Each tile shows the rendered RGB view; the bottom-right is the corresponding ground-truth depth converted to inverse-depth (disparity). Rows / columns follow the shared yaw $\times$ pitch grid (yaws $\{-90°, -45°, 0°, +45°, +90°\}$, pitches $\{-20°, 0°, +20°\}$).

Camera grid. Each scene renders $N_{\text{view}}{=}15$ views arranged on a fixed (yaw, pitch) grid centred on the scene origin and *shared across the whole dataset*. The grid uses 5 yaw offsets $\in \{-90°, -45°, 0°, +45°, +90°\}$ crossed with 3 pitch offsets $\in \{-20°, 0°, +20°\}$ about a reference elevation of $30°$, with focal length 35 mm and a radius of $\sim$12 m from the scene origin.

Per-scene generation procedure. We provide a visual demonstration of a single generated scene in Figure 6 and outline the generation process via pseudocode in Algorithm 1.

A.3 Mixed Real-World Training Dataset

Composition. The primary fine-tuning runs use a weighted mixture of two real-world datasets: DL3DV and RE10K. We train on a selected subset of these datasets, with the exact number of scenes reported in Table 2. The mixture is governed by a single *per-source* weight w_k (Table 2). Operationally, the data loader proceeds as follows:

1. It converts each source weight into a *per-scene* weight $w_k/|\mathcal{S}_k|$, where $|\mathcal{S}_k|$ denotes the number of scenes available from source k on disk.
2. It passes these per-scene weights to the `WeightedSceneGroupedSampler`, which samples the next scene with a probability proportional to its per-scene weight.

Source	Num. of Scenes Loaded	Weight	Min gap	Max gap
DL3DV-10K	8,559	0.4	5	25
RealEstate10K	26,628	0.6	10	50

Table 2: **Mixed training mixture.** Two real-world sources are sampled in proportion to the listed *weight*. *Min/Max gap* is the temporal stride (in frames) between the source and target frame of a training pair; the wider RE10K range reflects its faster intra-scene camera motion.

Algorithm 1 Procedural rendering of one Blender + Kubric scene.

Require: seed s, object count $N_{\text{obj}} \in \{5, 6\}$, view count N_{view}=15, resolution 1024×1024, shared view grid $\mathcal{V} = \{(\text{yaw}_i, \text{pitch}_i)\}_{i=1}^{N_{\text{view}}}$, colour palette $\mathcal{C}$

RNG $\leftarrow \text{seed}(s)$
scene $\leftarrow$ KB.SCENE(resolution, frame_end= N_{view})
ADDGROUNDPLANE(scene)
ADDLIGHTS(scene) ▷ CLEVR rig (jitter 0.5) + ambient 0.05
for $i \leftarrow 1$ to N_{obj} **do**
    cls $\leftarrow$ RNG.choice({`Cube`, `Sphere`})
    scale $\leftarrow$ RNG.uniform(0.8, 1.6)
    colour $\leftarrow \mathcal{C}[i \bmod |\mathcal{C}|]$
    $\theta \leftarrow 2\pi i/N_{\text{obj}}$; $r \leftarrow$ RNG.uniform(1.0, 2.5)
    pos $\leftarrow (r\cos\theta,\ r\sin\theta,\ \text{scale})$
    $o \leftarrow$ cls(scale, pos, PrincipledBSDF(colour, m, ρ))
    scene $+= o$
end for
SETUPCAMERAKEYFRAMES(scene, $\mathcal{V}$, ref_elev= $30°$, focal= 35mm)
SETUPCOMPOSITORFORDEPTH(scene)
for $k \leftarrow 1$ to N_{view} **do**
    RENDER(scene, frame= k)
    SAVE(rgb$_k$, depth$_k$)
end for

Depth and Pose Pre-extraction. We pre-extract depth maps and camera trajectories (intrinsics and extrinsics) offline using the DepthAnything-3 (DA3) (Lin et al., 2025) backbone (`depth-anything/da3nested-giant-large`) in "video" mode with `process_res_method="upper_bound_resize"`. During preprocessing, we observe that DA3 estimations exhibit greater stability in multi-view settings than in monocular contexts; specifically, increasing the sequence length significantly enhances prediction precision and reliability. Motivated by this, we partition the DL3DV and RE10K datasets into 60-frame chunks, processing each independently at 960 px to maximize the temporal context provided to DA3 simultaneously.

Because DA3 jointly estimates both scene depth and camera geometry, the outputs within each chunk inherently share a consistent coordinate frame. To maintain this consistency, training pairs are sampled exclusively from within the same chunk using these DA3-estimated poses. For evaluation, however, we follow standard benchmark protocols and utilize the ground-truth COLMAP or Structure-from-Motion (SfM) poses provided natively by the datasets. Crucially, this offline precomputation allows the training loaders to stream static on-disk artifacts directly, eliminating the runtime computational overhead of re-running DA3..

Pre-processing. All RGB video frames are resized to 384×384. Depth supervision is provided using the precomputed DA3 predictions as soft targets. To normalize the camera configurations, we map them to a physical disparity range of $[d_{\min}, d_{\max}] = [0.02, 2.0]$.

Training-Time Pair Selection. To construct training pairs, both dataset loaders employ a *chunk-aware random sampling* strategy. For any selected scene, a frame pair $(f_{\text{src}}, f_{\text{tgt}})$ is generated via the following pipeline:

1. **Anchor Selection:** We randomly sample a source anchor frame f_{src} from the set of frames covered by a valid DA3 chunk.
2. **Candidate Filtering:** We construct a valid target candidate set $\mathcal{C} = \{f : |f - f_{\text{src}}| \in [g_{\min}, g_{\max}]\}$. Critically, this set is strictly restricted to the same chunk as f_{src} to prevent mixing inconsistent coordinate frames across chunk boundaries.
3. **Target Sampling:** The final target frame is drawn uniformly from the candidate set, denoted as $f_{\text{tgt}} \sim \mathcal{U}(\mathcal{C})$.

A.4 Evaluation Pair Selection

Per-Benchmark Evaluation Configurations. Table 3 summarizes the configurations used across our evaluation sets. Pairs are sampled from within the 60-frame chunks defined during DA3 preprocessing, adhering to the dataset-specific stride and frame-gap constraints listed below.For DL3DV, we utilize the entire evaluation set, excluding a single corrupted scene. For RealEstate10K (RE10k), we evaluate on a test subset. Finally, for the Tanks & Temples dataset, our evaluation is strictly restricted to the following seven scenes: `Barn`, `Caterpillar`, `Church`, `Courthouse`, `Ignatius`, `Meetingroom`, and `Truck`.

Dataset	Stride	Min Gap	Max Gap	Num. Pairs	Num. Scenes
DL3DV-10K	5	5	25	28,780	54
RealEstate10K	10	10	50	18,084	1,991
Tanks & Temples	1	1	25	5,642	7

Table 3: Evaluation benchmark configurations, denoting the sequence frame stride, minimum/maximum frame separation gaps, total evaluated pairs, and scene counts.

B Training Configurations

The full training configuration used to produce the evaluated checkpoints is reported in Table 4 for the mixed-training run.

Hyper-parameter	Value
Base model	`Qwen/Qwen-Image-Edit-2509`
Transformer backbone	`linoyts/Qwen-Image-Edit-Rapid-AIO`
Lightning weights	`Qwen-Image-Edit-2511-Lightning-8steps-V1.0-bf16`
LoRA rank / α	128 / 128 (64 / 64 for Blender training)
LoRA target modules	q, k, v, out, MLP, mod, in, proj_out
LoRA learning rate	5×10^{-5}
ControlNet layers / scale	5 / 1.0
Learning rate	1×10^{-4} (cosine, min $= 10^{-6}$)
Mixed precision	bf16
Gradient clipping	1.0
EMA decay (from step 10k)	0.99995
Loss weights	w_{fm}=1.0, $w_{\text{fm,tgt-rgb}}$=1.0, $w_{\text{fm,tgt-dep}}$=0.75, w_{recon}=0.5, w_{lpips}=0.25, w_{dino}=0.02
Timestep sampling	logit-normal, shift $= 1.5$, base $= 4096$

Table 4: **Mixed training configuration.** Values reproduced verbatim from the released `config.json` (mixed-DA3 run).

B.1 Plücker Ray Map Conditioning

To condition the ControlNet, target camera rays are processed and injected via three steps:

- **Plücker Mapping:** Source and target rays are back-projected into a 6D Plücker coordinate map.
- **Lossless Tokenization:** The map is deterministically reshaped into flat geometric tokens, bypassing learned convolutions.
- **Zero-Init Injection:** A zero-initialized linear layer projects these tokens to the transformer width. They are then added element-wise to the image latents, ensuring a stable "no-op" initialization at the start of training.

C PCC: Supplementary Document

C.1 Scale Jitter Ablation

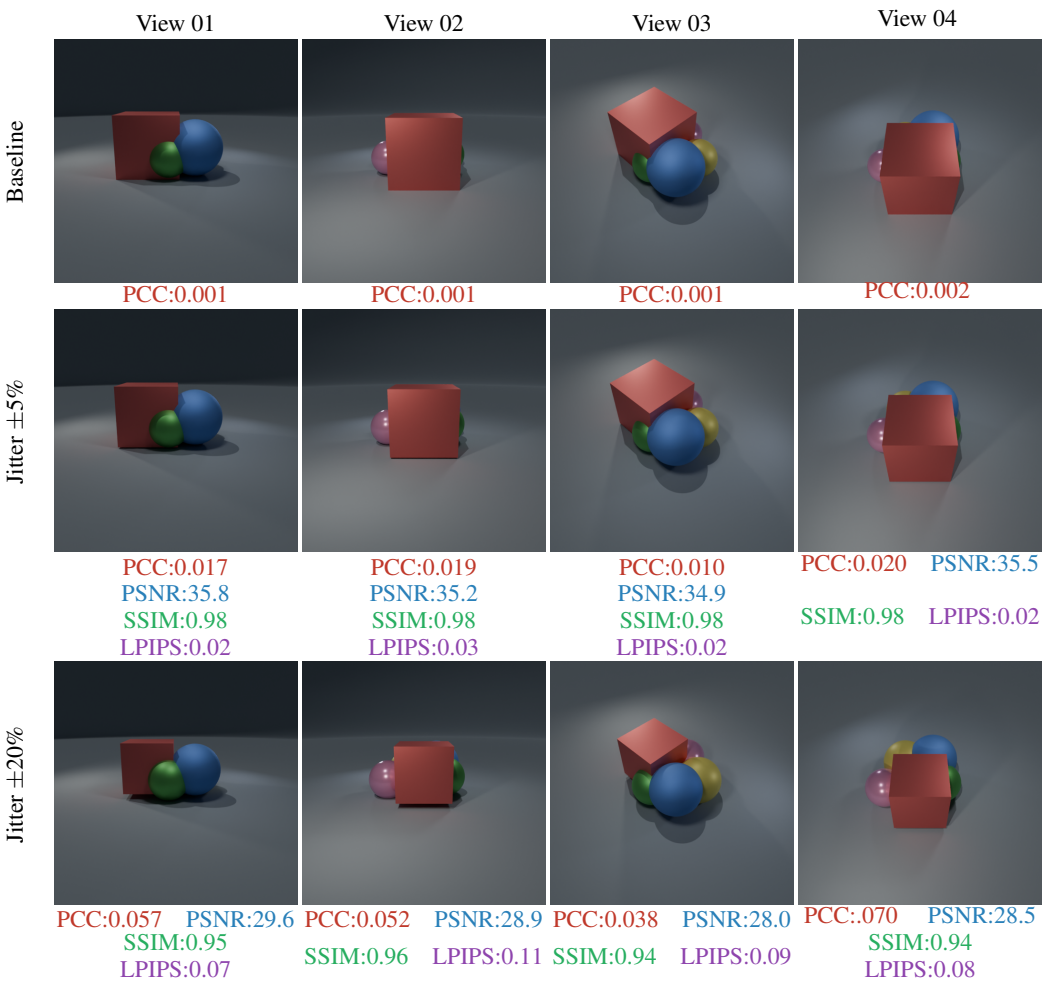

Figure 7: Scale jitter experiment. Baseline (top) with no jitter, and two variants where each object is randomly scaled by ±5% and ±20%. All metrics degrade together: PCC increases (worse 3D consistency), while PSNR/SSIM/LPIPS decrease (worse visual similarity). Unlike the occlusion experiment, scale jitter is detectable by all metrics since it changes both geometry and appearance uniformly.

C.2 Scene Visualizations of the Occlusion Ablation

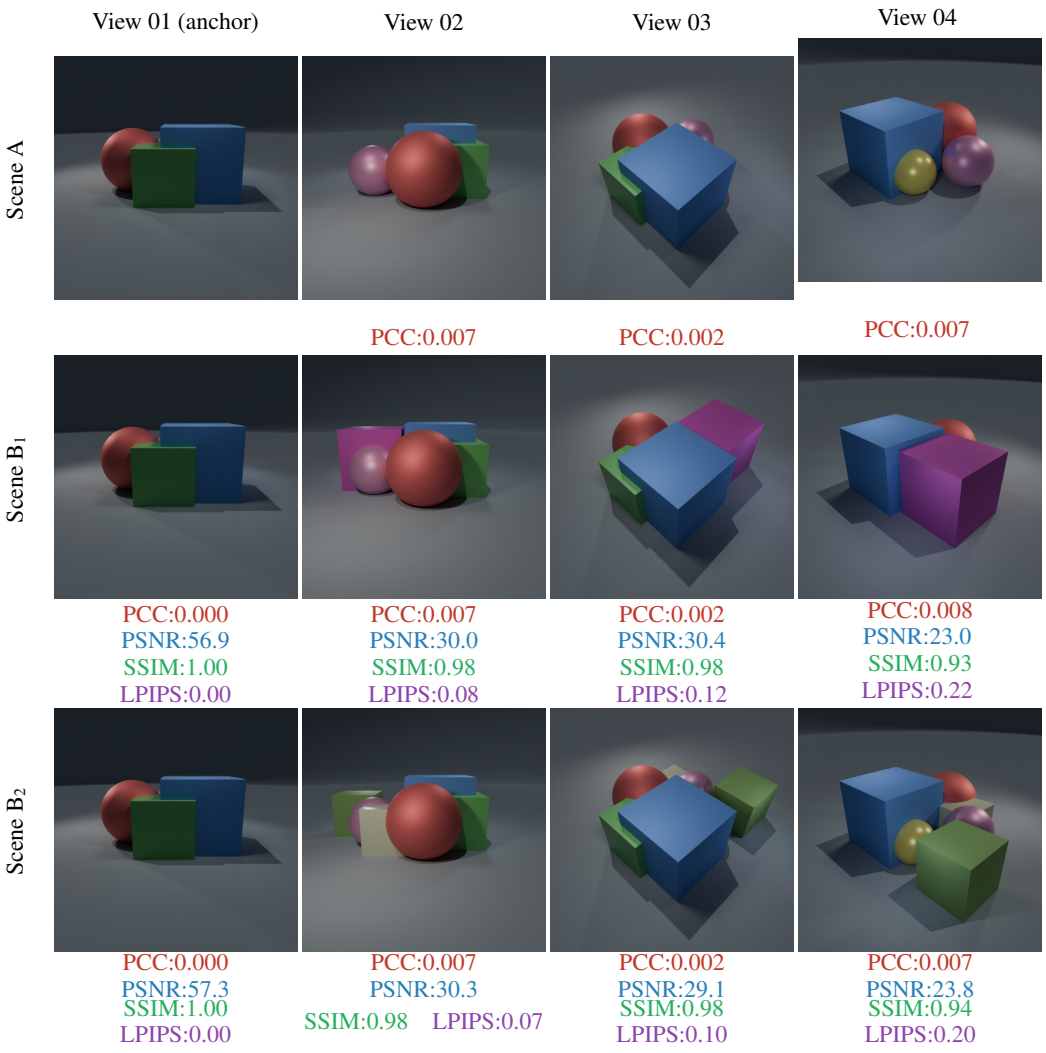

Figure 8: Occlusion experiment. Scene A (top) and two variants B_1, B_2 where objects are added behind the anchor camera (View 01). Metrics: PCC / PSNR / SSIM / LPIPS. For Scene A, 2D metrics are perfect since each view is compared to itself. For Scene B variants, PCC remains unchanged (≈Scene A) because each scene is internally 3D-consistent, while PSNR/SSIM/LPIPS correctly detect the visual differences from added objects.

C.3 SCALE ALIGNMENT FOR PCC

Because the multi-view geometry estimator (i.e DA3 (Lin et al., 2025)) jointly predicts depth and camera poses from RGB input alone, its outputs are defined only up to a global similarity transformation. Since each inference step minimizes multi-view photometric and geometric inconsistencies, processing frames from two different versions of a scene (e.g., a reference and a perturbed variant) forces the model to artificially deform the recovered geometry to ensure the reconstructions overlap. As a result, the estimator silently absorbs the true 3D inconsistency into its pose and scale predictions, meaning the final measured residual reflects only the leftover error rather than the actual 3D mismatch.

To resolve this ambiguity, we fit a single affine transformation, shared by the source and target views, that aligns the DA3-predicted depths to ground-truth metric depths over all valid pixels in both views:

$$\alpha^\star,\, \beta^\star \;=\; \arg\min_{\alpha,\,\beta} \sum_{v\in\{s,t\}} \sum_{p\in\mathcal{V}_v} \big(\alpha\, \hat{D}_v(p) + \beta - D_v^{\mathrm{gt}}(p)\big)^2, \tag{8}$$

where $v \in \{s, t\}$ indexes the source (s) or target (t) view, p denotes a pixel location, $\hat{D}_v(p)$ is the depth predicted by DA3 at pixel p in view v, and $D_v^{\mathrm{gt}}(p)$ is the corresponding ground-truth metric depth. The set $\mathcal{V}_v = \{p \mid \hat{D}_v(p) > 0 \wedge D_v^{\mathrm{gt}}(p) > 0\}$ contains pixels for which both depths are valid and positive. The variables α and β are, respectively, the shared multiplicative scale and additive shift, and $(\alpha^\star, \beta^\star)$ are their least-squares-optimal values. Using one shared pair for both views preserves their relative depth scale rather than aligning each depth map independently.

The fitted parameters (α^*, β^*) are then applied to both the source- and target-view depths:

$$\tilde{\mathbf{D}} \;=\; \max\big(\alpha^* \hat{\mathbf{D}} + \beta^*,\; 0\big).$$

This operation ensures that the relative depth structure predicted by DA3 across the two views is preserved, while the global scale and offset remain strictly locked to the GT metric scale.

Foreground/background scale alignment. In synthetic scenes with a plain backdrop, we fit *two* independent affines, one for the foreground and one for the background. The foreground/background partition is obtained automatically by 1-D *Otsu thresholding* (Otsu, 1979) on the pooled GT depths of the source and target views: Otsu picks the threshold t^* that maximizes the between-class variance of the depth histogram. Within each set we pool DA3/GT pixel pairs from both views and solve a 2×2 least-squares system for $(\alpha_{\mathrm{fg}}, \beta_{\mathrm{fg}})$ and $(\alpha_{\mathrm{bg}}, \beta_{\mathrm{bg}})$. At inference time each pixel is aligned with the affine corresponding to its own GT bin:

$$\hat{d}_{\mathrm{aligned}}(p) = \begin{cases} \alpha_{\mathrm{bg}}\, \hat{d}_{\mathrm{DA3}}(p) + \beta_{\mathrm{bg}}, & \text{if } d_{\mathrm{GT}}(p) \geq t^*, \\ \alpha_{\mathrm{fg}}\, \hat{d}_{\mathrm{DA3}}(p) + \beta_{\mathrm{fg}}, & \text{otherwise.} \end{cases}$$

This keeps foreground geometry tight while still placing the dome at a plausible distance behind it, removing the FG/BG cluster split that a single global affine would otherwise produce.

C.4 FAR DEPTH PRUNING

Multi-view geometry estimators frequently produce unbounded or thresholded depth predictions in sky and far-background regions. To prevent these outliers from skewing both the scale-alignment computation and the reprojection loss, we occasionally mask out the highest $\rho = 3 - 5\%$ of valid *target-view* depths prior to alignment. We specify in the experimental section whenever this far-depth masking is utilized.

C.5 CAMERA-POSE ACCURACY

To quantify camera-pose accuracy, we compare the DA3-predicted camera poses from the two views against the ground truth. We adopt established pose evaluation metrics (Lin et al., 2025; Wang et al., 2024a; He et al., 2024; Song et al., 2026), utilizing DA3 as our designated geometry estimator.

Specifically, we calculate the relative rotation error (RPE-R) e_R and relative translation error(RPE-T) e_t for the relative pose between views:

$$e_R = \arccos\left(\frac{\mathrm{tr}\big(\hat{\mathbf{R}}_{\mathrm{rel}}^{\top}\mathbf{R}_{\mathrm{rel}}^{\mathrm{gt}}\big) - 1}{2}\right),$$

$$e_t = \arccos\big(\hat{\mathbf{t}}_{\mathrm{rel}} \cdot \mathbf{t}_{\mathrm{rel}}^{\mathrm{gt}}\big),$$

where $\mathbf{R}_{\mathrm{rel}}$ and $\mathbf{t}_{\mathrm{rel}}$ represent the rotation and unit translation direction of the relative transformation, and $\hat{\cdot}$ denotes DA3 estimates. The maximum of these forms the pairwise pose error: $e_{\mathrm{pose}} = \max(e_R, e_t)$. We report the area under the cumulative error curve (AUC) up to threshold θ:

$$\mathrm{AUC@}\theta \;=\; \frac{1}{\theta}\int_0^{\theta} \mathbf{1}\big[e_{\mathrm{pose}} \le \alpha\big]\,\mathrm{d}\alpha,$$

for $\theta \in \{3, 5, 15, 30\}$. Higher AUC values confirm that the generated view possesses sufficient geometric consistency for accurate relative pose recovery.

C.6 Visualization of Scale-Alignment for DA3

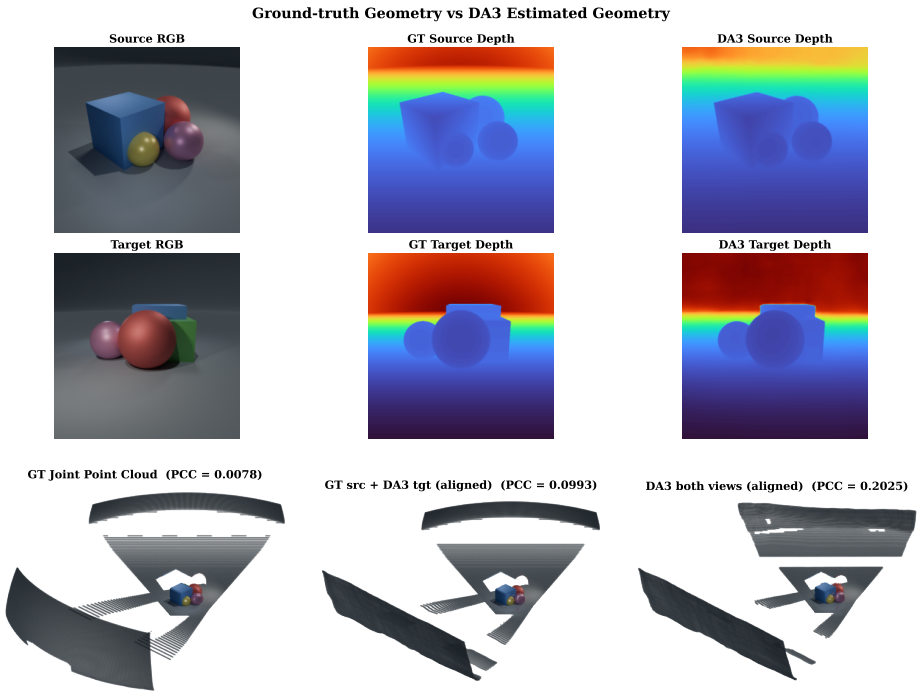

Figure 9: **Ground-truth versus DA3-estimated geometry for a two-view pair.** We visualize three distinct 3D reconstructions of a two-view scene. **Top two rows:** The source and target RGB inputs alongside their corresponding ground-truth (GT) depth maps (from Blender) and scale-aligned DA3 depth predictions, all color-coded by distance. **Bottom row (left to right):** (1) The reference GT joint point cloud, obtained by unprojecting both GT depths using GT camera poses (PCC = 0.008). (2) A mixed reconstruction utilizing the source GT depth and the target DA3 depth. The target depth is affine-aligned to the GT scale using a foreground/background split (determined via Otsu's thresholding on the pooled GT depth) to specifically isolate the error introduced by DA3 (PCC = 0.099). (3) The fully estimated point cloud, unprojected using DA3 depths and GT cameras for both views(PCC = 0.203).

C.7 PCC Evaluation in Practice

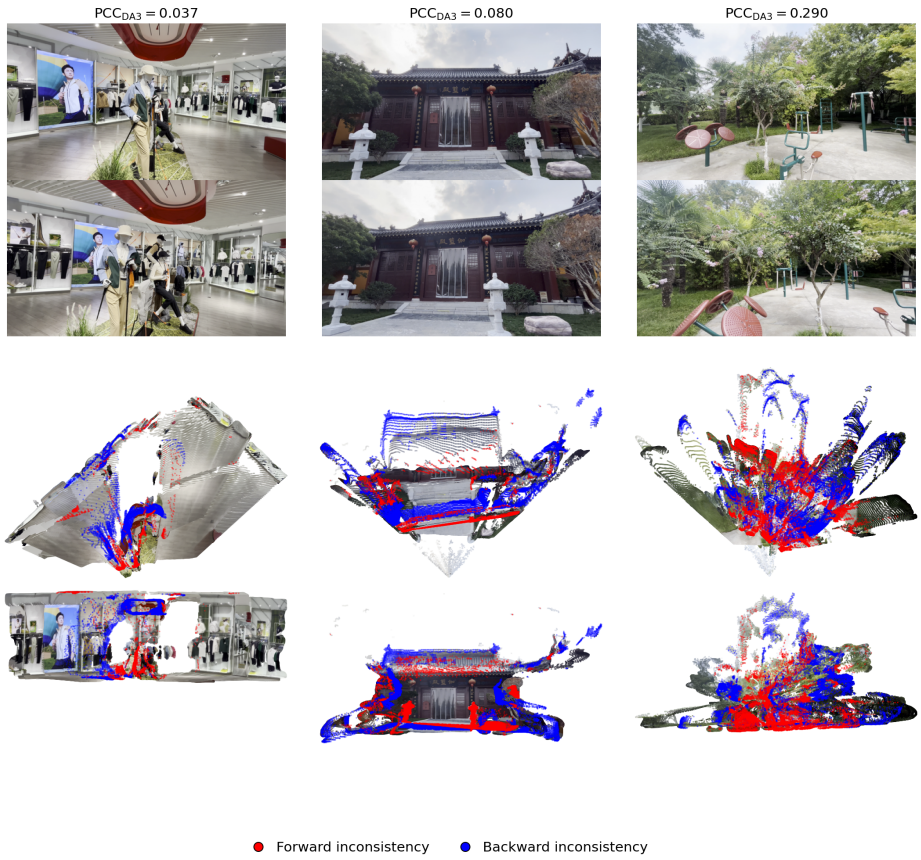

Figure 10: **Projection Consistency Constraint (PCC$_{DA3}$) on DL3DV.** Each column shows a source–target view pair: the top two rows are the source and target RGB frames, and the bottom two rows show the fused two-view point cloud (top-down and front) built from the GT source depth and DA3-estimated target depth (FG/BG affine-aligned), reprojected under the ground-truth poses. Consistent points keep their RGB color; red/blue highlight forward/backward reprojection inconsistencies, and the per-column PCC$_{DA3}$ score (lower is better) quantifies them.

Computing the PCC requires four primary inputs: the source depth, the target depth, and their respective camera poses. Crucially, the origin of these inputs—specifically, whether they are extracted from ground-truth data or estimated by a 3D reconstruction model—heavily influences the resulting metric values.

Because we employ a 3D reconstruction framework (DA3) to obtain these inputs, we outline our specific input configurations, dataset-dependent processing pipelines, and baselines below.

Camera Pose Configuration While an alternative approach would be to utilize the camera poses estimated directly by DA3, we strictly enforce the use of the dataset's ground-truth camera poses (derived from either COLMAP or the simulation environment) during the PCC evaluation. This design choice is critical for alignment with our core task: since our objective is to evaluate whether the generated output maintains 3D consistency conditioned on a specific camera prompt, we must measure performance against the exact geometric trajectory requested by that prompt.

Depth Processing and DA3 View-Sensitivity The pipeline for obtaining source and target depths depends on whether the underlying dataset is simulated or real-world. A key characteristic of DA3 is its high sensitivity to the number of input views; its depth and geometry estimations become significantly more stable and accurate as the number of viewpoints increases. Because Blender

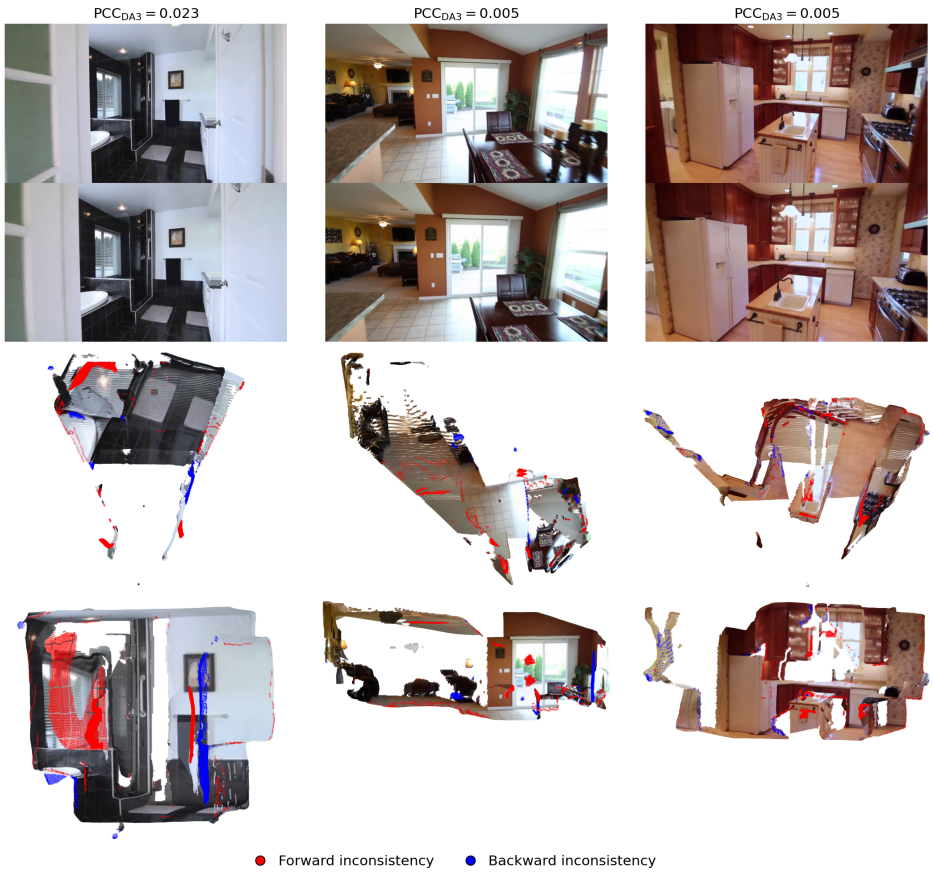

Figure 11: **Projection Consistency Constraint (PCC$_{DA3}$) on re10k.** Each column shows a source–target view pair: the top two rows are the source and target RGB frames, and the bottom two rows show the fused two-view point cloud (top-down and front) built from the GT source depth and DA3-estimated target depth (FG/BG affine-aligned), reprojected under the ground-truth poses. Consistent points keep their RGB color; red/blue highlight forward/backward reprojection inconsistencies, and the per-column PCC$_{DA3}$ score (lower is better) quantifies them.

datasets provide exact ground-truth depth while real-world, in-the-wild datasets do not, we handle them using two distinct paradigms:

- **Blender Simulated Scenes:** In these environments, exact ground-truth depth is available for the source view. Although DA3 can process a sparse 2-view input (the source and the generated target view) to estimate both depths simultaneously, its performance degrades severely in such few-view settings due to inherent reconstruction ambiguities (as illustrated in the rightmost panel of Figure 9). To prevent this degradation from polluting our evaluation, we only utilize DA3 to estimate the depth of the generated target view. We then align this estimated target depth with Blender's ground-truth target depth using the foreground-background alignment algorithm detailed in Appendix C.3. Figure 9 visualizes the final 3D reconstructed scene alongside its corresponding PCC value.
- **In-the-Wild Static Scenes:** Real-world video sequences lack ground-truth depth entirely. To leverage DA3's strength in dense-view scenarios, we preprocess the video data by partitioning sequences into 60-frame chunks. We treat this high-fidelity, 60-view reconstruction as the pseudo-ground-truth depth for each view, subsequent to which we apply the same alignment and PCC computation workflow used for the Blender scenes.

Error Propagation and the Oracle PCC Baseline It is important to note that because our evaluation pipeline relies on DA3-estimated depths, any inherent reconstruction errors from the model

Algorithm 2 Projection Consistency Check (PCC)

Require: source/target depth D_s, D_t; camera poses T_s, T_t; intrinsics K_s, K_t
Params: dead zone τ; max clamp c
Ensure: consistency score (lower is better; $0 =$ perfectly consistent)

function REPROJECT(D, T_{from}, T_{to})
  $P \leftarrow$ UNPROJECT(D, T_{from}) ▷ depth → 3D points in world
  return z-buffer render of P into the T_{to} camera
end function
▷ Forward: reproject source into target, compare to target depth
$\hat{D}_{s\to t},\ M_f \leftarrow$ REPROJECT(D_s, T_s, T_t)
$\Delta_f \leftarrow (D_t - \hat{D}_{s\to t})$ **on valid pixels** M_f
▷ Backward: reproject target into source, compare to source depth
$\hat{D}_{t\to s},\ M_b \leftarrow$ REPROJECT(D_t, T_t, T_s)
$\Delta_b \leftarrow (D_s - \hat{D}_{t\to s})$ **on valid pixels** M_b
▷ Robust, thresholded penalty (ignore noise, saturate outliers)
$\phi(\Delta) \leftarrow \min\big(\log(1 + \max(\Delta - \tau,\, 0)),\ c\big)$
$\text{score} \leftarrow \dfrac{\sum_{M_f} \phi(\Delta_f) + \sum_{M_b} \phi(\Delta_b)}{|M_f| + |M_b|}$
return score

will systematically affect the final PCC values. This limitation is visible even when validating on the original dataset pairs (where the target views are the actual ground-truth frames); due to estimation noise in DA3, the metric fails to achieve a perfect error score of 0 (see Figure 10 and Figure 11).To decouple our model's generation quality from the systematic errors of the reconstruction tool, we establish and report an Oracle PCC baseline across all experiments (Table 11 and Table 9). This oracle score represents the practical lower bound of the metric under DA3's geometric constraints, providing a realistic benchmark for perfect generation.

C.8 PCC PSEUDO CODE

The pseudocode for the PCC algorithm is provided in Algorithm 2.

C.9 DIFFERENTIABLE PCC LOSS VIA INVERSE RENDERING

The PCC score used for *evaluation* renders a depth map from one view into the other with a hard z-buffer: every source pixel is unprojected to 3D, projected into the target camera, rounded to the nearest integer pixel, and the closest point per target pixel is kept by sorting the splatted depths and letting nearer points overwrite farther ones. Writing $\mathrm{round}(\cdot)$ for the pixel snapping and $\mathrm{argsort}$ for the depth ordering, the rendered target depth is

$$D_t^{\text{zbuf}}(p) \;=\; \min_{q\,:\,\mathrm{round}(\pi_{s\to t}(q))\,=\,p} z_{s\to t}(q), \tag{9}$$

where $\pi_{s\to t}$ is the source-to-target reprojection and $z_{s\to t}$ the transformed depth. The three operations that make Eq. (9) an honest occlusion test — the $\mathrm{round}(\cdot)$ pixel snapping, the $\mathrm{argsort}$ depth ordering, and the scatter-assignment (a per-pixel $\min$ over a data-dependent index set) — are all piecewise-constant in the depth, so $\partial D_t^{\text{zbuf}}/\partial D = 0$ almost everywhere. The metric is therefore non-differentiable and cannot be used as a training objective on generated depth.

Inverse rendering via differentiable inverse warping. To obtain a trainable surrogate we replace the forward z-buffer *splat* (which scatters points *to* the target grid at rounded coordinates) with an *inverse warp* that, for every target pixel, analytically computes where it lands in the other view and *gathers* the other view's depth from those sub-pixel coordinates by bilinear sampling. Let $\mathbf{P}_s, \mathbf{P}_t \in SE(3)$ denote the OpenGL camera-to-world poses (X right, Y up, camera looking down $-Z$) and $K_v = (f_x^v, f_y^v, c_x^v, c_y^v)$ the pinhole intrinsics of view v. For a pixel $p = (u, v)$ with depth d

we unproject to the source camera frame,

$$\mathbf{x}_c \;=\; \Big(\tfrac{(u-c_x)\,d}{f_x},\; \tfrac{(c_y-v)\,d}{f_y},\; -d\Big)^{\top}, \tag{10}$$

transform it into the other camera with the single relative pose $\mathbf{T}_{f\to t} = \mathbf{P}_t^{-1}\mathbf{P}_f$, giving $\mathbf{x}' = (X', Y', Z')^{\top} = \mathbf{T}_{f\to t}\,\mathbf{x}_c$ and a reprojected depth $z' = -Z'$, and project back down,

$$u' = f'_x\,\frac{X'}{z'} + c'_x, \qquad v' = f'_y\,\frac{Y'}{z'} + c'_y, \qquad \mathbf{g}(p) = \Big(\tfrac{2u'}{W} - 1,\; \tfrac{2v'}{H} - 1\Big), \tag{11}$$

where $\mathbf{g}(p) \in [-1, 1]^2$ is the normalized sampling location (`align_corners=False`). A pixel is *valid* when its reprojection is finite, in front of the camera ($z' > \epsilon$), and lands inside the image. Given a depth map D and the grid $\mathbf{g}$, the inverse warp is the bilinear sampling operator $\mathcal{S}(D; \mathbf{g})$. Unlike Eq. (9), $\mathcal{S}$ is differentiable both with respect to the sampled *values* and to the sampling *coordinates* $\mathbf{g}$, so gradients propagate back into the predicted depth through the full projection chain in Eqs. (10)–(11).

Bidirectional gradient routing. Let $\hat{D}_t$ be the generated (target-view) depth we optimize and D_s the ground-truth source depth. We accumulate two terms, each of which places the gradient on $\hat{D}_t$ but routes it differently, so that the predicted surface is penalized whenever it pokes *in front of* the reprojected reference.

(i) Forward, source → target. We reproject the GT source depth into the target view. Because the coordinates are driven by GT data, the grid $\mathbf{g}_{s\to t}$ and the transformed reference depth z_t^{ref} are computed under $\mathrm{stop\text{-}grad}$; the gradient enters only through the *gathered prediction*

$$\hat{D}_t^{\rightarrow}(p) \;=\; \mathcal{S}\big(\hat{D}_t;\, \mathbf{g}_{s\to t}(p)\big), \qquad \Delta_{\mathrm{fwd}}(p) \;=\; \hat{D}_t^{\rightarrow}(p) - z_t^{\mathrm{ref}}(p). \tag{12}$$

(ii) Backward, target → source. We reproject the *predicted* target depth into the source view. Here both the grid $\mathbf{g}_{t\to s}$ and the transformed depth z_s^{pred} depend on $\hat{D}_t$ (gradient flows through the projection), while the GT source depth is gathered as a detached reference,

$$D_s^{\leftarrow}(p) \;=\; \mathcal{S}\big(D_s;\, \mathbf{g}_{t\to s}(p)\big), \qquad \Delta_{\mathrm{bwd}}(p) \;=\; D_s^{\leftarrow}(p) - z_s^{\mathrm{pred}}(p). \tag{13}$$

Signed penalty. Both residuals are passed through the same one-sided, saturated $\log$ penalty as the evaluation metric. With a scale s that maps native depth units to metres, a dead-zone δ (optionally depth-proportional, $\delta = \delta_{\mathrm{rel}}\,|D_{\mathrm{ref}}|\,s$), and an upper clamp τ,

$$\ell(\Delta) \;=\; \min\Big(\log\big(1 + \max(s\,\Delta - \delta,\, 0)\big),\; \tau\Big). \tag{14}$$

Only the positive part beyond δ is penalized (the predicted surface in front of the reference); the scaling acts on the *difference* alone, so the geometry in Eqs. (10)–(11) stays in native units. Summing over the valid pixels $\mathcal{V}$ of both directions and normalizing by their count gives the per-sample loss

$$\mathcal{L}_{\mathrm{PCC}} \;=\; \frac{\displaystyle\sum_{p\in\mathcal{V}_{\mathrm{fwd}}} \ell\big(\Delta_{\mathrm{fwd}}(p)\big) + \sum_{p\in\mathcal{V}_{\mathrm{bwd}}} \ell\big(\Delta_{\mathrm{bwd}}(p)\big)}{|\mathcal{V}_{\mathrm{fwd}}| + |\mathcal{V}_{\mathrm{bwd}}|}. \tag{15}$$

An optional per-pixel confidence map $w(p) \in [0, 1]$, sampled at $\mathbf{g}_{s\to t}$ for the forward term and used natively for the backward term, multiplies each violation so that low-confidence regions contribute less.

Remark (surrogate gap). Because the inverse warp gathers depth by bilinear interpolation without a z-test, it is occlusion-naïve and bleeds across depth discontinuities. $\mathcal{L}_{\mathrm{PCC}}$ is therefore a differentiable *surrogate* of Eq. (9) and is not expected to match the metric numerically; we optimize $\mathcal{L}_{\mathrm{PCC}}$ during training but always report the hard z-buffer PCC at evaluation time.

C.10 Quantitative Results of the Jitter and Occlusion Experiments

As shown in Table 5, all four metrics degrade monotonically with increasing jitter. At $\pm 40\%$ jitter, PCC rises from 0.001 to 0.044, PSNR drops from ∞ to 25.5 dB, SSIM from 1.0 to 0.935, and LPIPS from 0 to 0.149. This confirms that when geometric distortion directly changes pixel content, all metrics respond in agreement.

	2D image metrics				3D / camera pose metrics			
	PSNR↑	SSIM↑	LPIPS↓	PCC_{GT}↓	PCC_{DA3}↓	AUC@5↑	RPE-R↓	RPE-T↓
Baseline	∞	1.000	0.000	0.002	0.074	0.743	12.79	0.77
Jitter ±1%	42.1	0.996	0.004	0.004	0.095	0.700	13.08	0.79
Jitter ±2%	39.1	0.990	0.008	0.006	0.123	0.686	12.93	0.93
Jitter ±5%	34.6	0.976	0.021	0.015	0.232	0.486	13.23	2.22
Jitter ±10%	31.1	0.962	0.043	0.029	0.346	0.329	13.87	4.12
Jitter ±20%	28.1	0.948	0.088	0.052	0.501	0.086	14.31	8.76
Jitter ±40%	25.5	0.935	0.149	0.083	0.615	0.057	14.70	12.35
Occlusion (avg.)	34.7	0.978	0.064	0.005	–	–	–	–

Table 5: Mean metrics across views for the scale-jitter and occlusion experiments. Under scale jitter, 2D quality (PSNR/SSIM/LPIPS) and the DA3-based geometry/pose metrics (PCC_{DA3}, AUC@5, RPE-T) all degrade monotonically as the perturbation grows; rotation error RPE-R stays in a narrow ∼13–15 band, confirming that scale ambiguity manifests primarily as translation drift. Under occlusion, where each scene variant is internally 3D-consistent but visually different from the baseline, the 2D metrics respond: **PCC_{GT} remains at the baseline level** ($0.005 \approx 0.002$), correctly recognising each variant as geometrically consistent.

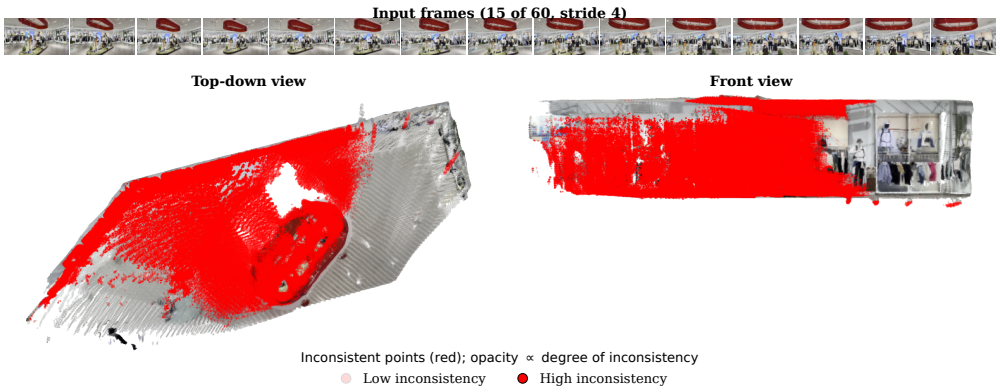

Figure 12: **Multi-view geometric inconsistency of a 60-frame accumulated point cloud (DL3DV). Top:** 15 of the 60 input frames (stride 4) used to build the cloud. **Bottom:** the accumulated point cloud, unprojected from the per-frame DA3 depth and DA3-estimated camera poses and shown from two viewpoints (top-down and front, in the reference-camera frame). Each frame's depth is scored by reprojecting the global cloud into that view (z-buffer) and measuring the relative depth violation $\text{rel} = (D - \hat{D})/\hat{D}$; points with $\text{rel} > \tau$ (τ=0.15) are marked red, with **opacity proportional to the degree of inconsistency** $\max(\text{rel} - \tau, 0)$ (faint = borderline, solid = strongly inconsistent). Geometrically consistent points keep their RGB color.

C.11 Multi-View PCC

Setup. Consider a scene with $V = N + M$ posed views, comprising N source views and M generated views. For each view v_i, where $i \in \{1, \ldots, V\}$, we are given an RGB image I_{v_i}, a depth map $D_{v_i} : \Omega \to \mathbb{R}_{\geq 0}$ (with image domain Ω), and a camera-to-world pose $C_{v_i} \in \text{SE}(3)$.

Global reprojection. For any given view v_i, we construct a global point cloud P_{global} aggregated from all the views v_j:

$$P_{\text{global}} = \bigcup_i P_{v_i}, \qquad \text{where} \quad P_{v_i} = \left\{ C_{v_i}\, \pi^{-1}(\mathbf{u},\, D_{v_i}(\mathbf{u})) \;\middle|\; \mathbf{u} \in \Omega \right\}, \tag{16}$$

and $\pi^{-1}(\mathbf{u}, d)$ unprojects pixel $\mathbf{u} \in \Omega$ at depth d into the camera coordinate frame. We then project P_{global} into camera v_i and rasterize it using a z-buffer (where the nearest point per pixel wins). This

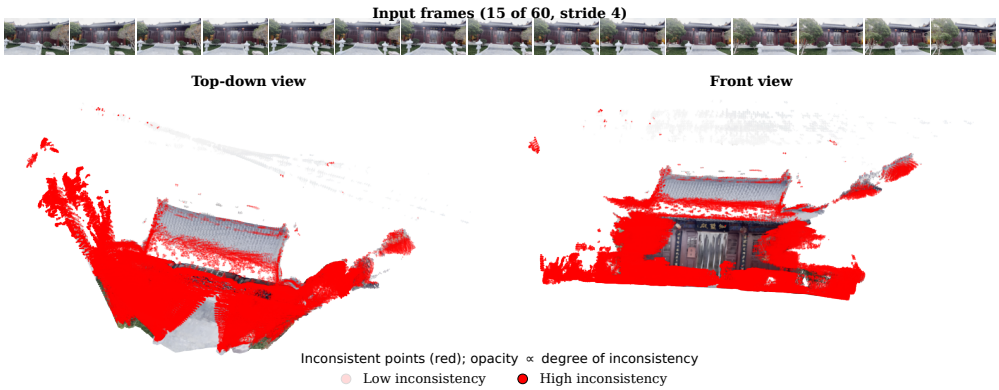

Figure 13: **Multi-view geometric inconsistency of a 60-frame accumulated point cloud (DL3DV).**

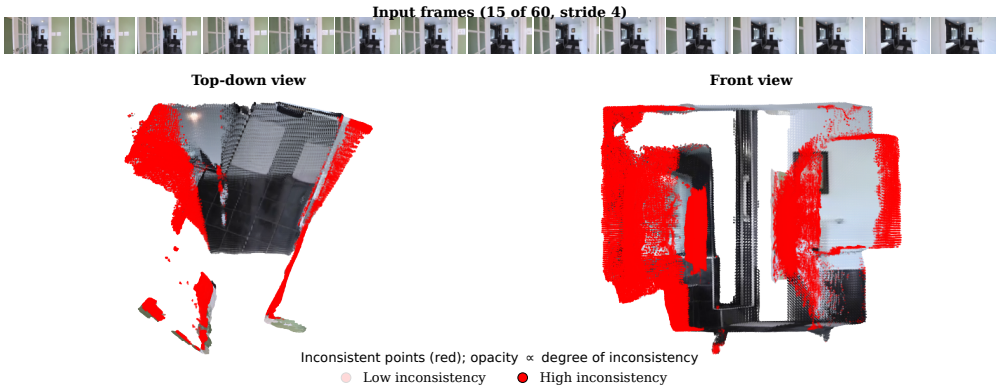

Figure 14: **Multi-view geometric inconsistency of a 60-frame accumulated point cloud (DL3DV).**

yields the *reprojected depth* of the scene from the perspective of v_i:

$$\widehat{D}_{v_i}(\mathbf{u}) \;=\; \min_{\substack{\mathbf{p}\in P_{\text{global}} \\ \pi(C_{v_i}^{-1}\mathbf{p})=\mathbf{u}}} \big[\, C_{v_i}^{-1}\mathbf{p} \,\big]_z. \tag{17}$$

Per-pixel consistency criterion. For opaque scenes, the only physically admissible relationship between the observed depth D_{v_i} and the reprojected depth $\widehat{D}_{v_i}$ is:

$$D_{v_i}(\mathbf{u}) \;\le\; \widehat{D}_{v_i}(\mathbf{u}) + \delta, \qquad \forall\, \mathbf{u} \in M_i, \tag{18}$$

where $\delta \ge 0$ is a tolerance for depth noise and rasterization quantization.

Why the test suffices. Although Eq. (18) is one-sided, applying it across *all* views systematically detects both primary error modes in novel view generation. Let v_i be a generated view containing a hallucinated surface $H \subset \mathbb{R}^3$ at pixel $\mathbf{u}_i \in \Omega$.

- **(a) Occlusion Violation** (*H lies behind the rest of the scene*). At view v_i, the reprojected depth $\widehat{D}_{v_i}(\mathbf{u}_i)$ records the closer, real surface seen by other views, whereas the predicted depth is $D_{v_i}(\mathbf{u}_i) = \text{depth}(H)$. Because H is behind the real surface, $D_{v_i}(\mathbf{u}_i) > \widehat{D}_{v_i}(\mathbf{u}_i)$, triggering a violation directly at view i.
- **(b) Free-Space Violation** (*H lies in front of the rest of the scene; a "floater"*). At view v_i, $D_{v_i}(\mathbf{u}_i) = \text{depth}(H) \le \widehat{D}_{v_i}(\mathbf{u}_i)$, so no violation is raised at i (from v_i's own perspective, the scene appears self-consistent). However, for any *other* view v_j whose ray passes

through H at pixel $\mathbf{u}_j$ first before any other points. The z-buffer pulls the reprojected depth $\widehat{D}_{v_j}(\mathbf{u}_j)$ forward to $\mathrm{depth}(H)$, while the real-scene observation $D_{v_j}(\mathbf{u}_j)$ corresponds to the true, farther surface. Thus, $D_{v_j}(\mathbf{u}_j) > \widehat{D}_{v_j}(\mathbf{u}_j)$, and the violation is successfully caught at view j.

Multiview PCC loss. We turn the one-sided criterion of Eq. (18) into a score by penalizing only its violations. For each view v_i define the per-pixel free-space residual

$$r_{v_i}(\mathbf{u}) \;=\; \max\big(0,\; D_{v_i}(\mathbf{u}) - \widehat{D}_{v_i}(\mathbf{u}) - \delta\big), \tag{19}$$

which is positive exactly when the observed depth lies *behind* the z-buffered scene depth by more than the tolerance δ. To bound the influence of gross outliers (e.g. depth-discontinuity fringes), we apply a saturating penalty

$$\rho(r) \;=\; \min\big(\log(1+r),\; \tau\big), \tag{20}$$

with cap τ. The multi-view PCC loss is the mean penalty over all covered pixels, aggregated across every view:

$$\mathcal{L}_{\mathrm{PCC}} \;=\; \frac{\displaystyle\sum_{i=1}^{V}\sum_{\mathbf{u}\in M_i} \rho\big(r_{v_i}(\mathbf{u})\big)}{\displaystyle\sum_{i=1}^{V}|M_i|}, \qquad M_i \;=\; \big\{\,\mathbf{u}\in\Omega \;\big|\; \widehat{D}_{v_i}(\mathbf{u}) < \infty\,\big\}, \tag{21}$$

where M_i is the set of pixels in view v_i that receive at least one reprojected point under the z-buffer. Lower $\mathcal{L}_{\mathrm{PCC}}$ indicates greater multi-view consistency.

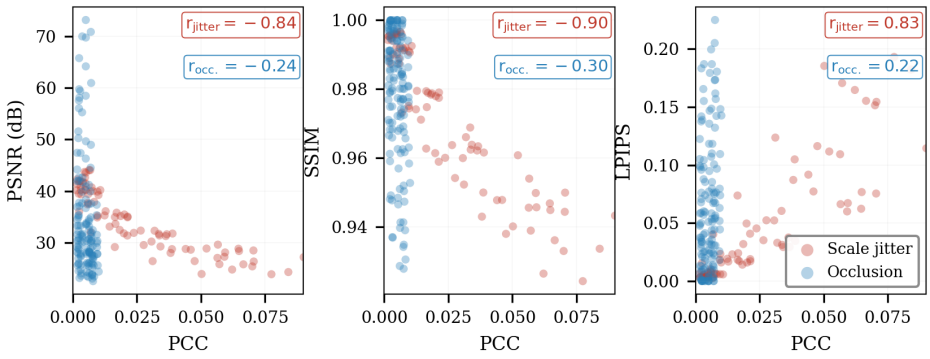

Figure 15: **Metric correlation analyses.** Per-view correlation between PCC and 2D metrics across both experiments. Each point represents a single (variant, view) pair. Red: scale jitter experiment—PCC correlates strongly with all 2D metrics (r=−0.84 for PSNR, −0.90 for SSIM, +0.83 for LPIPS), since geometric distortion simultaneously changes appearance. Blue: occlusion experiment—correlations weaken substantially (r=−0.24 for PSNR, −0.30 for SSIM, +0.22 for LPIPS), as views with large visual differences (high LPIPS, low PSNR) maintain low PCC because each scene is internally 3D-consistent. The strong-to-weak correlation shift between experiments demonstrates that PCC and 2D metrics measure fundamentally orthogonal properties.

C.12 PCC vs Standard Evaluation Metrics

A reliable 3D metric must correctly penalize geometric distortions while remaining invariant to purely visual changes in occluded regions. To prove PCC meets these criteria, we systematically evaluate its performance against established 2D perceptual and camera-pose metrics. Using controlled synthetic scenes, we demonstrate that PCC captures the true spatial consistency that standard evaluation protocols fundamentally miss. Unless stated otherwise, we perform the evaluations on a custom synthetic Blender simulation dataset featuring simple geometric primitives (Blender Online Community, 2026; Greff et al., 2022).

C.12.1 Evaluating PCC Against 2D Perception Metrics

We design two controlled sets of experiment using synthetic scenes rendered with 15 camera views distributed on a hemisphere around a scene of objects.

Experiment 1: Scale jitter. We render a baseline scene containing five geometric primitives and create six *jittered* variants by applying per-object random scale jitter of $\pm j\%$, with $j \in \{1, 2, 5, 10, 20, 40\}$. Each object is independently scaled by a factor drawn uniformly from $[1-j/100,\ 1+j/100]$, so some objects enlarge while others shrink within the same scene. Since objects change size, both the rendered appearance and the underlying 3D geometry are altered relative to the baseline. We compute two types of metrics comparing the jittered and baseline scenes: (1) 2D perceptual metrics (PSNR, SSIM, LPIPS). Note that for the unmodified baseline scene, these metrics are trivially perfect (PSNR $= \infty$, SSIM $= 1$, LPIPS $= 0$). (2) Our PCC metric, computed by reprojecting the baseline anchor view's depth (view 0) into each jittered variant's novel view i: $\text{PCC}(\mathbf{D}_0^{\text{base}}, \mathbf{D}_i^{\text{jitter}_j}, \mathbf{T}_{0\to i})$, where $\mathbf{T}_{0\to i} \in \text{SE}(3)$ denotes the relative camera transformation matrix mapping from the coordinate frame of view 0 to view i. This measures whether the baseline geometry and the jittered variant geometry agree when viewed from the same camera.

Experiment 2: Occlusion-invariant. We design a scenario to expose the limitations of 2D perceptual metrics in evaluating 3D consistency. From a baseline scene A, we create 10 modified variants (scene B_j, $j = 1, \ldots, 10$) by introducing or removing objects within regions unobserved by the anchor camera. By design, the rendered images from the anchor viewpoint remain nearly indistinguishable across all variants (achieving a PSNR > 50 dB compared to scene A). [3] The 2D perceptual

[3] The slight variance in PSNR results solely from subtle changes in global lighting reflections (see leftmost column in Figure 8).

metrics compare view i of scene A directly against view i of scene Bj. Because the actual scene content differs, these metrics should heavily penalize the novel viewpoints where the new objects are revealed. Conversely, PCC evaluates geometric consistency by bidirectionally reprojecting depth between the anchor view and the novel views. Because the added or removed objects in scene Bj are strictly occluded from the anchor perspective, they introduce no depth conflicts or geometric contradictions between the two views. As a result, PCC should correctly recognize that the multi-view geometry remains perfectly aligned, yielding values nearly identical to the baseline.

Analysis (Figure 15). The per-view scatter plots of PCC against each 2D perceptual metric reveal the critical distinction between the two experiments. Under scale jitter (red points), PCC and the 2D perceptual metrics are strongly correlated: as geometric distortion increases, both PCC and LPIPS rise together ($r = +0.83$), while PCC and PSNR exhibit negative correlation since PSNR decreases with distortion ($r = -0.84$), and similarly for SSIM ($r = -0.90$). This is expected as scale jitter simultaneously changes geometry and appearance, making all metrics respond in agreement. For additional details, please refer to Table 5 in Appendix C.1. Under occlusion (blue points), the correlation structure fundamentally changes. The blue points cluster at low PCC values (< 0.03) but spread widely along the 2D metric axes, reflecting the large visual differences caused by added objects in views that see them. The Pearson correlations weaken or invert: PCC vs. PSNR becomes $r = +0.64$ (positive, opposite to jitter), PCC vs. SSIM becomes $r = +0.48$, and PCC vs. LPIPS becomes $r = -0.45$ (negative, opposite to jitter). Views with high LPIPS (large visual change due to added objects) show *low* PCC (the scene is geometrically consistent). This decorrelation demonstrates that PCC and 2D metrics measure fundamentally orthogonal properties: 2D metrics capture *whether images look different*, while PCC captures *whether the underlying 3D geometry is consistent across views*. For details, please refer to Table 5 in Appendix C.2. These experiments establish that PCC captures multi-view 3D consistency—a property that 2D metrics cannot measure. A novel-view synthesis model that hallucinates plausible but geometrically incoherent content across views may achieve high PSNR and SSIM per-view while exhibiting elevated PCC. Conversely, a model that produces a 3D-consistent scene differing from the ground truth in object composition will show degraded 2D metrics but low PCC. We therefore advocate reporting PCC alongside standard 2D perceptual metrics to provide a more complete evaluation of novel-view synthesis quality.

C.12.2 Evaluating PCC Against Camera Pose Metrics

Evaluating the 3D consistency of generated views using PCC relies heavily on accurate depth estimation, which can be natively predicted by the generation model or extracted via an external 3D reconstruction tool. In practice, these estimations are rarely flawless. Prone to scale drift, depth boundary errors, and pose inaccuracies, such underlying artifacts inevitably compromise the reliability of the PCC measurement. Given that camera-pose accuracy metrics similarly depend on the geometric estimations of SfM or 3D reconstruction models, this section evaluates how depth inaccuracies impact PCC and compares its robustness against standard camera-pose metrics.

To extract scene geometry, we process the source image I_{s} and the generated target image $\hat{I}_{\mathrm{t}}$ through a multi-view geometry estimator $\mathcal{E}$, specifically utilizing Depth-Anything3-LARGE-1.1 (Lin et al., 2025). This estimator jointly predicts the metric depth and camera parameters for both views:

$$\hat{\mathbf{D}}_s,\ \hat{\mathbf{D}}_t,\ \hat{\mathbf{T}}_s,\ \hat{\mathbf{T}}_t,\ \hat{\mathbf{K}}_s,\ \hat{\mathbf{K}}_t\ =\ \mathcal{E}\big(\mathbf{I}_s,\ \hat{\mathbf{I}}_t\big),$$

where $\hat{\mathbf{D}} \in \mathbb{R}^{H\times W}$ are the scale-aligned depth maps (details in Appendix C.3), $\hat{\mathbf{T}} \in \mathrm{SE}(3)$ are the camera-to-world extrinsics, and $\hat{\mathbf{K}} \in \mathbb{R}^{3\times 3}$ represent the intrinsic matrices. We use the estimated depth maps for computing PCC and the estimated camera poses for computing relative pose errors (RPEs; see Appendix C.5).

We follow the experimental protocol Section C.12.1 and evaluate the PCC scores against camera-pose accuracies for the scale jitter variants. While the experiments in Section C.12.1 utilize ground-truth depth generated by Blender, this section adopts a more practical setup using the DA3-estimated geometry of the synthetic scenes to compare PCC against standard camera-pose metrics. As shown in Figure 16, PCC, pose-AUC, and relative translation error (RPE-T) all worsen monotonically with increasing jitter. On the other hand, RPE-R remains essentially flat – an expected outcome since the actual camera poses remained static while only the object scales changed. Importantly, RPE-T alone is insufficient to capture overall camera-pose accuracy; a complete assessment requires evaluating both translation and rotation. Consequently, the RPE metric suite as a whole fails to serve as a

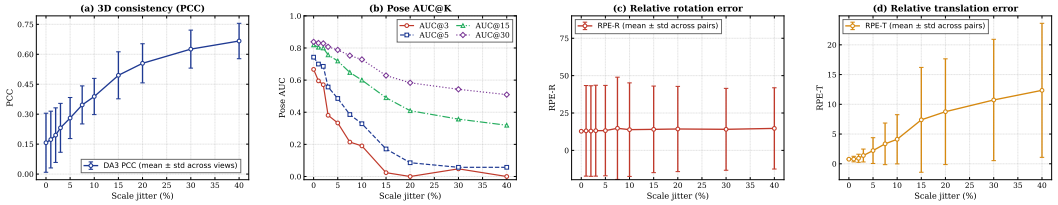

Figure 16: **Sensitivity of the DA3 PCC metric to controlled scale perturbations.** Starting from a clean reference scene, we re-render each object with a random per-object scale jitter of 1–40% and measure (a) PCC, (b) pose AUC@3, 5, 15, 30, (c) relative rotation error (RPE-R), and (d) relative translation error (RPE-T), with depth and cameras predicted by DA3 and FG/BG-split affine alignment (no top-% pruning). With one exception, all evaluated metrics track geometric error monotonically alongside increasing jitter magnitude. Specifically, PCC rises from 0.07 at 0% jitter to 0.62 at 40%, AUC@30 falls from 0.84 to 0.51 (with AUC@3 collapsing from 0.67 to 0 by the 20% mark), and RPE-T increases from 0.77 to 12.35. RPE-R is the only metric that fails to reliably reflect the introduced noise, remaining largely stagnant and breaking monotonicity at 2%, 7%, and 30%.

reliable 3D consistency metric under the jitter test. Furthermore, although pose-AUC appears to pass the test by degrading monotonically, it achieves this by aggregating pairwise RPEs into a single score, which masks the underlying failure of the rotation estimates. However, in general practice, any flaw in either RPE-R or RPE-T might affect the overall reliability of the AUC score since the metric relies on both components.

C.12.3 Evaluation on a Real-World Scene

We conduct an experiment in a real-world scene to evaluate whether DA3 depth estimates can be used to detect 3D inconsistencies under a counterfactual setting. We define a *Real Pair*, composed of a source and a target view, depicting a 3D-consistent scene in which all objects remain stationary while the camera moves between two viewpoints. We also define an *Edited Pair* sharing the same source image as the Real Pair and an edited target image, retaining the same camera viewpoint as the one of the Real Pair target. The edited target displaces objects in the scene, creating a counterfactual scene whose object configuration is inconsistent with the 3D structure observed in the source image. An example is shown in Figure 17. As shown in the figure, applying our PCC evaluation using these DA3 depth estimates demonstrates that the metric can effectively isolate regions of structural inconsistency. Specifically, the forward projection from the source to the target view detects the absence of objects at their original spatial coordinates—such as the initial position of the stool, as well as the chair which was moved almost entirely out of the frame—flagging these vacated regions with projection errors (visualized as blue pixels). Conversely, the backward projection from the target back to the source identifies the conflicting geometries of the displaced objects at their newly introduced locations, such as the moved stool (visualized as red pixels). This bidirectional validation confirms that, even without ground-truth geometry, DA3-guided PCC can reliably pinpoint pixel-level hallucinations and physical displacements, effectively distinguishing geometrically coherent view synthesis from counterfactual 2D manipulations. It is worth noting, however, that off-the-shelf DA3 depth estimates are still subject to inherent noise. As observed in the Real Pair, minor misdetections emerge as sparse error pixels even when the underlying scene exhibits perfect multi-view consistency.

D Sensitivity Analyses

D.1 How Do Frame Gap and Camera Displacement Affect Performance?

Because the DL3DV dataset consists of continuous videos of static scenes, evaluating novel view synthesis requires sampling discrete source and target frame pairs from a sequence (see Appendix A.4 for exact sampling configurations). Assuming a relatively uniform camera velocity, the frame gap between the selected views serves as a proxy for the magnitude of the camera pose baseline. A larger frame gap translates to a more pronounced geometric shift, significantly increas-

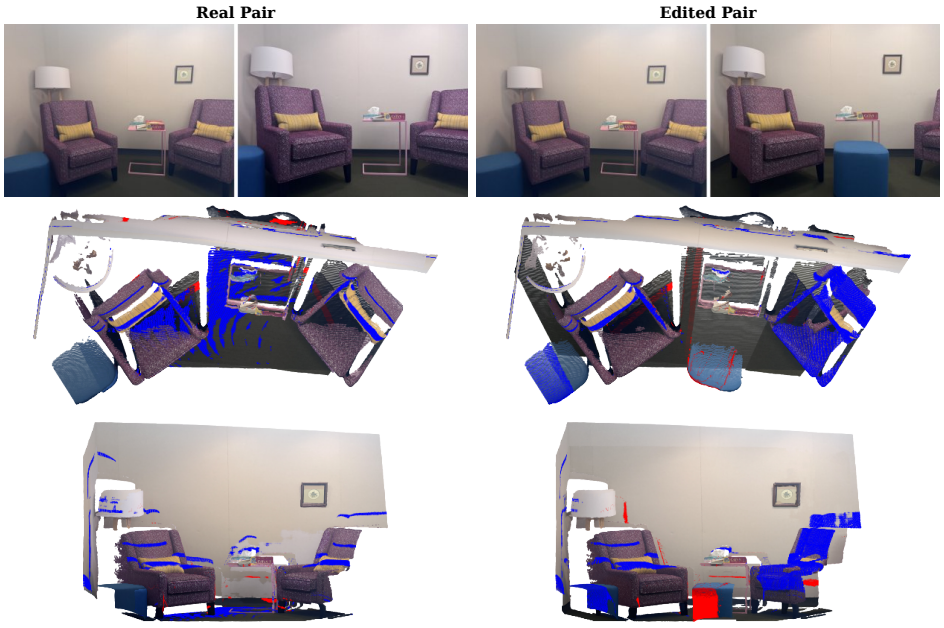

Figure 17: **Geometrically inconsistent point detections in a counterfactual scene.** The *real pair* depicts a 3D-consistent scene in which all objects remain stationary as the camera moves between the two viewpoints. The *edited pair* shares the same source image (same left image), and its target image (right image) is rendered from the same camera viewpoint as in the consistent pair. However, one or more objects are displaced in the second's target, producing a counterfactual scene that is incompatible with the 3D structure observed in source. By projecting the point cloud derived from each image into the other view, the PCC-based procedure identifies points whose projected geometry is inconsistent with the modified scene. These inconsistencies are highlighted as red and blue pixels, corresponding to inconsistencies detected by the forward and backward projections, respectively.

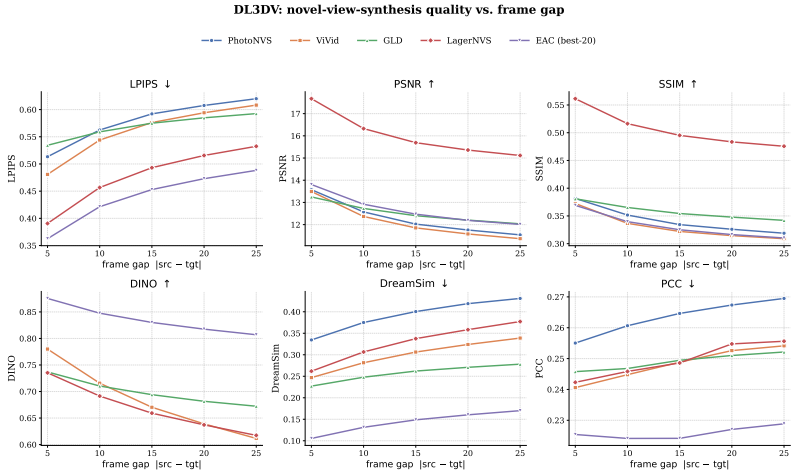

Figure 18: **Effect of frame gap on model performance.** Performance consistently degrades across all models as the frame gap increases. Nevertheless, our method outperforms all baselines at every frame-gap setting on all metrics except PSNR and SSIM, for which LagerNVS achieves the best results.

ing the difficulty of the synthesis task. For this ablation, we evaluate model resilience to baseline changes by varying the frame gap from 5 to 25 frames using a stride of 5.

Model	Perceptual Quality			Geometry
	LPIPS ↓	DINO-Sim ↑	DreamSim ↓	PCC ↓
DL3DV Dataset				
ESC-Shadow	0.4952 ± 0.1386	0.8035 ± 0.1540	0.1676 ± 0.1304	0.2533 ± 0.1640
ESC-Base	0.4448 ± 0.1569	0.8366 ± 0.1389	0.1420 ± 0.1252	0.2448 ± 0.1646
ESC-PCC-BoN	$\mathbf{0.4340 \pm 0.1556}$	$\mathbf{0.8386 \pm 0.1388}$	$\mathbf{0.1402 \pm 0.1232}$	$\mathbf{0.2257 \pm 0.1556}$
RE10K Dataset				
ESC-Shadow	0.3166 ± 0.1328	0.8982 ± 0.0766	0.0780 ± 0.0568	0.0870 ± 0.1433
ESC-Base	0.2613 ± 0.1249	0.9157 ± 0.0652	0.0669 ± 0.0497	0.0773 ± 0.1430
ESC-PCC-BoN	$\mathbf{0.2150 \pm 0.1220}$	$\mathbf{0.9222 \pm 0.0667}$	$\mathbf{0.0655 \pm 0.0510}$	$\mathbf{0.0656 \pm 0.1359}$
Tanks & Temples Dataset				
ESC-Shadow	0.6336 ± 0.0859	0.6549 ± 0.2078	0.2928 ± 0.1512	0.4291 ± 0.4094
ESC-Base	0.5835 ± 0.1016	0.7242 ± 0.1872	0.2362 ± 0.1371	0.3826 ± 0.3875
ESC-PCC-BoN	$\mathbf{0.5596 \pm 0.1164}$	$\mathbf{0.7339 \pm 0.1854}$	$\mathbf{0.2232 \pm 0.1343}$	$\mathbf{0.2849 \pm 0.3139}$

Table 6: **Ablation study of the ESC framework.** Quantitative evaluation comparing ESC-shadow, ESC-base, and the full ESC-PCC-BoN pipeline across the DL3DV, RE10K, and Tanks & Temples datasets.

As plotted in Figure 18, we evaluate the impact of this frame gap across several core metrics: PSNR, SSIM, LPIPS, DINO-Sim, DreamSim and PCC. We observe a universal degradation in performance across all models as the frame gap widens. Notably, this performance drop is non-linear; the decline is steepest at narrower gaps (e.g., moving from 5 to 10 frames) and gradually smooths out as the baseline expands further. This trajectory and rate of degradation remain remarkably consistent across all evaluated architectures and metrics, indicating a uniform response to increasing geometric difficulty. Our model outperforms all baselines across every frame gap setting on all metrics except PSNR and SSIM, where LagerNVS scores highest. Rather than reflecting superior geometric synthesis, this discrepancy stems from two key factors: (1) *Generative Blurring:* As shown qualitatively in Figure 4, Figure 20, and Figure 21, LagerNVS systematically produces smoother, blurred outputs. Because PSNR and SSIM heavily penalize high-frequency artifacts, they favor smooth regressions over sharp but slightly misaligned generations. (2) *Rigid Pixel Comparisons:* In novel view synthesis, models must hallucinate occluded regions, which have infinite valid configurations. Because PSNR and SSIM require exact pixel-to-pixel alignment, they severely penalize high-quality, plausible generations if they deviate from the specific ground-truth pixels.

D.2 Escaping the 2D Cave: Does Depth Conditioning Truly Benefit Novel View Synthesis?

We first evaluate the necessity of depth tracking by contrasting ESC-shadow against our depth-aware variants. ESC-shadow retains the exact same model architecture and adapter footprint (sharing identical LoRA and ControlNet structures) as the other variants, but lacks the input and output channels for depth tokens.[4] As demonstrated by the quantitative ablation results in Table 6, incorporating explicit depth estimation yields substantial performance gains. Specifically, our depth-aware models (ESC-base and ESC-PCC-BoN) consistently outperform ESC-shadow across both perceptual metrics (LPIPS, DINO-Sim, DreamSim) and geometric consistency (PCC).

On the other hand, in our Best-of-N (ESC-PCC-BoN) inference strategy, the PCC metric serves as a geometry-guided scoring function to select the most spatially consistent sample from a candidate pool of generated views. Computing this PCC score inherently requires depth maps for both the anchor and the generated target views. Rather than relying on an external depth estimator to process the generated RGB images—which would introduce significant computational overhead during inference—we directly utilize the depth maps natively predicted by our ESC framework alongside the RGB generations.

[4] We denote the depth-free baseline as **ESC-shadow**, as it is forced to infer geometry solely from 2D pixel projections (the "shadows cast on the wall").

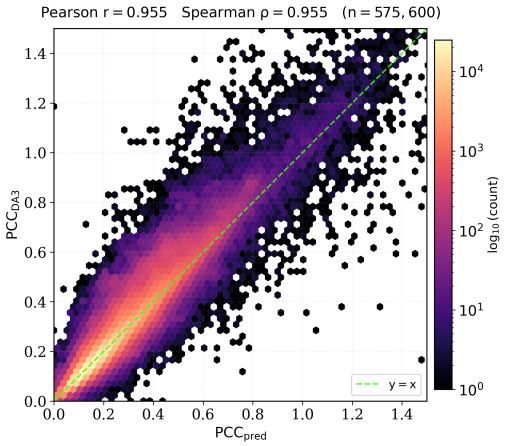

Figure 19: **Correlation between DA3-derived and predicted PCC metrics.**

To validate the reliability of our internally predicted depth for driving this 3D consistency evaluation, we analyze its alignment with established, state-of-the-art geometry estimators. As illustrated in Figure 19, we compare the PCC scores calculated using our model's predicted depth against those computed using depth estimates extracted via Depth-Anything 3 (DA3). We observe a remarkably strong positive correlation (Pearson $r = 0.955$, Spearman $\rho = 0.955$) between the two metrics across the evaluated samples. This strong alignment confirms that our model's native depth predictions accurately capture the underlying multi-view geometry, proving them to be highly effective and sufficient for the Best-of-N selection process without the need for auxiliary depth estimation tools.

E Baseline Models

We compare our ESC model against a representative set of recent novel-view-synthesis (NVS) baselines spanning the main generative paradigms. **OVIE** (Rahary et al., 2026) and **LagerNVS** (Szymanowicz et al., 2026) are feed-forward regressors that predict a target view in a single forward pass (OVIE additionally performs a 16-scale search), prioritizing speed over photorealism. **VIVID** (Elata et al., 2024) is a latent diffusion model with classifier-free guidance, while **PhotoNVS** (Yu et al., 2023) is a pixel-space DDPM requiring many denoising steps. **GLD** (Jang et al., 2026) adopts a multi-stage cascade diffusion design, and **Gen3C** (Ren et al., 2025) casts NVS as conditional video diffusion at high resolution. For each baseline we use the authors' released checkpoints and recommended inference settings. We evaluate two variants of our own model: **ESC-Blender**, specialized on the Blender scenes, and **ESC-Mixed**, trained on the mixture of RealEstate10K and DL3DV training sets and evaluated on Tanks-and-Temples, RealEstate10K, and DL3DV.

E.1 Inference Efficiency

Table 7 compares the inference paradigm, total number of function evaluations (NFE), native resolution, and per-view latency across all evaluated methods. Feed-forward approaches (OVIE, LagerNVS) offer the lowest latency, generating novel views in well under one second. Conversely, diffusion-based methods prioritize fidelity, with computational cost largely dictated by the number of function evaluations; consequently, PhotoNVS (2000 steps) and Gen3C (50 steps at 704×1280) are one to two orders of magnitude slower. Our ESC models strike a favorable balance, synthesizing a 384×384 view in approximately three seconds using only 20 NFE. This approach achieves latency comparable to VIVID while requiring fewer function evaluations and remains significantly more efficient than cascade and video-diffusion baselines. Notably, this inference latency remains consistent for both our Blender-specialized (LoRA rank 64) and mixed-data (LoRA rank 128) variants.

Model	Paradigm	NFE	Native Res.	Per-view Time
OVIE	feed-forward + 16-scale search	$\sim$1 ($\times$16)	256	0.1–0.3 s
LagerNVS	feed-forward	$\sim$1	512	0.2–0.5 s
ESC-Blender (rank 64)	diffusion	20	384	$\sim$3 s
ESC-Mixed (rank 128)	diffusion	20	384	$\sim$3 s
VIVID	diffusion + CFG (g=1.5)	32 ($\sim$64)	384	3–4 s
GLD	cascade diffusion (L1/L2/L3)	multi-stage	—	5–15 s
PhotoNVS	DDPM	2000	384	60–120 s

Table 7: Comparison of inference paradigms, total number of function evaluations (NFE), native resolution, and per-view latency. Per-view times are normalized to an H200 GPU and represent startup-corrected estimates. For our ESC models, we observe similar inference latency across the evaluated LoRA ranks (rank 64 for Blender; rank 128 for Mixed).

F Additional Results

F.1 Compute Time

Using the standard estimate $C \approx 6ND$ with $N \approx 2.2 \times 10^{10}$ active parameters (the frozen Qwen-Image-Edit backbone plus the trainable LoRA and 5-layer ControlNet adapter) and $D \approx 5.2 \times 10^{10}$ tokens (177k steps $\times$ a global batch of 128×2304 latent tokens per 384×384 sample), our full training run costs approximately 6.8×10^{21} FLOPs.

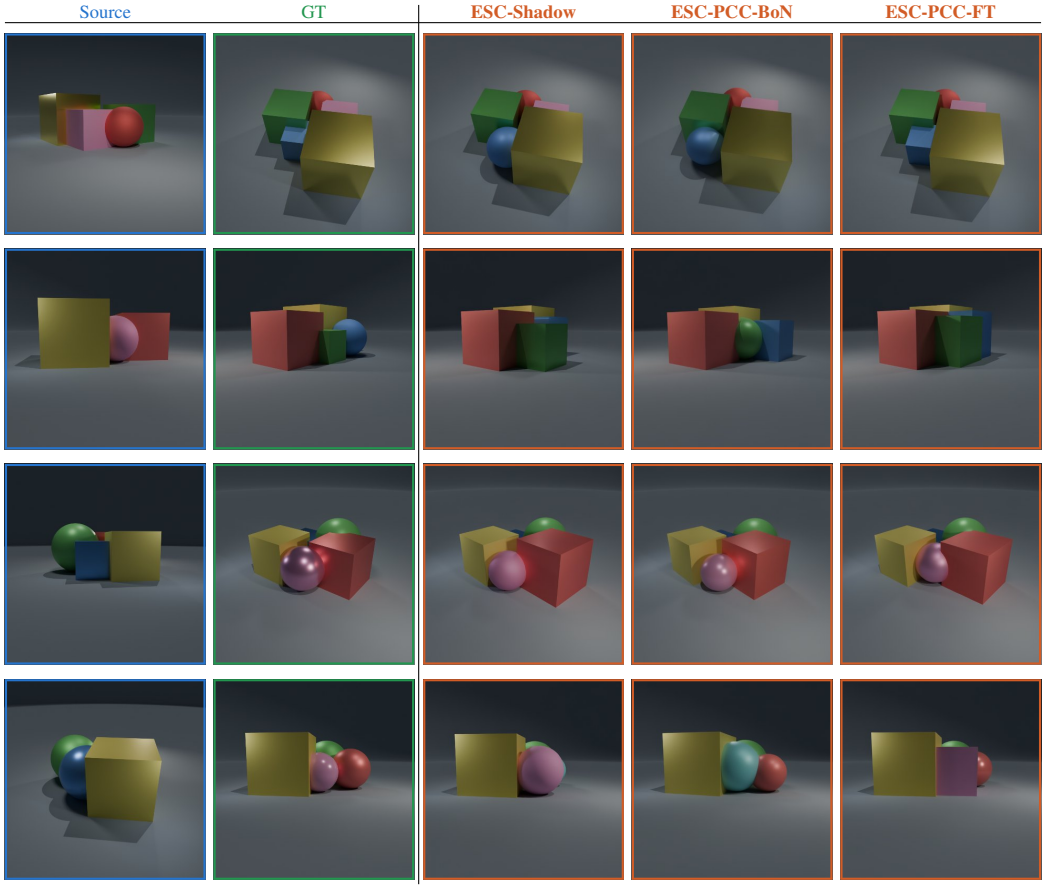

Table 8: **Qualitative comparison of ESC variants on the synthetic Blender benchmark.** Each row shows the source view (blue border), the ground-truth target (green border), and three ESC variants (orange border): ESC-Shadow, ESC-PCC-BoN, and ESC-PCC-FT. PCC-aware selection and fine-tuning yield novel views that most faithfully reproduce the scene geometry.

	Synthesis Quality		Perceptual Quality			Geometry	Pose & View Prompt Consistency			
Model	PSNR ↑	SSIM ↑	LPIPS ↓	DINO-Sim ↑	DreamSim ↓	PCC ↓	AUC@15 ↑	AUC@30 ↑	RPE-R ↓	RPE-T ↓
DL3DV Dataset										
PhotoNVS	12.38	0.3465	0.5743	0.5647	0.3876	0.2628	0.0148	0.0558	70.79	47.18
VIVID	10.88	0.3001	0.6407	0.6909	0.2953	0.2475	0.0551	0.1949	22.61	37.21
OVIE	11.58	0.3365	0.7099	0.2755	0.6480	0.3122	0.0084	0.0403	54.14	49.18
GLD	12.58	0.3598	0.5665	0.7019	0.2549	0.2487	0.1134	0.2894	16.03	31.58
LagerNVS	**16.14**	**0.5102**	0.4714	0.6734	0.3231	0.2487	**0.1231**	**0.2932**	16.74	**30.70**
ESC-shadow	12.16	0.3082	0.4952	0.8035	0.1676	0.2533	0.0492	0.2012	19.62	32.91
ESC (*Ours*)	12.69	0.3343	0.4448	0.8366	0.1420	0.2448	0.0886	0.2575	12.97	33.28
ESC-PCC-BoN (*Ours*)	12.76	0.3347	**0.4340**	**0.8386**	**0.1402**	**0.2257**	0.0922	0.2643	**12.91**	32.52
Oracle (*GT-vs-DA3*)	·	·	·	·	·	0.1199	0.4090	0.6631	5.14	9.87
RE10K Dataset										
PhotoNVS	17.51	0.5886	0.3660	0.7669	0.2176	0.0899	0.4734	0.6234	3.96	16.4
VIVID	16.46	0.5514	0.3741	0.8841	0.1088	0.0879	0.3997	0.6119	1.51	14.61
OVIE	13.80	0.4638	0.5009	0.7168	0.7881	0.1344	0.0243	0.0923	12.93	47.12
GLD	15.77	0.5527	0.4413	0.8128	0.1394	0.0898	**0.7273**	**0.8422**	0.95	**5.86**
LagerNVS	**18.46**	**0.6350**	0.3239	0.8838	0.1256	0.0783	0.6800	0.7980	**0.56**	8.02
ESC-shadow	15.74	0.5180	0.3166	0.8982	0.0780	0.0870	0.6142	0.7679	1.61	8.48
ESC (*Ours*)	16.61	0.5464	0.2613	0.9157	0.0669	0.0773	0.5661	0.7346	1.09	9.97
ESC-PCC-BoN (*Ours*)	17.73	0.5912	**0.2150**	**0.9222**	**0.0655**	**0.0656**	0.5787	0.7429	0.98	9.68
Oracle (*GT-vs-DA3*)	·	·	·	·	·	0.0539	0.8098	0.8862	0.45	4.62
Tanks & Temples Dataset										
PhotoNVS	10.13	0.2692	0.7688	0.2345	0.5902	0.4467	0.0275	0.0796	49.70	43.41
VIVID	10.38	0.2859	0.7527	0.3793	0.5040	0.4416	0.0833	0.2310	33.89	31.86
OVIE	10.58	0.3093	0.8219	0.0640	0.7881	0.5102	0.0052	0.0229	53.01	53.35
GLD	10.88	0.2891	0.6365	0.5099	0.4027	0.4184	**0.4989**	**0.6208**	15.92	**15.33**
LagerNVS	**13.48**	**0.4669**	0.6420	0.2738	0.6630	0.4901	0.1937	0.2736	35.25	33.84
ESC-shadow	10.78	0.2692	0.6336	0.6549	0.2928	0.4291	0.1694	0.3837	22.00	20.08
ESC (*Ours*)	11.27	0.2882	0.5835	0.7242	0.2362	0.3826	0.3287	0.5501	**14.62**	15.64
ESC-PCC-BoN (*Ours*)	11.42	0.2892	**0.5596**	**0.7339**	**0.2232**	**0.2849**	0.3329	0.5482	14.74	16.28
Oracle (*GT-vs-DA3*)	·	·	·	·	·	0.1899	0.7700	0.8357	7.55	6.33

Table 9: **Full Comprehensive Evaluation across View Synthesis Benchmarks.** We present the complete evaluation metrics across DL3DV, RE10K, and Tanks & Temples datasets. Arrow (↓ / ↑) indicates the direction of better performance. The top-three entries within each dataset are highlighted using distinct progressive pastel red backgrounds, with the 1st place values also emphasized in **bold** text. "Oracle" indicates the ground-truth label compared against the paired DA3 estimate on the source images.

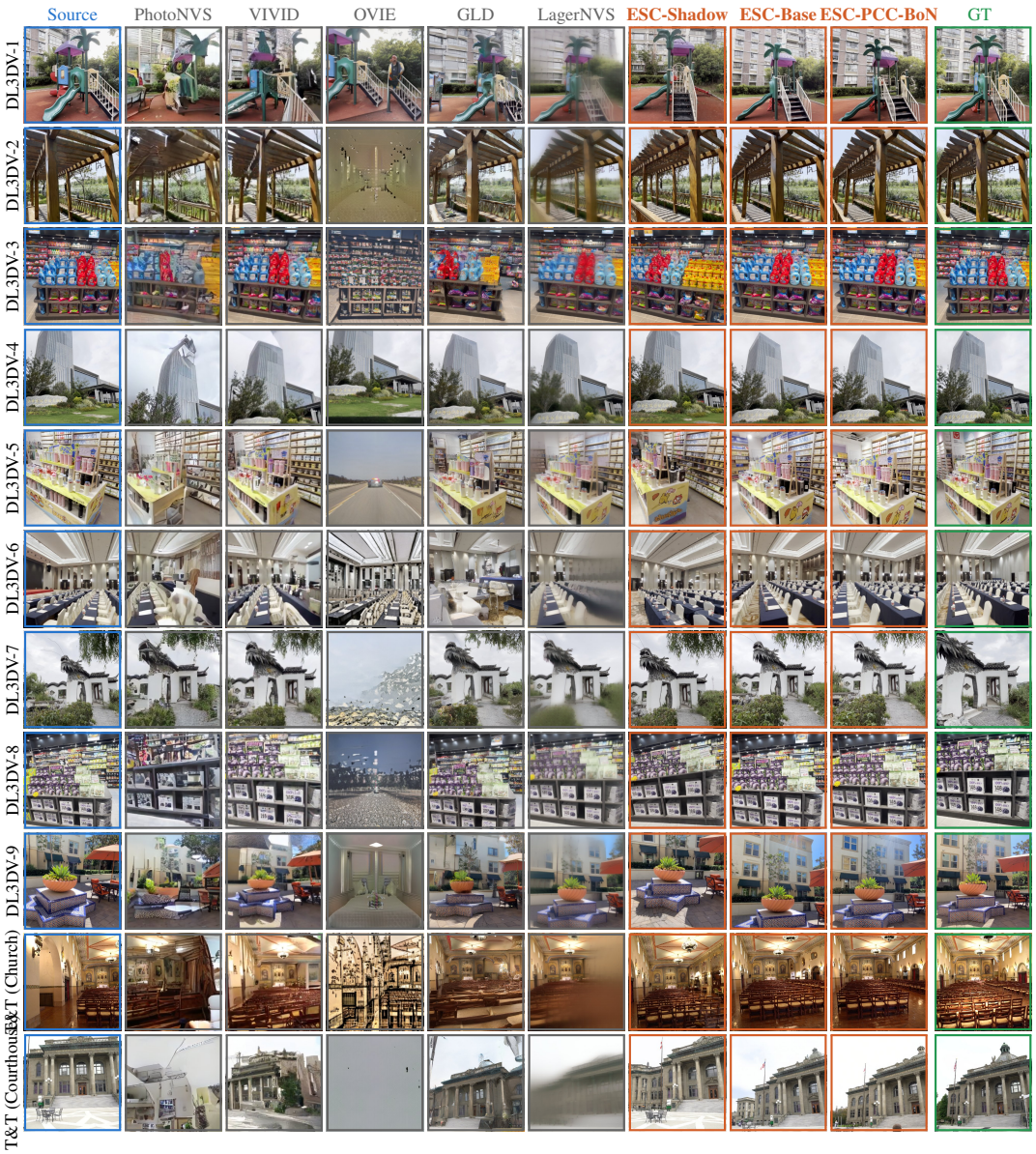

Figure 20: **Extended qualitative comparison on novel view synthesis (DL3DV and Tanks & Temples).** For each scene we show the input source view (blue border), the renderings produced by the baselines (gray border: PhotoNVS, VIVID, OVIE, GLD, LagerNVS), our ESC variants (orange border: ESC-Shadow, ESC-Base, and the geometry-guided ESC-PCC-BoN (Best-of-N)), and the ground-truth target view (green border). Rows 1–9 are scenes from DL3DV, and the last two rows show the *Church* and *Courthouse* scenes from Tanks & Temples. Across datasets, our variants reproduce scene geometry and fine detail more faithfully than the baselines, with ESC-PCC-BoN consistently best matching the ground-truth target.

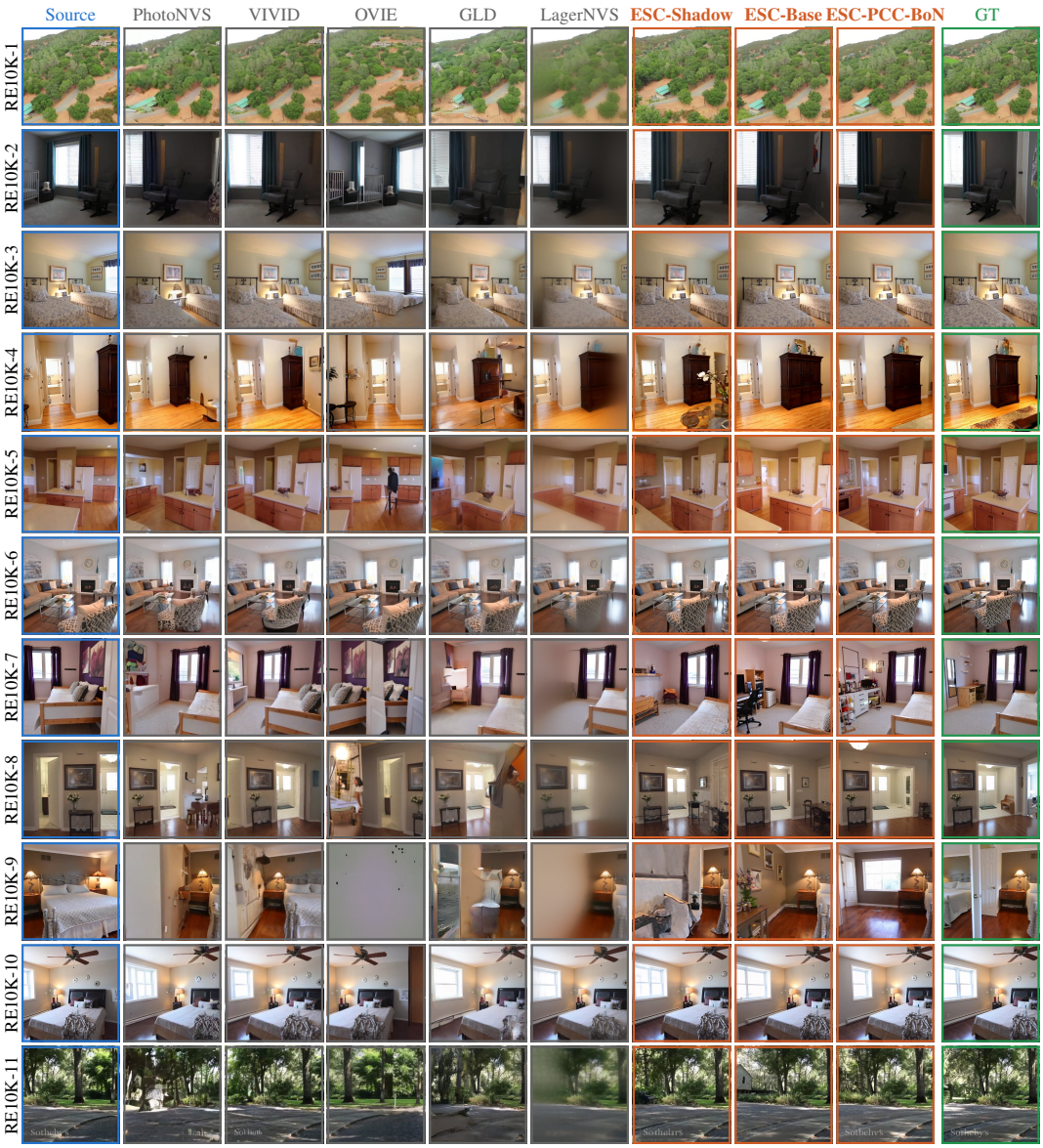

Figure 21: **Extended qualitative comparison on novel view synthesis (RE10K).** For each scene we show the input source view (blue border), the renderings produced by the baselines (gray border: PhotoNVS, VIVID, OVIE, GLD, LagerNVS), our ESC variants (orange border: ESC-Shadow, ESC-Base, and the geometry-guided ESC-PCC-BoN (Best-of-N)), and the ground-truth target view (green border). All rows are scenes from the RealEstate10K (RE10K) benchmark. Our variants reproduce indoor scene geometry and fine detail more faithfully than the baselines, with ESC-PCC-BoN consistently best matching the ground-truth target.

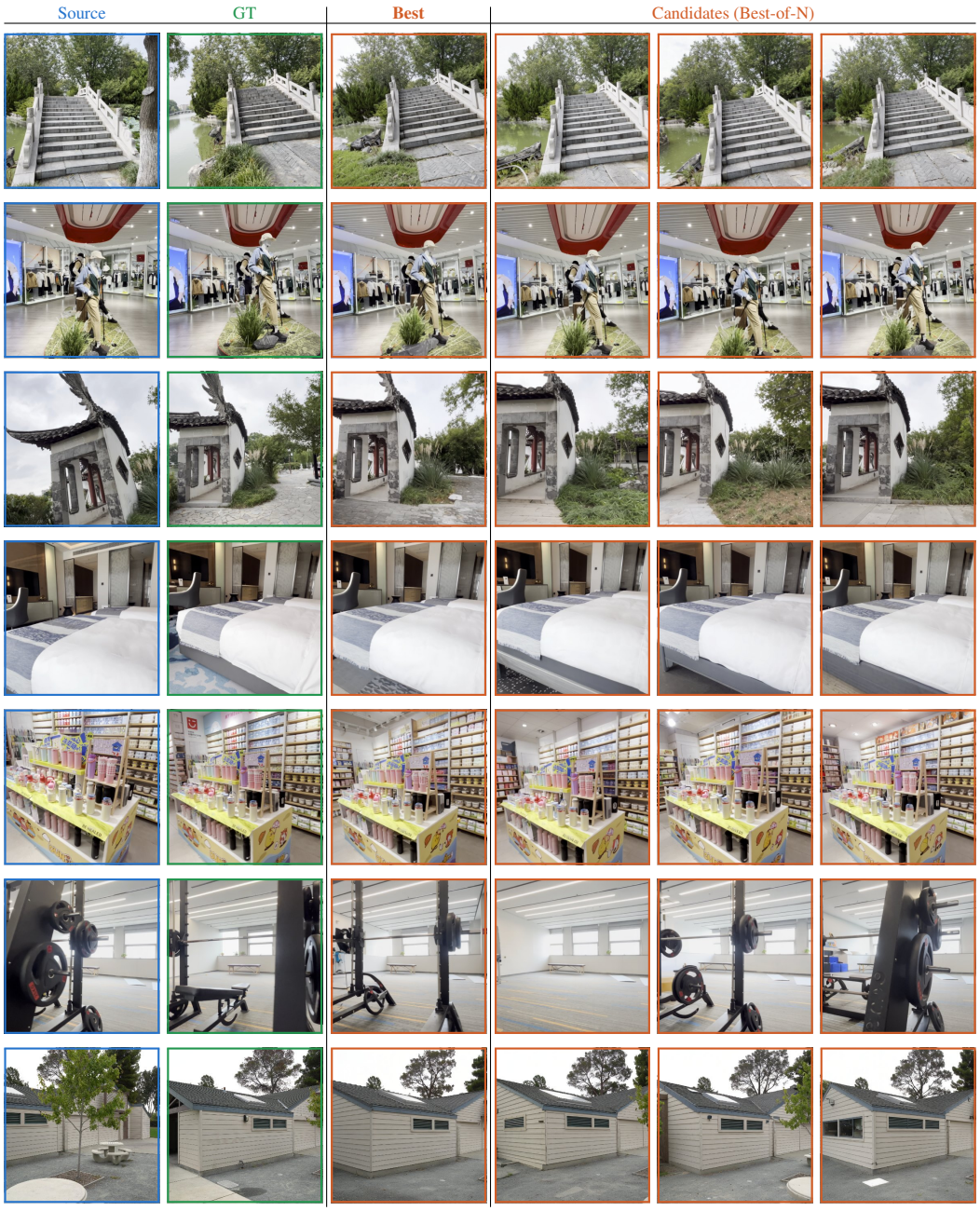

Figure 22: **Best-of-N novel-view synthesis on DL3DV.** Each row shows the source view (blue border), ground truth (green border), the PCC-selected best sample (orange border), and three additional candidates (orange border) from the same Best-of-N pool. The first six rows are DL3DV scenes; the last row is the *Barn* scene from Tanks & Temples.

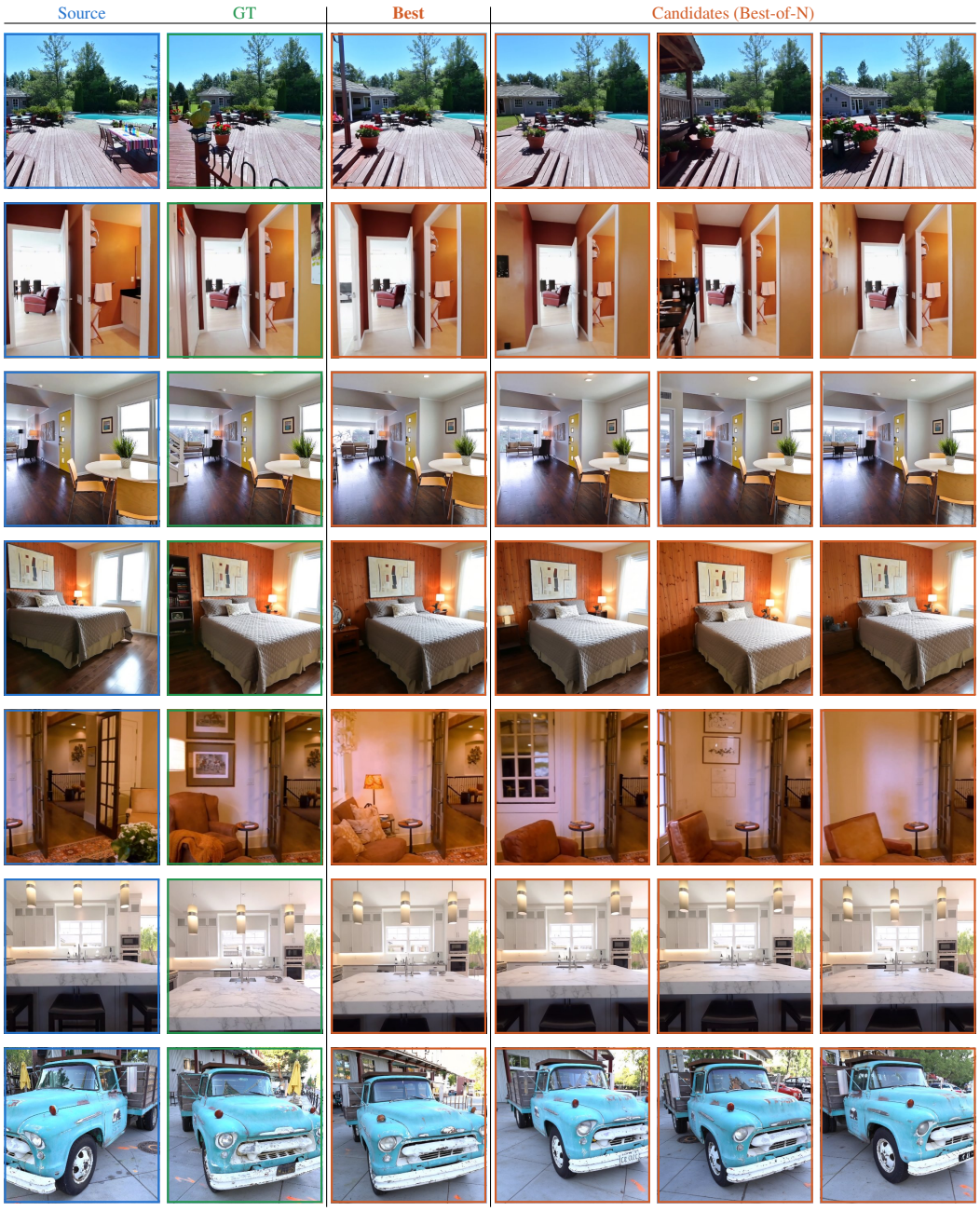

Figure 23: **Additional Best-of-N novel-view synthesis.** Each row shows the source view (blue border), ground truth (green border), the PCC-selected best sample (orange border), and three additional candidates (orange border) from the same Best-of-N pool. The first six rows are RE10K scenes; the last row is the *Truck* scene from Tanks & Temples.

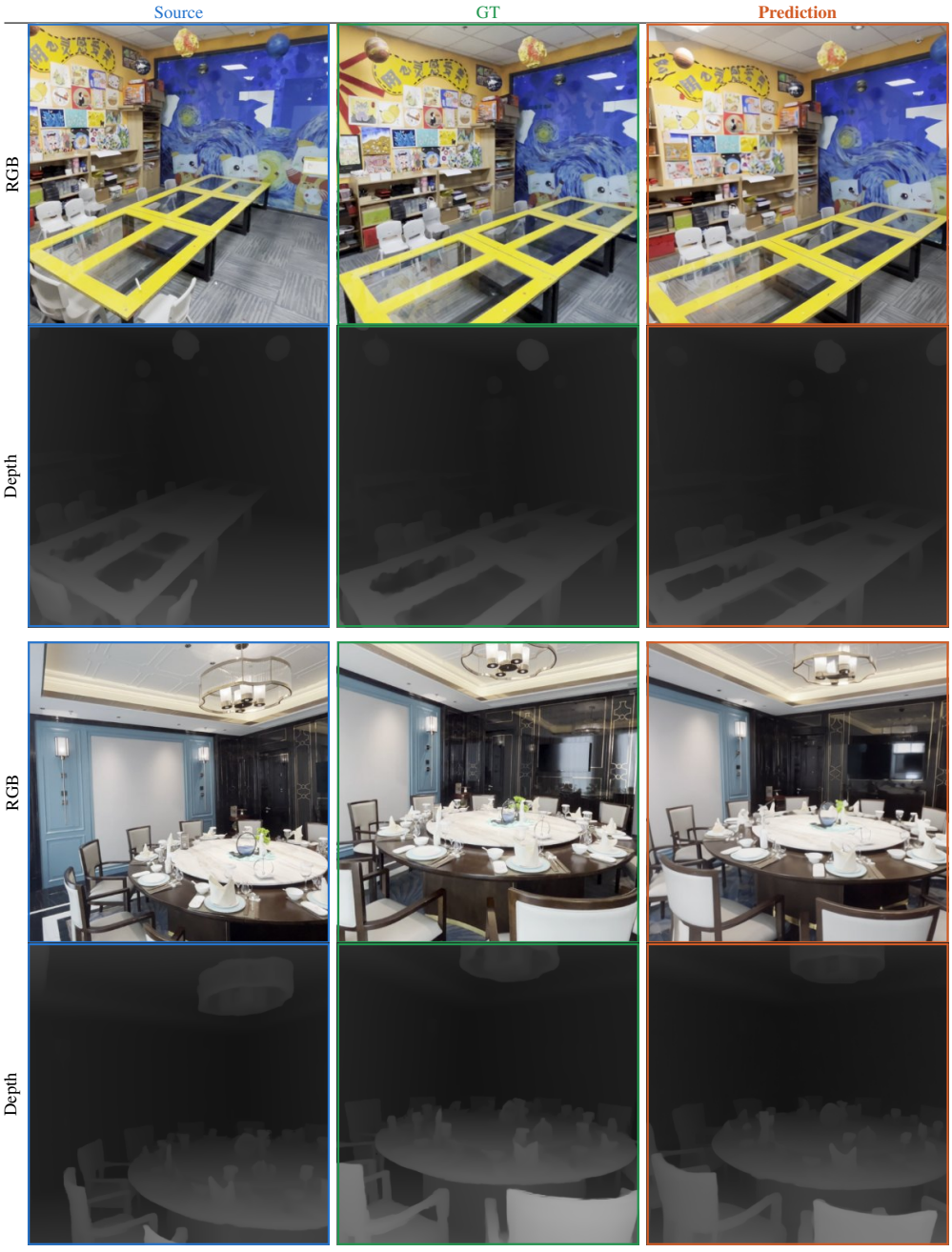

Table 10: **RGB and depth visualization.** For each of two scenes we show the source view (blue border), the ground-truth target (green border), and our prediction (orange border), as an RGB row and the corresponding depth row. Our predictions closely match the ground-truth targets in both appearance and geometry.

Metric	ESC-shadow @130k iters	ESC-Base @100k iters	ESC-PCC-BoN 100k iters	ESC-PCC-FT @100k+30k iters	Oracle
Synthesis Quality					
PSNR ↑	31.90	30.73	31.59	**33.65**	·
SSIM ↑	0.9607	0.9532	0.9578	**0.9693**	·
Perceptual Quality					
LPIPS ↓	0.0419	0.0584	0.0460	**0.0324**	·
DINO-Sim ↑	0.9715	0.9650	0.9652	**0.9746**	·
Geometric Fidelity					
PCC ↓	0.1057	0.1220	0.0915	**0.0721**	0.0576
Trajectory & View Prompt Consistency					
AUC@5 ↑	0.5986	0.4403	0.5283	**0.6132**	0.6250
AUC@15 ↑	0.8028	0.7067	0.7758	**0.8095**	0.7500
AUC@30 ↑	0.8598	0.8046	0.8462	**0.8641**	0.7948
RPE-R ↓	10.69	12.34	11.03	**10.37**	15.20
RPE-T ↓	**1.58**	2.69	1.82	1.71	1.15

Table 11: **Ablation of ESCvariants on the Blender benchmark.** We compare four ESCconfigurations and an oracle across four metric groups: synthesis quality (PSNR, SSIM), perceptual quality (LPIPS, DINO similarity), geometric fidelity (PCC), and trajectory & view-prompt consistency (pose AUC at $5°/15°/30°$, and relative pose errors RPE-R in degrees and RPE-T). Arrows indicate whether higher (↑) or lower (↓) is better. *ESC-Base* uses a single generated sample; *ESC-PCC-BoN* selects the best of 20 samples per view by geometric consistency (PCC); *ESC-PCC-FT* additionally fine-tunes with the PCC objective; and *ESC-shadow* denotes a model variant we trained without depth conditioning and prediction. *Oracle* indicates the ground-truth label compared against the paired DA3 estimate on the source images. The best, second-best, and third-best results among the ESCvariants in each row are highlighted. PCC-aware selection and fine-tuning consistently improve perceptual quality and geometric fidelity, with ESC-PCC-FT achieving the best results across nearly all metrics. More qualitative results are shown in Table 8.

F.2 How does ESC perform in simulation with ground-truth geometry?

A primary advantage of utilizing our synthetic Blender dataset is that it inherently provides pristine, exact ground-truth depth maps. Unlike the noisy pseudo-ground-truth depth maps extracted from alternative 3D reconstruction or estimation frameworks, these clean structural priors facilitate exceptionally smooth optimization dynamics and stable convergence across all model variants. Furthermore, this controlled simulation environment grants us the unique flexibility to generate diverse scene variations specifically within the occluded regions of the original viewpoint. This allows us to explicitly evaluate the network's generative completion capabilities under severe disocclusions, with qualitative examples demonstrated in Figure 8.

In this ablation experiment, we explore using the Projection Consistency Constraint (PCC) directly as a learning objective to fine-tune our model. Initializing from the pre-trained ESC-Base EMA checkpoint, we incorporate the PCC regularizer (detailed further in Section 3.1) to explicitly enforce strict geometric alignment across views[5]. To avoid disrupting the core flow-matching process at high noise levels, this PCC objective is applied exclusively over a truncated timestep range. This constraint ensures that the geometric regularization effectively shapes only the later, more structurally resolved stages of denoising.

Overall, all trained model configurations produce highly reasonable and geometrically plausible novel-view predictions, with the supervised fine-tuning (*ESC-PCC-FT*) variant consistently delivering the highest visual and structural fidelity (See Table 11 for the quantitative results and Table 8 for the qualitative results). To rigorously assess the geometric integrity of these generations, we overlay epipolar lines across the synthesized views in Figure 24. The precise correspondence of visual

[5]We utilize the model's own predicted depth map for PCC evaluation during this training phase and Best-of-N selection. Conversely, for final benchmarking, we employ the DA3 estimated depth.

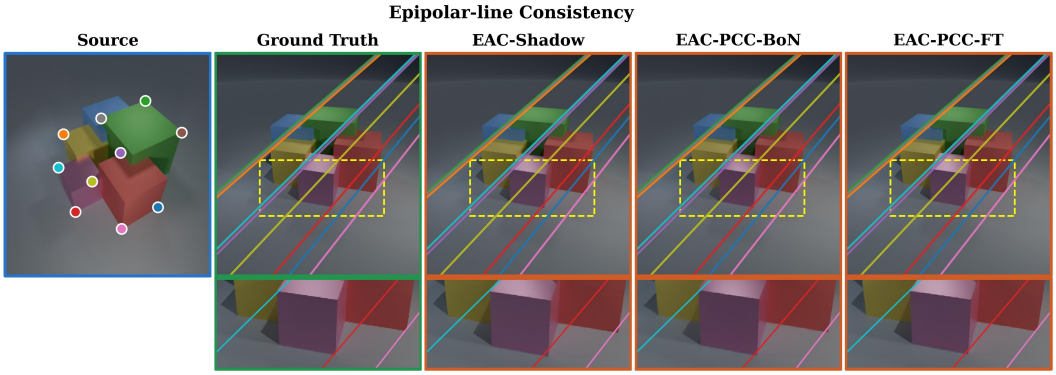

Figure 24: **Qualitative comparison of ESCvariants on a simulation environment.** Panels show the source view (blue border), ground-truth (green border), and three ESCvariant generations (orange border). The bottom row magnifies the yellow dashed region to illustrate 3D geometric consistency using color-matched epipolar lines derived from the source keypoints. While the bottom-right corner of the pink(red) cube nearly touches the red(pink) epipolar line in the ground-truth, it sits nearly flush in ESC-PCC-FT, shows a minor pixel gap in ESC-PCC-BoN, and leaves a small but more prominent gap in ESC-shadow, establishing the order of geometric alignment quality across the variants in this example.

features along these epipolar lines demonstrates that our ControlNet adapter effectively interprets Plücker tokens to generate novel views that conform to the underlying camera geometry. This alignment highlights the model's 3D consistency with the original scene. PCC vs Standard Evaluation Metrics