# ONE CAPTURE, ONE CAMERA FAMILY: STRUCTURED CAMERA MODELING FOR LIGHT FIELD GAUSSIAN SPLATTING

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

3D Gaussian Splatting (3DGS) has demonstrated remarkable performance for novel view synthesis, yet its conventional camera formulation is poorly matched to light field (LF) reconstruction. In particular, treating the sub-aperture views captured in a single LF exposure as independent cameras disregards their shared acquisition geometry and leads to inconsistent scene structure across views. To address this, we propose StructLFGS, a reconstruction framework built on the principle of *one capture, one camera family*. Specifically, we model all sub-aperture views as observations generated by a single structured camera family, which compresses the camera solution space from per-view 6-DoF states to a compact shared parameterization. Crucially, this reduction allows us to determine the canonical camera parameters directly from the observed angular trajectories. This defines a canonical frame shared by all views and eliminates per-view camera optimization. With this canonical family fixed, the trajectories are lifted into Gaussian primitives. Their disparity estimates further constrain optimization, making cross-view geometric consistency intrinsic to the representation. Extensive experiments on four LF benchmarks show that StructLFGS outperforms GS-based methods and remains competitive against LF-specific methods.

## 1 INTRODUCTION

3D Gaussian Splatting (3DGS) (Kerbl et al., 2023) has demonstrated remarkable performance in scene reconstruction and novel view synthesis when provided with multi-view inputs. Light field (LF) cameras capture both the intensity and direction of light rays in a single exposure (Levoy & Hanrahan, 1996; Adelson & Wang, 1992), thereby providing dense multi-view observations. However, the sub-aperture views are separated by a narrow baseline, which makes it difficult for structure-from-motion (SfM) (Schönberger & Frahm, 2016) to estimate the per-view camera poses required by 3DGS (Zhang et al., 2017). Consequently, this powerful 3D representation cannot be directly applied to a natural source of multi-view observations.

To address this, existing pipelines adopt two mainstream strategies. Joint optimization methods (Huang et al., 2025; Fu et al., 2024) circumvent SfM by jointly optimizing camera poses with the Gaussian scene. This added flexibility can leave each view's 6-DoF pose underconstrained. Consequently, the poses can drift when the narrow baseline provides insufficient parallax. Moreover, the poses can absorb view-specific appearance rather than the underlying geometry. By contrast, feed-forward methods (Jiang et al., 2025; Ye et al., 2026) predict poses in a single forward pass. This eliminates per-scene optimization altogether. Yet their per-view pose outputs do not explicitly enforce the regular angular lattice that defines a rectified LF. Both strategies remain within a *view-wise camera formulation*. This formulation is well suited to freely moving cameras. However, it does not explicitly encode the coupled acquisition geometry of a rectified LF.

A rectified LF is not a collection of independent views. Under the two-plane parameterization, the sub-aperture views form a regular angular lattice of virtual cameras. These cameras share coplanar centers and parallel optical axes (Fig. 1). Their intrinsics also share a focus-dependent offset (Dansereau et al., 2013). They are therefore coupled, each determined by its angular index and a compact set of shared parameter groups. Together they form a *structured camera family*. This



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.

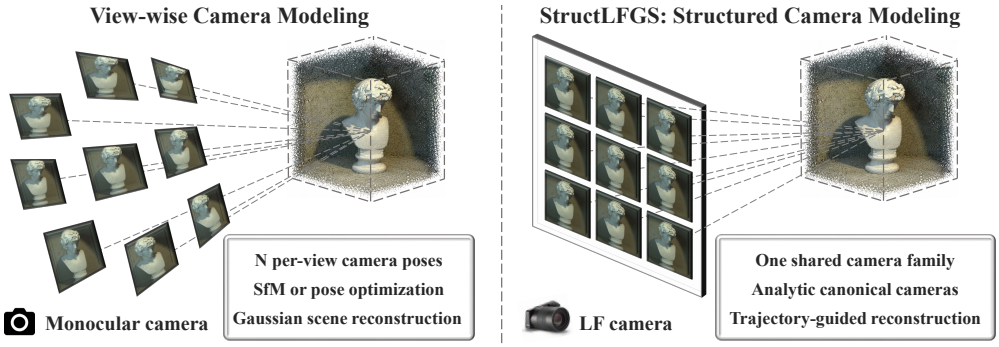


Figure 1: Camera modeling comparison. View-wise pipelines estimate camera poses through SfM or joint optimization. StructLFGS represents LF views with a shared camera family and analytically constructs canonical cameras from angular disparities. The same angular trajectories support Gaussian initialization and disparity supervision during fixed-camera optimization.

coupling also manifests in the cross-view image geometry. For a scene point visible across views, its projection varies affinely with the angular coordinates. It therefore traces a linear trajectory on the epipolar plane image (EPI), whose slope encodes angular disparity (Bolles et al., 1987; Wanner & Goldluecke, 2012). This trajectory structure shows that camera geometry and scene depth are linked by a single scalar per point. A view-wise parameterization does not enforce this coupling by construction, introducing degrees of freedom beyond those required by rectified LF acquisition.

Motivated by this, we propose StructLFGS, a reconstruction framework built on the principle of *one capture, one camera family*. We model all sub-aperture views as observations generated by a single structured camera family, compressing the camera solution space from independent 6-DoF states to a compact shared parameterization $(f, \mathbf{c}, B, o)$. This shared parameterization links the observed angular trajectories to scene depth through a single disparity relation. Specifically, we fix $f$ and $\mathbf{c}$ by a coordinate convention and determine $B_c$ and $o_c$ analytically by mapping the disparities to a prescribed canonical inverse-depth interval. The resulting camera family establishes a common frame in which the same trajectories are lifted directly into Gaussian primitives, preserving their correspondence-derived geometry at initialization. Their disparities then guide Gaussian optimization through supervision at reliable correspondences, while the fixed camera family preserves the regular acquisition geometry. In this way, one set of angular trajectories connects camera construction, Gaussian initialization, and geometric supervision. StructLFGS thus eliminates the need for external SfM or per-view camera optimization.

StructLFGS outperforms the evaluated joint optimization and feed-forward 3DGS pipelines on both synthetic and real-world LF benchmarks. Against LF-specific methods, it achieves the best results among the evaluated methods on synthetic LFs and remains competitive on real-world captures.

Our main contributions are summarized as follows:

- We model each LF capture as a structured camera family, compressing the camera solution space from per-view 6-DoF poses to a compact shared parameterization.
- We determine the canonical camera family analytically from angular disparities estimated from EPI trajectories, without SfM or iterative pose optimization.
- We reuse the recovered trajectories to initialize the Gaussian scene and regularize its optimization with a disparity-consistency loss.

## 2 Related Work

### 2.1 Light Field Reconstruction

LF records radiance over two spatial and two angular dimensions (Levoy & Hanrahan, 1996; Gortler et al., 1996); a single lenslet-camera exposure decodes into regularly indexed sub-aperture views sharing a common acquisition geometry (Adelson & Wang, 1992; Ng et al., 2005; Dansereau et al.,

2013). This spatial–angular redundancy supports depth estimation, view synthesis, and angular super resolution (Wu et al., 2017; Han et al., 2021); classical methods exploit epipolar-plane geometry and disparity-guided warping (Bolles et al., 1987; Wanner & Goldluecke, 2014), while learning methods estimate disparity for view warping (Kalantari et al., 2016), apply spatial–angular or 4D convolutions (Yeung et al., 2018; Wang et al., 2018; 2020), aggregate multi-view features with attention or geometry (Jin et al., 2020b;a; Wu et al., 2021; Liu et al., 2022), or model epipolar structure with transformers (Liang et al., 2026). These methods primarily synthesize missing views through image, feature, or ray-space processing. We study explicit Gaussian reconstruction with a structured camera family as an alternative, and compare against these LF-specific methods in Sec. 4.3.

### 2.2 Explicit 3D Scene and Camera Modeling

Unlike neural radiance fields rendered through volumetric integration (Mildenhall et al., 2021), 3DGS represents a scene as anisotropic Gaussian primitives with real-time differentiable rendering (Kerbl et al., 2023), but it requires a camera for every input view. Most pipelines obtain these cameras through SfM (e.g., COLMAP (Schönberger & Frahm, 2016)). However, limited parallax can make SfM unreliable on narrow-baseline LF inputs. Joint optimization methods therefore optimize view-wise cameras with the Gaussian scene, such as 3R-GS (Huang et al., 2025) and CF-3DGS (Fu et al., 2024), while feed-forward methods estimate poses in a single forward pass, such as AnySplat (Jiang et al., 2025) and YoNoSplat (Ye et al., 2026). Related methods adopt similar camera strategies (Zhao et al., 2024; Ji & Yao, 2025; Huang & Mikolajczyk, 2025; Charatan et al., 2024). For rectified LFs, unconstrained per-view poses introduce degrees of freedom beyond those required by the regular acquisition geometry. Two-plane parameterization and EPI geometry establish the relation between angular disparity and scene depth, while plenoptic calibration estimates the physical imaging geometry (Dansereau et al., 2013). Recent plenoptic 3DGS methods (Yim et al., 2025) delegate camera handling to patch extraction, yet none of these derives the camera analytically from the angular lattice. We instead build on these relations. With intrinsics fixed by convention, we map recovered angular disparities to a prescribed inverse-depth interval and analytically determine the canonical baseline and focus offset. Our formulation connects canonical camera construction, Gaussian initialization, and disparity supervision through the same angular trajectories. The resulting camera family remains fixed during optimization, eliminating external SfM and pose refinement. We evaluate against SfM-based, joint optimization, and feed-forward GS strategies in Sec. 4.2.

## 3 Method

We present StructLFGS, a light field Gaussian splatting framework that reconstructs a single rectified LF from a sparse subset of its sub-aperture views. We model all sub-aperture views as observations generated by a shared structured camera family, whose disparity depends on scene depth and scales the angular shift across views. Given a coordinate convention and a canonical inverse-depth interval, the recovered disparities analytically determine a canonical camera. The resulting trajectories are lifted into Gaussian primitives and reused as disparity supervision during fixed-camera optimization, yielding a unified pipeline from camera construction to Gaussian reconstruction without external SfM or per-view pose optimization. Fig. 2 summarizes the complete framework.

### 3.1 Structured Camera Modeling

Let $\mathbf{a} = (a_x, a_y) \in \mathcal{A}$ be an angular index on an $A \times A$ lattice, measured in units of adjacent views, with $\mathbf{a} = \mathbf{0}$ defining the reference frame. We describe the camera array by shared parameters $\boldsymbol{\theta} = (f, \mathbf{c}, B, o)$, where $f = f_x = f_y$ is the focal length, $\mathbf{c} = (c_x, c_y)$ is the principal point, $B$ is the baseline per angular step, and $o$ is the focus-induced disparity offset, corresponding to the disparity of a point at infinite depth. The camera at angular index $\mathbf{a}$ is

$$\mathbf{C}_\mathbf{a} = B[a_x, a_y, 0]^\mathsf{T}, \qquad \mathbf{R}_\mathbf{a} = \mathbf{I}, \qquad \mathbf{K}_\mathbf{a} = \begin{bmatrix} f & 0 & c_x + a_x o \\ 0 & f & c_y + a_y o \\ 0 & 0 & 1 \end{bmatrix}. \tag{1}$$

The angular terms in the principal point encode the focus-induced offset $o$ (Dansereau et al., 2013), which induces a depth-independent image shift, while $B$ controls the depth-dependent component

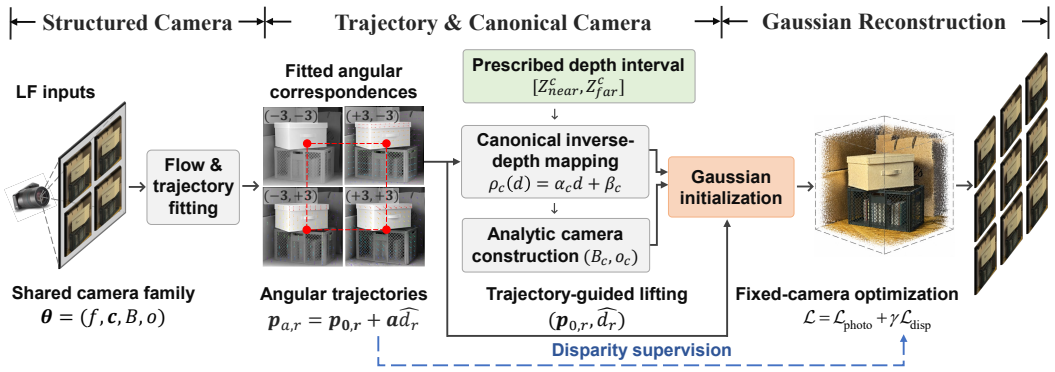


Figure 2: Overview of StructLFGS. A structured camera family models all LF views with a compact shared parameterization, replacing independent 6-DoF poses. Cross-view optical flow yields angular trajectories, whose disparities determine the canonical camera family under a prescribed inverse-depth convention. These trajectories initialize the Gaussian scene and provide disparity supervision during optimization with the cameras fixed.

of disparity. Substituting these parameters into the pinhole projection of $\mathbf{X} = (X, Y, Z)^\mathsf{T} \in \mathbb{R}^3$ gives

$$\mathbf{p_a} = \begin{bmatrix} f\dfrac{X - Ba_x}{Z} + c_x + a_x o \\ f\dfrac{Y - Ba_y}{Z} + c_y + a_y o \end{bmatrix} = \mathbf{p_0} + \begin{bmatrix} a_x \\ a_y \end{bmatrix} \left( o - \frac{fB}{Z} \right), \tag{2}$$

where $\mathbf{p_0} = \left[ \frac{fX}{Z} + c_x, \frac{fY}{Z} + c_y \right]^\mathsf{T}$ is the reference-view projection and $d = o - \frac{fB}{Z}$ the per-step disparity. We fix $f$ and $\mathbf{c}$ by a coordinate convention; the remaining baseline $B$ and offset $o$ are determined only up to an affine gauge in inverse depth, which is fixed in Sec. 3.2.

### 3.2 Canonical Camera and Trajectory Construction

We estimate angular disparities from cross-view flow and use them to construct a canonical camera and trajectory set. Because disparity is affine in inverse depth, the observations determine a family of camera–depth configurations rather than a unique metric calibration. We fix this affine gauge by mapping the robust disparity range to a prescribed canonical inverse-depth interval.

For two input anchor views $i$ and $j$, let $\Delta\mathbf{a}_{ij} = \mathbf{a}_j - \mathbf{a}_i$ be their angular separation and $\mathbf{F}_{i\to j}$ the optical flow estimated by a frozen RAFT network (Teed & Deng, 2020). Under the structured camera model, $\mathbf{F}_{i\to j}(\mathbf{x}) \approx \Delta\mathbf{a}_{ij} d(\mathbf{x})$. The least-squares per-step disparity estimate is

$$d_{ij}(\mathbf{x}) = \frac{\Delta\mathbf{a}_{ij}^\mathsf{T} \mathbf{F}_{i\to j}(\mathbf{x})}{\|\Delta\mathbf{a}_{ij}\|_2^2}. \tag{3}$$

We robustly aggregate pairwise estimates along each correspondence track (App. A), obtaining per track a disparity $\hat{d}_r$, an uncertainty $\sigma_r$, and a confidence $q_r$. For a track anchored at pixel $\mathbf{x}_r$ in view $\mathbf{a}_r$, its reference-view position is $\mathbf{p}_{0,r} = \mathbf{x}_r - \mathbf{a}_r \hat{d}_r$. The resulting trajectories support Gaussian initialization and disparity supervision.

We map the robust disparity range $[d_{\min}, d_{\max}]$ to a prescribed canonical inverse-depth interval via the affine chart

$$\rho_c(d) = \alpha_c d + \beta_c, \qquad Z_c(d) = \rho_c(d)^{-1}, \tag{4}$$

where $\rho_c = 1/Z_c$ is the canonical inverse depth and $\alpha_c, \beta_c \in \mathbb{R}$ the coefficients. The endpoint constraints $\rho_c(d_{\min}) = 1/Z^c_{\text{near}}$ and $\rho_c(d_{\max}) = 1/Z^c_{\text{far}}$ determine

$$\alpha_c = \frac{1/Z^c_{\text{far}} - 1/Z^c_{\text{near}}}{d_{\max} - d_{\min}}, \qquad \beta_c = \frac{1}{Z^c_{\text{near}}} - \alpha_c d_{\min}, \tag{5}$$

where $0 < Z^c_{\text{near}} < Z^c_{\text{far}}$ are prescribed, so $d_{\min} < d_{\max}$ gives $\alpha_c < 0$.

Substituting $\rho_c(d) = \alpha_c d + \beta_c$ into $o_c - fB_c/Z_c(d) = d$ and matching coefficients yields

$$B_c = -\frac{1}{f\alpha_c}, \qquad o_c = -\frac{\beta_c}{\alpha_c}. \tag{6}$$

The resulting camera family reproduces the fitted disparities of the retained tracks in the chosen canonical gauge, without representing metric calibration. Because $fB_c = -1/\alpha_c$, consistent back-projection preserves the fitted angular trajectories for any positive canonical focal length.

By construction, the canonical camera reproduces the fitted angular trajectory of each retained track:

$$\mathbf{p}^c_{\mathbf{a}} = \mathbf{p}^c_{\mathbf{0}} + \mathbf{a}\,\hat{d}_r, \tag{7}$$

where $\mathbf{p}^c_{\mathbf{0}} = \mathbf{p}_{0,r}$ is the reference-view position and $Z_c(\hat{d}_r)$ provides the canonical depth used for Gaussian lifting.

### 3.3 Trajectory-Guided Gaussian Reconstruction

Given the trajectory set and canonical camera, we reconstruct the Gaussian scene under fixed cameras. Fused trajectories initialize Gaussian geometry, while fitted disparity maps on the anchor views provide geometric supervision. Both operations use the same disparity-to-depth mapping.

We retain tracks whose canonical depths lie in $[Z^c_{\text{near}}, Z^c_{\text{far}}]$. We associate the retained tracks using reciprocal-neighbor, disparity-consistency, and cross-view projection checks, and fuse each retained group by confidence-weighted averaging (App. A). For each fused group $m$ with reference position $(\bar{x}_m, \bar{y}_m)$ and disparity $\bar{d}_m$, we initialize one anisotropic Gaussian at

$$\boldsymbol{\mu}_m = Z_c(\bar{d}_m)\,[(\bar{x}_m - c_x)/f \quad (\bar{y}_m - c_y)/f \quad 1]^{\mathsf{T}}. \tag{8}$$

Uncertainty propagation initializes the Gaussian covariance, and the fused color initializes its appearance.

During optimization, only Gaussian parameters are updated. The photometric objective is the standard $\ell_1$–SSIM loss $\mathcal{L}_{\text{photo}}$. To retain the measured angular geometry, we additionally supervise the rendered disparity. For a Gaussian at canonical depth $Z$, its disparity is

$$d(Z) = \frac{1/Z - \beta_c}{\alpha_c}. \tag{9}$$

Alpha-composited disparity moments yield the rendered mean $D(\mathbf{u})$ and variance $V(\mathbf{u})$ (App. A). We apply the confidence-weighted Huber loss

$$\mathcal{L}_{\text{disp}} = \frac{\sum_{\mathbf{u}\in\mathcal{U}} q(\mathbf{u})\,\rho_H\left(\dfrac{D(\mathbf{u}) - \hat{d}(\mathbf{u})}{\sigma(\mathbf{u}) + \epsilon_\sigma}\right)}{\sum_{\mathbf{u}\in\mathcal{U}} q(\mathbf{u})}, \tag{10}$$

where $\hat{d}(\mathbf{u})$ and $\sigma(\mathbf{u})$ are the fitted disparity and uncertainty, $q(\mathbf{u})$ the confidence, and $\epsilon_\sigma$ an uncertainty floor. The supervision set $\mathcal{U}$ is selected using trajectory confidence, accumulated opacity, and rendered disparity variance; the exact thresholds are in App. A. The complete objective is

$$\mathcal{L} = \mathcal{L}_{\text{photo}} + \gamma\mathcal{L}_{\text{disp}}, \tag{11}$$

with disparity weight $\gamma$, optimized with MCMC-based Gaussian relocation and capacity control (Kheradmand et al., 2024).

## 4 Experiments

### 4.1 Experimental Setup

**Datasets and protocols.** We evaluate on four LF benchmarks: synthetic *HCInew* (4 scenes) (Honauer et al., 2016) and *HCIold* (5 scenes) (Wanner et al., 2013), and real-world *30scenes*

Table 1: Comparison with GS-based methods on angular reconstruction setting ($2 \times 2 \rightarrow 7 \times 7$ and $3 \times 3 \rightarrow 7 \times 7$). Each cell reports PSNR↑/SSIM↑/LPIPS↓ ($\times 10$) on full-frame RGB. We marked the best, second-best, and third-best, respectively.

| Method | HCInew | HCIold | 30scenes | Occlusions |
|---|---|---|---|---|
| 3R-GS | 18.91/0.4111/2.853 | 25.40/0.6332/1.963 | 28.98/0.8214/1.007 | 26.83/0.8104/1.032 |
| CF-3DGS | 8.80/0.2483/4.614 | 11.78/0.3603/4.221 | 15.56/0.3641/3.396 | 15.94/0.3085/3.602 |
| AnySplat | 19.81/0.4664/2.949 | 17.94/0.4173/3.583 | 19.62/0.6247/3.017 | 17.64/0.4636/3.864 |
| YoNoSplat | 21.30/0.5524/2.804 | 24.89/0.6761/2.013 | 25.89/0.7730/1.973 | 25.06/0.7503/2.054 |
| StructLFGS ($2 \times 2$) | 38.31/0.9706/0.097 | 42.28/0.9834/0.070 | 39.30/0.9803/0.153 | 35.63/0.9680/0.218 |
| 3R-GS | 19.69/0.4591/2.948 | 28.34/0.7376/1.821 | 33.07/0.8935/0.720 | 31.65/0.9130/0.692 |
| CF-3DGS | 23.59/0.6774/2.792 | 28.24/0.7906/1.802 | 33.00/0.9278/1.019 | 29.76/0.9050/1.015 |
| AnySplat | 20.68/0.5014/2.524 | 18.81/0.4279/3.299 | 21.28/0.6791/2.584 | 19.38/0.5386/3.148 |
| YoNoSplat | 21.53/0.5873/2.614 | 24.76/0.6615/2.234 | 26.35/0.7889/2.059 | 24.86/0.7432/2.229 |
| StructLFGS ($3 \times 3$) | 41.64/0.9802/0.074 | 43.34/0.9860/0.076 | 41.07/0.9853/0.157 | 38.26/0.9804/0.153 |

(30 scenes) (Kalantari et al., 2016) and *Occlusions* (25 scenes) (Raj et al., 2016). We consider $2 \times 2 \rightarrow 7 \times 7$ and $3 \times 3 \rightarrow 7 \times 7$ reconstruction. The $2 \times 2$ inputs are the four corners of the central $7 \times 7$ lattice; the $3 \times 3$ inputs use rows and columns $\{1, 4, 7\}$. The remaining 45 and 40 views, respectively, are used for evaluation. GS comparisons and ablations use full-frame RGB PSNR, SSIM, and LPIPS. For LF-specific comparisons, PSNR and SSIM follow the cropped Y-channel protocol (Liang et al., 2026), while LPIPS is evaluated on full-frame RGB images.

**Baselines.** We compare against two complementary groups. *GS baselines* represent three camera strategies: SfM-based method (vanilla 3DGS), joint optimization methods (3R-GS (Huang et al., 2025) and CF-3DGS (Fu et al., 2024)), and feed-forward methods (AnySplat (Jiang et al., 2025) and YoNoSplat (Ye et al., 2026)). *LF-specific baselines* include Kalantari et al. (Kalantari et al., 2016), LFASR-geo (Jin et al., 2020b), FS-GAF (Jin et al., 2020a), Yeung et al. (Yeung et al., 2018), Pseudo4DCNN (Wang et al., 2018), P4DCNN (Wang et al., 2020), SAA-Net (Wu et al., 2021), LF-EASR (Liu et al., 2022), and EPIT-ASR (Liang et al., 2026). These methods provide task-specific references for the $2 \times 2 \rightarrow 7 \times 7$ angular reconstruction setting.

**Implementation details.** StructLFGS reconstructs each scene independently without LF-specific training or fine-tuning. A frozen pretrained RAFT-Large network (Teed & Deng, 2020) estimates flow between the selected input views, from which we recover angular trajectories through projection and IRLS refinement. The canonical chart uses the 2nd–98th disparity percentiles and a fixed depth interval of $[0.5, 5.0]$. The Gaussian scene is initialized from the fused trajectories and optimized for 10K steps with MCMC-based relocation and capacity control (Kheradmand et al., 2024). The disparity-consistency loss starts at step 1500, warms up over 1000 steps, and reaches a maximum weight of 0.05. Additional experiments (i.e., ground-truth camera comparisons, efficiency measurements, and hyperparameter sensitivity analyses) are provided in App. C–E. All experiments are conducted on a single NVIDIA GeForce RTX 4090 GPU with 24 GB memory.

### 4.2 Main Results on GS-based Reconstruction

Table 1 compares StructLFGS with executable GS baselines under their method-specific evaluation protocols, details in App. B. Vanilla 3DGS is omitted because COLMAP fails on these LF inputs. StructLFGS achieves the best PSNR, SSIM, and LPIPS on all four datasets, with PSNR margins of 8.80–17.01 dB for $2 \times 2$ inputs and 6.61–18.05 dB for $3 \times 3$ inputs. These results characterize the practical reconstruction performance of StructLFGS relative to the evaluated GS pipelines. The contribution of the structured camera family is examined separately through the controlled comparisons in Table 3. Fig. 3 illustrates a geometric failure mode associated with this performance gap. GS baselines produce blurred textures and distorted structures, while StructLFGS preserves sharper details. In the shown HCInew *dishes* case, CF-3DGS collapses the camera centers into a tight cluster; in 30scenes *IMG_1324*, its centers become scattered. These configurations depart from the regular angular lattice, whereas StructLFGS preserves the prescribed lattice by construction. The examples

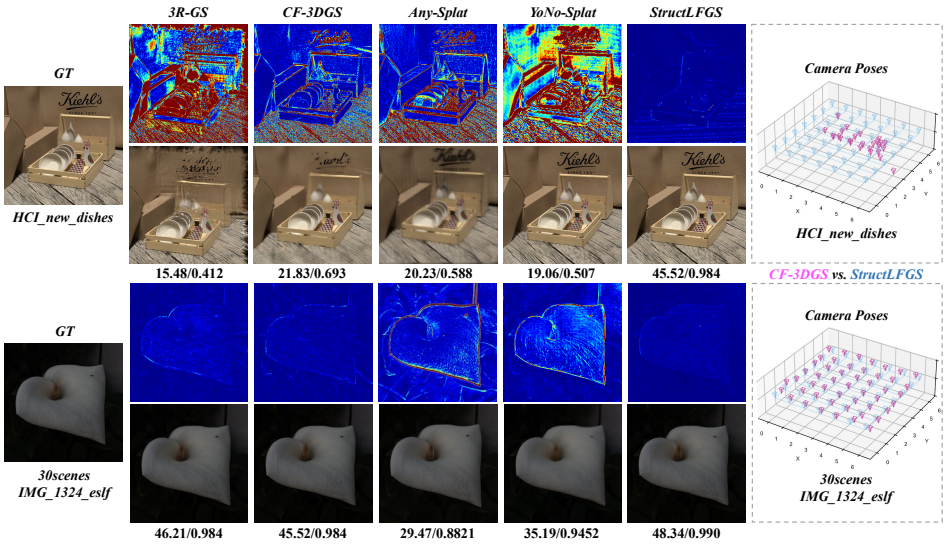


Figure 3: Qualitative comparison with GS-based methods on $3 \times 3$ protocol. Baselines produce blurred textures and distorted structures, whereas StructLFGS recovers sharper details. The right panel visualizes the camera poses: CF-3DGS yields collapsed or scattered centers, while StructLFGS preserves the regular angular lattice.

Table 2: Comparison with LF-specific methods on angular reconstruction setting ($2 \times 2 \rightarrow 7 \times 7$). Each cell reports PSNR↑/SSIM↑/LPIPS↓ ($\times 10$); PSNR and SSIM use the cropped Y channel, while LPIPS uses full-frame RGB. We marked the best, second-best, and third-best, respectively.

| Method | HCInew | HCIold | 30scenes | Occlusions |
|---|---|---|---|---|
| Kalantari2016 | 28.91/0.9150/0.893 | 34.93/0.9405/0.508 | 37.49/0.9839/0.178 | 33.45/0.9689/0.304 |
| LFASR-geo | 32.23/0.9555/0.423 | 38.40/0.9707/0.231 | 38.45/0.9853/0.175 | 35.28/0.9795/0.228 |
| FS-GAF | 34.28/0.9725/0.189 | 39.21/0.9808/0.166 | 40.52/0.9918/0.106 | 36.77/0.9849/0.141 |
| Yeung2018 | 32.09/0.9430/0.418 | 40.44/0.9619/0.189 | 42.21/0.9933/0.096 | 38.09/0.9872/0.126 |
| Pseudo4DCNN | 27.59/0.8752/1.102 | 35.52/0.9327/0.627 | 38.20/0.9758/0.258 | 35.55/0.9756/0.274 |
| P4DCNN | 29.56/0.9113/0.679 | 36.60/0.9467/0.414 | 39.75/0.9860/0.165 | 36.52/0.9813/0.207 |
| SAA-Net | 30.10/0.9230/0.836 | 36.14/0.9490/0.526 | 39.77/0.9921/0.111 | 36.68/0.9850/0.147 |
| LF-EASR | 33.65/0.9638/0.345 | 40.83/0.9681/0.146 | 41.47/0.9918/0.111 | 37.65/0.9863/0.142 |
| EPIT-ASR | 34.76/0.9676/0.279 | 42.26/0.9869/0.141 | 43.54/0.9951/0.070 | 39.43/0.9907/0.084 |
| StructLFGS | 39.85/0.9873/0.097 | 44.73/0.9940/0.070 | 41.88/0.9932/0.153 | 37.80/0.9871/0.218 |

highlight a limitation of unconstrained view-wise camera optimization under narrow baselines: its additional degrees of freedom can weaken the shared geometry needed for consistent novel-view synthesis.

### 4.3 Comparison with LF-specific Methods

Table 2 compares StructLFGS with LF-specific angular super-resolution methods under the cropped-Y evaluation protocol. StructLFGS achieves the best PSNR, SSIM, and LPIPS on both synthetic benchmarks, exceeding the strongest PSNR baseline by 5.09 dB on HCInew and 2.47 dB on HCIold. These results show that trajectory-derived geometry and the shared camera family offer an alternative to LF-specific learned priors without requiring LF-specific network training or fine-tuning. On real captures, StructLFGS remains competitive in PSNR and SSIM but performs less well in LPIPS. The gap may partly reflect flow errors and inter-view capture discrepancies. LF-specific training may help learned methods accommodate these discrepancies, which our fixed camera model does not explicitly represent. Nevertheless, the qualitative results in Fig. 4 show that StructLFGS preserves sharper local structures, illustrating the benefit of explicit LF geometry in these examples.

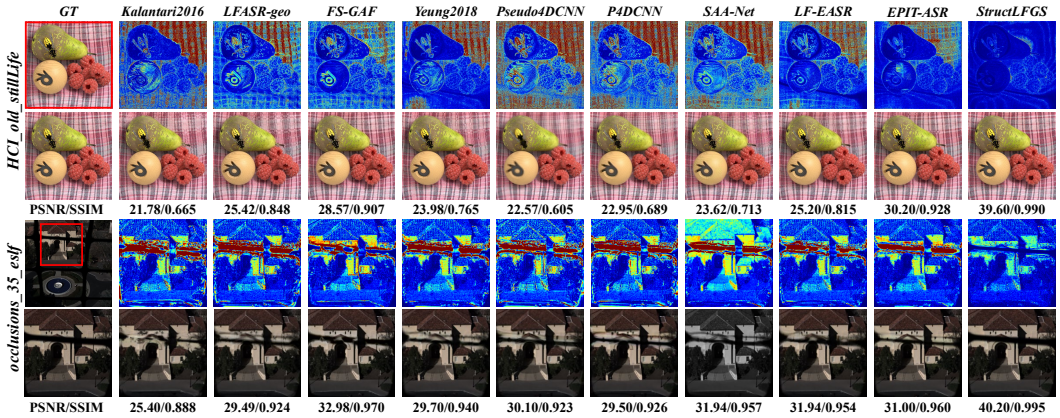


Figure 4: Qualitative comparison with LF-specific methods. Baselines show blur and artifacts, while StructLFGS recovers sharper details and the highest PSNR/SSIM on the shown scenes.

Table 3: Component ablations. Camera: fixed family (Fixed), shared $(B, o)$ refinement (Shared-refined), or independent input poses (Free-input). Init.: trajectory (T), point-count-matched random (R), or matched depth shuffle (S). Disp.: disparity supervision.

| ID | Camera | Init. | Disp. | HCInew (4) | 30scenes (4 of 30) |
|---|---|---|---|---|---|
| **A (Full)** | Fixed | T | ✓ | **38.31/0.9706/0.097** | **37.40/0.9812/0.144** |
| B | Fixed | T | – | 38.02/0.9685/0.105 | 37.25/0.9804/0.153 |
| C | Shared-refined | T | – | 37.05/0.9572/0.132 | 36.72/0.9715/0.173 |
| D | Free-input | T | – | 29.91/0.8925/0.487 | 31.59/0.9147/0.676 |
| E | Fixed | R | ✓ | 25.74/0.8235/0.684 | 34.18/0.9608/0.337 |
| F | Fixed | R | – | 24.37/0.7435/0.872 | 32.42/0.9372/0.526 |
| $E_m$ | Fixed | S | ✓ | 32.61/0.9340/0.250 | 36.55/0.9708/0.198 |
| $F_m$ | Fixed | S | – | 31.07/0.9037/0.364 | 36.21/0.9698/0.253 |

### 4.4 Ablation Studies

Table 3 evaluates camera constraints, trajectory initialization, and disparity supervision under $2 \times 2 \rightarrow 7 \times 7$ reconstruction on HCInew and four real scenes from 30scenes (*IMG_1324*, *Cars*, *Flower1*, and *Seahorse*). Paired gains below refer to HCInew and this real subset, respectively.

**Camera-family constraint.** The fixed canonical family yields the best reconstruction among the camera variants. B, C, and D share the canonical-camera and trajectory initialization, a 10K budget, and disabled disparity supervision. C refines shared $(B, o)$, while D optimizes independent 6-DoF input poses and interpolates target poses; neither uses held-out RGB. C exceeds D by $7.14/5.13$ dB, showing the benefit of retaining the shared family even from identical initialization. Fixing the family further improves PSNR by $0.97/0.53$ dB (B versus C), with better SSIM and LPIPS in both comparisons. These results support constructing the camera family analytically and keeping it fixed during Gaussian optimization.

**Trajectory initialization.** Trajectory initialization benefits reconstruction through the correspondence between reference pixels and depth. It outperforms point-count-matched random initialization by $12.57/3.22$ dB (A versus E). Shuffling depths while preserving reference samples, Gaussian attributes, and the depth distribution reduces PSNR by $5.70/0.85$ dB with disparity supervision (A versus $E_m$) and $6.95/1.04$ dB without it (B versus $F_m$). Thus, assigning depth consistently with angular trajectories matters beyond matching the point count and depth distribution. Fig. 5(b) shows the corresponding EPI differences.

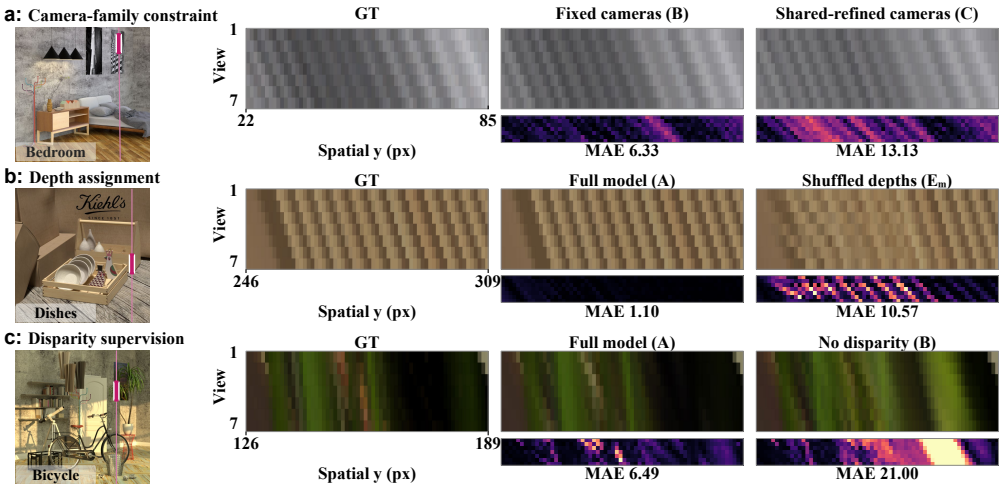


Figure 5: Qualitative ablations on EPI crops. (a) Fixed camera-family geometry (B) preserves more coherent structure than shared camera refinement (C). (b) Trajectory-consistent depth initialization (A) outperforms shuffled-depth initialization ($E_m$). (c) Disparity supervision (A) reduces local EPI artifacts compared with its removal (B).

Table 4: Replacing the disparity-derived camera with ground-truth calibration on HCInew. Each cell reports mean full-frame RGB PSNR↑/SSIM↑/LPIPS↓ ($\times 10$); $\Delta$ is GT minus ours.

| Protocol | Ours | GT calib. | $\Delta$PSNR | $\Delta$SSIM | $\Delta$LPIPS |
|---|---|---|---|---|---|
| $2 \times 2 \rightarrow 7 \times 7$ | 38.31/0.9706/0.097 | 39.01/0.9721/0.092 | 0.70 | 0.0016 | -0.005 |
| $3 \times 3 \rightarrow 7 \times 7$ | 41.64/0.9802/0.074 | 41.80/0.9805/0.074 | 0.16 | 0.0002 | 0.000 |

**Disparity supervision.** Disparity supervision improves all three initialization settings: PSNR rises by $0.29/0.15$ dB with trajectory initialization (A versus B), $1.54/0.34$ dB with shuffled depths ($E_m$ versus $F_m$), and $1.37/1.76$ dB with random initialization (E versus F), alongside better SSIM and LPIPS. The larger gains with shuffled or random initialization show that supervision helps compensate for poorer initial geometry, while the full model remains strongest. Fig. 5(c) shows reduced local EPI artifacts.

**Canonical camera vs. calibration.** Replacing the canonical camera with ground-truth calibration improves mean PSNR on HCInew by $0.70$ dB with $2 \times 2$ inputs and $0.16$ dB with $3 \times 3$ inputs (Table 4). Both variants use the same estimated trajectories and optimization budget. The canonical construction approaches the calibrated reference, particularly with nine inputs, while requiring no metric calibration. Per-scene results are in App. C.

## 5 Conclusion

We propose StructLFGS, a Gaussian splatting framework built on the principle of *one capture, one camera family*. StructLFGS explicitly models the shared acquisition geometry of a light field. A canonical camera family follows analytically from angular disparities under a prescribed inverse-depth convention, eliminating external SfM and per-view pose optimization. The same angular trajectories connect camera construction, Gaussian initialization, and disparity supervision within one reconstruction pipeline. On four LF benchmarks, StructLFGS outperforms the evaluated GS-based methods. It leads LF-specific methods on synthetic data and remains competitive on real captures, without LF-specific training or fine-tuning. These results demonstrate the value of incorporating acquisition structure into camera modeling and scene reconstruction.

## AI USE STATEMENT

Generative AI tools were used only to improve the readability and presentation of the manuscript. All AI-assisted edits were reviewed and verified by the authors, who take full responsibility for the final content.

## REFERENCES


Edward H. Adelson and John Y. A. Wang. Single lens stereo with a plenoptic camera. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):99–106, 1992.

Robert C. Bolles, Harlyn H. Baker, and David H. Marimont. Epipolar-plane image analysis: An approach to determining structure from motion. *International Journal of Computer Vision*, 1(1): 7–55, 1987.

David Charatan, Sizhe Li, Andrea Tagliasacchi, and Vincent Sitzmann. pixelsplat: 3d gaussian splats from image pairs for scalable generalizable 3d reconstruction. In *CVPR*, pp. 19457–19467, 2024.

Donald G. Dansereau, Oscar Pizarro, and Stefan B. Williams. Decoding, calibration and rectification for lenselet-based plenoptic cameras. In *CVPR*, pp. 1027–1034, 2013.

Yang Fu, Xiaolong Wang, Sifei Liu, Amey Kulkarni, and Jan Kautz. COLMAP-free 3d gaussian splatting. In *CVPR*, pp. 20796–20805, 2024.

Steven J. Gortler, Radek Grzeszczuk, Richard Szeliski, and Michael F. Cohen. The lumigraph. In *SIGGRAPH*, pp. 43–54, 1996.

Kang Han, Wei Xiang, Eric Wang, and Tao Huang. A novel occlusion-aware vote cost for light field depth estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(11): 8022–8035, 2021.

Katrin Honauer, Ole Johannsen, Daniel Kondermann, and Bastian Goldluecke. A dataset and evaluation methodology for depth estimation on 4D light fields. In *ACCV*, pp. 19–34, 2016.

Ranran Huang and Krystian Mikolajczyk. No pose at all: Self-supervised pose-free 3d gaussian splatting from sparse views. In *ICCV*, pp. 27947–27957, 2025.

Zhisheng Huang, Peng Wang, Jingdong Zhang, Yuan Liu, Xin Li, and Wenping Wang. 3R-GS: Best practice in optimizing camera poses along with 3DGS. *ArXiv:2504.04294*, 2025.

Bo Ji and Angela Yao. SfM-free 3d gaussian splatting via hierarchical training. In *CVPR*, pp. 21654–21663, 2025.

Lihan Jiang, Yucheng Mao, Linning Xu, Tao Lu, Kerui Ren, Yichen Jin, Xudong Xu, Mulin Yu, Jiangmiao Pang, Feng Zhao, Dahua Lin, and Bo Dai. AnySplat: Feed-forward 3d gaussian splatting from unconstrained views. *ACM Transactions on Graphics*, 44(6):1–16, 2025.

Jing Jin, Junhui Hou, Jie Chen, Huanqiang Zeng, Sam Kwong, and Jingyi Yu. Deep coarse-to-fine dense light field reconstruction with flexible sampling and geometry-aware fusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(4):1819–1836, 2020a.

Jing Jin, Junhui Hou, Hui Yuan, and Sam Kwong. Learning light field angular super-resolution via a geometry-aware network. In *AAAI*, pp. 11141–11148, 2020b.

Nima Khademi Kalantari, Ting-Chun Wang, and Ravi Ramamoorthi. Learning-based view synthesis for light field cameras. *ACM Transactions on Graphics*, 35(6):1–10, 2016.

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkuehler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4):139–1, 2023.

Shakiba Kheradmand, Daniel Rebain, Gopal Sharma, Weiwei Sun, Yang-Che Tseng, Hossam Isack, Abhishek Kar, Andrea Tagliasacchi, and Kwang Moo Yi. 3d gaussian splatting as markov chain monte carlo. In *NeurIPS*, volume 37, pp. 80965–80986, 2024.

Vincent Leroy, Yohann Cabon, and Jérôme Revaud. Grounding image matching in 3d with MASt3R. In *ECCV*, pp. 71–91, 2024.

Marc Levoy and Pat Hanrahan. Light field rendering. In *SIGGRAPH*, 1996.

Zhengyu Liang, Yingqian Wang, Longguang Wang, Jungang Yang, Yulan Guo, Li Liu, Shilin Zhou, and Wei An. Diving into epipolar transformers for light field super-resolution and disparity estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 48(7):8726–8743, 2026.

Gaosheng Liu, Huanjing Yue, Jiaming Wu, and Jingyu Yang. Efficient light field angular super-resolution with sub-aperture feature learning and macropixel upsampling. *IEEE Transactions on Multimedia*, 25:6588–6600, 2022.

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. *Communications of the ACM*, 65(1):99–106, 2021.

Ren Ng, Marc Levoy, Mathieu Brédif, Gene Duval, Mark Horowitz, and Pat Hanrahan. *Light field photography with a hand-held plenoptic camera*. PhD thesis, Stanford university, 2005.

Abhilash Sunder Raj, Michael Lowney, Raj Shah, and Gordon Wetzstein. Stanford lytro light field archive. 2016.

Johannes L. Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *CVPR*, pp. 4104–4113, 2016.

Zachary Teed and Jia Deng. RAFT: Recurrent all-pairs field transforms for optical flow. In *ECCV*, pp. 402–419, 2020.

Yunlong Wang, Fei Liu, Zilei Wang, Guangqi Hou, Zhenan Sun, and Tieniu Tan. End-to-end view synthesis for light field imaging with pseudo 4DCNN. In *ECCV*, pp. 333–348, 2018.

Yunlong Wang, Fei Liu, Kunbo Zhang, Zilei Wang, Zhenan Sun, and Tieniu Tan. High-fidelity view synthesis for light field imaging with extended pseudo 4DCNN. *IEEE Transactions on Computational Imaging*, 6:830–842, 2020.

Sven Wanner and Bastian Goldluecke. Globally consistent depth labeling of 4d light fields. In *CVPR*, pp. 41–48, 2012.

Sven Wanner and Bastian Goldluecke. Variational light field analysis for disparity estimation and super-resolution. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(3):606–619, 2014.

Sven Wanner, Stephan Meister, and Bastian Goldluecke. Datasets and benchmarks for densely sampled 4d light fields. In *VMV*, pp. 225–226, 2013.

Gaochang Wu, Belen Masia, Adrian Jarabo, Yuchen Zhang, Liangyong Wang, Qionghai Dai, Tianyou Chai, and Yebin Liu. Light field image processing: An overview. *IEEE Journal of Selected Topics in Signal Processing*, 11(7):926–954, 2017.

Gaochang Wu, Yingqian Wang, Yebin Liu, Lu Fang, and Tianyou Chai. Spatial-angular attention network for light field reconstruction. *IEEE Transactions on Image Processing*, 30:8999–9013, 2021.

Botao Ye, Boqi Chen, Haofei Xu, Daniel Barath, and Marc Pollefeys. YoNoSplat: You only need one model for feedforward 3d gaussian splatting. In *ICLR*, pp. 39852–39871, 2026.

Henry Wing Fung Yeung, Junhui Hou, Jie Chen, Yuk Ying Chung, and Xiaoming Chen. Fast light field reconstruction with deep coarse-to-fine modeling of spatial-angular clues. In *ECCV*, pp. 137–152, 2018.

Jonghoon Yim, Byeungwoo Jeon, Roger Olsson, and Mårten Sjöström. 3d-gaussian splatting representation of rendered views from plenoptic 2.0 lenslet images. In *VCIP*, pp. 1–5, 2025.

Yingliang Zhang, Zhong Li, Wei Yang, Peihong Yu, Halting Lin, and Jingyi Yu. The light field 3d scanner. In *ICCP*, pp. 1–9. IEEE, 2017.

Lingzhe Zhao, Peng Wang, and Peidong Liu. BAD-gaussians: Bundle adjusted deblur gaussian splatting. In *ECCV*, pp. 233–250, 2024.

# APPENDIX: ONE CAPTURE, ONE CAMERA FAMILY: STRUCTURED CAMERA MODELING FOR LIGHT FIELD GAUSSIAN SPLATTING

This appendix complements the main paper in five parts. Sec. A details trajectory fitting, uncertainty fusion, Gaussian initialization, and disparity supervision. Sec. B specifies the evaluation and reproduction protocols for every baseline, including the camera procedures favorable to some baselines. Sec. C compares our disparity-derived camera family with ground-truth calibration through per-scene results and visual comparisons. Sec. D reports optimization time and peak GPU memory against GS-based baselines and specifies the measurement protocols. Sec. E examines sensitivity to the disparity-supervision weight.

## A METHOD IMPLEMENTATION DETAILS

**Trajectory fitting.** We use RAFT-Large with the torchvision `C_T_SKHT_V2` weights and their prescribed input normalization. Images are replicate-padded on the bottom and right to multiples of eight; the predicted flow is cropped to the original resolution.

For track $r$ anchored at pixel $\mathbf{x}$ in view $i$, let $\mathcal{J}_r$ index valid correspondences to the other three anchors. The backward flow is bilinearly sampled at the forward endpoint to compute $e_{ij} = \|\mathbf{F}_{i\to j}(\mathbf{x}) + \mathbf{F}_{j\to i}(\mathbf{x} + \mathbf{F}_{i\to j}(\mathbf{x}))\|_2$. A correspondence is valid when the endpoint lies inside the image, both flows are finite, $e_{ij} \le \tau_{ij} = 0.5 + 0.05\|\mathbf{F}_{i\to j}(\mathbf{x})\|_2$, and $\|\mathbf{F}_{i\to j}(\mathbf{x}) - d_{ij}(\mathbf{x})\Delta\mathbf{a}_{ij}\|_2 \le 0.5$ pixels. Pixels without valid correspondences are discarded before fitting. For the projected disparities $d_{\ell r}$ of Eq. 3, define $b_{\ell r} = \|\Delta\mathbf{a}_{\ell r}\|_2^2$ and initialize $\hat{d}_r^{(0)} = \sum_{\ell\in\mathcal{J}_r} b_{\ell r} d_{\ell r} / \sum_{\ell\in\mathcal{J}_r} b_{\ell r}$. Three iteratively reweighted least-squares (IRLS) updates then give

$$\omega_{\ell r}^{(t)} = b_{\ell r} \min\left(1, \frac{0.5}{\max(|d_{\ell r} - \hat{d}_r^{(t-1)}|, 10^{-8})}\right), \qquad \hat{d}_r^{(t)} = \frac{\sum_{\ell\in\mathcal{J}_r} \omega_{\ell r}^{(t)} d_{\ell r}}{\sum_{\ell\in\mathcal{J}_r} \omega_{\ell r}^{(t)}}, \quad t = 1, 2, 3. \tag{12}$$

For the final estimate $\hat{d}_r = \hat{d}_r^{(3)}$, let $W_r = \sum_{\ell\in\mathcal{J}_r} \omega_{\ell r}^{(3)}$ and $n_r = |\mathcal{J}_r|$. Its uncertainty and confidence are

$$\begin{aligned} \sigma_r^2 &= \frac{\sum_{\ell\in\mathcal{J}_r} \omega_{\ell r}^{(3)} (d_{\ell r} - \hat{d}_r)^2}{W_r} + 10^{-12}, \\ q_r &= \frac{n_r}{3} \exp\left[-\left(\frac{\sigma_r}{0.15}\right)^2\right] \frac{1}{n_r} \sum_{\ell\in\mathcal{J}_r} \exp\left[-\left(\frac{e_{\ell r}}{\tau_{\ell r}}\right)^2\right]. \end{aligned} \tag{13}$$

Here $e_{\ell r}$ and $\tau_{\ell r}$ are the corresponding forward–backward errors and thresholds. Only fits with $n_r \ge 2$ and $W_r \ge 64$ are retained. Disparities and their uncertainties are measured in pixels per angular step. The robust disparity range $[d_{\min}, d_{\max}]$ of Eq. 4 is the 2nd and 98th percentile of the retained disparities. The canonical depth interval is fixed to $[Z_{\text{near}}^c, Z_{\text{far}}^c] = [0.5, 5.0]$. This interval fixes the affine inverse-depth gauge. With consistent camera construction and back-projection, the fitted angular trajectories of the retained tracks are preserved.

**Association and uncertainty fusion.** Before association, we retain positive-confidence tracks with finite canonical depth in $[Z_{\text{near}}^c, Z_{\text{far}}^c]$. Two tracks from different anchors are linked when their reference positions are reciprocal nearest neighbors within one pixel and $|\hat{d}_r - \hat{d}_s| \le 3\sqrt{\sigma_r^2 + \sigma_s^2 + 10^{-12}}$. Each track must also project inside the other track's anchor image. Connected components define candidate groups, including isolated tracks. Within each group, we retain the highest-confidence track from each anchor. Let $\mathcal{G}_m$ denote this retained set and $w_r = q_r / \sum_{s\in\mathcal{G}_m} q_s$. These weights give the fused position, disparity, and color in Sec. 3.3; the fused variance is

$$\bar{\sigma}_{d,m}^2 = \max\left(0, \sum_{r\in\mathcal{G}_m} w_r(\sigma_r^2 + \hat{d}_r^2) - \bar{d}_m^2\right). \tag{14}$$

It includes both within-track uncertainty and disagreement among track estimates. The fused confidence is $\bar{q}_m = 1 - \prod_{r\in\mathcal{G}_m}(1 - \min(q_r, 1 - 10^{-12}))$. We discard a fused track if its projection leaves any contributing anchor image.

**Covariance initialization.** The lifting map in Eq. 8 uses $f = 1.5\max(H, W)$ and $(c_x, c_y) = ((W-1)/2, (H-1)/2)$ for images of size $H \times W$. At a fused track, write $Z = Z_c(\bar{d}_m)$ and suppress the group index on the following derivatives:

$$\mathbf{g}_x = \frac{Z}{f}(1,0,0)^\mathsf{T}, \qquad \mathbf{g}_y = \frac{Z}{f}(0,1,0)^\mathsf{T}, \qquad \mathbf{g}_d = -\alpha_c Z \boldsymbol{\mu}_m. \tag{15}$$

These are partial derivatives with respect to reference-image coordinates and disparity, holding the other inputs fixed. With local disparity slopes $d_x, d_y$, define $\mathbf{t}_x = \mathbf{g}_x + d_x \mathbf{g}_d$ and $\mathbf{t}_y = \mathbf{g}_y + d_y \mathbf{g}_d$. The covariance before eigenvalue clipping is

$$\boldsymbol{\Sigma}_m = \sigma_p^2(\mathbf{t}_x\mathbf{t}_x^\mathsf{T} + \mathbf{t}_y\mathbf{t}_y^\mathsf{T}) + \bar{\sigma}_{d,m}^2 \mathbf{g}_d\mathbf{g}_d^\mathsf{T} + s_{\min}^2 \mathbf{I}_3. \tag{16}$$

Here $\sigma_p = 0.5$ pixels; the minimum Gaussian scale is $s_{\min} = \max(0.02Z/f, 10^{-6})$. To estimate $d_x, d_y$, we round fused reference positions to the nearest pixel and clamp them to the image bounds. At each occupied pixel, disparities are averaged using $\bar{q}_m$, and uncertainty is taken from the highest-confidence track. A central difference uses the two immediate neighbors only when both are valid and their disparity difference satisfies the same $3\sigma$ gate as association. Otherwise, the corresponding slope is zero, including at image boundaries. We clip covariance eigenvalues to $[s_{\min}^2, Z^2/f]$ and use their square roots as Gaussian scales. The eigenvectors form the rotation; a sign flip enforces determinant $+1$ before conversion to a unit quaternion.

**Rendered disparity and supervision mask.** For the Gaussian disparities $d_k$ in Eq. 9, we render the features $(d_k, d_k^2)$ using the same Gaussian rasterizer. At pixel $\mathbf{u}$, let $\alpha_k(\mathbf{u})$ be the opacity contribution, including the projected Gaussian footprint, with Gaussians ordered from front to back. Define $T_k(\mathbf{u}) = \prod_{j<k}(1-\alpha_j(\mathbf{u}))$, $a_{\mathrm{acc}}(\mathbf{u}) = \sum_k T_k(\mathbf{u})\alpha_k(\mathbf{u})$, and $M_j(\mathbf{u}) = \sum_k T_k(\mathbf{u})\alpha_k(\mathbf{u})d_k^j$ for $j = 1, 2$. For finite moments and finite $a_{\mathrm{acc}} > 10^{-8}$, the normalized moments are

$$D(\mathbf{u}) = \frac{M_1(\mathbf{u})}{a_{\mathrm{acc}}(\mathbf{u})}, \qquad V(\mathbf{u}) = \max\left(0, \frac{M_2(\mathbf{u})}{a_{\mathrm{acc}}(\mathbf{u})} - D(\mathbf{u})^2\right). \tag{17}$$

Otherwise, the implementation returns zero for both $D$ and $V$. The targets $\hat{d}(\mathbf{u})$, $\sigma(\mathbf{u})$, and $q(\mathbf{u})$ in Eq. 10 are the fitted maps of the corresponding anchor, before cross-anchor fusion for Gaussian initialization. The set $\mathcal{U}$ contains valid fitted pixels with finite rendered $D, V, a_{\mathrm{acc}}, q \geq 0.95$, $a_{\mathrm{acc}} \geq 0.05$, and $V \leq 0.05$. The confidence gate $q \geq 0.95$ keeps pixels with reliable fitted disparity. The coverage gate $a_{\mathrm{acc}} \geq 0.05$ removes pixels where few Gaussians overlap. The variance gate $V \leq 0.05$ drops pixels whose composited Gaussians disagree in disparity. The variance $V$ is used only for this selection, never as a supervision target. An empty set contributes zero loss.

## B Baseline Evaluation and Reproducibility

**Common evaluation protocol.** We evaluate all methods on the same four LF benchmarks and the same angular reconstruction tasks. For $2 \times 2 \to 7 \times 7$, the four corner views of the central $7 \times 7$ lattice are used as inputs, and the remaining 45 views are held out for evaluation. For $3 \times 3 \to 7 \times 7$, the input views lie on rows and columns $1, 4, 7$, and the remaining 40 views are held out. Every method renders the same held-out views for a given protocol.

The GS-based comparison in Table 1 uses full-frame RGB PSNR, SSIM, and AlexNet LPIPS. Scores are averaged over held-out views within each scene and then equally weighted across scenes. The LF-specific comparison in Table 2 follows the protocol used by EPIT-ASR (Liang et al., 2026): PSNR and SSIM are computed on the cropped Y channel, whereas LPIPS is computed on full-frame RGB images. The crop definition, color conversion, metric implementations, and scene/view aggregation are identical across all LF-specific methods.

**Input format and rectification.** StructLFGS assumes a rectified LF whose sub-aperture views form a regular angular lattice with coplanar centers and parallel optical axes. The four benchmarks provide this format directly or through their released preprocessing. The synthetic HCI scenes are rendered from virtual camera arrays and are rectified by construction. The 30scenes and Occlusions datasets are decoded and rectified from Lytro Illum captures with the released toolbox (Kalantari et al., 2016). For a raw plenoptic capture, plenoptic calibration and rectification are required before applying the method (Dansereau et al., 2013).

**Vanilla 3DGS.** Vanilla 3DGS (Kerbl et al., 2023) obtains cameras through COLMAP (Schönberger & Frahm, 2016). Under the rectified LF inputs used here, the limited inter-view parallax prevents COLMAP from recovering a valid camera configuration for the available input views. We count a scene as a failure when COLMAP does not provide a valid camera for every input view or produces non-finite or otherwise invalid camera parameters. This occurred for all 64 scenes in the four benchmarks (4 HCInew, 5 HCIold, 30 30scenes, and 25 Occlusions), so vanilla 3DGS has no valid result under this input setting.

**3R-GS.** We run the official 3R-GS (Huang et al., 2025) with its released MASt3R-SfM pipeline (Leroy et al., 2024) in place of COLMAP. Input-view cameras are initialized with the released ViT-Large MASt3R checkpoint. The Gaussian scene is optimized jointly with the MLP pose optimizer and epipolar loss for 10,000 steps at the original training resolution. For each held-out view, we retain the released `optim_camtoworlds` procedure for 100 additional steps using that view's RGB image to fit its target camera. Thus, target-view RGB enters the reported 3R-GS evaluation through target-camera adaptation.

**CF-3DGS.** The official CF-3DGS method (Fu et al., 2024) targets dense video, so we adapt its progressive schedule to the sparse-view LF setting. Scene optimization runs for 5,000 iterations on the sparse input views. We then retain the released target-view pose-fitting stage for 8,000 additional iterations using the held-out RGB images. The reported result therefore includes target-camera adaptation under the method's released evaluation procedure.

**Feed-forward baselines.** AnySplat (Jiang et al., 2025) and YoNoSplat (Ye et al., 2026) use their released pretrained weights without per-scene optimization. For AnySplat, the released camera-prediction stage receives the full angular lattice, including the held-out RGB images in our execution, and predicts the camera lattice used for rendering. We report this result with the corresponding target-image input explicitly retained. YoNoSplat predicts cameras from the sparse input views and obtains held-out poses by bilinear interpolation over the $7 \times 7$ lattice, with the orientation fixed to the first input camera; it does not use held-out RGB images. Both methods render the same held-out views. Their image preprocessing, rendering resolution, and released checkpoints are kept fixed across scenes and are reported with the corresponding evaluation scripts.

**LF-specific baselines.** LF-specific baselines receive the same sparse input views and predict the same held-out views. We use their released implementations, checkpoints, and standard preprocessing when available, without providing held-out RGB images as inputs or optimization targets. Following the protocol used by EPIT-ASR (Liang et al., 2026), PSNR and SSIM are computed on the cropped Y channel, whereas LPIPS is computed on full-frame RGB images. The crop definition, color conversion, metric implementations, and scene/view aggregation are kept identical across all LF-specific methods.

**Fairness and interpretation.** The comparison is controlled at the task level: methods use the same datasets, sparse-view protocols, held-out views, and evaluation aggregation. It is not controlled at the information level because some GS baselines use held-out RGB to estimate or adapt target-view cameras, whereas StructLFGS and YoNoSplat do not. We retain these procedures to evaluate the released or executable pipelines under their intended operating modes and disclose them for reproducibility. Accordingly, Table 1 measures practical pipeline performance and does not by itself isolate the effect of the camera-family parameterization. That effect is assessed by the matched component comparisons in Table 3.

## C Comparison with Ground-Truth Camera Calibration

We measure the reconstruction gap between our disparity-derived camera family and a calibrated reference on all four HCInew scenes. The reference uses the published focal length, sensor size, baseline, and focus distance to set $f_{\mathrm{gt}}$, $B_{\mathrm{gt}}$, and $o_{\mathrm{gt}} = f_{\mathrm{gt}} B_{\mathrm{gt}} / Z_{\mathrm{focus}}$; the principal point follows the centered pixel convention. Its depth chart is $Z_{\mathrm{gt}}(d) = f_{\mathrm{gt}} B_{\mathrm{gt}} / (o_{\mathrm{gt}} - d)$. The calibrated and canonical charts assign different depth coordinates to the same fitted disparities. We uniformly rescale the calibrated depths and baselines so that the median calibrated depth matches the median

canonical depth over the retained trajectory samples. A uniform scale preserves every calibrated projection, so this only aligns the depth scale between the two charts. Both variants retain the same four-corner RAFT trajectories, disparity supervision, initialization support, seed, loss schedules, Gaussian capacity, and 10K budget. Initialization is lifted consistently with each camera chart; GT depth is not used. We reevaluate both variants with identical full-frame RGB PSNR, SSIM, and AlexNet LPIPS. Per-view scores are averaged within each scene and then weighted equally across scenes.

Table C.1: Camera calibration reference on HCInew at 10K steps. Both variants use the same estimated trajectories. Metrics are full-frame RGB. Per-scene rows report scene-wise means. LPIPS is scaled by 10. ΔPSNR is calibrated minus ours, computed before rounding.

| | Ours (disparity-derived) | | | GT calibration | | | |
|---|---|---|---|---|---|---|---|
| Scene | PSNR↑ | SSIM↑ | LPIPS↓ | PSNR↑ | SSIM↑ | LPIPS↓ | ΔPSNR |
| $2 \times 2 \rightarrow 7 \times 7$ (45 held-out views) | | | | | | | |
| bedroom | 37.51 | 0.9581 | 0.120 | 37.51 | 0.9569 | 0.119 | +0.00 |
| bicycle | 35.08 | 0.9677 | 0.113 | 35.73 | 0.9713 | 0.100 | +0.66 |
| dishes | 43.65 | 0.9946 | 0.028 | 45.11 | 0.9953 | 0.021 | +1.47 |
| herbs | 37.00 | 0.9618 | 0.127 | 37.68 | 0.9651 | 0.126 | +0.68 |
| **Mean** | **38.31** | **0.9706** | **0.097** | **39.01** | **0.9721** | **0.092** | **+0.70** |
| $3 \times 3 \rightarrow 7 \times 7$ (40 held-out views) | | | | | | | |
| bedroom | 39.33 | 0.9681 | 0.127 | 39.13 | 0.9674 | 0.121 | -0.20 |
| bicycle | 38.71 | 0.9802 | 0.077 | 39.03 | 0.9814 | 0.068 | +0.32 |
| dishes | 48.34 | 0.9963 | 0.013 | 48.85 | 0.9968 | 0.009 | +0.51 |
| herbs | 40.19 | 0.9763 | 0.078 | 40.19 | 0.9763 | 0.096 | -0.01 |
| **Mean** | **41.64** | **0.9802** | **0.074** | **41.80** | **0.9805** | **0.074** | **+0.16** |

**Reconstruction gap.** Table C.1 shows that the mean PSNR gap decreases from 0.70 dB with $2 \times 2$ inputs to 0.16 dB with $3 \times 3$ inputs; the mean SSIM gap decreases from 0.0016 to 0.0002. Calibration yields the largest PSNR gains on *dishes* (1.47/0.51 dB), while *bedroom* is nearly tied with four inputs and favors ours by 0.20 dB with nine inputs. On *herbs*, the nine-input PSNR is also nearly tied, and ours has lower LPIPS (0.078 versus 0.096, both scaled by 10). Thus, calibrated cameras do not uniformly improve every scene or metric under a fixed optimization budget.

The smaller gap with nine inputs is consistent with stronger photometric constraints from the five additional views, while the trajectory estimates and direct disparity targets remain unchanged. These results show that the disparity-derived family is an effective geometric parameterization for novel-view synthesis without metric calibration, with quality close to the calibrated reference under the $3 \times 3$ protocol. The calibrated variant therefore serves as a reconstruction reference under the same optimization budget.

**Visual comparison.** Fig. C.1 compares the same held-out view and crop across both protocols. Both camera variants preserve the main structures; the error maps expose remaining local differences. In the selected *bicycle* crop, the MAE gap narrows from 0.40 to 0.03 intensity levels with nine inputs. The *bedroom* crop has slightly lower error with ours under both protocols. These local observations complement the full-view metrics and illustrate why calibration gains are scene- and metric-dependent.

## D Efficiency Comparison

Table D.1 reports the mean total runtime and peak GPU memory over the four HCInew scenes. For every method, we measure the wall-clock time of the complete per-scene pipeline, including preprocessing, camera estimation, scene initialization and optimization, target-view camera adaptation when applicable, rendering, and metric evaluation. This includes trajectory extraction and cache construction for StructLFGS, MASt3R-SfM correspondence extraction for 3R-GS, and progressive

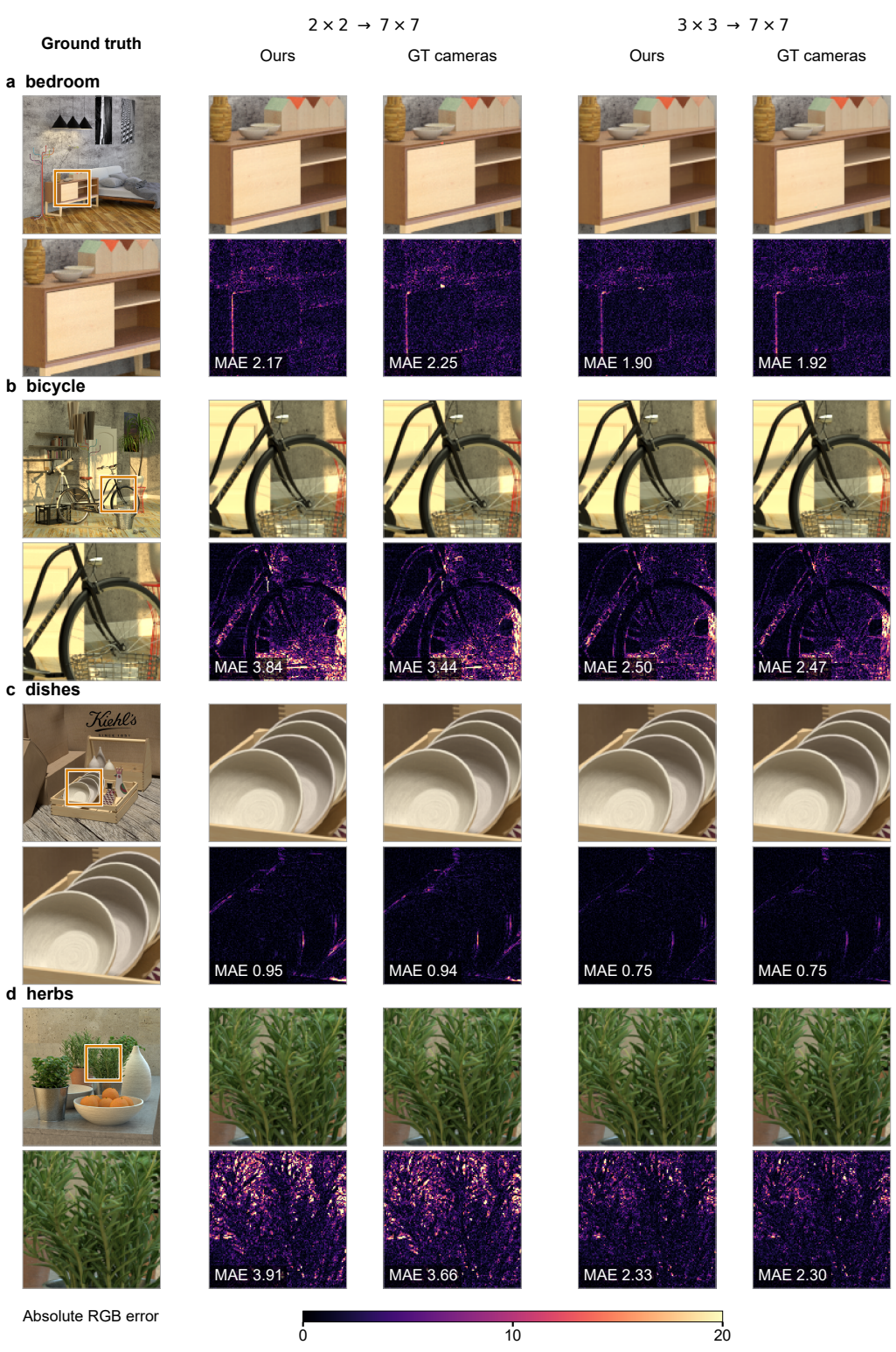


Figure C.1: Camera calibration reference on all four HCInew scenes. Cam050 is held out under both protocols. Each scene shows the GT overview and its marked $128 \times 128$ crop (left), followed by matched reconstructions and error maps. MAE is the mean absolute RGB error on the 0–255 scale; all maps share the 0–20 color range.

Table D.1: Efficiency comparison on HCInew. Total runtime includes all stages of each per-scene pipeline, from preprocessing through target-view rendering and metric evaluation, and is averaged over four scenes. Peak GPU memory is reported in GB.

| Method | Input protocol | Total runtime (s/scene) | Peak GPU memory (GB) |
|---|---|---|---|
| StructLFGS | $2 \times 2 \rightarrow 7 \times 7$ | 190.61 | 1.56 |
| StructLFGS | $3 \times 3 \rightarrow 7 \times 7$ | 199.81 | 1.55 |
| 3R-GS | $2 \times 2 \rightarrow 7 \times 7$ | 670.25 | 3.13 |
| 3R-GS | $3 \times 3 \rightarrow 7 \times 7$ | 610.67 | 3.06 |
| CF-3DGS | $2 \times 2 \rightarrow 7 \times 7$ | 1425.25 | 3.38 |
| CF-3DGS | $3 \times 3 \rightarrow 7 \times 7$ | 1549.50 | 4.01 |

reconstruction and target-view pose optimization for CF-3DGS. All methods are measured on the same NVIDIA GeForce RTX 4090 GPU.

StructLFGS requires $190.61$–$199.81$ s per scene, corresponding to $3.06$–$3.52\times$ and $7.48$–$7.75\times$ speedups over 3R-GS and CF-3DGS, respectively. Its measured peak memory is $1.55$–$1.56$ GB, compared with $3.06$–$4.01$ GB reported for the baselines. Increasing the inputs from four to nine raises optimization time by $4.8\%$. By fixing the structured camera family before Gaussian optimization, StructLFGS removes per-view pose variables from the training loop. The measured costs demonstrate the practical efficiency of this formulation under the corresponding as-run protocols.

## E Hyperparameter Sensitivity

We evaluate the maximum disparity-supervision weight $\gamma_{\max} \in \{0, 0.01, 0.05, 0.10\}$ on the four HCInew scenes under $3 \times 3 \rightarrow 7 \times 7$ reconstruction; $\gamma_{\max} = 0$ disables the loss. All variants share the disparity-derived cameras, four-corner trajectories, initialization, seed 42, and 10K-step budget. Positive weights start at step 1500 and warm up over 1000 steps. The default $0.05$ setting reuses its existing checkpoint. Every setting is evaluated with the same full-frame RGB protocol.

Table E.1: Disparity-supervision control and weight sensitivity on HCInew under $3 \times 3 \rightarrow 7 \times 7$. Results average full-frame RGB metrics over 40 held-out views per scene and then equally over four scenes. LPIPS is scaled by $10$.

| $\gamma_{\max}$ | PSNR↑ | SSIM↑ | LPIPS↓ |
|---|---|---|---|
| 0 (disabled) | 41.32 | 0.9754 | 0.0802 |
| 0.01 | 41.58 | 0.9802 | 0.0742 |
| 0.05 (default) | 41.64 | 0.9802 | 0.0736 |
| 0.10 | 41.59 | 0.9801 | 0.0754 |

Table E.1 shows that all tested positive weights improve all three metrics over the zero-weight control. The default $0.05$ raises PSNR by $0.32$ dB and SSIM by $0.0048$, and reduces LPIPS from $0.0802$ to $0.0736$ ($\times 10$). With cameras and initialization fixed, these gains demonstrate the contribution of disparity supervision during Gaussian optimization.

Across positive weights, mean PSNR varies by only $0.06$ dB, less than the $0.27$–$0.32$ dB gains over the zero-weight control. The default achieves the highest PSNR/SSIM and lowest LPIPS. Thus, disparity supervision improves reconstruction across the tested weights, with limited sensitivity to their precise choice.