LATENT BRIDGE TRAJECTORY DISTILLATION FOR 3D SUPER-RESOLUTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Synthesizing high-resolution (HR) novel views from given low-resolution (LR) observations with known camera poses is intrinsically ill-posed. Existing 3D Gaussian Splatting (3DGS) super-resolution (SR) methods typically incorporate pretrained restoration priors into the optimization process. However, conventional diffusion-based restoration methods follow a noise-to-HR trajectory that is poorly aligned with the 3DGS optimization initialized from LR observations, and HR details inferred independently across views can be inconsistent. To address these issues, we propose Latent Bridge Trajectory Distillation (LBTD), which progressively distills restoration knowledge along an LR-anchored restoration trajectory into the 3D representation. Specifically, a multi-view latent bridge transports LR observations directly toward the HR distribution, while a cross-view adapter enables pose-nearest views to exchange complementary information for more consistent detail recovery. Trajectory distillation then maintains a persistent bridge trajectory across successive 3DGS optimization stages, where clean HR images are estimated at intermediate steps and updated 3D renderings are fed back to advance the trajectory. For detail enhancement, we propose detail-aware Gaussian splitting, which allocates finer primitives along directions where SR supervision reveals structures beyond the evidence available in the LR observations. Experiments show that LBTD achieves state-of-the-art performance, outperforming 3DSR by 0.31–0.89 dB in PSNR while using fewer Gaussians. In addition, LBTD recovers richer details with fewer distortions, leading to sharper and more faithful HR renderings.

1 INTRODUCTION

Novel view synthesis aims to generate photorealistic images of a 3D scene from unseen viewpoints given multi-view observations with known camera poses, and is widely used in applications such as virtual reality, cinematic production, and digital content creation. Recently, 3D Gaussian Splatting (3DGS) (Kerbl et al., 2023) has emerged as a dominant approach for this task because of its high rendering quality and real-time efficiency. In practical scenarios, however, multi-view images are often captured at limited resolution due to sensor constraints, whereas downstream applications increasingly demand high-resolution (HR) renderings with fine-grained details. Therefore, reconstructing a high-quality 3D representation from low-resolution (LR) observations with known poses for photorealistic HR novel-view rendering is an important problem with substantial industrial value.

In the absence of HR observations, optimizing 3DGS solely from LR inputs is insufficient to generate high-frequency details of a 3D scene, which often results in over-smoothed outputs. Existing 3D super-resolution (3D SR) methods therefore incorporate pretrained image or video SR priors, including pseudo-HR supervision with sub-pixel constraints (Feng et al., 2024), reliability-aware SR guidance (Asthana et al., 2026), and video-based generative models for improved cross-view consistency (Shen et al., 2024; Ko et al., 2025). More recent approaches tighten the coupling between diffusion-based SR and 3D reconstruction by distilling generative priors into 3DGS (Yu et al., 2024a) or interleaving diffusion sampling with 3D optimization (Chen et al., 2025). Despite this tighter coupling, existing generative 3D SR methods typically follow a noise-to-HR trajectory that is poorly aligned with an LR-initialized 3D representation. Their intermediate guidance is not explicitly anchored to the observed LR scene, which weakens the transfer of high-frequency details and introduces structural artifacts during 3D optimization. Additionally, as SR is inherently ill-posed,

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.

the independently restored HR images from different viewpoints are often inconsistent, providing conflicting supervision for 3DGS optimization and potentially causing distortions and floaters, as illustrated in Fig. 1. These limitations motivate an LR-anchored, multi-view restoration trajectory that co-evolves with the 3D representation.

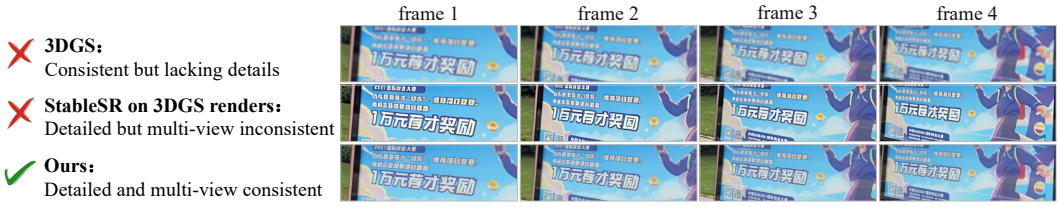

Figure 1: Comparison of multi-view consistency between existing methods and our method.

In this paper, we propose Latent Bridge Trajectory Distillation (LBTD), which reformulates 3DGS SR as progressive distillation along an LR-to-HR restoration trajectory. LBTD consists of three components: a multi-view latent bridge SR model, trajectory distillation, and detail-aware Gaussian splitting. Specifically, the proposed multi-view latent bridge SR model directly transports LR observations toward the corresponding HR domain, while a cross-view adapter enables pose-nearest views to exchange complementary information throughout the trajectory to improve multi-view consistency. This model effectively improves rendering quality while generating multi-view consistent content. During 3DGS optimization, trajectory distillation progressively transfers the LR-to-HR restoration process into 3DGS by maintaining persistent bridge states across successive optimization stages. Pseudo-HR images estimated at intermediate steps supervise 3D reconstruction, while updated 3D renderings are fed back to advance the subsequent bridge states. This closed-loop interaction allows the restoration trajectory and the 3D representation to evolve jointly.

The main contributions of this paper are summarized as follows:

1. We propose a multi-view latent bridge SR model that directly transports LR observations toward the HR domain while enabling information interaction among pose-nearest views, effectively improving output consistency and rendering quality.
2. We introduce trajectory distillation to progressively transfer the restoration process into 3DGS through stage-wise supervision and 3D feedback, allowing SR guidance and the 3D representation to evolve jointly.
3. We develop a detail-aware Gaussian splitting strategy that allocates finer primitives along directions where SR supervision reveals details absent from LR observations. Experiments show that LBTD achieves state-of-the-art performance, outperforming 3DSR by 0.31–0.89 dB in PSNR while using fewer Gaussians. Our method produces richer details with fewer distortions, yielding superior visual quality.

2 Related Works

2.1 3D Gaussian Splatting and Density Control

Novel view synthesis reconstructs a scene representation from posed images for rendering unseen viewpoints. NeRF (Mildenhall et al., 2020) represents scenes using implicit radiance fields, while 3D Gaussian Splatting (3DGS) (Kerbl et al., 2023) employs explicit anisotropic Gaussian primitives and differentiable rasterization for real-time rendering. Extensions such as Mip-Splatting (Yu et al., 2024b) further improve rendering robustness across sampling rates and output resolutions. Since the initial Gaussian set is insufficient to represent all scene details, 3DGS periodically performs adaptive densification according to accumulated positional gradients. Subsequent methods improve this process by designing better densification signals. AbsGS (Ye et al., 2024) alleviates gradient cancellation using homodirectional gradients, Pixel-GS (Zhang et al., 2024) incorporates pixel coverage and camera distance, and GDAGS (Zhou et al., 2026) exploits gradient-direction coherence to distinguish splitting from cloning. SteepGS (Wang et al., 2025) further modifies the splitting operation by determining offspring placement using a per-Gaussian splitting matrix. In contrast, our method

targets the additional representational demand introduced specifically by super-resolution. We retain the underlying densification criterion but determine the splitting direction from the accumulated pseudo-HR positional-gradient component that is orthogonal to the LR-fidelity gradient, thereby allocating finer primitives along directions where new high-frequency structures are required.

2.2 3D Super-Resolution and Bridge Models

3D super-resolution reconstructs an HR scene representation from posed LR observations. Early NeRF-based methods recover missing details through multi-view sub-pixel information (Wang et al., 2022), view-consistent detail generation (Han et al., 2025), or iterative diffusion-guided restoration (Lee et al., 2024). More recent methods incorporate pretrained image or video SR priors into 3DGS. SRGS (Feng et al., 2024) combines pseudo-HR supervision with sub-pixel constraints, while GaussianSR (Yu et al., 2024a) distills a diffusion prior through score distillation. SuperGaussian (Shen et al., 2024) and Sequence Matters (Ko et al., 2025) exploit video SR to obtain more coherent multi-view supervision, whereas SplatSuRe (Asthana et al., 2026) selectively applies SR guidance according to the sampling fidelity of the LR observations. 3DSR (Chen et al., 2025) establishes a tighter interaction by alternating diffusion-based SR sampling with 3DGS optimization and feeding reconstructed views into subsequent denoising states.

Bridge models provide an alternative to noise-initialized restoration by directly transporting samples between source and target distributions. I^2SB (Liu et al., 2023) constructs Schrödinger bridges for paired image restoration, while Diffusion Schrödinger Bridge Matching (Shi et al., 2023) and Denoising Diffusion Bridge Models (Zhou et al., 2024) develop related source-to-target diffusion formulations. Latent Bridge Matching (Chadebec et al., 2025) performs bridge matching in the latent space of a pretrained autoencoder, enabling efficient high-resolution image-to-image translation. Our method differs from both existing 3D SR and bridge-based restoration approaches. We construct a multi-view LR-to-HR latent bridge that enables the retrieval of complementary information from pose-nearest observations and progressively distill a persistent bridge trajectory into 3DGS, allowing the restoration process and the 3D representation to evolve jointly.

3 Proposed Methodology

3.1 Preliminaries

3D Gaussian Splatting 3D Gaussian Splatting (3DGS) (Kerbl et al., 2023) represents a scene as a set of N anisotropic 3D Gaussians and renders novel views through differentiable rasterization. The i-th Gaussian is centered at $\mu_i \in \mathbb{R}^3$ with covariance $\Sigma_i \in \mathbb{R}^{3\times3}$ and is defined as $G_i(\mathbf{x}) = \exp\big(-\frac{1}{2}(\mathbf{x}-\mu_i)^\top \Sigma_i^{-1}(\mathbf{x}-\mu_i)\big)$. The covariance is factorized as $\Sigma_i = R_i S_i S_i^\top R_i^\top$, where $S_i = \mathrm{diag}(s_i)$ and R_i determine the scales and orientation of the principal axes, respectively. In addition, each Gaussian stores an opacity o_i and spherical-harmonic coefficients c_i for view-dependent appearance. During rendering, the Gaussians are projected onto the image plane and composited in depth order, i.e., $C(\mathbf{p}) = \sum_{i=1}^{N} c_i \alpha_i \prod_{j=1}^{i-1}(1-\alpha_j)$, where $\alpha_i = o_i G'_i(\mathbf{p})$ and G'_i denotes the projected 2D Gaussian evaluated at pixel $\mathbf{p}$. Given a supervision image Y and a rendered image R, the Gaussian attributes are optimized with the loss defined as follows:

$$\mathcal{L}_{\mathrm{GS}}(R,Y) = (1-\lambda)\|R-Y\|_1 + \lambda\mathcal{L}_{\text{D-SSIM}}(R,Y). \tag{1}$$

Latent Bridge Matching Latent Bridge Matching (LBM) (Chadebec et al., 2025) learns a generative transport process between paired source and target distributions in the latent space of a pretrained autoencoder. Given paired samples $(x^{\mathrm{src}}, x^{\mathrm{tgt}}) \sim \Pi$, where Π denotes a coupling between the source and target distributions, these two samples are encoded as $z^{\mathrm{src}} = E(x^{\mathrm{src}})$ and $z^{\mathrm{tgt}} = E(x^{\mathrm{tgt}})$, where $E(\cdot)$ denotes a pre-trained encoder. For $t \in [0,1]$, a Brownian bridge connecting the source at $t=0$ and the target at $t=1$ is defined as

$$z_t = (1-t)z^{\mathrm{src}} + t z^{\mathrm{tgt}} + \eta\sqrt{t(1-t)}\,\epsilon, \qquad \epsilon \sim \mathcal{N}(0,I), \tag{2}$$

where η controls the bridge stochasticity. The perturbation vanishes at both endpoints, yielding $z_0 = z^{\mathrm{src}}$ and $z_1 = z^{\mathrm{tgt}}$. Conditioned on the target endpoint, the corresponding bridge drift is

$$v^*(z_t, t; z^{\mathrm{tgt}}) = \frac{z^{\mathrm{tgt}} - z_t}{1-t} \qquad 0 \le t < 1. \tag{3}$$

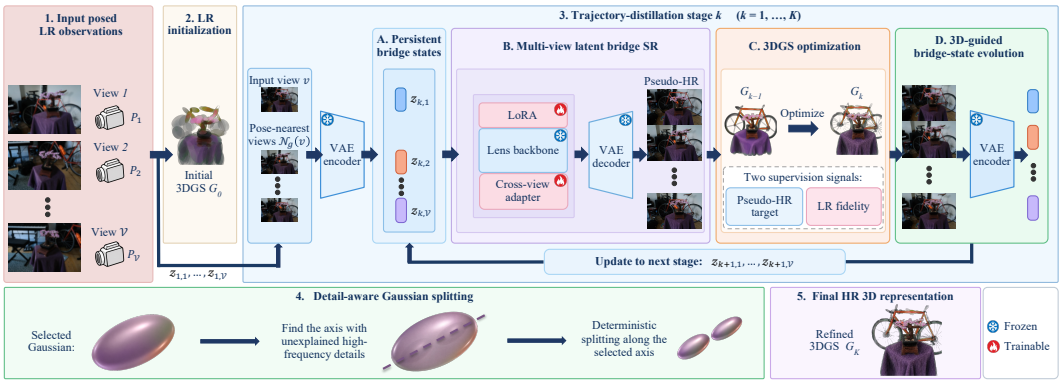

Figure 2: **Overall pipeline of LBTD.** After LR initialization, LBTD alternates multi-view bridge prediction, 3DGS distillation with detail-aware splitting, and rendering feedback along a persistent LR-to-HR restoration trajectory.

LBM learns a neural drift field v_θ by minimizing the loss defined as follows:

$$\mathcal{L}_{\text{LBM}} = \mathbb{E}\left[\left\|v_\theta(z_t, t) - v^*(z_t, t; z^{\text{tgt}})\right\|_2^2\right]. \tag{4}$$

The learned velocity field transports the source data toward the corresponding target distribution and, from any intermediate state, the target endpoint is estimated as follows:

$$\hat{z}^{\text{tgt}} = z_t + (1-t)v_\theta(z_t, t). \tag{5}$$

The corresponding target image is obtained by decoding $\hat{x}^{\text{tgt}} = D(\hat{z}^{\text{tgt}})$. By performing transport in the latent space, LBM substantially reduces the dimensionality of the transport process and enables efficient high-resolution image-to-image generation. In Sec. 3.2, we specialize this formulation to super-resolution by treating LR and HR images as the source and target distributions, respectively, and further extend it to a multi-view bridge model.

3.2 Latent Bridge Trajectory Distillation (LBTD)

Given $\mathcal{V}$ posed LR observations $\mathcal{D} = \{(I_\nu^{\text{LR}}, P_\nu)\}_{\nu=1}^{\mathcal{V}}$, where I_ν^{LR} and P_ν denote the ν-th LR image and its corresponding camera pose, respectively, our goal is to reconstruct a 3D Gaussian representation $\mathcal{G}$ for HR novel-view rendering. Since corresponding HR observations are unavailable during scene reconstruction, we propose Latent Bridge Trajectory Distillation (LBTD), which couples a multi-view latent bridge SR model with 3DGS optimization and distills an LR-to-HR restoration trajectory into the evolving 3D representation, as illustrated in Fig. 2.

Specifically, LBTD initializes both the 3D Gaussian representation and persistent bridge states from the LR observations, and then alternates among multi-view pseudo-HR prediction, 3DGS optimization with detail-aware Gaussian splitting, and 3D rendering feedback over K stages. Through this recurrent interaction, the restoration trajectory and the 3D representation evolve jointly toward a coherent HR reconstruction. We next describe the multi-view latent bridge SR model, trajectory distillation, and detail-aware Gaussian splitting in detail.

3.2.1 Multi-view Latent Bridge SR Model

We first specialize the latent bridge formulation introduced in Sec. 3.1 to image super-resolution and further extend it to multi-view restoration. Given an LR observation I_v^{LR}, we bicubically upsample it to the target resolution and encode it as the source endpoint, $z_v^{\text{src}} = E(\mathcal{F}_{\uparrow s}(I_v^{\text{LR}}))$, while its corresponding HR image defines the target endpoint, $z_v^{\text{tgt}} = E(I_v^{\text{HR}})$. We instantiate the bridge model using Lens (Chen et al., 2026), which consists of a frozen autoencoder and a pretrained multimodal Transformer backbone. Rather than fine-tuning the entire backbone, we adopt low-rank adaptation (LoRA) modules (Hu et al., 2022) and train them on 2D super-resolution data. Following SeeSR (Wu et al., 2024), we use a degradation-aware prompt extractor to obtain semantic tags from I_v^{LR}. These tags are inserted into the fixed prompt "a high-resolution, sharp, photorealistic

photograph of [tags]". The resulting prompt is encoded by GPT-OSS-20B (Agarwal et al., 2025) to provide the semantic condition c_v.

Although this LR-anchored bridge provides effective single-image restoration, independently restoring different views may still produce inconsistent content across views. We therefore extend it to a multi-view bridge by allowing each input view to retrieve complementary information from its pose-nearest observations. Specifically, we form fixed groups of G views using camera-pose distances. For a view v in group $\mathcal{V}_g$, we define $\mathcal{N}_g(v) = \mathcal{V}_g \setminus \{v\}$ as its reference-view set. For the tokens of target view v at block ℓ, we construct the reference-token bank $H_{\ell,\mathcal{N}_g(v)}$ by concatenating the tokens of its neighboring views:

$$H_{\ell,\mathcal{N}_g(v)} = \mathrm{Concat}\left(\{h_{\ell,u}\}_{u\in\mathcal{N}_g(v)}\right), \tag{6}$$

where $h_{\ell,u} \in \mathbb{R}^{L\times d}$ denotes the L tokens of view u at block ℓ, each token of dimension d. We use a cross-view adapter, using the target-view tokens as queries to retrieve complementary information from the reference bank through cross attention, which is computed as follows:

$$a_{\ell,v} = \mathrm{CrossAttn}_\ell\left(h_{\ell,v}, H_{\ell,\mathcal{N}_g(v)}\right), \qquad h^{\mathrm{out}}_{\ell,v} = h^{\mathrm{base}}_{\ell,v} + \gamma_\ell a_{\ell,v}, \tag{7}$$

where $h^{\mathrm{base}}_{\ell,v}$ denotes the output of the original Lens branch, $a_{\ell,v}$ is the retrieved cross-view feature, and γ_ℓ is a learnable scalar. The same operation is applied symmetrically to all views in $\mathcal{V}_g$, allowing each view to generate its own representation while selectively aggregating complementary information from the other views in the group. Consequently, more consistent high-frequency details can be inferred using information shared across overlapping views, further enhancing visual quality for subsequent 3D reconstruction.

The multi-view latent SR bridge is optimized using the bridge-matching objective together with an auxiliary perceptual loss, defined as follows:

$$\mathcal{L}_{\mathrm{SR}} = \mathcal{L}_{\mathrm{LBM}} + \lambda_p \mathcal{L}_{\mathrm{LPIPS}}\left(\widehat{I}^{\mathrm{HR}}_v, I^{\mathrm{HR}}_v\right), \tag{8}$$

where $\widehat{I}^{\mathrm{HR}}_v$ denotes the estimated HR image.

3.2.2 Trajectory Distillation

To transfer the trained multi-view latent bridge SR model into the 3D representation, we introduce trajectory distillation, which couples the LR-to-HR restoration trajectory with successive stages of 3DGS optimization. Rather than supervising 3DGS only with the final SR output, we align the bridge sampling intervals with K stages of 3DGS optimization and maintain a persistent bridge state for each input view. As a result, the LR-to-HR restoration trajectory evolves jointly with the 3D representation, rather than being independently restarted at each stage.

Initialization and stage-wise distillation. We first initialize the Gaussian representation using the observed LR images upsampled to the target resolution by minimizing the loss:

$$\mathcal{L}_{\mathrm{init}} = \frac{1}{\mathcal{V}}\sum_{v=1}^{\mathcal{V}} \mathcal{L}_{\mathrm{GS}}\left(\mathcal{R}(\mathcal{G}, P_v), \mathcal{F}_{\uparrow s}(I^{\mathrm{LR}}_v)\right), \tag{9}$$

yielding the LR-initialized representation $\mathcal{G}_0$. We adopt an increasing bridge schedule $0 = t_1 < t_2 < \cdots < t_{K+1} = 1$ and initialize the bridge state of view v as $z_{1,v} = z^{\mathrm{src}}_v$. At stage k, a sampling step updates $z_{k,v}$ to $z_{k+1,v}$ and returns the clean HR prediction $\widehat{I}^{\mathrm{HR}}_{k,v} = \widehat{I}^{\mathrm{HR}}_v(z_{k,v}, t_k)$ for each view. We optimize the Gaussian representation using these predictions and the original LR observations:

$$\mathcal{L}_k = \frac{1}{\mathcal{V}}\sum_{v=1}^{\mathcal{V}}\left[\mathcal{L}_{\mathrm{GS}}\left(\mathcal{R}(\mathcal{G}, P_v), \widehat{I}^{\mathrm{HR}}_{k,v}\right) + \lambda_{\mathrm{LR}}\mathcal{L}_{\mathrm{GS}}\left(\mathcal{F}_{\downarrow s}\left(\mathcal{R}(\mathcal{G}, P_v)\right), I^{\mathrm{LR}}_v\right)\right]. \tag{10}$$

We update the Gaussian representation from $\mathcal{G}_{k-1}$ to $\mathcal{G}_k$ by minimizing Eq. (10). The pseudo-HR term guides the model to recover missing high-frequency information, while the LR-fidelity term anchors the output to the actual LR observations.

3D-guided trajectory evolution. After each distillation stage, the updated scene is rendered as $R_{k,v} = \mathcal{R}(\mathcal{G}_k, P_v)$ and fed back into the bridge trajectory. To achieve this, we estimate the velocity of the rendered output obtained from intermediate steps, computed as follows:

$$\widetilde{v}_{k,v} = \begin{cases} v^{\mathrm{MV}}_{\theta,\phi}\left(z_{1,v}; \mathbf{Z}^{g}_{t_1,\mathcal{N}_g(v)}, t_1, c_v\right), & k = 1, \\ \dfrac{E(R_{k-1,v}) - z_{k,v}}{1 - t_k}, & k \geq 2, \end{cases} \tag{11}$$

where $v^{\mathrm{MV}}_{\theta,\phi}$ denotes the velocity estimated by the multi-view latent bridge SR model, and $\mathbf{Z}^{g}_{t_1,\mathcal{N}_g(v)}$ denotes the latent representations of the pose-nearest views at time t_1. For $k = 1$, the velocity predicted by the multi-view bridge model is directly adopted, while for $k \geq 2$, the velocity is computed using the encoded rendering as the target estimate. The persistent bridge state is then updated from t_k to t_{k+1} using $z_{k+1,v} = z_{k,v} + (t_{k+1} - t_k)\widetilde{v}_{k,v}$. In this way, the evolving 3D representation directly influences the subsequent restoration trajectory, constraining bridge states with content that can be coherently represented in the shared scene. By repeatedly performing this process, LBTD threads a single persistent LR-to-HR trajectory through successive 3DGS optimization stages, which allows the restoration guidance and the 3D representation to progressively refine each other.

3.2.3 Detail-Aware Gaussian Splitting

By leveraging the trained multi-view bridge SR model, trajectory distillation effectively introduces high-frequency information into the 3D representation. This requires 3DGS to allocate additional representational capacity to regions where finer structures are needed. However, conventional 3DGS places the children of a selected Gaussian by randomly sampling from the parent distribution, without considering the spatial directions in which SR supervision requires additional refinement. To address this limitation, we propose detail-aware Gaussian splitting, which uses the discrepancy between pseudo-HR and LR supervision to determine how each selected Gaussian should be refined.

Detail-specific splitting direction. During trajectory distillation, the pseudo-HR and the LR-fidelity terms in Eq. (10) provide complementary supervision: the former introduces missing high-frequency structures, whereas the latter constrains content supported by the observations. For the i-th Gaussian at the n-th optimization iteration, let $g^{\mathrm{HR}}_{i,n}$ and $g^{\mathrm{LR}}_{i,n}$ denote the corresponding gradients with respect to its position μ_i. We extract the component of the pseudo-HR gradient that is orthogonal to the LR-fidelity gradient as follows:

$$q_{i,n} = g^{\mathrm{HR}}_{i,n} - \frac{\left\langle g^{\mathrm{HR}}_{i,n}, g^{\mathrm{LR}}_{i,n} \right\rangle}{\left\| g^{\mathrm{LR}}_{i,n} \right\|_2^2} g^{\mathrm{LR}}_{i,n}, \tag{12}$$

where $q_{i,n}$ represents the position-update direction given by the component of the pseudo-HR positional gradient orthogonal to the current LR-fidelity gradient. Since individual gradients may be noisy and view-dependent, we accumulate their outer products between consecutive densification operations:

$$M_i = \sum_{n \in \mathcal{T}_i} q_{i,n} q_{i,n}^{\top}, \quad j_i^{\star} = \arg\max_{j \in \{1,2,3\}} d_{i,j}^{\top} M_i d_{i,j}, \quad a_i = d_{i,j_i^{\star}}, \tag{13}$$

where $\mathcal{T}_i$ denotes the set of iterations during which Gaussian i is visible, and $\{d_{i,1}, d_{i,2}, d_{i,3}\}$ are the principal axes defined by R_i. Thus, a_i corresponds to the principal direction receiving the strongest accumulated SR-specific refinement signal.

Deterministic splitting. Given the selected axis a_i, we replace random child placement with a symmetric deterministic split. The child Gaussian centers μ_i' are initialized as follows:

$$\mu_i' = \mu_i \pm \beta s_{i,a} a_i, \tag{14}$$

where β controls the displacement of the child Gaussians from the parent center. Meanwhile, the scale along the selected direction is reduced as $s'_{i,a} = \frac{s_{i,a}}{\kappa}$, where $\kappa > 1$ is the shrinkage factor, while the scales along the remaining two principal axes remain unchanged. Both children are initialized with an opacity of $o_i' = 1 - \sqrt{1 - o_i}$ and directly inherit the remaining attributes from the parent. In essence, the proposed strategy preserves the standard densification criterion while replacing random child placement with SR-guided directional refinement. Consequently, additional Gaussian capacity is allocated along directions where trajectory distillation consistently reveals missing high-frequency structures, without introducing any additional learnable parameters.

Table 1: Quantitative results on Mip-NeRF 360 (Barron et al., 2022), DL3DV (Ling et al., 2024), and LLFF (Mildenhall et al., 2019) under the $\times 8 \rightarrow \times 2$ setting. The best, second-best, and third-best results within each dataset are highlighted in red, orange, and yellow, respectively.

Dataset	Methods	PSNR↑	SSIM↑	LPIPS↓	NIQE↓	DISTS↓	FID↓	DreamSim↓	CMMD↓
Mip-NeRF 360	3DGS+StableSR	24.25	0.6914	0.2624	5.106	0.1439	40.24	0.0260	0.2406
	Mip-Splatting	25.97	0.7391	0.2635	5.824	0.1441	24.88	0.0160	0.1194
	SRGS	26.68	0.7624	0.2302	5.265	0.1287	34.11	0.0195	0.1361
	3DSR	26.03	0.7380	0.2330	5.109	0.1280	23.53	0.0152	0.0866
	SplatSuRe	26.31	0.7392	0.2205	3.923	0.1385	29.99	0.0160	0.1156
	LBTD (Ours)	26.53	0.7568	0.1939	4.883	0.1083	20.07	0.0132	0.0518
DL3DV	3DGS+StableSR	25.33	0.8231	0.1688	5.433	0.1203	21.77	0.0298	0.4596
	Mip-Splatting	28.35	0.8638	0.1758	5.506	0.1171	12.45	0.0150	0.1884
	SRGS	28.73	0.8796	0.1406	5.476	0.1101	17.21	0.0207	0.2194
	3DSR	28.22	0.8662	0.1378	5.123	0.1071	13.18	0.0162	0.2047
	SplatSuRe	28.21	0.8554	0.1379	4.066	0.1092	15.73	0.0182	0.2279
	LBTD (Ours)	29.11	0.8840	0.1224	5.207	0.0933	11.75	0.0140	0.1356
LLFF	3DGS+StableSR	23.44	0.7450	0.2063	5.126	0.1408	31.63	0.0143	0.2756
	Mip-Splatting	24.44	0.7616	0.2527	5.870	0.1509	25.36	0.0143	0.1741
	SRGS	24.01	0.7732	0.1850	4.946	0.1240	36.72	0.0169	0.1952
	3DSR	24.35	0.7626	0.1745	4.789	0.1278	21.40	0.00851	0.0937
	SplatSuRe	23.41	0.7436	0.1978	4.306	0.1281	34.10	0.0153	0.1860
	LBTD (Ours)	24.66	0.7730	0.1708	5.017	0.1194	23.00	0.00901	0.1034

4 Experiments and Analysis

4.1 Experiment Setup

Datasets. We evaluate our method on four benchmarks: Mip-NeRF 360 (Barron et al., 2022), DL3DV (Ling et al., 2024), LLFF (Mildenhall et al., 2019), and Tanks & Temples (Knapitsch et al., 2017), following the 3DGS train/test protocol (Kerbl et al., 2023). For Mip-NeRF 360, DL3DV, and LLFF, the input images are downsampled by a factor of 8 to approximately 500×375, 479×268, and 504×378, respectively, and reconstructed at $\times4$ the LR resolution. For Tanks & Temples, images are downsampled by $\times4$ to approximately 245×136 and reconstructed at $\times4$ the LR resolution.

Baselines and metrics. We compare our method against Mip-Splatting (Yu et al., 2024b), SRGS (Feng et al., 2024), 3DSR (Chen et al., 2025), and SplatSuRe (Asthana et al., 2026). We additionally report results for 3DGS+StableSR, which first independently super-resolves the input views using StableSR (Wang et al., 2024) before performing 3DGS optimization. We evaluate reconstruction fidelity using PSNR and SSIM. Perceptual quality is assessed using LPIPS (Zhang et al., 2018), DISTS (Ding et al., 2022), DreamSim (Fu et al., 2023), FID (Heusel et al., 2017), CMMD (Jayasumana et al., 2024), and NIQE (Mittal et al., 2013).

Implementation details. We use Lens-Base (Chen et al., 2026) as the bridge SR teacher. For each scene, we optimize 3DGS for 50K iterations, consisting of 30K iterations for LR initialization followed by $K{=}4$ trajectory-distillation stages of 5K iterations each. We use the bridge schedule $\{t_k\}_{k=1}^{K+1} = \{0, 0.25, 0.5, 0.75, 1\}$ and process groups of $G{=}4$ pose-nearest views. The cross-view adapter is trained on the DL3DV-Benchmark. For detail-aware splitting, we set $\kappa{=}1.6$ and $\tau_{\mathrm{pd}}{=}0.005$. Additional training and implementation details are provided in Appendix C.

4.2 Comparison with State-of-the-Art Methods

Table 1 reports the quantitative results on Mip-NeRF 360, DL3DV, and LLFF, while the results on Tanks & Temples are provided in Appendix Table 3. Qualitative comparisons are shown in Figs. 3 and 4. The numbers of Gaussians and the rendering speeds are reported in Appendix D.2.

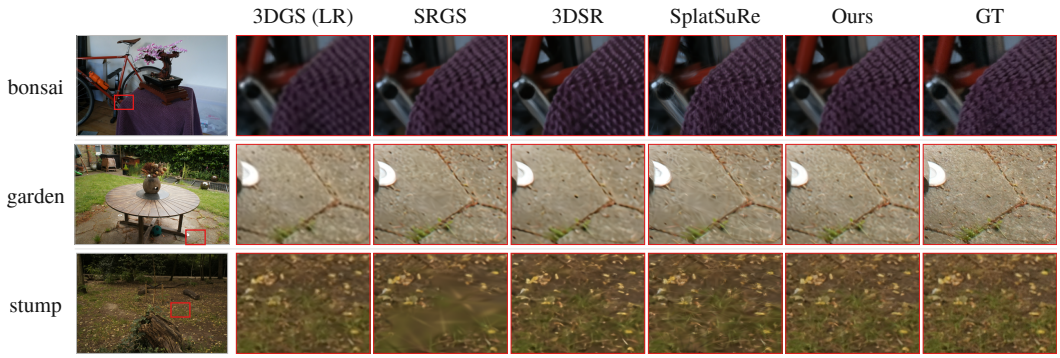

Figure 3: Qualitative comparison on Mip-NeRF 360 (Barron et al., 2022) under the $\times 8 \rightarrow \times 2$ setting.

Mip-NeRF 360. LBTD performs best in LPIPS, DISTS, FID, DreamSim, and CMMD, while ranking second in PSNR, SSIM, and NIQE. Compared with 3DSR (Chen et al., 2025), LBTD improves all eight metrics, including a 0.50 dB gain in PSNR and reductions in LPIPS from 0.2330 to 0.1939 and FID from 23.53 to 20.07. As shown in Fig. 3, LBTD recovers finer textures with fewer distorted or hallucinated details, particularly in regions containing repetitive high-frequency patterns.

DL3DV. LBTD achieves the best performance in seven of the eight metrics. Relative to 3DSR, it improves PSNR from 28.22 to 29.11 and SSIM from 0.8662 to 0.8840, while reducing LPIPS from 0.1378 to 0.1224, DISTS from 0.1071 to 0.0933, and FID from 13.18 to 11.75. Although SplatSuRe (Asthana et al., 2026) achieves the best NIQE, LBTD outperforms it on all remaining metrics. Fig. 4 further shows that LBTD better preserves fine textures and structural boundaries while suppressing inconsistent high-frequency details.



Figure 4: Qualitative comparison on DL3DV (Ling et al., 2024) under the $\times 8 \rightarrow \times 2$ setting.

LLFF. On the forward-facing LLFF dataset, LBTD achieves the best PSNR, LPIPS, and DISTS and attains an SSIM of 0.7730, comparable to the best result of 0.7732 achieved by SRGS (Feng et al., 2024). Compared with 3DSR, LBTD improves PSNR from 24.35 to 24.66 and reduces DISTS from 0.1278 to 0.1194. Additional qualitative comparisons are provided in Appendix Fig. 12.

4.3 Ablation Study

Multi-view latent bridge SR. We first evaluate the contributions of the latent bridge SR prior and the cross-view adapter in Table 2, with qualitative comparisons provided in Appendix Fig. 7. Compared with LR initialization alone, introducing the single-view latent bridge substantially improves reconstruction quality, increasing PSNR from 25.94 to 26.52 while reducing LPIPS from 0.2635 to 0.1980 and DISTS from 0.1442 to 0.1080. Incorporating the cross-view adapter further improves NIQE from 5.039 to 4.883 and FID from 20.22 to 20.07. These results show that the

Table 2: Quantitative ablation studies on the Mip-NeRF 360 dataset (Barron et al., 2022) under the $\times 8 \rightarrow \times 2$ setting: (a) multi-view latent bridge SR, (b) trajectory distillation, and (c) detail-aware Gaussian splitting. The best result within each group and column is highlighted in **bold**.

	Variant	PSNR↑	SSIM↑	LPIPS↓	NIQE↓	DISTS↓	FID↓
(a)	LR Initialization Only	25.94	0.7390	0.2635	5.828	0.1442	24.67
	Single-View Latent Bridge SR	26.52	0.7562	0.1980	5.039	**0.1080**	20.22
	Multi-View Latent Bridge SR (Ours)	**26.53**	**0.7568**	**0.1939**	**4.883**	0.1083	**20.07**
(b)	No trajectory distillation	26.37	0.7502	0.1990	**4.739**	0.1151	20.63
	Trajectory Distillation (K=1)	26.41	0.7532	0.2099	5.073	0.1137	21.03
	Trajectory Distillation (K=2)	26.47	0.7551	0.2005	4.987	0.1094	20.78
	Trajectory Distillation (K=4, Ours)	**26.53**	**0.7568**	**0.1939**	4.883	**0.1083**	**20.07**
(c)	Standard Splitting	26.51	0.7561	0.1947	**4.856**	0.1088	20.71
	Detail-Aware Gaussian Splitting (Ours)	**26.53**	**0.7568**	**0.1939**	4.883	**0.1083**	**20.07**

bridge SR prior substantially improves reconstruction quality over LR initialization alone, while the multi-view model further improves the perceptual quality and consistency (Appendix D.3).

Trajectory distillation. Table 2 studies the effect of progressively distilling the bridge trajectory into 3DGS, with qualitative results shown in Appendix Fig. 8. Increasing the number of distillation stages generally improves reconstruction quality, with K=4 achieving the best SSIM, LPIPS, DISTS, and FID in Table 2(b). The variant without trajectory distillation uses the same optimization steps but restarts 4-step bridge sampling from the original LR observations. Relative to this baseline, trajectory distillation with K=4 improves PSNR from 26.37 to 26.53 while reducing FID from 20.63 to 20.07. This comparison verifies that the performance gain is not merely due to repeated SR guidance; rather, it arises from continuously coupling bridge-state evolution with the progressively refined 3D representation.

Detail-aware Gaussian splitting. Finally, we evaluate the proposed detail-aware splitting strategy in Table 2, with qualitative comparisons provided in Appendix Fig. 9. Replacing standard Gaussian splitting with our detail-aware strategy improves PSNR from 26.51 to 26.53 and reduces FID from 20.71 to 20.07. These results suggest that directing Gaussian refinement according to SR-specific gradient signals improves reconstruction quality over standard random splitting, particularly in terms of the perceptual quality of the reconstructed scene.

5 Conclusion

In this paper, we address the problem of 3DGS super-resolution, which aims to synthesize high-frequency details from posed low-resolution (LR) observations. Existing generative priors are often poorly aligned with LR-initialized 3D optimization and may introduce cross-view inconsistencies. To overcome these limitations, we propose Latent Bridge Trajectory Distillation (LBTD), which couples an LR-anchored multi-view restoration trajectory with an evolving 3D representation. The proposed multi-view latent bridge directly transports LR observations toward the high-resolution (HR) domain while exchanging complementary information among pose-nearest views. Trajectory distillation maintains persistent bridge states across successive 3DGS optimization stages and feeds refined 3D renderings back into the restoration process. In addition, detail-aware Gaussian splitting allocates additional representational capacity to regions where SR supervision reveals missing structures. Experiments on four benchmarks show that LBTD outperforms 3DSR by 0.31–0.89 dB in PSNR while achieving consistent improvements in SSIM, LPIPS, and DISTS, all while using fewer Gaussians. Qualitative results further demonstrate that LBTD produces richer details and fewer structural distortions in HR novel-view rendering.

AI USE STATEMENT

Large language models were used solely as writing aids for grammar checking and language polishing.

REFERENCES

Sandhini Agarwal, Lama Ahmad, Jason Ai, Sam Altman, Andy Applebaum, Edwin Arbus, Rahul K Arora, Yu Bai, Bowen Baker, Haiming Bao, et al. gpt-oss-120b & gpt-oss-20b model card. arXiv preprint arXiv:2508.10925, 2025.

Eirikur Agustsson and Radu Timofte. NTIRE 2017 challenge on single image super-resolution: Dataset and study. In 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 1122–1131, 2017. doi: 10.1109/CVPRW.2017.150.

Mohammad Asim, Christopher Wewer, Thomas Wimmer, Bernt Schiele, and Jan Eric Lenssen. MEt3R: Measuring multi-view consistency in generated images. In IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 6034–6044, 2025.

Pranav Asthana, Alex Hanson, Allen Tu, Tom Goldstein, Matthias Zwicker, and Amitabh Varshney. SplatSuRe: Selective super-resolution for multi-view consistent 3d gaussian splatting. In IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2026. arXiv:2512.02172.

Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 5460–5469. IEEE, 2022.

Clément Chadebec, Onur Tasar, Sanjeev Sreetharan, and Benjamin Aubin. LBM: Latent bridge matching for fast image-to-image translation. In IEEE/CVF International Conference on Computer Vision (ICCV), 2025. arXiv:2503.07535.

Dong Chen, Fangyun Wei, Ziyu Wan, Dongdong Chen, Jiawei Zhang, Jinjing Zhao, Sirui Zhang, Yang Yue, Zhiyang Liang, Baining Guo, et al. Lens: Rethinking training efficiency for foundational text-to-image models. arXiv preprint arXiv:2605.21573, 2026.

Yi-Ting Chen, Ting-Hsuan Liao, Pengsheng Guo, Alexander Schwing, and Jia-Bin Huang. Bridging diffusion models and 3d representations: A 3d consistent super-resolution framework. In IEEE/CVF International Conference on Computer Vision (ICCV), 2025. arXiv:2508.04090.

Keyan Ding, Kede Ma, Shiqi Wang, and Eero P Simoncelli. Image quality assessment: Unifying structure and texture similarity. IEEE transactions on pattern analysis and machine intelligence, 44(5):2567–2581, 2022.

Xiang Feng, Yongbo He, Yubo Wang, Yan Yang, Wen Li, Yifei Chen, Zhenzhong Kuang, Jiajun Ding, Jianping Fan, and Jun Yu. SRGS: Super-resolution 3d gaussian splatting. arXiv preprint arXiv:2404.10318, 2024. URL https://arxiv.org/abs/2404.10318.

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. arXiv preprint arXiv:2306.09344, 2023.

Shuhang Gu, Andreas Lugmayr, Martin Danelljan, Manuel Fritsche, Julien Lamour, and Radu Timofte. DIV8K: DIVerse 8k resolution image dataset. In 2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW), pp. 3512–3516, 2019. doi: 10.1109/ICCVW.2019.00435.

Yuqi Han, Tao Yu, Xiaohang Yu, Di Xu, Binge Zheng, Zonghong Dai, Changpeng Yang, Yuwang Wang, and Qionghai Dai. Super-NeRF: View-consistent detail generation for nerf super-resolution. IEEE Transactions on Visualization and Computer Graphics, 31(9):6053–6066, 2025. doi: 10.1109/TVCG.2024.3490840.

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 Advances in Neural Information Processing Systems (NeurIPS), 2017.

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

Sadeep Jayasumana, Srikumar Ramalingam, Andreas Veit, Daniel Glasner, Ayan Chakrabarti, and Sanjiv Kumar. Rethinking fid: Towards a better evaluation metric for image generation. In 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 9307–9315. IEEE, 2024.

Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4401–4410, 2019. doi: 10.1109/CVPR.2019.00453.

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

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.

Hyun-kyu Ko, Dongheok Park, Youngin Park, Byeonghyeon Lee, Juhee Han, and Eunbyung Park. Sequence matters: Harnessing video models in 3d super-resolution. In AAAI Conference on Artificial Intelligence (AAAI), 2025.

Wei-Sheng Lai, Jia-Bin Huang, Oliver Wang, Eli Shechtman, Ersin Yumer, and Ming-Hsuan Yang. Learning blind video temporal consistency. In European conference on computer vision, pp. 179–195. Springer, 2018.

Jie Long Lee, Chen Li, and Gim Hee Lee. DiSR-NeRF: Diffusion-guided view-consistent super-resolution nerf. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 20561–20570, 2024.

Yawei Li, Kai Zhang, Jingyun Liang, Jiezhang Cao, Ce Liu, Rui Gong, Yulun Zhang, Hao Tang, Yun Liu, Denis Demandolx, Rakesh Ranjan, Radu Timofte, and Luc Van Gool. LSDIR: A large scale dataset for image restoration. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, pp. 1775–1787, 2023. doi: 10.1109/CVPRW59228.2023.00178.

Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee. Enhanced deep residual networks for single image super-resolution. In 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 1132–1140, 2017. doi: 10.1109/CVPRW.2017.151.

Lu Ling, Yichen Sheng, Zhi Tu, Wentian Zhao, Cheng Xin, Kun Wan, Lantao Yu, Qianyu Guo, Zixun Yu, Yawen Lu, et al. DL3DV-10K: A large-scale scene dataset for deep learning-based 3d vision. In IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2024.

Guan-Horng Liu, Arash Vahdat, De-An Huang, Evangelos A. Theodorou, Weili Nie, and Anima Anandkumar. I^2SB: Image-to-image schrödinger bridge. In International Conference on Machine Learning (ICML), 2023.

Ben Mildenhall, Pratul P. Srinivasan, Rodrigo Ortiz-Cayon, Nima Khademi Kalantari, Ravi Ramamoorthi, Ren Ng, and Abhishek Kar. Local light field fusion: Practical view synthesis with prescriptive sampling guidelines. ACM Transactions on Graphics, 38(4), 2019.

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

Anish Mittal, Rajiv Soundararajan, and Alan C Bovik. Making a "completely blind" image quality analyzer. IEEE Signal processing letters, 20(3):209–212, 2013. doi: 10.1109/LSP.2012.2227726.

Yuan Shen, Duygu Ceylan, Paul Guerrero, Zexiang Xu, Niloy J. Mitra, Shenlong Wang, and Anna Frühstück. SuperGaussian: Repurposing video models for 3d super resolution. In European Conference on Computer Vision (ECCV), 2024.

Yuyang Shi, Valentin De Bortoli, Andrew Campbell, and Arnaud Doucet. Diffusion schrödinger bridge matching. In Advances in Neural Information Processing Systems (NeurIPS), 2023.

Chen Wang, Xian Wu, Yuan-Chen Guo, Song-Hai Zhang, Yu-Wing Tai, and Shi-Min Hu. NeRF-SR: High quality neural radiance fields using supersampling. In Proceedings of the 30th ACM International Conference on Multimedia, pp. 6445–6454. ACM, 2022. doi: 10.1145/3503161.3547808.

Jianyi Wang, Zongsheng Yue, Shangchen Zhou, Kelvin C.K. Chan, and Chen Change Loy. Exploiting diffusion prior for real-world image super-resolution. International Journal of Computer Vision, 2024.

Peihao Wang, Yuehao Wang, Dilin Wang, Sreyas Mohan, Zhiwen Fan, Lemeng Wu, Ruisi Cai, Yu-Ying Yeh, Zhangyang Wang, Qiang Liu, and Rakesh Ranjan. Steepest descent density control for compact 3d gaussian splatting. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 26663–26672, 2025.

Xintao Wang, Liangbin Xie, Chao Dong, and Ying Shan. Real-ESRGAN: Training real-world blind super-resolution with pure synthetic data. In IEEE/CVF International Conference on Computer Vision Workshops (ICCVW), 2021.

Rongyuan Wu, Tao Yang, Lingchen Sun, Zhengqiang Zhang, Shuai Li, and Lei Zhang. SeeSR: Towards semantics-aware real-world image super-resolution. In IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2024.

Zongxin Ye, Wenyu Li, Sidun Liu, Peng Qiao, and Yong Dou. AbsGS: Recovering fine details for 3d gaussian splatting, 2024.

Xiqian Yu, Hanxin Zhu, Tianyu He, and Zhibo Chen. GaussianSR: 3d gaussian super-resolution with 2d diffusion priors. arXiv preprint arXiv:2406.10111, 2024a. URL https://arxiv.org/abs/2406.10111.

Zehao Yu, Anpei Chen, Binbin Huang, Torsten Sattler, and Andreas Geiger. Mip-splatting: Alias-free 3d gaussian splatting. In 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 19447–19456. IEEE, 2024b.

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

Zheng Zhang, Wenbo Hu, Yixing Lao, Tong He, and Hengshuang Zhao. Pixel-GS: Density control with pixel-aware gradient for 3d gaussian splatting. In European Conference on Computer Vision (ECCV), 2024.

Linqi Zhou, Aaron Lou, Samar Khanna, and Stefano Ermon. Denoising diffusion bridge models. In International Conference on Learning Representations (ICLR), 2024.

Zheng Zhou, Yu-Jie Xiong, Jia-Chen Zhang, Chun-Ming Xia, Xihe Qiu, and Hongjian Zhan. Gradient-direction-aware density control for 3d gaussian splatting. In The Fourteenth International Conference on Learning Representations, 2026. URL https://openreview.net/forum?id=6qDxK4Gz7F.

A APPENDIX

This appendix provides additional methodological details in Appendix B, implementation details in Appendix C, supplementary experimental results in Appendix D, qualitative ablation studies in Appendix E, and additional qualitative comparisons in Appendix F.

B ADDITIONAL METHOD DETAILS

We provide a detailed description of the multi-view latent bridge SR model, trajectory distillation, and detail-aware Gaussian splitting, supplementing Sec. 3.2. Additional training settings and hyperparameters are provided separately in Appendix C.

B.1 MULTI-VIEW LATENT BRIDGE SR MODEL

LR-to-HR latent bridge. Given an LR observation I_v^{LR}, we first bicubically upsample it to the target resolution and encode it as the source endpoint:

$$z_v^{\mathrm{src}} = E\big(\mathcal{F}_{\uparrow s}(I_v^{\mathrm{LR}})\big), \tag{B.1}$$

where $\mathcal{F}_{\uparrow s}(\cdot)$ denotes bicubic upsampling by a factor of s, and $E(\cdot)$ is the frozen VAE encoder. During teacher training, the corresponding HR image defines the target endpoint:

$$z_v^{\mathrm{tgt}} = E(I_v^{\mathrm{HR}}). \tag{B.2}$$

Following latent bridge matching (Chadebec et al., 2025), we construct an intermediate bridge state for $t \in [0, 1]$ as

$$z_{t,v} = (1-t)z_v^{\mathrm{src}} + t z_v^{\mathrm{tgt}} + \eta\sqrt{t(1-t)}\,\epsilon, \quad \epsilon \sim \mathcal{N}(0, I), \tag{B.3}$$

where η controls the bridge stochasticity. The perturbation vanishes at both endpoints, such that $t = 0$ corresponds to the LR source and $t = 1$ corresponds to the HR target. Unlike conventional diffusion-based restoration, in which the generative state is initialized from Gaussian noise and the LR image is introduced primarily as a condition, the bridge starts directly from the encoded LR observation and transports its content toward the HR distribution.

Conditioned on the target endpoint, the drift of the corresponding Brownian bridge is defined as follows:

$$v^*(z_{t,v}, t; z_v^{\mathrm{tgt}}) = \frac{z_v^{\mathrm{tgt}} - z_{t,v}}{1-t} \qquad 0 \le t < 1. \tag{B.4}$$

We instantiate the bridge velocity field using Lens (Chen et al., 2026), a pretrained text-to-image model consisting of a frozen autoencoder and a multimodal diffusion-Transformer backbone. Rather than fine-tuning the entire model, we insert low-rank adaptation (LoRA) modules (Hu et al., 2022) into the Transformer and train these modules on generic 2D super-resolution data. Given the intermediate state $z_{t,v}$, the single-view bridge predicts $v_\theta(z_{t,v}, t, c_v)$, where θ denotes the SR bridge parameters and c_v denotes the semantic condition extracted from the LR observation. Following SeeSR (Wu et al., 2024), we use a degradation-aware prompt extractor to extract semantic tags from I_v^{LR}. These tags are inserted into the fixed prompt "a high-resolution, sharp, photorealistic photograph of [tags]". The resulting prompt is encoded as the semantic condition c_v using GPT-OSS-20B (Agarwal et al., 2025).

An important property of the bridge formulation is that the prediction at any intermediate state provides an estimate of the clean HR endpoint:

$$\widehat{z}_v^{\mathrm{tgt}} = z_{t,v} + (1-t)v_\theta(z_{t,v}, t, c_v). \tag{B.5}$$

The corresponding HR prediction is obtained by decoding the estimated target latent:

$$\widehat{I}_v^{\mathrm{HR}} = D(\widehat{z}_v^{\mathrm{tgt}}), \tag{B.6}$$

where D denotes the frozen VAE decoder. We optimize the SR LoRA using the bridge-matching loss, together with an auxiliary perceptual loss on the decoded target estimate:

$$\mathcal{L}_{\mathrm{SR}} = \mathcal{L}_{\mathrm{LBM}} + \lambda_p \mathcal{L}_{\mathrm{LPIPS}}\left(\widehat{I}_v^{\mathrm{HR}}, I_v^{\mathrm{HR}}\right), \tag{B.7}$$

We learn the Markovian drift $v_\theta(z_{t,v}, t, c_v)$ by minimizing the bridge-matching loss:

$$\mathcal{L}_{\mathrm{LBM}} = \mathbb{E}\left[\left\| v_\theta(z_{t,v}, t, c_v) - \frac{z_v^{\mathrm{tgt}} - z_{t,v}}{1-t} \right\|_2^2\right]. \tag{B.8}$$

This stage produces a single-view LR-to-HR latent bridge that preserves the image prior of the pretrained backbone while specializing it for image super-resolution.

Multi-view restoration. Although the single-view bridge provides an LR-anchored restoration process, it still predicts the restoration trajectory of each view independently. This is undesirable for 3D reconstruction because pose-nearest views usually observe overlapping scene content and therefore provide complementary evidence for resolving high-frequency details that are ambiguous in an individual LR image.

To incorporate such cross-view evidence directly into the restoration process, we extend the single-view bridge to a multi-view formulation. Given a group of G pose-nearest views

$$\mathcal{V}_g = \{v_1, v_2, \ldots, v_G\}, \tag{B.9}$$

their current bridge states are jointly processed:

$$\mathbf{Z}_t^g = \{z_{t,u} \mid u \in \mathcal{V}_g\}. \tag{B.10}$$

For a particular view v, we define its reference-view set as

$$\mathcal{N}_g(v) = \mathcal{V}_g \setminus \{v\}, \tag{B.11}$$

and we denote the states of this reference-view set as

$$\mathbf{Z}_{t,\mathcal{N}_g(v)}^g = \{z_{t,u} \mid u \in \mathcal{N}_g(v)\}. \tag{B.12}$$

The velocity prediction is then generalized from the single-view prediction to

$$v_{\theta,\phi}^{\mathrm{MV}}(z_{t,v}; \mathbf{Z}_{t,\mathcal{N}_g(v)}^g, t, c_v), \tag{B.13}$$

where ϕ denotes the parameters of the proposed cross-view adapter. Accordingly, the multi-view-aware HR endpoint estimate is given by

$$\hat{z}_v^{\mathrm{tgt}} = z_{t,v} + (1-t) v_{\theta,\phi}^{\mathrm{MV}}\left(z_{t,v}; \mathbf{Z}_{t,\mathcal{N}_g(v)}^g, t, c_v\right). \tag{B.14}$$

Different from a conventional single-view bridge, the restoration of view v is therefore conditioned not only on its own bridge state, but also on representations extracted from neighboring observations. As a result, ambiguous high-frequency details can be inferred jointly from multiple observations of the same scene content rather than independently for each view.

Cross-view adapter. We realize the multi-view interaction in Eq. (B.13) using a gated cross-view adapter in the Transformer blocks. Let

$$h_{\ell,v} \in \mathbb{R}^{L \times d}, \tag{B.15}$$

denote the L image tokens of view v entering the ℓ-th Transformer block, where d denotes the token dimension. The original Lens branch processes each view independently. In parallel, for each view v, we concatenate the token sequences of its reference-view set $\mathcal{N}_g(v)$ into a reference-token bank:

$$H_{\ell,\mathcal{N}_g(v)} = \mathrm{Concat}\left(\{h_{\ell,u}\}_{u \in \mathcal{N}_g(v)}\right) \in \mathbb{R}^{(G-1)L \times d}. \tag{B.16}$$

The tokens of view v then retrieve complementary information from the reference-token bank through multi-head cross-attention:

$$a_{\ell,v} = \mathrm{CrossAttn}_\ell\left(h_{\ell,v}, H_{\ell,\mathcal{N}_g(v)}\right), \tag{B.17}$$

where the tokens of view v provide the queries, while the reference-token bank provides the keys and values. The resulting cross-view features are incorporated into the original single-view branch through a gated residual connection:

$$h_{\ell,v}^{\mathrm{out}} = h_{\ell,v}^{\mathrm{base}} + \gamma_\ell a_{\ell,v}, \tag{B.18}$$

where $h_{\ell,v}^{\text{base}}$ denotes the output of the original Lens branch and γ_ℓ is a learnable scalar initialized to zero. This initialization ensures that the augmented block initially behaves identically to the pretrained single-view model, while allowing cross-view information to be gradually incorporated during training. The same operation is performed symmetrically for all $v \in \mathcal{V}_g$, such that each view preserves its own representation while selectively aggregating complementary information from neighboring observations.

Since pose-nearest views typically share substantial scene content, the cross-view adapter provides additional information for restoring high-quality details. Consequently, the high-frequency content is inferred jointly across related views, promoting more consistent pseudo-HR predictions before they are distilled into the shared 3D representation.

Training strategy. We train the proposed multi-view latent bridge SR model in two stages. First, we freeze the Lens backbone and VAE and optimize only the SR LoRA on generic 2D super-resolution data, which leads to a single-view LR-to-HR restoration prior. Second, we freeze both the pretrained backbone and the learned SR LoRA and optimize only the cross-view adapter on multi-view training data. This decoupled training strategy preserves the pretrained image prior while progressively specializing the model for image super-resolution and cross-view information exchange.

During 3D reconstruction, the complete multi-view latent bridge SR model is kept frozen and serves as the teacher for trajectory distillation. At each trajectory-distillation stage, it jointly processes the persistent bridge states of pose-nearest views and produces the corresponding multi-view-aware HR endpoint estimates. These predictions provide the stage-wise pseudo-HR supervision used to progressively optimize the 3D Gaussian representation, as described in Sec. B.2.

B.2 Trajectory Distillation

Although the proposed multi-view latent bridge model provides cross-view-aware HR predictions, using only its final SR output as supervision would reduce the teacher to a fixed pseudo-label generator and discard the progressive restoration knowledge encoded along the LR-to-HR transport process. We therefore propose trajectory distillation, which couples a persistent bridge trajectory with successive stages of 3DGS optimization. Rather than independently restarting the SR process at each stage, we maintain one bridge state for every input view and progressively advance these states from the LR source endpoint toward the HR target endpoint. At each stage, the current bridge states produce clean HR endpoint estimates for supervising 3DGS; the optimized 3D representation is subsequently re-rendered and fed back into the bridge-state evolution. In this way, a single LR-to-HR restoration trajectory is threaded through the evolving 3D representation, allowing the SR guidance and the reconstructed scene to progressively refine each other.

Initialization and stage-wise distillation. Before introducing generative HR supervision, we first initialize the Gaussian representation using only the observed LR images. Let

$$\widetilde{I}_v^{\text{LR}} = \mathcal{F}_{\uparrow s}(I_v^{\text{LR}}), \tag{B.19}$$

denote the LR observation bicubically upsampled to the target rendering resolution. The initial Gaussian representation $\mathcal{G}_0$ is optimized with

$$\mathcal{L}_{\text{init}}(\mathcal{G}) = \frac{1}{\mathcal{V}} \sum_{v=1}^{\mathcal{V}} \mathcal{L}_{\text{GS}} \left(\mathcal{R}(\mathcal{G}, P_v), \widetilde{I}_v^{\text{LR}} \right), \tag{B.20}$$

where $\mathcal{V}$ denotes the number of input views, and $\mathcal{L}_{\text{GS}}$ is the photometric loss defined in Eq. (1). $\mathcal{R}(\mathcal{G}, P_v)$ denotes the rendering of the Gaussian representation $\mathcal{G}$ from camera pose P_v. Minimizing this loss yields $\mathcal{G}_0$. This LR initialization establishes the scene geometry and appearance from the observed content before the missing high-frequency information is introduced by the SR teacher.

We then define an increasing bridge-time schedule

$$0 = t_1 < t_2 < \cdots < t_{K+1} = 1, \tag{B.21}$$

where K denotes the number of trajectory-distillation stages. At stage k, we denote the bridge state at time t_k by $z_{k,v}$. For every view v, we maintain a persistent bridge state initialized at the LR source endpoint, which is defined as follows:

$$z_{1,v} = z_v^{\text{src}}. \tag{B.22}$$

Different from independently restarting the bridge sampling process at every 3D optimization stage, the state $z_{k+1,v}$ is evolved from $z_{k,v}$ and retained for the subsequent stage. Hence, the K stages collectively follow one persistent LR-to-HR restoration trajectory.

At the k-th trajectory-distillation stage, the multi-view latent bridge teacher jointly processes the current states of a group of pose-nearest views. The sampling step returns the updated bridge state $z_{k+1,v}$ together with a clean HR endpoint estimate. Following Eq. (B.14), this estimate is computed from the input state $z_{k,v}$ at time t_k:

$$\widehat{z}^{\mathrm{HR}}_{k,v} = z_{k,v} + (1 - t_k) v^{\mathrm{MV}}_{\theta,\phi}\left(z_{k,v}; \mathbf{Z}^{g}_{t_k,\mathcal{N}_g(v)}, t_k, c_v\right). \tag{B.23}$$

The estimated latent is then decoded into the stage-wise pseudo-HR target:

$$\widehat{I}^{\mathrm{HR}}_{k,v} = D\big(\widehat{z}^{\mathrm{HR}}_{k,v}\big). \tag{B.24}$$

Importantly, $\widehat{I}^{\mathrm{HR}}_{k,v}$ is the teacher's estimate of the clean HR endpoint from the current bridge state, rather than the intermediate bridge state itself. Each stage therefore supervises 3DGS with an HR endpoint prediction, while the bridge states used to obtain these predictions follow one persistent LR-to-HR trajectory.

The pseudo-HR predictions provide the missing high-frequency supervision, but the generated details should remain consistent with the captured LR scene. We therefore optimize 3DGS using both the stage-wise pseudo-HR targets and the original LR observations. The pseudo-HR loss is defined as

$$\mathcal{L}^{\mathrm{HR}}_{k,v}(\mathcal{G}) = \mathcal{L}_{\mathrm{GS}}\left(\mathcal{R}(\mathcal{G}, P_v), \widehat{I}^{\mathrm{HR}}_{k,v}\right), \tag{B.25}$$

and the LR-fidelity term is defined as

$$\mathcal{L}^{\mathrm{LR}}_{k,v}(\mathcal{G}) = \lambda_{\mathrm{LR}} \mathcal{L}_{\mathrm{GS}}\left(\mathcal{F}_{\downarrow s}\left(\mathcal{R}(\mathcal{G}, P_v)\right), I^{\mathrm{LR}}_{v}\right), \tag{B.26}$$

where $\mathcal{F}_{\downarrow s}$ denotes anti-aliased downsampling by a factor of s, and λ_{LR} controls the LR-fidelity term.

The per-view distillation loss is

$$\mathcal{L}_{k,v}(\mathcal{G}) = \mathcal{L}^{\mathrm{HR}}_{k,v}(\mathcal{G}) + \mathcal{L}^{\mathrm{LR}}_{k,v}(\mathcal{G}), \tag{B.27}$$

and the scene-level loss can be written as

$$\mathcal{L}_k(\mathcal{G}) = \frac{1}{\mathcal{V}} \sum_{v=1}^{\mathcal{V}} \mathcal{L}_{k,v}(\mathcal{G}). \tag{B.28}$$

The pseudo-HR term transfers the high-frequency restoration knowledge predicted by the bridge teacher, whereas the LR-fidelity term anchors the optimized scene to the actual LR observations. In particular, the LR-fidelity term discourages the generative prior from introducing appearance or geometric changes that are inconsistent with the captured scene. Starting from $\mathcal{G}_{k-1}$, we minimize $\mathcal{L}_k(\mathcal{G})$ for a fixed number of iterations to obtain $\mathcal{G}_k$.

3D-guided trajectory evolution. A defining component of trajectory distillation is that the optimized 3D scene not only receives supervision from the bridge teacher, but also feeds its current reconstruction back into the subsequent restoration trajectory. After the k-th 3D optimization stage, we render the updated representation as

$$R_{k,v} = \mathcal{R}(\mathcal{G}_k, P_v), \tag{B.29}$$

and encode the rendering into the latent space with $E(R_{k,v})$. Recall that, under the increasing bridge-time convention, a target estimate $\widehat{z}^{\mathrm{tgt}}$ associated with a state $z_{k,v}$ at $t_k < 1$ implies the velocity defined as follows:

$$v^*(z_{k,v}, t_k; \widehat{z}^{\mathrm{tgt}}) = \frac{\widehat{z}^{\mathrm{tgt}} - z_{k,v}}{1 - t_k}. \tag{B.30}$$

For sampling steps $2 \le k \le K$, we use the encoded rendering from the previous distillation stage as the target estimate and define the bridge-state update velocity as follows:

$$\widetilde{v}_{k,v} = \frac{E(R_{k-1,v}) - z_{k,v}}{1 - t_k}. \tag{B.31}$$

The persistent bridge state is then updated from t_k to t_{k+1} using

$$z_{k+1,v} = z_{k,v} + (t_{k+1} - t_k)\widetilde{v}_{k,v}, \tag{B.32}$$

or equivalently,

$$z_{k+1,v} = z_{k,v} + (t_{k+1} - t_k)\frac{E(R_{k-1,v}) - z_{k,v}}{1 - t_k}. \tag{B.33}$$

Eq. (B.32) has an intuitive interpretation: the optimized 3D rendering is treated as the target content currently supported by the shared scene representation, and the next bridge state is updated toward this 3D-consistent content. The bridge trajectory is therefore no longer determined solely by 2D predictions; its evolution is progressively constrained by what can be coherently represented in 3D.

For the first update from t_1 to t_2, we use the velocity predicted by the multi-view bridge teacher:

$$z_{2,v} = z_{1,v} + (t_2 - t_1)v^{\mathrm{MV}}_{\theta,\phi}\left(z_{1,v}; \mathbf{Z}^{g}_{t_1,\mathcal{N}_g(v)}, t_1, c_v\right). \tag{B.34}$$

From the second distillation stage onward, the persistent states are updated using the 3D-guided rule in Eq. (B.32). During trajectory distillation, we update the bridge states deterministically without adding further noise. For $2 \le k \le K$, the k-th stage can therefore be summarized as the following recurrent process:

$$\left(z_{k,v}, R_{k-1,v}\right) \xrightarrow{\text{sampling}} \left(z_{k+1,v}, \widehat{I}^{\mathrm{HR}}_{k,v}\right) \xrightarrow{\text{3D distillation}} \mathcal{G}_k \xrightarrow{\text{render}} R_{k,v}. \tag{B.35}$$

Repeating this recursion over the stages threads a single persistent LR-to-HR bridge trajectory through the evolving 3D representation. As the bridge states progressively approach the HR endpoint, the teacher provides restoration supervision from progressively later stages of the LR-to-HR transport, while the reconstructed 3D scene feeds its cross-view-consistent content back into the subsequent bridge states. Consequently, trajectory distillation transfers not merely a fixed final SR result, but the progressive restoration knowledge encoded along the entire bridge trajectory into 3DGS. For $k < K$, the updated states are retained for the subsequent stage. After the final stage, we use $\mathcal{G}_K$ for HR novel-view rendering.

B.3 Detail-Aware Gaussian Splitting

Trajectory distillation progressively introduces high-frequency supervision that is absent from the original LR observations. Effectively representing these newly recovered details, however, requires not only optimizing the attributes of the existing Gaussians but also allocating sufficient representational capacity in the spatial directions where finer structures are required. In standard 3DGS densification, once a large Gaussian is selected for splitting, the positions of its children are randomly sampled from the spatial distribution of the parent:

$$\mu_i' \sim \mathcal{N}(\mu_i, \Sigma_i), \tag{B.36}$$

where μ_i and Σ_i denote the mean and covariance of the i-th Gaussian, respectively. Consequently, the actual refinement direction is determined by the parent geometry and random sampling, without explicitly considering where the SR supervision requires additional fine-scale representation. We therefore introduce detail-aware Gaussian splitting, which uses the discrepancy between pseudo-HR and LR supervision to determine how each candidate Gaussian should be refined.

Detail-specific splitting direction. During the k-th trajectory-distillation stage, the Gaussian representation is jointly optimized using the pseudo-HR target and the original LR observation. The two supervision terms for view v are defined in Eqs. (B.25) and (B.26). The LR term constrains scene content that is directly supported by the LR observations, whereas the pseudo-HR term additionally introduces high-frequency structures predicted by the SR teacher. We use their positional gradients to guide the splitting direction.

For a candidate Gaussian i visible at optimization iteration n, let $g^{\mathrm{HR}}_{i,n} = \frac{\partial \mathcal{L}^{\mathrm{HR}}_{k,v}}{\partial \mu_i}$ and $g^{\mathrm{LR}}_{i,n} = \frac{\partial \mathcal{L}^{\mathrm{LR}}_{k,v}}{\partial \mu_i}$ denote the positional gradients induced by the pseudo-HR and LR supervision, respectively. Directly using $g^{\mathrm{HR}}_{i,n}$ would mix the high-frequency detail introduced by the SR teacher with structures that

are already constrained by the LR observations. We instead remove from the pseudo-HR gradient its projection onto the LR-fidelity gradient:

$$q_{i,n} = g_{i,n}^{\mathrm{HR}} - \frac{\langle g_{i,n}^{\mathrm{HR}}, g_{i,n}^{\mathrm{LR}} \rangle}{\|g_{i,n}^{\mathrm{LR}}\|_2^2} g_{i,n}^{\mathrm{LR}}. \tag{B.37}$$

This projection gives

$$\langle q_{i,n}, g_{i,n}^{\mathrm{LR}} \rangle = 0, \tag{B.38}$$

such that $q_{i,n}$ retains the component of the pseudo-HR positional gradient orthogonal to the current LR-fidelity gradient. When $\|g_{i,n}^{\mathrm{LR}}\|_2^2 \leq 10^{-12}$, we skip this projection and set $q_{i,n} = g_{i,n}^{\mathrm{HR}}$. We refer to $q_{i,n}$ as a detail-specific gradient and use its accumulated squared projections to guide splitting-axis selection.

The gradient observed at a single optimization iteration may be noisy and view-dependent. We therefore aggregate the detail-specific gradients over the optimization interval between two consecutive densification operations. For Gaussian i, we accumulate their outer products as

$$M_i = \sum_{n \in \mathcal{T}_i} q_{i,n} q_{i,n}^\top \in \mathbb{R}^{3\times 3}, \tag{B.39}$$

where $\mathcal{T}_i$ denotes the set of iterations during which Gaussian i is visible since the previous densification step. We reset M_i to zero after each densification operation.

For an arbitrary unit direction d, Eq. (B.39) gives

$$d^\top M_i d = \sum_{n \in \mathcal{T}_i} \langle d, q_{i,n} \rangle^2. \tag{B.40}$$

Hence, $d^\top M_i d$ measures the accumulated strength of the detail-specific supervision along direction d. A large value indicates that the SR teacher repeatedly requests refinement along this direction.

Rather than introducing an arbitrary splitting direction, we evaluate this directional signal along the three principal axes of the Gaussian. Let

$$R_i = [d_{i,1} \quad d_{i,2} \quad d_{i,3}] \tag{B.41}$$

denote the rotation matrix of Gaussian i, where $d_{i,1}, d_{i,2}, d_{i,3}$ are its principal axes. We determine the splitting axis by

$$j_i^\star = \arg\max_{j \in \{1,2,3\}} d_{i,j}^\top M_i d_{i,j}, \tag{B.42}$$

and define

$$a_i = d_{i,j_i^\star}. \tag{B.43}$$

The selected axis therefore corresponds to the principal direction receiving the strongest accumulated detail-specific supervision. We use this signal to guide refinement of details that may not be fully constrained by the LR observations. When the total accumulated gradient energy is negligible, i.e., $\mathrm{tr}(M_i) \leq 10^{-16}$, we use the longest principal axis as the splitting direction, avoiding random direction selection.

Deterministic splitting. Given the selected direction a_i, we deterministically place two child Gaussians symmetrically around the parent. Let $s_{i,a}$ denote the scale of the parent Gaussian along a_i. The child centers are defined as

$$\mu_i' = \mu_i \pm \beta s_{i,a} a_i, \tag{B.44}$$

where β controls the displacement of the child Gaussians from the parent center. Meanwhile, the scale along the selected direction is reduced as

$$s_{i,a}' = \frac{s_{i,a}}{\kappa}, \qquad \kappa > 1, \tag{B.45}$$

while the scales along the remaining two principal axes remain unchanged.

To preserve the parent variance along the splitting direction, we determine the displacement factor β from the shrinkage factor κ. The variance of the equally weighted mixture of the two normalized child Gaussians along a_i is

$$\frac{s_{i,a}^2}{\kappa^2} + \beta^2 s_{i,a}^2. \tag{B.46}$$

Matching this quantity to the variance $s_{i,a}^2$ of the parent gives

$$\kappa^{-2} + \beta^2 = 1, \tag{B.47}$$

and therefore, we have

$$\beta = \sqrt{1 - \kappa^{-2}}. \tag{B.48}$$

This construction replaces one coarse Gaussian with two finer primitives while preserving its variance along the selected direction.

We initialize both children with the same opacity o_i', which should satisfy the condition:

$$(1 - o_i')^2 = 1 - o_i. \tag{B.49}$$

Solving for the child opacity gives

$$o_i' = 1 - \sqrt{1 - o_i}. \tag{B.50}$$

The remaining attributes are inherited from the parent. The child attributes are subsequently updated during 3DGS optimization.

Overall, the proposed splitting strategy preserves the standard densification criterion for selecting candidate Gaussians, but replaces the random placement of their children with a deterministic, detail-guided refinement. The detail-specific gradient in Eq. (B.37) provides a signal for identifying spatial refinements requested by the pseudo-HR supervision that may not be fully constrained by the LR observations. The accumulated matrix in Eq. (B.39) identifies the principal axis along which this signal is strongest, and Eqs. (B.44)–(B.50) define the attributes of the resulting child Gaussians. As a result, additional Gaussian capacity is allocated along directions indicated by the accumulated SR supervision, rather than being determined by random sampling. The proposed strategy introduces no learnable parameters.

C Additional Implementation Details

All experiments are conducted on NVIDIA A100 GPUs with 80 GB memory.

C.1 Multi-view Latent Bridge SR Model

We build the bridge SR teacher on the 3.8B-parameter Lens-Base model (Chen et al., 2026). For SR adaptation, we insert a rank-32 LoRA into the query, key, value, attention output, and feed-forward projections of all 48 Transformer blocks, resulting in approximately 80M trainable LoRA parameters. The cross-view adapter uses 8 attention heads with a dimension of 64 per head in each block.

We first freeze the Lens backbone and VAE and optimize the SR LoRA on generic 2D SR data. The SR LoRA is trained using 512×512 crops from LSDIR (Li et al., 2023), DIV2K (Agustsson & Timofte, 2017), Flickr2K (Lim et al., 2017), DIV8K (Gu et al., 2019), and FFHQ (Karras et al., 2019), with $\times4$ Real-ESRGAN degradations (Wang et al., 2021) and a batch size of 4 images per GPU. We set the bridge stochasticity to $\eta = 5\times10^{-3}$, sample t uniformly from $\{0, 0.25, 0.5, 0.75\}$, and use $\lambda_p = 10$ in Eq. (8).

After SR adaptation, we freeze the Lens backbone and the learned SR LoRA and train only the cross-view adapter on 512×512 crops from the DL3DV-Benchmark (Ling et al., 2024). We use 14 scenes for evaluation and the remaining scenes for cross-view adapter training. Evaluation results are reported in Table 1 and Figs. 4 and 11. We initialize the learnable scalar γ_ℓ in Eq. (7) to zero. This initialization preserves the behavior of the single-view SR model at the start of adapter training while allowing cross-view information to be gradually incorporated during training. We train the cross-view adapter with a batch size of one group per GPU, with $G = 4$ views per group. We use AdamW with a learning rate of 5×10^{-5} in both training stages. The SR LoRA and the cross-view adapter are trained for $15{,}500$ and $30{,}000$ steps, respectively.

Table 3: Quantitative results on Tanks & Temples (Knapitsch et al., 2017) under the $\times 4 \rightarrow \times 1$ setting. The best, second-best, and third-best results are highlighted in red, orange, and yellow, respectively.

Methods	PSNR↑	SSIM↑	LPIPS↓	NIQE↓	DISTS↓	FID↓	DreamSim↓	CMMD↓
3DGS (LR)	21.51	0.6860	0.3893	6.548	0.2115	43.85	0.0579	1.160
3DGS+StableSR	21.29	0.7136	0.2245	4.032	0.1401	35.28	0.0573	1.023
Mip-Splatting	22.28	0.7422	0.2764	4.681	0.1545	25.34	0.0405	1.162
SRGS	22.95	0.7844	0.1843	4.404	0.1271	25.16	0.0458	1.360
3DSR	22.53	0.7520	0.2094	4.131	0.1395	28.84	0.0477	1.295
SplatSuRe	22.88	0.7677	0.1613	3.390	0.1150	20.68	0.0350	1.096
LBTD (Ours)	22.94	0.7712	0.1853	4.017	0.1235	21.52	0.0364	0.937

C.2 3DGS Optimization

During scene reconstruction, both the SR bridge parameters θ and the cross-view adapter parameters ϕ remain frozen. For each scene, we optimize the Gaussian representation for 50K iterations, consisting of 30K iterations for LR initialization followed by $K = 4$ trajectory-distillation stages of 5K iterations each. We use the bridge schedule $\{t_k\}_{k=1}^{K+1} = \{0, 0.25, 0.5, 0.75, 1\}$ and process groups of $G = 4$ views. We set $\lambda_{\mathrm{LR}} = 1.0$ in Eq. (10). For detail-aware Gaussian splitting, we set the shrinkage factor to $\kappa = 1.6$ and the size threshold to $\tau_{\mathrm{pd}} = 0.005$. The displacement factor β is determined by Eq. (B.48).

View grouping. For each scene, we construct the view groups once from the input camera poses and reuse them throughout trajectory distillation. For views v and u, we define the pose distance as

$$d(v, u) = w_{\mathrm{t}} \|\mathbf{c}_v - \mathbf{c}_u\|_2 + w_{\mathrm{r}} \arccos\big(2\langle \mathbf{r}_v, \mathbf{r}_u \rangle^2 - 1\big), \tag{C.1}$$

where $\mathbf{c}_v$ and $\mathbf{r}_v$ denote the camera center and unit rotation quaternion, respectively. The weights w_{t} and w_{r} control the contributions of translation and rotation to the pose distance. We set $w_{\mathrm{t}} = w_{\mathrm{r}} = 1$. For numerical stability, we clip the argument of $\arccos$ to $[-1, 1]$.

At each step, we select the unassigned view with the smallest index in the input camera list, add up to $G - 1$ nearest unassigned views to this view according to $d(v, u)$, and remove these views from the unassigned set. We repeat this process until every view has been assigned to one group. If the final group contains fewer than G assigned views, we complete its input by borrowing views from other groups as context. Only predictions for the views assigned to that group are retained; predictions for borrowed context views are discarded.

D Additional Experimental Results

D.1 Results on Tanks & Temples

Table 3 reports quantitative results on the Truck and Train scenes of Tanks & Temples (Knapitsch et al., 2017) under the $\times 4 \rightarrow \times 1$ setting. LBTD outperforms 3DSR across all eight metrics, improving PSNR from 22.53 to 22.94 and reducing FID from 28.84 to 21.52. Fig. 5 provides the corresponding qualitative comparisons. Since this evaluation contains only two scenes, we regard these results as complementary evidence to the larger benchmarks.

D.2 Number of Gaussians and Rendering Speed

Table 4 summarizes the average number of Gaussians per scene across all four datasets. Compared with 3DSR (Chen et al., 2025), LBTD uses approximately 13.6%–27.1% fewer Gaussians on average. Tables 5 and 6 report the number of Gaussians and rendering speed for individual scenes on Mip-NeRF 360 (Barron et al., 2022) and DL3DV (Ling et al., 2024), respectively. Compared with 3DSR (Chen et al., 2025), LBTD uses fewer Gaussians in every evaluated scene of both datasets. On Mip-NeRF 360, the average number of Gaussians decreases from 7.374M to 6.374M, while the



Figure 5: Qualitative comparison on Tanks & Temples (Knapitsch et al., 2017) under the $\times 4 \rightarrow \times 1$ setting. LBTD recovers fine scene structures and coherent high-frequency details while reducing distortions produced by competing SR-based methods.

Table 4: Number of Gaussians compared with 3DSR (Chen et al., 2025). #G denotes the average number of Gaussians per scene in millions. The lower count within each dataset is highlighted in **bold**.

	3DSR	LBTD (Ours)
Dataset	#G (M)↓	#G (M)↓
Mip-NeRF 360	7.374	**6.374**
DL3DV	2.183	**1.805**
LLFF	2.047	**1.572**
Tanks & Temples	2.154	**1.571**

Table 5: Number of Gaussians and rendering speed on Mip-NeRF 360 (Barron et al., 2022) under the $\times 8 \rightarrow \times 2$ setting. #G denotes the number of Gaussians in millions, and FPS denotes frames per second. The better result within each row and metric is highlighted in **bold**.

	3DSR		LBTD (Ours)	
Scene	#G (M)↓	FPS↑	#G (M)↓	FPS↑
bicycle	15.805	22.73	**13.880**	**26.05**
bonsai	3.019	97.00	**2.320**	**112.45**
counter	1.975	96.43	**1.695**	**118.68**
flowers	8.440	36.07	**6.975**	**41.25**
garden	7.144	35.57	**6.453**	**37.60**
kitchen	2.556	**98.37**	**2.322**	97.35
room	2.913	90.85	**2.436**	**107.94**
stump	12.303	33.96	**10.186**	**38.06**
treehill	12.208	30.33	**11.097**	**33.33**
Mean	7.374	60.15	**6.374**	**68.08**

mean rendering speed increases from 60.15 to 68.08 FPS. On DL3DV, the average number of Gaussians decreases from 2.183M to 1.805M, while the mean rendering speed increases from 65.60 to 69.51 FPS.

D.3 Multi-view Consistency Evaluation

We evaluate multi-view consistency at two levels: SR outputs and final novel-view renderings, by measuring MEt3R (Asim et al., 2025) and E_{warp} (Lai et al., 2018).

Table 6: Number of Gaussians and rendering speed on DL3DV (Ling et al., 2024) under the $\times 8 \rightarrow \times 2$ setting. #G denotes the number of Gaussians in millions, and FPS denotes frames per second. The better result within each row and metric is highlighted in **bold**.

	3DSR		LBTD (Ours)	
Scene	#G (M)↓	FPS↑	#G (M)↓	FPS↑
1da888bd	1.309	84.76	**1.070**	**92.02**
26fd2335	2.032	**60.12**	**1.715**	59.92
280abf7b	1.283	84.33	**1.113**	**86.39**
2f3e1c0f	2.599	**52.42**	**2.230**	43.66
484c0aca	1.084	78.62	**0.899**	**87.12**
513e4ea2	2.733	**56.46**	**2.326**	46.65
5f0041e5	1.305	71.18	**1.026**	**94.99**
946f49be	2.515	50.56	**2.111**	**52.83**
9e9a89ae	2.014	84.91	**1.602**	**94.29**
adf35184	1.300	69.62	**1.025**	**74.54**
b5faa2a8	2.849	53.35	**2.345**	**53.54**
b6d1134c	3.593	44.38	**3.102**	**52.52**
e5684b32	3.084	67.16	**2.484**	**73.37**
ec305787	2.867	60.57	**2.220**	**61.26**
Mean	2.183	65.60	**1.805**	**69.51**

Table 7: Multi-view consistency of SR outputs on DL3DV-Benchmark (Ling et al., 2024). The best result for each metric is highlighted in **bold**.

Methods	MEt3R↓	E_{warp} (10^{-3})↓
StableSR	0.5724	4.582
Single-View Latent Bridge SR	0.5503	2.973
Multi-View Latent Bridge SR (Ours)	**0.5403**	**2.881**

Table 8: Multi-view consistency of novel-view renderings on DL3DV-Benchmark (Ling et al., 2024). StableSR is applied independently to the renderings of a 3DGS model trained on LR images. The best result for each metric is highlighted in **bold**.

Methods	MEt3R↓	E_{warp} (10^{-3})↓
StableSR on 3DGS renders	0.6167	26.06
3DSR	0.5979	22.03
LBTD (Ours)	**0.5969**	**20.59**

Consistency of SR outputs. Table 7 compares the multi-view consistency of SR outputs from StableSR and our single-view and multi-view latent bridge SR models. Compared with StableSR, the single-view model reduces MEt3R from 0.5724 to 0.5503 and E_{warp} from 4.582×10^{-3} to 2.973×10^{-3}. Adding the cross-view adapter further reduces these metrics to 0.5403 and 2.881×10^{-3}, respectively. These results indicate that cross-view interaction further improves the consistency of the restored views.

Consistency of novel-view outputs. Table 8 compares LBTD with StableSR applied to 3DGS renderings and with 3DSR. LBTD achieves lower MEt3R and E_{warp} than both methods, indicating improved consistency across novel-view renderings.

D.4 SR Models and Trajectory Distillation

Fig. 6 compares StableSR and the multi-view latent bridge SR model with and without trajectory distillation. For StableSR, trajectory distillation improves NIQE and DISTS, but worsens the other

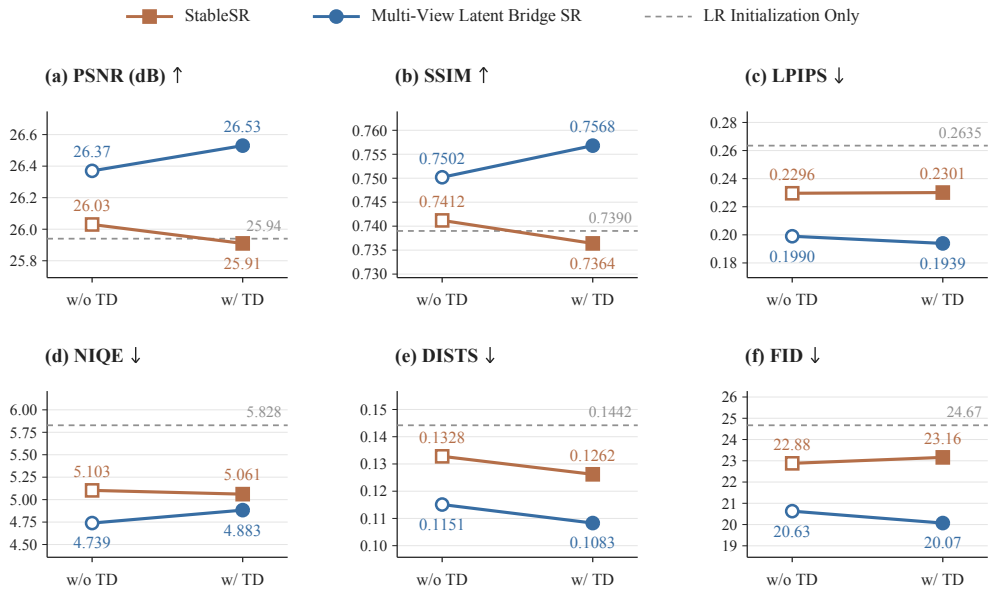

Figure 6: Quantitative comparison of SR configurations and trajectory distillation on Mip-NeRF 360 (Barron et al., 2022). TD denotes trajectory distillation. Hollow and filled markers indicate results without and with TD, respectively.

four metrics. For the multi-view latent bridge SR model, trajectory distillation increases PSNR from 26.37 to 26.53 and reduces LPIPS from 0.1990 to 0.1939 and FID from 20.63 to 20.07. Unlike conventional diffusion-based restoration, in which the generative state is initialized from Gaussian noise and the LR image is introduced primarily as a condition, the bridge starts directly from the encoded LR observation and transports its content toward the HR distribution. These results support our motivation for an LR-anchored, multi-view restoration trajectory that co-evolves with the 3D representation.

D.5 Additional Ablation Studies on Trajectory Distillation

Table 9 compares configurations that differ in the initial bridge source, the source of the sampling state in later stages, the number of SR sampling steps per stage, and rendering feedback. We use the same frozen multi-view latent bridge SR model and cross-view adapter. These experiments complement the SR-model comparison in Appendix D.4. We evaluate on the Mip-NeRF 360 dataset (Barron et al., 2022) under the $\times 8 \rightarrow \times 2$ setting. All variants use the same Gaussian initialization, losses, optimizer, and 4-stage optimization schedule, with 5K Gaussian updates per stage and the same detail-aware splitting. The results of variant (b) and LBTD are reused from Table 2(b). In Table 9, Initial Source refers to the source used to construct the first bridge state. Later-stage State Source specifies whether subsequent stages restart from LR observations or continue from the bridge state retained from the preceding stage. Rendering Feedback indicates whether encoded 3D renderings are used as target estimates for bridge-state updates.

Variant (a) performs complete four-step bridge sampling once from the LR observations and uses the final pseudo-HR targets throughout all Gaussian-optimization stages. Variant (b) corresponds to "No trajectory distillation" in Table 2(b). At each stage, it restarts complete 4-step bridge sampling from the original LR observations, and the clean HR prediction supervises the subsequent 5K Gaussian updates. The scene is rendered at each stage of the bridge trajectory and the rendering is used to update the bridge state, including the first sampling step of the first stage. The bridge state is discarded before the next stage. Variant (c) retains the persistent bridge trajectory and stage-wise pseudo-HR supervision, but without rendering feedback. Variant (d) retains the persistent trajectory and rendering feedback of LBTD, but initializes the bridge state from the rendering of the LR-initialized 3DGS representation.

Table 9: Additional ablation studies on trajectory distillation on the Mip-NeRF 360 dataset (Barron et al., 2022) under the $\times 8 \rightarrow \times 2$ setting. The best result for each metric is highlighted in **bold**.

Variant	Initial Source	Later-stage State Source	SR Steps per Stage	Rendering Feedback	PSNR↑	SSIM↑	LPIPS↓	NIQE↓	DISTS↓	FID↓
(a)	LR	–	$(4, 0, 0, 0)$	w/o	26.41	0.7520	**0.1910**	4.792	**0.1051**	20.38
(b)	LR	LR	$(4, 4, 4, 4)$	w/	26.37	0.7502	0.1990	4.739	0.1151	20.63
(c)	LR	Retained state	$(1, 1, 1, 1)$	w/o	26.43	0.7526	0.1918	4.817	0.1055	20.58
(d)	Render	Retained state	$(1, 1, 1, 1)$	w/	23.96	0.6548	0.2449	**3.594**	0.1761	23.95
LBTD (Ours)	LR	Retained state	$(1, 1, 1, 1)$	w/	**26.53**	**0.7568**	0.1939	4.883	0.1083	**20.07**

Results and analysis. Table 9 shows that LBTD achieves the best PSNR, SSIM, and FID among the evaluated variants. Compared with variant (a), LBTD improves PSNR from 26.41 to 26.53 and reduces FID from 20.38 to 20.07. Variant (c) slightly improves PSNR and SSIM over variant (a). Adding rendering feedback to variant (c) further increases PSNR from 26.43 to 26.53 and SSIM from 0.7526 to 0.7568, while reducing FID from 20.58 to 20.07. The comparison between variant (c) and LBTD shows improved reconstruction fidelity and FID with rendering feedback. Compared with variant (b), LBTD improves five of the six reported metrics. Variant (d) reduces PSNR from 26.53 to 23.96 and increases FID from 20.07 to 23.95 compared with LBTD. This comparison supports using the observed LR images as the initial source for the bridge.

E Qualitative Ablation Studies

We provide qualitative comparisons for the three components evaluated in Table 2: multi-view latent bridge SR, trajectory distillation, and detail-aware Gaussian splitting. All comparisons use Mip-NeRF 360 under the $\times 8 \rightarrow \times 2$ setting.

Multi-view latent bridge SR. Fig. 7 complements Table 2(a). We compare LR initialization only, single-view latent bridge SR, and multi-view latent bridge SR with the cross-view adapter. The qualitative results show the effect of introducing the SR prior and cross-view interaction on the reconstructed image details.

Trajectory distillation. Fig. 8 complements Table 2(b) by comparing the variant without trajectory distillation and variants with different numbers of distillation stages. The variant without trajectory distillation uses the same total number of Gaussian-optimization steps as our full configuration, but performs four-step sampling from the original LR observations at each stage without retaining the bridge state.

Detail-aware Gaussian splitting. Fig. 9 complements Table 2(c). We compare standard splitting with the proposed detail-aware Gaussian splitting to examine their effects on the recovered fine details. The qualitative results show our method recovers finer details on the packaging in the Counter scene.

F Additional Qualitative Comparisons

Figs. 10 and 11 provide additional qualitative comparisons on Mip-NeRF 360 and DL3DV, supplementing Figs. 3 and 4 in the main paper. Fig. 12 presents qualitative comparisons on LLFF. We compare our method with 3DGS (LR) (Kerbl et al., 2023), SRGS (Feng et al., 2024), 3DSR (Chen et al., 2025), and SplatSuRe (Asthana et al., 2026).

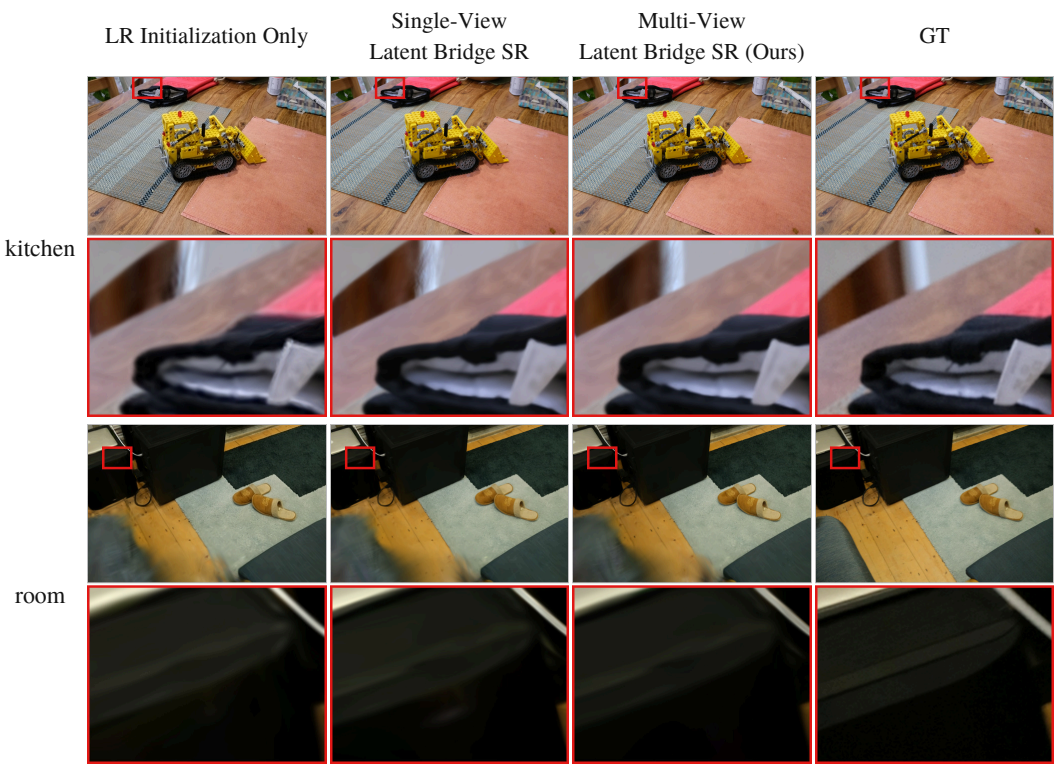

Figure 7: Qualitative ablation study of multi-view latent bridge SR on Mip-NeRF 360 (Barron et al., 2022) under the $\times 8 \rightarrow \times 2$ setting. From left to right, we show LR initialization only, single-view latent bridge SR, multi-view latent bridge SR (Ours), and the ground truth (GT).

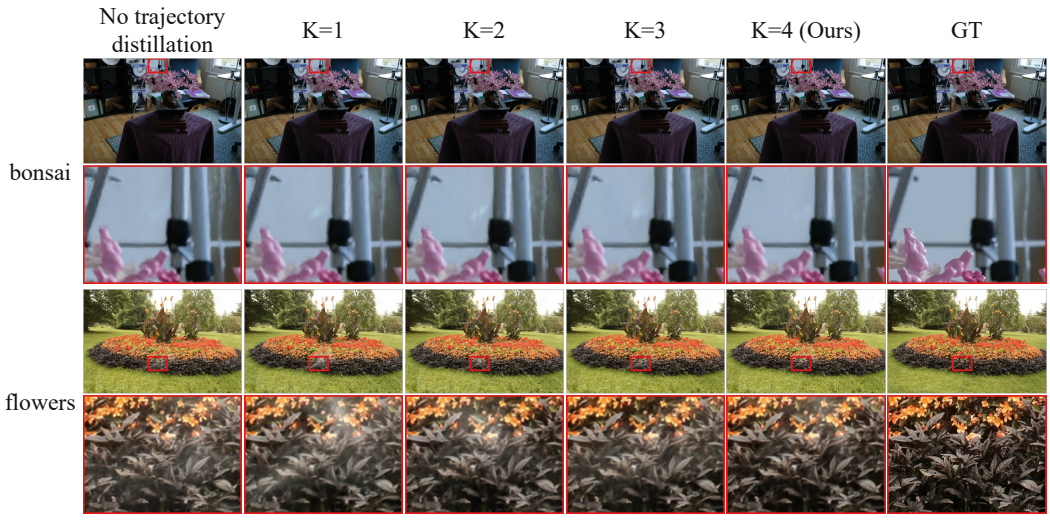

Figure 8: Qualitative ablation study of trajectory distillation on Mip-NeRF 360 (Barron et al., 2022) under the $\times 8 \rightarrow \times 2$ setting. From left to right, we show the variant without trajectory distillation, variants with $K = 1$, $K = 2$, $K = 3$, and $K = 4$ (Ours), and the ground truth (GT).

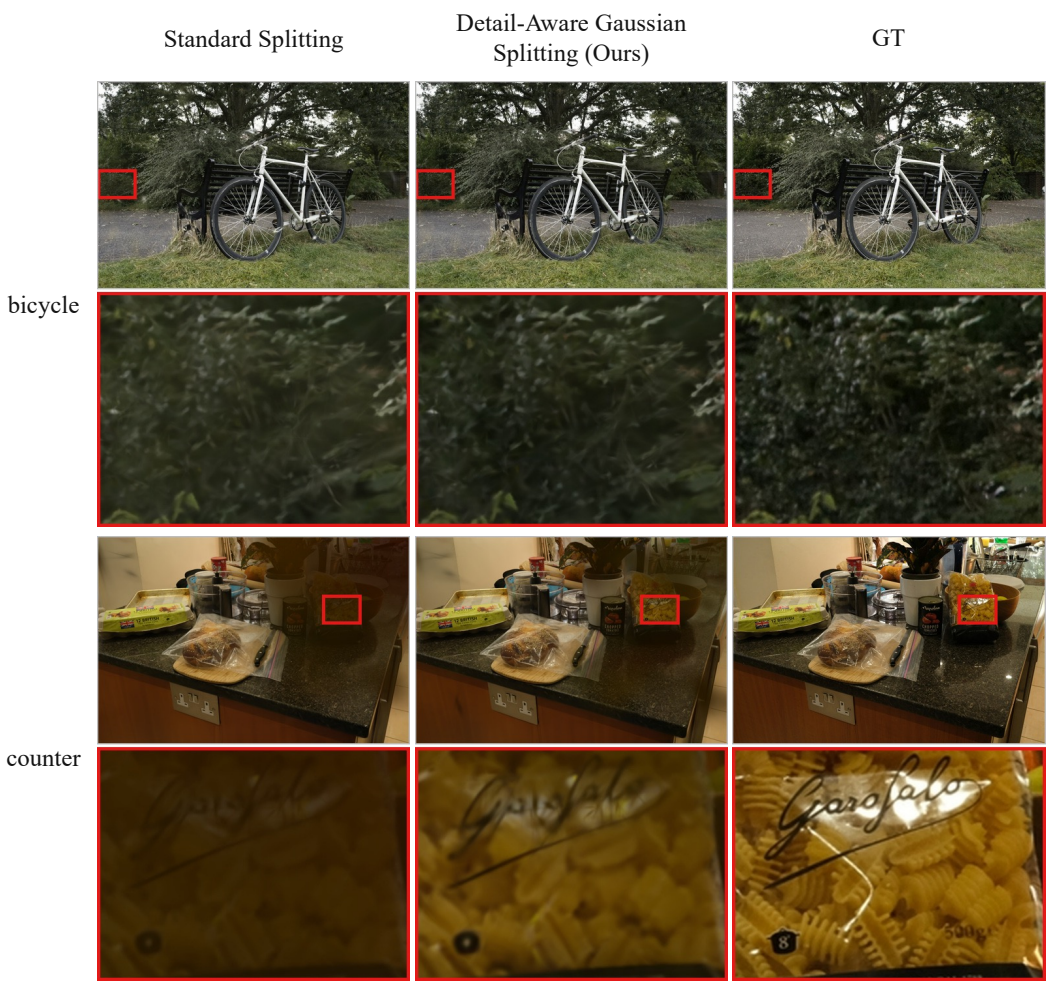

Figure 9: Qualitative ablation study of detail-aware Gaussian splitting on Mip-NeRF 360 (Barron et al., 2022) under the $\times 8 \rightarrow \times 2$ setting. From left to right, we show standard splitting, detail-aware Gaussian splitting (Ours), and the ground truth (GT).

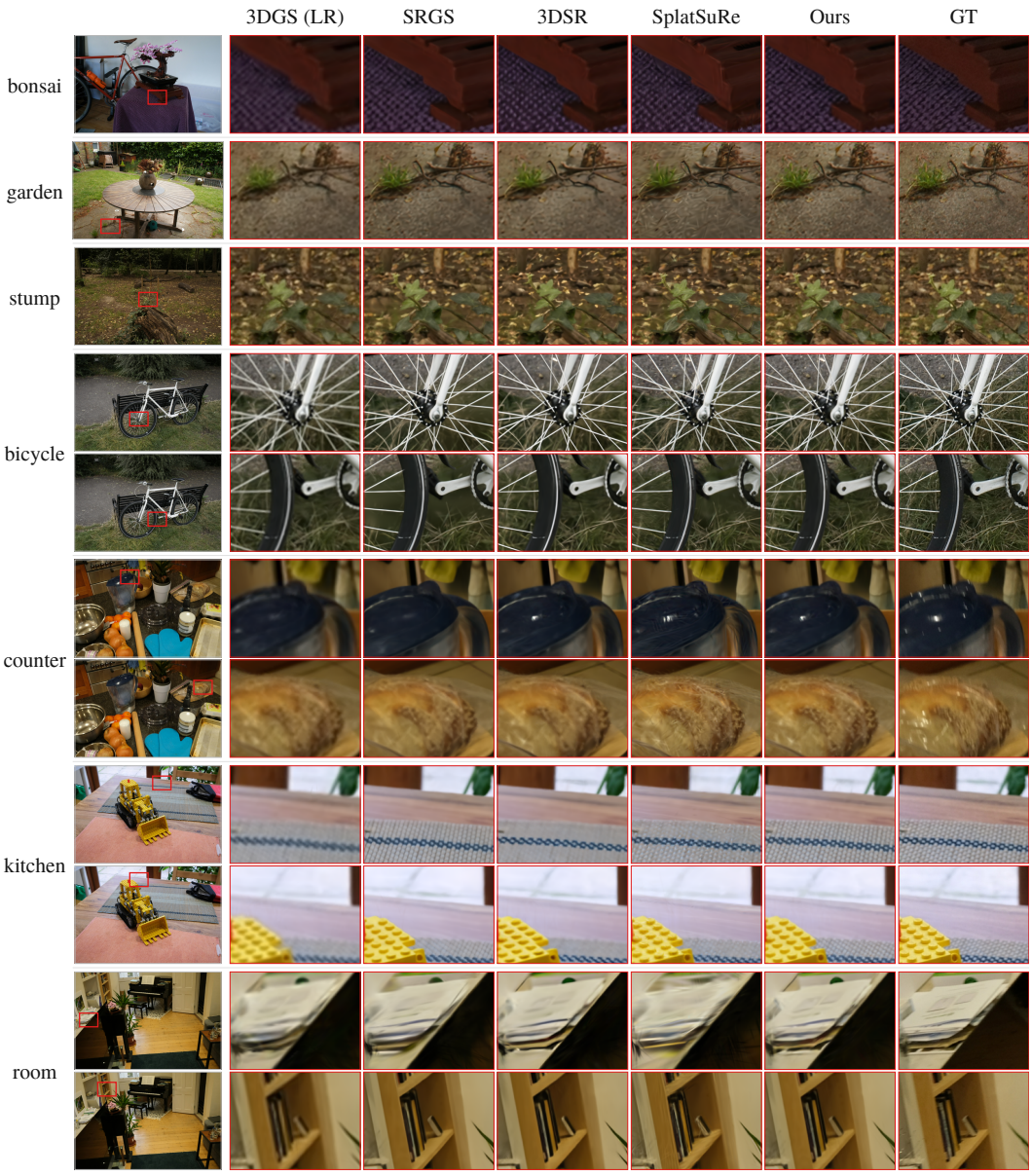

Figure 10: Additional qualitative comparison on Mip-NeRF 360 (Barron et al., 2022) under the $\times 8 \rightarrow \times 2$ setting.

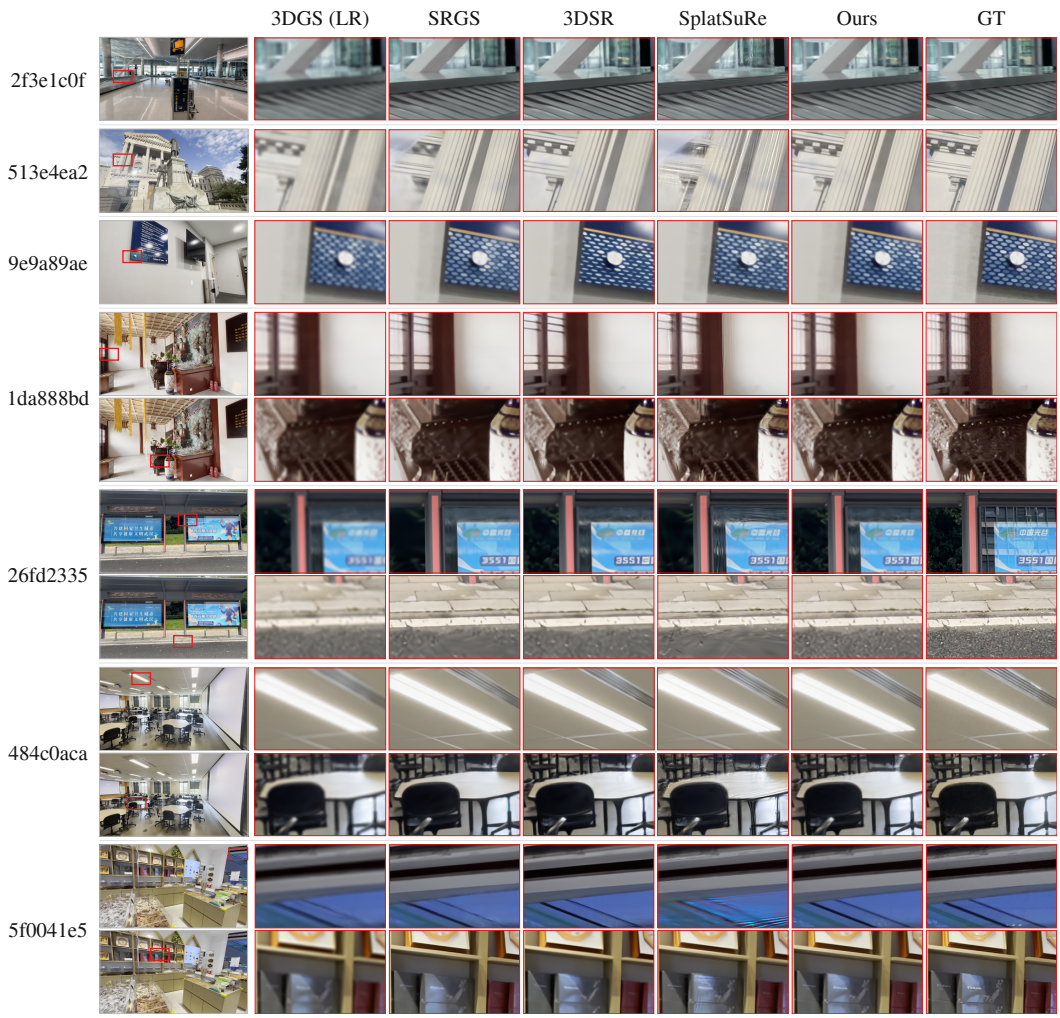

Figure 11: Additional qualitative comparison on DL3DV (Ling et al., 2024) under the $\times 8 \rightarrow \times 2$ setting.

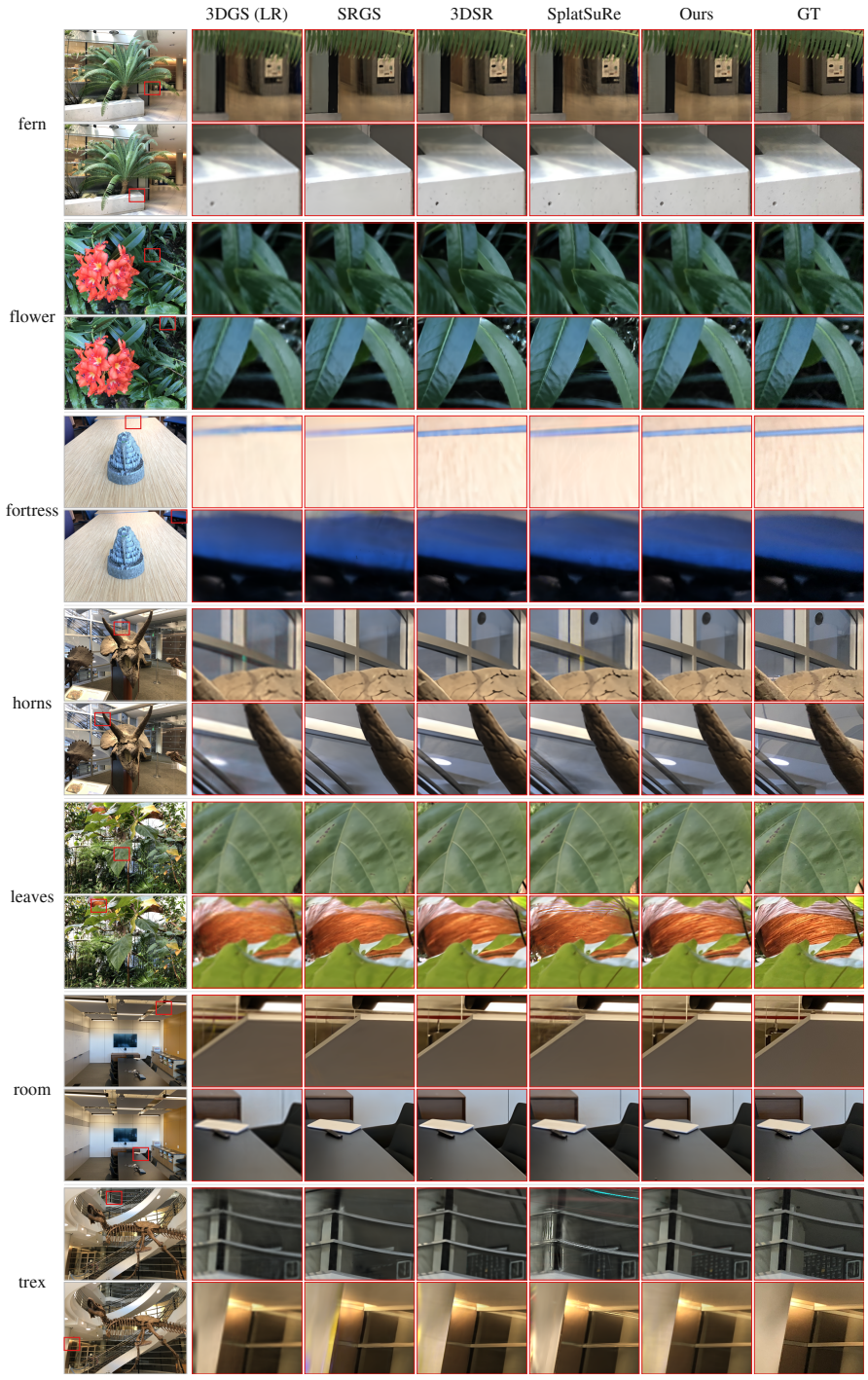

Figure 12: Qualitative comparison on LLFF (Mildenhall et al., 2019) under the $\times 8 \rightarrow \times 2$ setting.