UNIVERSITY OF
CAMBRIDGE
Faculty of Computer
Science and Technology

Machine Visual Perception

Course Project Report

Improved 2D Gaussian Splatting Reconstruction from Sparse-view Images Using Diffusion Models

Authors:
Yifei Shi
Kyle Fram
Albert Kwok
Jacob Georgis

Group number: 10

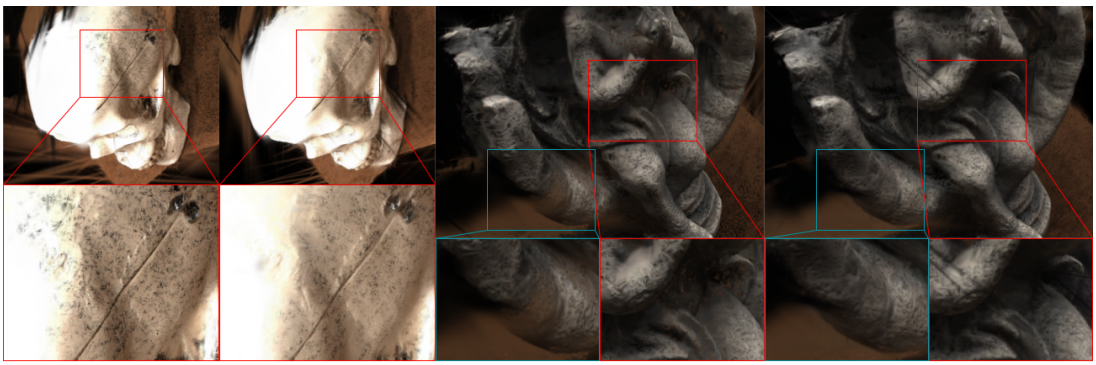


Figure 1: Our Geometric-Constrained Generative Pipeline significantly improves the consistency of side surfaces seen between sparse views in comparison to our baseline, Sparse2DGS [20].

Abstract

Sparse-view 3D reconstruction remains a significant challenge, as limited observational coverage often leads to geometric artifacts and blurred textures in explicit representations like 3D Gaussian Splatting. While recent approaches such as Sparse2DGS introduce geometric regularization, they struggle to synthesize plausible semantic details in occluded regions. Conversely, the naive integration of generative priors often induces structural inconsistencies. To address these limitations, we propose a Geometry-Constrained Generative Enhancement framework. Our pipeline leverages a diffusion-based refinement model to synthesize high-fidelity pseudo-views from interpolated camera poses. Crucially, to prevent geometric drift caused by generative hallucinations, we introduce a hybrid constraint system that enforces active Monocular Depth (DPT) regularization while applying a passive adaptive supervision strategy to filter out unreliable noise. Extensive evaluation on the DTU Sparse-View benchmark demonstrates that our approach significantly enhances both perceptual fidelity and structural integrity. Quantitatively, we report a 10.2% reduction in LPIPS on observed views, alongside a 2.64% improvement in Local Geometric Smoothness, validating the effective suppression of floating artifacts and the densification of the reconstructed surface. Code is available at https://github.com/MKiyoaki/l335-project.

Chapter 1: Introduction and Motivation

1.1: Introduction to the Problem

Passive 3D reconstruction from 2D imagery enables geometric recovery without relying on active sensors like LIDAR. Due to their flexibility, these techniques have become pivotal in domains such as VR/AR and robotics, which simultaneously demand high computational efficiency and reconstruction fidelity [16].

Conventionally, approaches towards geometry reconstruction, such as structure-from-motion (SfM) [12] and multi-view stereo (MVS) [4], rely on identifying corresponding image features (e.g., via SIFT [8]) to establish depth. However, a fundamental limitation of these methods is their strict reliance on dense observational coverage; specifically, they require high overlap between frames to ensure sufficient feature matching. This requirement renders them impractical in dynamic scenarios, such as autonomous driving or mobile robotics, where data acquisition is inevitably constrained by bandwidth or physical movement limitations [28].

To overcome these coverage limitations, recent advancements have shifted towards deep-learning-based approaches [17]. Initially, Neural Radiance Fields (NeRF) [9] revolutionized view synthesis; more recently, however, explicit representations such as 3D Gaussian Splatting (3DGS) [5] have emerged as a powerful alternative. Unlike implicit fields, 3DGS represents geometry using anisotropic Gaussians, allowing for real-time differentiable rendering and explicit structural manipulation.

Despite these algorithmic advancements, a critical research gap remains in the regime of sparse-view reconstruction (typically fewer than 5 views). While 3DGS excels with dense inputs, its performance degrades rapidly when observations are sparse: the optimization tends to overfit to the few available training views, inevitably resulting in floating artifacts, incorrect depth placement, and blurred textures in unobserved regions. In response, recent works like Sparse2DGS [20] attempt to mitigate this by flattening Gaussians into 2D disks to enforce a planar regularization. Nevertheless, geometric regularization alone is insufficient to hallucinate missing semantic details in large occluded areas, leaving the fundamental problem of information scarcity unresolved.

This persistent challenge motivates the integration of generative priors. Generative models, trained on massive datasets, encapsulate rich geometric and appearance knowledge that can potentially fill the gaps left by sparse sensors. In this work, we propose a framework designed to bridge this gap. By leveraging generative diffusion models, we aim to synthesize novel views that preserve geometric consistency, thereby enabling robust reconstruction even in the absence of dense observational coverage.

1.1.1 Approach and Contribution

To effectively address the limitations of sparse-view reconstruction, we propose a framework that bridges the gap between explicit 3D reconstruction and the high-fidelity priors of generative diffusion models. Our core insight is that while generative models possess a rich understanding of scene statistics, they are prone to hallucinating geometrically inconsistent artifacts. Consequently, naively integrating these priors often destabilizes the reconstruction.

Guided by this insight, we adopt 2D Gaussian Splatting as our baseline and introduce a **Geometry-Constrained Generative Pipeline**. Functionally, we first employ an error-driven strategy to identify key frames with noticeable artifacts. Then, we generate pseudo-views via manifold-preserving camera pose interpolation and employ a diffusion-based refinement model (Difix3D [19]) to synthesize consistent imagery. Crucially, to prevent geometric collapse from hallucinated inconsistencies, we introduce a Hybrid Geometric Constraint Algorithm. This system enforces structural integrity using an active Monocular Depth (DPT) [13] prior while simultaneously applying a passive adaptive supervision strategy (loss decoupling and re-weighting) to filter out unreliable generative noise.

The main contributions of this project are summarized as follows:

1. We propose a complete generative data augmentation pipeline for Sparse-View 2D Gaussian Splatting, utilizing an error-driven strategy to target weak viewpoints for diffusion-based refinement.

2. We introduce a Hybrid Geometric Constraint Strategy that combines active Monocular Depth (DPT) regularization with a passive adaptive weighting scheme. This effectively bridges the gap between hallucinated 2D priors and consistent 3D geometry, preventing the structural drift often observed in naive generative augmentation.

3. We demonstrate through quantitative and qualitative experiments that our approach enhances perceptual fidelity and geometric stability. Specifically, our method reduces perceptual artifacts (improving LPIPS) and suppresses geometric noise (e.g., floaters) compared to the sparse baseline, validating that our constraints successfully integrate generative priors without disrupting the alignment with input observations.

1.2: Related Works

Neural Radiance Fields under Sparse Views The field of novel view synthesis was originally revolutionized by Neural Radiance Fields (NeRF) [9], which utilize implicit neural representations to achieve photorealistic rendering. Despite its capability, standard NeRF relies heavily on dense input views; consequently, when constrained to sparse observations (e.g., fewer than 3 views), it suffers from severe overfitting and geometric collapse. To mitigate this issue, subsequent works introduced explicit regularizations. For instance, RegNeRF [10] utilizes a depth-smoothness regularization on unseen views, while SparseNeRF [18] distills robust depth rankings to guide the geometry. Nevertheless, these implicit methods generally suffer from slow inference speeds and limited editability. More critically, while they successfully regularize existing geometry, they fundamentally lack the generative capability to hallucinate plausible high-frequency textures in large occluded regions—a limitation our explicit framework addresses by integrating diffusion priors.

Gaussian Splatting and Geometric Regularization To address the efficiency bottlenecks of implicit representations, 3D Gaussian Splatting (3DGS) [5] has emerged as a powerful alternative, enabling real-time rendering via explicit anisotropic Gaussians. However, similar to NeRF, standard 3DGS tends to generate "floaters" in the sparse-view regime due to insufficient constraints. Building on this, Sparse2DGS [20] (our backbone) addresses the instability by flattening Gaussians into 2D surfels and integrating priors from unsupervised Multi-View Stereo networks [21]. Although Sparse2DGS effectively enforces metric geometry through surface alignment, it remains bound by the information present in the input images. Unlike our approach, it cannot synthesize missing semantic details in blind spots, inevitably resulting in structurally accurate but texturally blurred reconstructions in unobserved areas.

Generative Priors for 3D Reconstruction Motivated by the need to compensate for information scarcity, recent research has shifted towards integrating 2D diffusion models into 3D pipelines. Approaches such as Gen-3D [22] attempt to learn scene distributions directly, while Kong et al. [7] propose an iterative framework (GS-GS) that cycles between generative hallucination and Gaussian optimization. Complementary to these, Difix3D [19] specifically focuses on removing rendering artifacts via image-to-image translation. Crucially, however, a key limitation persists in these generative approaches: the risk of geometric inconsistency, where hallucinated details conflict with the physical 3D structure. Distinguishing itself from these methods, our work introduces a hybrid constraint system—specifically utilizing active Monocular Depth (DPT) priors and decoupling MVS losses—to strictly ground these generative hallucinations within a consistent 3D geometry, thereby preventing the structural drift observed in naive augmentation.

1.3: Overview of the Project

Figure 1.1 provides a high-level roadmap of our proposed system, illustrating the closed-loop workflow from baseline diagnosis to generative augmentation and final integration. Guided by this architecture, the remainder of this report is organized as follows:

Chapter 2 details the technical implementation of our Geometry-Constrained Generative Pipeline. We first introduce the Sparse2DGS baseline algorithm. Subsequently, we elaborate on our algorithmic improvements, including the error-driven key frame identification, diffusion-based pseudo-view synthesis, and the core Hybrid Geometric Constraint Algorithm designed to stabilize optimization.

Chapter 3 presents the empirical validation of our framework. We describe the experimental setup using the DTU sparse-view benchmark and analyze the training results. This chapter provides a comprehensive quantitative comparison focusing on fitting stability and perceptual quality, supported by qualitative visualizations and rigorous ablation studies.

Chapter 4 summarizes our findings regarding the role of geometric constraints in generative augmentation. We also discuss current limitations, such as texture hallucination shifts, and propose potential avenues for future research in texture synthesis and depth refinement.

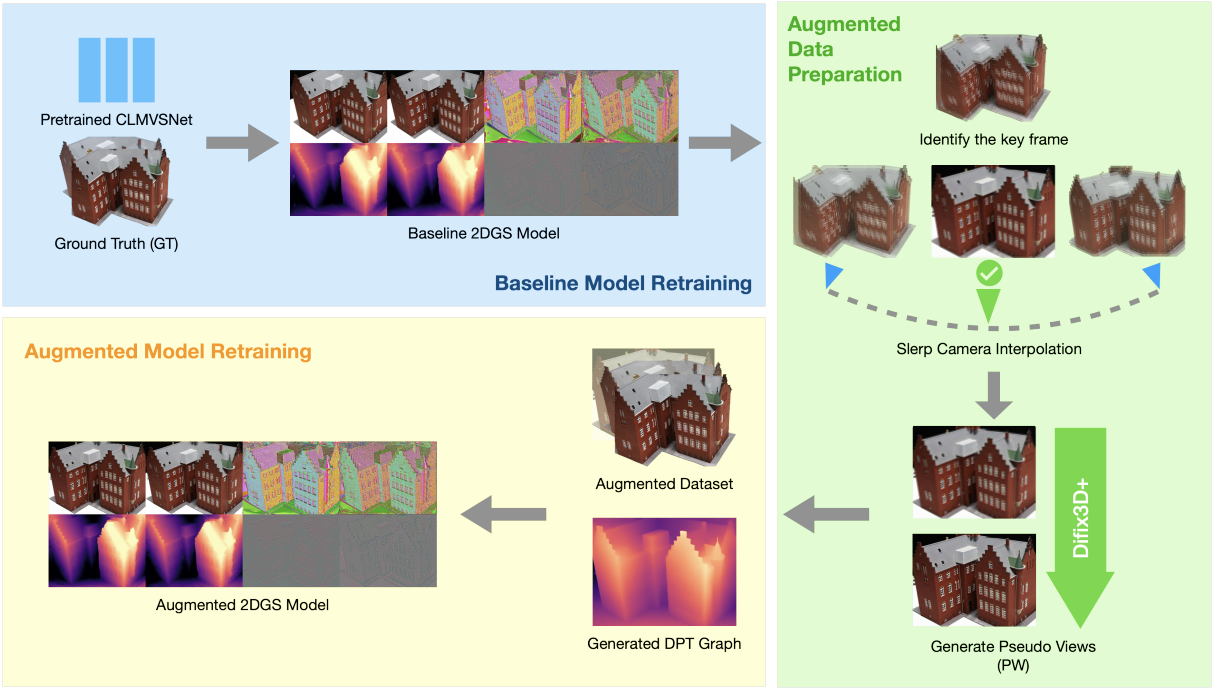


Figure 1.1: **System Architecture Overview.** The workflow follows a cyclic process involving Baseline Diagnosis (identifying artifacts), Geometry-Aware Generative Augmentation (synthesizing pseudo-views with DPT priors), and Dual-Branch Integration (retraining with adaptive supervision).

Chapter 2: Methodology

2.1: Baseline Algorithm

We adopt Sparse2DGS [20] as our backbone framework. This method specifically addresses the aforementioned shortcomings of sparse-view synthesis by incorporating geometric priors from a pre-trained unsupervised Multi-View Stereo (MVS) network [27, 21] into the 2D Gaussian Splatting pipeline. An overview of the system is illustrated in Figure 2.1.

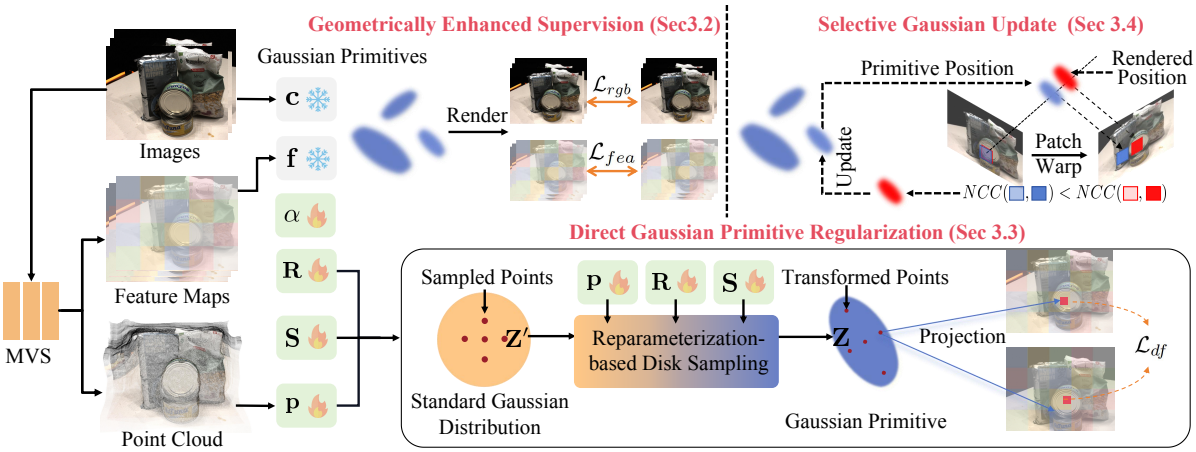


Figure 2.1: Overview of the Sparse2DGS system. Source [20].

2.1.1 2D Gaussian Primitive Representation

Unlike standard 3DGS [5], which models the scene using volumetric ellipsoids, Sparse2DGS utilizes oriented planar disks (surfels) to represent scene geometry. This explicit surface representation is particularly advantageous for enforcing geometric consistency in data-scarce regimes. Formally, each primitive is explicitly parameterized by:

- **Center Position:** $\mu \in \mathbb{R}^3$, initialized from the sparse MVS point cloud.
- **Orientation:** Defined by a local tangent frame with axes $\mathbf{t}_u, \mathbf{t}_v$ and a normal vector $\mathbf{t}_n = \mathbf{t}_u \times \mathbf{t}_v$. These vectors form a rotation matrix $\mathbf{R} = [\mathbf{t}_u, \mathbf{t}_v, \mathbf{t}_n]$.
- **Scaling:** A 2D scaling vector $\mathbf{s} = (s_u, s_v)$ controlling the spatial extent of the splat on the tangent plane.

Based on these parameters, a point on the primitive's local tangent plane in world space is defined as:

$$P(u, v) = \mu + s_u \mathbf{t}_u u + s_v \mathbf{t}_v v \tag{2.1}$$

where the opacity follows a 2D Gaussian distribution:

$$\mathcal{G}(u, v) = \exp\left(-\frac{u^2 + v^2}{2}\right) \tag{2.2}$$

Crucially, distinct from the rasterization-based projection used in 3DGS, Sparse2DGS employs a differentiable ray-splat intersection mechanism. This modification allows for precise depth and normal rendering, which is essential for aligning the geometry with MVS priors [20].

2.1.2 Unsupervised MVS Backbone: CL-MVSNet

To provide geometric supervision in the absence of ground-truth depth, the framework integrates CL-MVSNet [21]. Conventionally, unsupervised MVS methods [6] rely on photometric consistency—warping images between views based on estimated depth and minimizing colour difference. However, this assumption often fails in regions with low texture, repetitive patterns, or view-dependent effects such as specular highlights.

CL-MVSNet mitigates these limitations by incorporating Contrastive Learning. Instead of relying solely on pixel-wise colour matching, it learns robust feature representations that are invariant to lighting changes and viewpoint variations. Specifically, the network employs a dual-branch strategy:

1. **Image-Level Branch:** Utilizes masking to encourage the network to exploit contextual information, improving performance in textureless regions.

2. **Scene-Level Branch:** Enforces consistency across different source views of the same scene, guiding the network to learn geometry rather than overfitting to lighting artifacts.

In our pipeline, we utilize the pre-trained, frozen CL-MVSNet to generate pseudo-depth maps (D_{mvs}) and pseudo-normal maps (N_{mvs}), which serve as dense geometric priors for the Gaussian optimization.

2.1.3 Geometric Regularization and Optimization

Sparse-view reconstruction suffers from inherent geometric ambiguity, often leading to floaters or depth collapse when optimizing standard photometric losses alone. To counteract this ambiguity, the baseline imposes explicit constraints derived from the CL-MVSNet priors.

The optimization minimizes a composite loss function:

$$\mathcal{L}_{total} = \mathcal{L}_{rgb} + \lambda_d \mathcal{L}_{depth} + \lambda_n \mathcal{L}_{normal} + \lambda_f \mathcal{L}_{fea} \tag{2.3}$$

The terms are defined as follows:

- $\mathcal{L}_{rgb}$: Measures the photometric reconstruction error between rendered images and ground truth.

- $\mathcal{L}_{depth}$: Enforces consistency between the rendered depth maps (via ray-splat intersection) and the pseudo-depth priors D_{mvs}.

- $\mathcal{L}_{normal}$: Aligns the accumulated surface normals of the Gaussian primitives with the pseudo-normals N_{mvs}, ensuring smooth and accurate surface orientation.

- $\mathcal{L}_{fea}$: Enforces semantic consistency using rendered MVS features to prevent overfitting in textureless regions.

Collectively, this regularization ensures that the 2D Gaussian primitives adhere to a plausible scene geometry, even in areas with sparse observational coverage.

2.2: Algorithm Improvements

In this section we will outline our algorithmic improvements. We utilised the Difix3D [19] model to enhance the synthesis of novel views for added supervision. The model allowed us to remove artefacts from rendered pseudo-views, enabling the generation of clean frames to enhance degenerate ones. Additionally, the DPT [13] model was utilised to generate depth information from our pseudo-views, allowing for more robustness against geometric degeneration from hallucinated inconsistencies. Our approach involves identifying degenerated frames using perceptual metrics (such as PSNR), interpolating views approaching these degenerated frames, synthesise clean pseudo-observations through applying Difix to the rendered image, and retraining our augmented dataset with additional supervision from the DPT model.

2.2.1 Camera Interpolation

Our camera interpolation algorithm for determining camera poses for pseudo-observations utilises a combination of standard Linear Interpolation (LERP) for translation, and Spherical Linear Interpolation (SLERP) for rotation.

- **Linear Interpolation (LERP):** For the translational component, we assume the camera moves along a straight path between keyframes. Given two camera centers $\mathbf{t}_0$ and $\mathbf{t}_1$, and an interpolation factor $\alpha \in [0, 1]$, the intermediate translation $\mathbf{t}_\alpha$ is calculated as:

$$\mathbf{t}_\alpha = (1 - \alpha)\mathbf{t}_0 + \alpha\mathbf{t}_1 \tag{2.4}$$

- **Spherical Linear Interpolation (SLERP):** Linear interpolation for rotation matrices often distorts the resulting rotation structure, causing "jerky" and inconsistent movement. Instead, we represent rotations as unit quaternions, and rotate with a constant angular velocity along an arc. Given quaternions $\mathbf{q}_0$ and $\mathbf{q}_1$, the interpolated quaternion $\mathbf{q}_\alpha$ is given by:

$$\mathbf{q}_\alpha = \frac{\sin((1 - \alpha)\theta)}{\sin(\theta)}\mathbf{q}_0 + \frac{\sin(\alpha\theta)}{\sin(\theta)}\mathbf{q}_1 \tag{2.5}$$

  where $\theta = \arccos(\mathbf{q}_0 \cdot \mathbf{q}_1)$ is the angle subtended by the arc between the two quaternions.

2.2.2 Diffusion Models: Difix3D

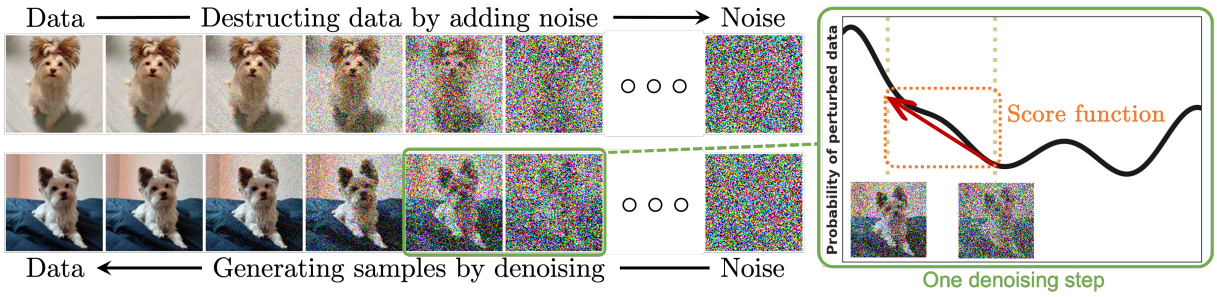


Figure 2.2: Overview of a typical diffusion model. Source [26].

Diffusion models are models which learn to *denoise* input into data. During training, noise progressively gets added to the data (usually Gaussian noise). The parameters of the model are then optimised with a *denoising score matching objective*:

$$\mathcal{L}_{\text{simple}} = \mathbb{E}_{x_0,\epsilon\sim\mathcal{N}(0,\mathbf{I}),t}\left[\|\epsilon - \epsilon_\theta(x_t, t)\|^2\right] \tag{2.6}$$

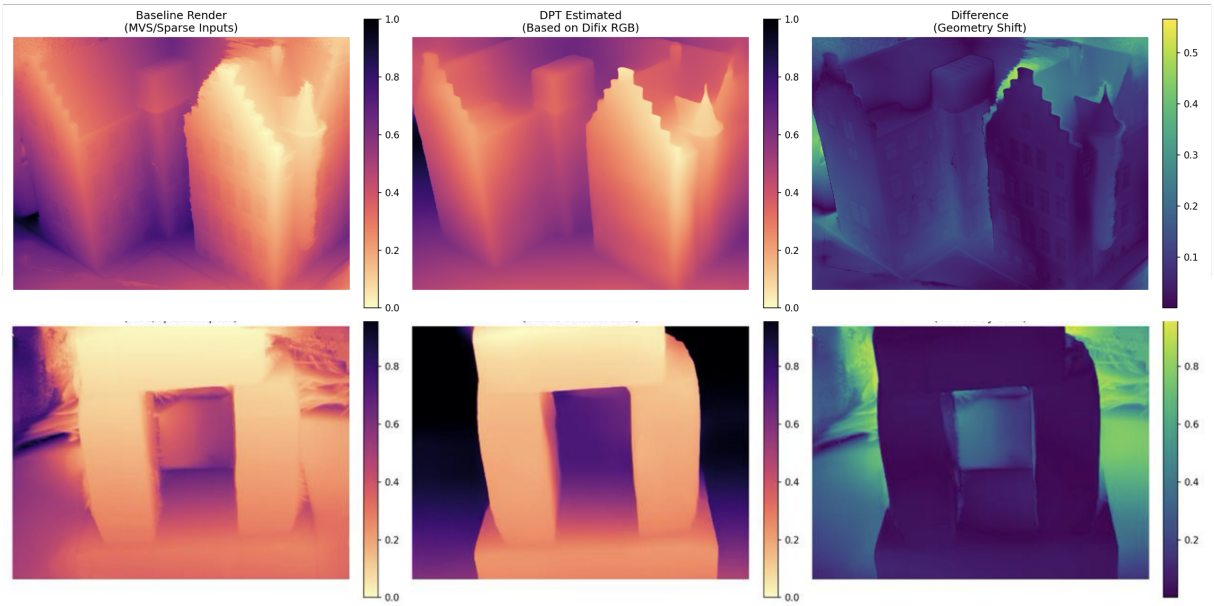


Figure 2.3: **Qualitative comparison of depth priors on Scene 24 (Top) and Scene 40 (Bottom). Left:** The baseline rendered depth derived from sparse MVS inputs. Note the noisy gradients and lack of structural coherence. **Middle:** The monocular depth estimated by DPT from the refined pseudo-view. **Right:** The difference map highlighting the geometric shift. While MVS captures high-frequency details (e.g., window edges), it often fails to maintain global geometric structure in sparse settings. In contrast, DPT provides a smooth, plausible surface prior that guides the reconstruction.

Difix is one such model, based on a pre-trained SD-Turbo (Stable Diffusion) [15] and fine-tuned using degraded rendered images. Difix formulates the artefact removal as a single-step diffusion denoising, outputting a cleaned image for the purpose of improved novel-view synthesis.

2.2.3 Geometric Guidance and Adaptive Supervision

The primary challenge in integrating generative priors is that the high-fidelity textures hallucinated by Difix may contain structural inconsistencies that conflict with the established multi-view geometry (MVS) of the scene. If pseudo-observations are treated as ground truth, they introduce noise that can lead to geometric collapse (floaters or fragmentation) during optimization. We visualize the qualitative difference between the sparse MVS depth and our DPT-estimated prior in Figure 2.3.

To mitigate this, our framework employs a **Hybrid Geometric Constraint Algorithm** that combines active regularization with passive adaptive supervision. This system consists of two complementary strategies:

1. **Active Geometric Constraint (DPT Prior):** We utilize a Dense Prediction Transformer (DPT) [13] to estimate monocular depth maps for each pseudo-view. This depth information acts as a structural prior, guiding the 2D Gaussian primitives to align onto a continuous, physically plausible surface.

   Following the depth regularization strategy proposed by Kong et al. [7], we enforce this geometric constraint using a structural similarity (SSIM) loss. Specifically, we define the loss as the standard structural dissimilarity (D-SSIM) between the rendered and aligned depths:

$$\mathcal{L}_{DPT-DSSIM} = \frac{1 - \text{SSIM}(D_{render}, D_{aligned})}{2} \tag{2.7}$$

where D_{render} is the rasterized depth map, and $D_{aligned}$ is the estimated monocular depth scaled to the scene metric.

2. **Passive Geometric Constraint (Adaptive Supervision):** To prevent the model from optimizing against unreliable geometric signals in the hallucinated views, we implement a dual-pronged adaptive strategy:

   - **MVS Loss Decoupling:** We intentionally *disable* geometric consistency losses derived from the MVS backbone (e.g., normal loss $\mathcal{L}_{normal}$ and MVS-depth loss $\mathcal{L}_{depth}$) for pseudo-views. This prevents erroneous gradients since these views lack true multi-view consistency.
   - **Uncertainty-Aware RGB Weighting:** We down-weight the photometric loss ($\mathcal{L}_{rgb}$) for pseudo-views. This acknowledges the inherent error (aleatoric uncertainty) in generative textures, allowing the model to learn colour style without overfitting to pixel-misaligned artifacts.

This specialized dual-branch optimization scheme ensures that the model leverages the perceptual quality of Difix3D without compromising the geometric integrity maintained by the Sparse2DGS backbone. The overall loss function is detailed fully in Section 2.5

2.3: Implementation Details

To ensure reproducibility and modularity, we implemented a unified experimentation framework based on `Hydra` [23]. The system architecture follows an orchestrator-worker pattern, where a central main script dispatches tasks to specialized runner classes.

Auxiliary Tooling Suite To support the closed-loop data pipeline, we developed a comprehensive suite of auxiliary utilities. This includes a keyframe extraction program that parses evaluation logs to identify viewpoints with high reconstruction error, and a manifold interpolator that computes intermediate camera poses using spherical and linear interpolation. We also created specialized inference scripts that wrap the Difix3D and DPT models to batch-process raw renders into high-fidelity RGB and depth maps. Finally, we implemented a dataset augmentation script that merges the synthesized pseudo-views with the original sparse dataset, handling file renaming and format standardization to prepare the hybrid dataset for retraining.

Modular Runner Architecture We encapsulated the distinct software environments of Sparse2DGS and Difix3D into separate runner modules named SparseRunner and DifixRunner. These runners inherit from an abstract BaseRunner and operate by constructing and executing command-line instructions via subprocess calls. This design effectively isolates dependency conflicts, such as incompatible CUDA versions, and enables the flexible scheduling of training, rendering, and inference tasks through a unified interface by dynamically building execution commands.

Configuration and Data Flow Management All experimental hyperparameters and path configurations are managed via hierarchical yaml files. We implemented an automated data flow alignment mechanism where output paths from upstream stages are dynamically passed as input arguments to downstream stages. This ensures pipeline continuity without manual intervention, allowing the system to seamlessly transition between baseline training, generative augmentation, and final retraining phases based on the passed configuration context.

Granular Evaluation Interface We modified the external interface of the Sparse2DGS codebase to enhance diagnostic capabilities. We added functionality to perform per-frame metric evaluation, allowing the system to compute and log performance statistics for every individual frame across all scenes rather than just global averages. Additionally, we implemented custom data loaders capable of ingesting arbitrary interpolated camera poses, enabling the rendering of specific diagnostic views defined by our interpolation algorithm outside the standard training or testing splits.

Adaptive Training Logic Most importantly, we refactored the core training loop of the baseline model to support our geometric guidance and adaptive supervision strategy. We introduced logic to explicitly distinguish between real and pseudo views within a training batch. This modification enables the selective application of the dual-branch loss function, specifically allowing the system to disable MVS gradients and enforce DPT-based geometric constraints exclusively on the generated pseudo-views during the optimization process.

2.4: Data Pipelines

This section details the algorithmic implementation of the **Augmented Data Preparation** module described from Section1.3. The pipeline transforms the coarse baseline reconstruction into a high-quality augmented dataset through four sequential steps:

Metric-Guided Keyframe Identification Instead of random sampling, we employ an error-driven selection strategy. We perform a diagnostic rendering of the baseline model and compute perceptual metrics (LPIPS, SSIM, PSNR) against the training views. Frames exhibiting the highest reconstruction error (e.g., lowest PSNR) are flagged as *Key Frames*, indicating regions where the geometry is most deficient.

Manifold-Preserving Interpolation To generate novel viewpoints around these key frames, we compute target poses P_{new}. We utilize **Spherical Linear Interpolation (Slerp)** for rotation quaternions to ensure smooth angular transitions and linear interpolation (Lerp) for translation vectors. This ensures the virtual camera moves along a plausible orbital path.

Generative Pseudo-View Synthesis This stage synthesizes the training targets. The interpolated poses are first rendered by the baseline model to produce raw, artifact-prone images. These are then processed by Difix3D [19], which acts as an image-to-image translation model to denoise artifacts and hallucinate realistic textures.

Depth Estimation and Dataset Construction To enforce geometric consistency during retraining, RGB supervision alone is insufficient. We generate corresponding geometry for the pseudo-views using a Monocular Depth Estimator (DPT). To align the scale-invariant DPT output with the metric scale of the scene, we apply a robust affine alignment based on median scaling before adding the pair to the final Augmented Dataset.

2.5: Training Procedures

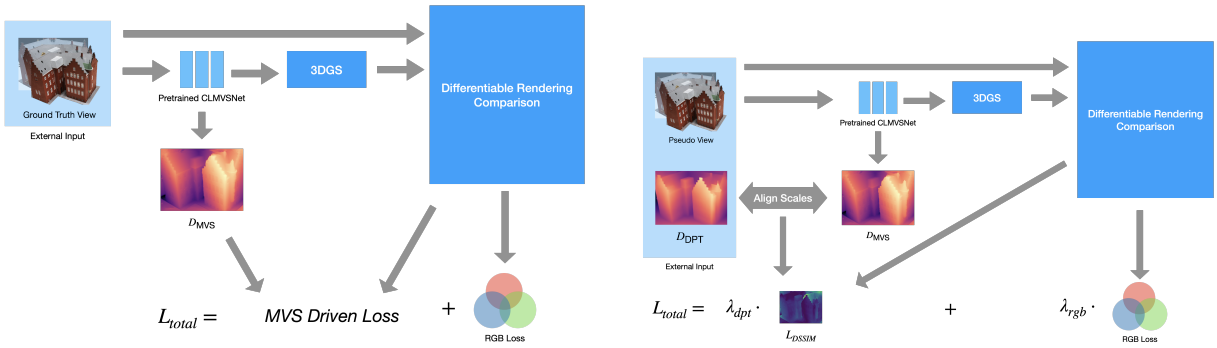
2.5.1 General Procedures

We implemented our training pipeline using the PyTorch framework, building upon the official Sparse2DGS codebase [20]. The optimization of 3D Gaussian primitives is performed using the Adam optimizer with $\beta_1 = 0.9$ and $\beta_2 = 0.999$. Following standard protocols [5], we employ an exponentially decaying learning rate scheduler for the position parameters.

The configuration for Sparse2DGS model training, evaluation and inference are kept at their default hyperparameters. We leverage the training experiment under 7,000 iterations with resolution downsampling scale factor as 2, for both training and evaluation.

2.5.2 Augmented Model Training Procedures

To realize the **Hybrid Geometric Constraint Algorithm** proposed in Section 2.2.3, we engineered a dynamic loss calculation logic within the training loop. The total objective function $\mathcal{L}_{total}$ is computed per iteration by detecting the source of the current viewpoint (Real vs. Pseudo). As illustrated in Figure 2.4, we employ distinct optimization pathways depending on the reliability of the geometric signal.



(a) **Ground Truth View Supervision** (b) **Pseudo View Supervision**

Figure 2.4: **Comparison of optimization strategies. (a)** For real views, the model follows the standard Sparse2DGS pipeline, utilizing MVS-derived normal and feature consistency for geometric regularization. **(b)** For generative pseudo-views, we decouple the potentially erroneous MVS constraints and introduce $\mathcal{L}_{DPT-DSSIM}$ instead.

We differentiate the supervision signals using a dynamic weighting scheme. The total loss is formulated as:

$$\mathcal{L}_{total} = \lambda_{rgb}\mathcal{L}_{photo} + \lambda_{dpt}\mathcal{L}_{DPT-DSSIM} + \mathbb{1}_{real} \cdot (\lambda_n\mathcal{L}_{normal} + \lambda_d\mathcal{L}_{dist} + \lambda_f\mathcal{L}_{fea}) \tag{2.8}$$

where $\mathbb{1}_{real}$ is an indicator function that equals 1 if the current view is a ground-truth image and 0 if it is a generated pseudo-view. This effectively deactivates MVS-dependent losses $(\mathcal{L}_{normal}, \mathcal{L}_{dist}, \mathcal{L}_{fea})$ for pseudo-views to prevent geometric collapse caused by texture-geometry misalignment. Based on the strategies defined in Section 2.2.3, the terms are applied as follows:

- **Ground Truth View Supervision ($\mathbb{1}_{real} = 1$).** For ground truth images, we maintain full MVS consistency. The MVS-derived losses $(\mathcal{L}_{normal}, \mathcal{L}_{dist}, \mathcal{L}_{fea})$ are active, anchoring the geometry to the reliable dense priors provided by CL-MVSNet. The RGB weight is set to standard $\lambda_{rgb} = 1.0$.
- **Pseudo View Supervision ($\mathbb{1}_{real} = 0$).** For generated views, the passive decoupling strategy is triggered. The indicator function $\mathbb{1}_{real}$ zeros out all MVS-derived losses to prevent geometric conflict. Simultaneously, the active DPT constraint $\mathcal{L}_{DPT-DSSIM}$ is enabled with weight λ_{dpt} to enforce smoothness. To handle generative uncertainty, the photometric weight is down-scaled to $\lambda_{rgb} = 0.1$.

This implementation ensures that the model learns high-frequency textures from the diffusion priors while relying strictly on the MVS backbone and DPT smoothness priors for geometric stability.

2.6: Testing and Validation Procedures

Training Validation Validation during training was performed by outputting a map showcasing the rendered image and geometric data after every 500th iteration, based from the original Sparse2DGS source code. This visualisation allows us to keep track of the rendered image output, as well as depth calculations.

Testing After training, we evaluate the resulting scene reconstruction quality. This is done through rendering the model at the camera poses we trained with using the aforementioned (Section 2.6.1) PSNR, SSIM, and LPIPS metrics, collecting an average as a representation for the scene.

2.6.1 Evaluation Metrics

We quantify the reconstruction frame quality through perceptual metrics in comparison to a ground truth. The following metrics are calculated during the evaluation stage:

1. **Peak Signal-to-Noise Ratio (PSNR):** PSNR determines the ratio between maximum signal power and the power of the noise. Higher PSNR indicates less noise distortion in the resulting image. It measures per-pixel differences but is unaffected by structural semantic information.

2. **Structural Similarity Index Measure (SSIM) [11]:** SSIM determines the similarity between two images based on luminance, contrast, and structural information. Producing scores in the [-1,1] range, it better aligns with human perception by accounting for structural correlations rather than just absolute errors.

3. **Learned Perceptual Image Patch Similarity (LPIPS) [29]:** LPIPS analyzes similarity through deep neural network feature extraction, trained on human judgement data. Lower LPIPS scores indicate higher perceptual similarity, effectively capturing high-frequency textural details that PSNR might miss.

4. **Local Geometric Smoothness [14]:** To evaluate the 3D structural integrity in the absence of ground truth point clouds, we introduce Local Smoothness as a non-reference geometric proxy. Defined as the mean Euclidean distance between each reconstructed point and its k-nearest neighbours ($k = 6$), this metric quantifies surface density. Lower values indicate a more compact and continuous surface with reduced high-frequency geometric noise (e.g., floaters), reflecting the effectiveness of our geometric constraints.

These metrics collectively enable us to judge not only the photometric fidelity of the renders but also the geometric stability of the underlying 3D reconstruction.

Chapter 3: Experiments and Evaluation

3.1: Datasets

We evaluate our framework on the **DTU Sparse-View Benchmark**, utilizing the specific split curated by Wu et al. [20]. Derived from the original DTU Large-Scale MVS dataset [3], this subset is designed to benchmark reconstruction algorithms under conditions of extreme data scarcity. An example can be found at Figure 3.1.

The dataset comprises 15 object-centric scenes exhibiting diverse geometric complexities and material properties, ranging from diffuse to specular surfaces. Crucially, the image data is provided in RGBA format with transparent backgrounds, facilitating the precise isolation of the foreground object from environmental clutter. Each scene contains sparse observations of the same target object from varying angles, accompanied by pre-aligned camera parameters. In our experimental setting, the training input is strictly limited to 3 sparse views. This setup presents a significant challenge, as the wide baseline and lack of overlap make it difficult for traditional methods to recover complete geometry.

For data augmentation, we generate one pseudo-view per key frame using the Difix3D inference pipeline (identified via the error-driven strategy described in Section 2.4). We adopt *199 iteration steps* as the standard configuration, adhering to the default protocol established by the original Difix3D implementation [19]. We empirically validated this choice as a trade-off between refinement quality and geometric fidelity. As demonstrated in Figure 3.2, fewer steps fail to fully denoise the artifacts from the raw render, while excessive steps risk deviating too far from the original scene structure. The 199-step setting consistently yields high-fidelity textures while preserving the underlying geometry necessary for 3D consistency.

3.2: Training and Testing Results

This section presents the quantitative and qualitative results that validate the effectiveness of our **Geometry-Constrained** Generative Enhancement framework. Our evaluation is two-fold: first, we compare the overall performance of our proposed method against the baseline model; second, we conduct a detailed ablation study to isolate the contribution of two key components—the diffusion-based texture enhancement (i.e., Difix3D) and the **hybrid geometric constraint strategy**, which encompasses both DPT regularization and adaptive supervision.

The quantitative results are analyzed via standard novel view synthesis metrics (PSNR, SSIM, LPIPS), while the qualitative analysis focuses on geometric fidelity through depth and normal map visualizations. The Figure 3.3 shows the baseline loss evolution.

3.2.1 Ablation Study Setup

To rigorously test the necessity of each component in our closed-loop pipeline, we define four distinct experimental conditions. All models are trained under the same initial conditions, augmentation, and retraining schedule.

- **BL: Baseline (MVS Only)**

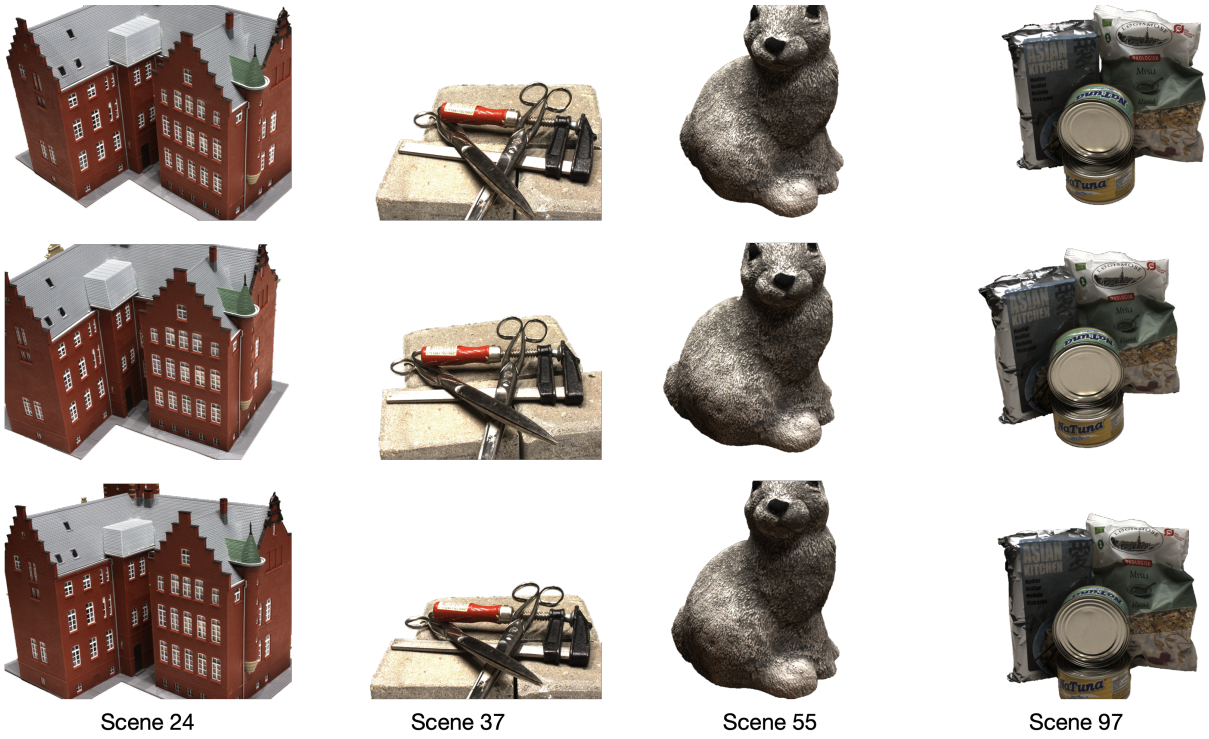


Figure 3.1: **Sample input views from the DTU sparse dataset.** The figure displays the three Ground Truth training views for Scene 24 (Red Brick House), Scene 37 (Tools), Scene 55 (Rabbit), and Scene 97 (Food), illustrating the diversity of objects and the sparsity of the provided angles.

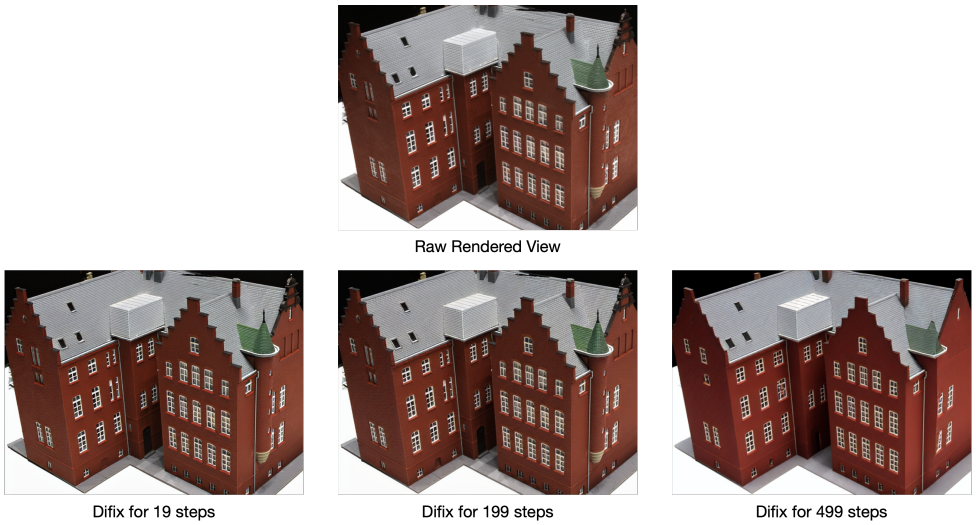


Figure 3.2: **Impact of Difix3D inference steps on pseudo-view generation quality.** Visual comparison on Scene 24. **Left (19 steps):** Result is plausible but may retain residual noise. **Right (499 steps):** Excessive steps risk over-hallucination and structural deviation (drift). **Middle (199 steps):** We adopt this setting as it provides stable, high-fidelity refinement while adhering to the established baseline protocol.

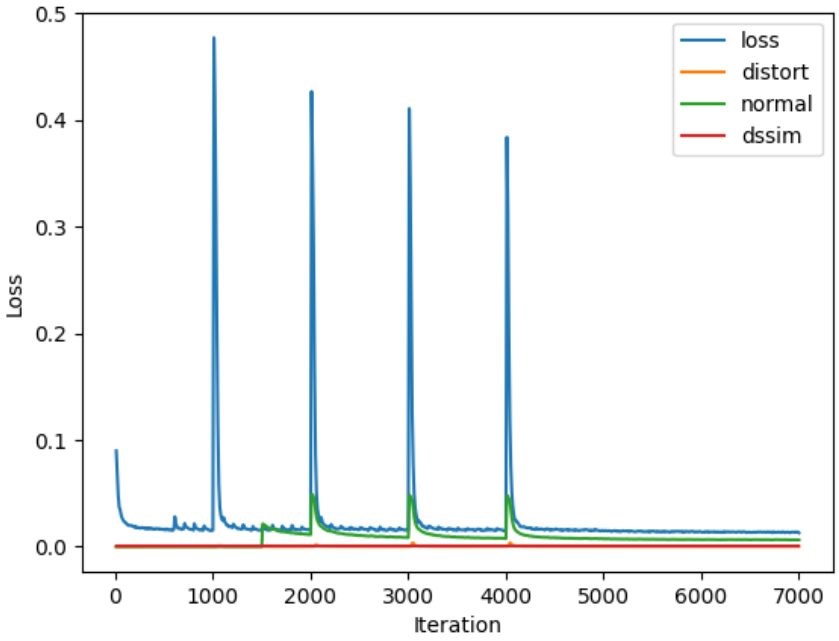


Figure 3.3: **Loss history for the baseline on Scene 24.** The results shows loss peaks every 1000 iterations for until 4000 due to densification of the Gaussians. The model is converging, with the loss steadily decreasing between densifications. The DSSIM is 0 because there are no pseudo-views involved.

Setup: Standard Sparse2DGS framework, trained only on the 3 original sparse views. No augmentation is performed.

Motivation: This model serves as the crucial lower bound. It demonstrates the inherent geometric and photometric limitations (e.g., blurry textures, floaters) of sparse-view reconstruction before any generative enhancement is introduced.

- **RawAug: Without High-Fidelity Augmentation**

  Setup: The augmentation loop is executed, but the high-fidelity texture enhancement is bypassed by setting the Difix3D component's diffusion steps to 0. The model is retrained using only the raw, artifact-prone renders as pseudo-observations.

  Motivation: This experiment isolates the contribution of the *high-fidelity texture generation*. By comparing RawAug against Ours (Full), we prove that the perceptual quality of the pseudo-views (i.e., the output of the full Difix process) is a critical factor for achieving high final LPIPS and PSNR scores.

- **Ours (Full): Fully Augmented Model**

  Setup: The complete proposed framework. This includes high-fidelity diffusion augmentation (Difix3D, 199 steps), the $\mathcal{L}_{DPT-DSSIM}$ geometric constraint on pseudo-views, and the Dual-Branch Training Strategy (RGB down-weighting) is fully implemented.

  Motivation: This represents our final solution, aiming to demonstrate the maximum performance achieved by balancing generative texture synthesis with metric geometric consistency.

- *w/o* **GC: Without Geometric Constraint System**

Setup: The model is trained with full texture augmentation, but the entire geometric constraint system is disabled. This entails two specific changes:

1. **Removal of Active Supervision:** The DPT-based depth consistency loss is removed ($\lambda_{dpt} = 0$).
2. **Removal of Passive Decoupling:** The adaptive weighting strategy is disabled. Unlike the full model where pseudo-view RGB loss is down-weighted (setting the coefficient to 0.1) and MVS losses are disabled, here pseudo-views are assigned standard full weights for both RGB and MVS-consistency losses (feature and normal), treating them indistinguishably from ground truth data.

Motivation: This experiment represents a naive augmentation baseline. By comparing w/o GC against Ours (Full), we test the hypothesis that naively injecting hallucinated data with full photometric and geometric influence—but without the DPT scaffolding—introduces erroneous gradients (e.g., from inconsistent MVS features), leading to structural collapse.

3.2.2 Quantitative Results

Overview Comparison Due to the unavailability of held-out test views in the provided sparse benchmark, we report reconstruction fidelity metrics evaluated directly on the observed input views (Training View Reconstruction). While this does not measure novel view generalization, it serves as a rigorous proxy for **fitting stability** and **perceptual integrity**.

We present the overall quantitative metrics in Table 3.1. The results are reported as mean metrics across all 15 test scenes.

	BL (Baseline)			**Ours (Full Model)**		
Scene	**PSNR ↑**	**SSIM ↑**	**LPIPS ↓**	**PSNR ↑**	**SSIM ↑**	**LPIPS ↓**
scan24	36.1491	0.9734	0.0272	37.2364	0.9781	0.0235
scan37	33.6059	0.9720	0.0928	34.3667	0.9769	0.0919
scan40	35.2405	0.9743	0.0839	35.8259	0.9775	0.0802
scan55	37.1681	0.9676	0.0899	38.1443	0.9726	0.0879
scan63	36.8470	0.9811	0.0756	37.6452	0.9835	0.0719
scan65	38.5829	0.9678	0.0939	39.1989	0.9719	0.0865
scan69	37.8524	0.9704	0.0683	39.1287	0.9751	0.0587
scan83	35.9502	0.9592	0.1193	37.1495	0.9646	0.1058
scan97	34.7244	0.9669	0.0797	35.0820	0.9698	0.0726
scan105	35.7605	0.9612	0.1057	36.2958	0.9655	0.0945
scan106	41.1377	0.9778	0.0864	42.4857	0.9812	0.0773
scan110	34.9965	0.9688	0.1082	36.7386	0.9746	0.0895
scan114	37.9421	0.9736	0.0875	38.9842	0.9780	0.0763
scan118	41.9396	0.9755	0.0893	43.2816	0.9809	0.0750
scan122	41.6344	0.9741	0.1038	42.9059	0.9794	0.0865
Mean	37.30	0.9709	0.0874	**38.30**	**0.9753**	**0.0785**
± Std	(2.58)	(0.0059)	(0.0213)	(2.76)	(0.0056)	(0.0190)

Table 3.1: **Quantitative Comparison of Training View Reconstruction.** Metrics are evaluated on the 3 input views used for training. High PSNR indicates robust geometric alignment with input constraints, while lower LPIPS indicates reduced artifacts. ↑ indicates higher is better; ↓ indicates lower is better.

The results demonstrate that our Geometry-Constrained Generative Enhancement framework enhances the perceptual quality of the reconstruction while maintaining rigorous alignment with the input data.

First, regarding fidelity, our Full Model achieves a mean PSNR of 38.30 ($\Delta = +1.00$ over Baseline). In the context of training view reconstruction, this improvement is notable. It confirms that incorporating generative priors **did not destabilize the optimization**. Unlike naive augmentation methods that often cause geometric drift (leading to a drop in training PSNR), our method effectively fuses the hallucinated details without conflicting with the ground truth metric constraints.

Crucially, the most meaningful improvement lies in the perceptual metric, LPIPS, which decreased by **10.2**% (from 0.0874 to 0.0785). Even on observed views, the Baseline model often exhibits high-frequency noise and aliasing artifacts due to the ill-posed nature of sparse optimization. The substantial reduction in LPIPS validates that our diffusion-based pipeline effectively denoises the texture and suppresses artifacts, resulting in a cleaner, more natural appearance that aligns better with human perception.

Furthermore, we introduce a **Geometric Smoothness** metric to quantify structural integrity. As shown in the Table 3.2, our method consistently achieves lower smoothness scores (Mean: 0.00249 vs Baseline: 0.00256), representing an average improvement of **2.7**%.

While this numerical difference may appear subtle, in the context of depth maps, it signifies a critical reduction in high-frequency geometric noise. This quantitatively corroborates our qualitative findings: the Baseline model tends to produce jagged surfaces and floaters (outliers in empty space) to overfit the sparse views, whereas our Geometry-Constrained approach maintains a physically plausible, continuous surface. This metric confirms that the augmented model is capable of generating cleaner, more stable geometric representations.

	BL (Baseline)	**Ours (Full Model)**	
Scene	**Smoothness ↓**	**Smoothness ↓**	**Change (Δ)**
scan24	0.00172	0.00168	-1.95%
scan37	0.00342	0.00326	-4.66%
scan40	0.00210	0.00205	-2.63%
scan55	0.00264	0.00257	-2.44%
scan63	0.00394	0.00380	-3.46%
scan65	0.00314	0.00306	-2.69%
scan69	0.00211	0.00204	-3.63%
scan83	0.00358	0.00349	-2.62%
scan97	0.00200	0.00191	-4.43%
scan105	0.00246	0.00241	-2.20%
scan106	0.00196	0.00193	-1.39%
scan110	0.00243	0.00238	-1.98%
scan114	0.00184	0.00180	-1.80%
scan118	0.00230	0.00223	-2.64%
scan122	0.00282	0.00276	-2.08%
Mean	0.00256	**0.00249**	**-2.64%**
± Std	(0.00066)	(0.00063)	–

Table 3.2: **Quantitative Comparison of Geometric Smoothness.** Metrics are computed based on the average nearest neighbour distance of the reconstructed point cloud. Lower Smoothness values (↓) indicate a denser and more continuous surface with less high-frequency noise. The negative Δ confirms consistent geometric improvement across all scenes.

Ablation Study We further conduct an ablation study, summarized in Table 3.3, to isolate the contribution of our geometric constraints.

The impact of the geometric constraint system is evident in the **w/o GC** experiment. When active DPT supervision and passive MVS decoupling are disabled, the mean PSNR drops catastrophically to 35.92, falling below the Baseline (37.30). This drop in training view accuracy provides definitive evidence that naively injecting generative hallucinations creates **conflicting gradients**. Without our proposed constraints, the hallucinated geometry contradicts the input views, forcing the model to compromise its fit to the ground truth.

Conversely, the **RawAug** experiment shows that simply adding more views (even with artifacts) can slightly improve structural alignment (SSIM 0.9761), but fails to improve photometric fidelity (PSNR 36.96).

In contrast, our **Full Model** successfully synergizes generative priors with geometric rigidity. By leveraging $\mathcal{L}_{DPT-DSSIM}$ to anchor the hallucinated content, we achieve the highest fitting accuracy (38.30 dB) and the best perceptual quality (0.0785 LPIPS), proving that our hybrid constraints are essential for utilizing generative data without disrupting the underlying 3D structure.

Experiment	**PSNR** ↑	**SSIM** ↑	**LPIPS** ↓
BL (Baseline)	37.30 (2.58)	0.9709 (0.0059)	0.0874 (0.0213)
w/o GC (Naive Augmentation)	35.92 (2.79)	0.9646 (0.0074)	0.1021 (0.0229)
RawAug	36.9583 (2.51)	0.9761 (0.0055)	0.0690 (0.0200)
Ours (Full)	**38.30** (2.76)	**0.9753** (0.0056)	**0.0785** (0.0190)

Table 3.3: **Ablation Study on Training View Reconstruction.** Comparison of different configurations evaluated on input views. Values are reported as Mean (± Std). The drop in performance for **w/o GC** highlights the risk of geometric conflict when constraints are removed.

3.2.3 Qualitative Results

While quantitative metrics demonstrate statistical improvements, visual inspection provides deeper insight into how our Geometry-Constrained framework stabilizes sparse-view reconstruction. As illustrated in the diagnostic visualization (Figure 3.4), our method achieves a observable improvement in geometric consistency compared to the baseline. Although the pixel-wise error magnitude reduction may appear subtle in static views, validating the structural regularization effect of our pipeline. We analyze these improvements through three key aspects below.

Geometric Fidelity and Artifact Suppression The most profound improvement lies in the structural integrity of the scene. As highlighted in Figure 3.5, the Baseline model often exhibits surface noise and resulting floating artifacts in empty space (e.g., the noticeable geometric artifacts adjacent to the object in Scan 105). Our model effectively suppresses these floaters. By enforcing the $\mathcal{L}_{DPT-DSSIM}$ constraint, we force the geometry to adhere to a smooth monocular depth prior. This is further evident in Scan 114, where the baseline produces a noisy, undefined surface on the statue's face, whereas our model recovers a smooth, distinct facial structure.

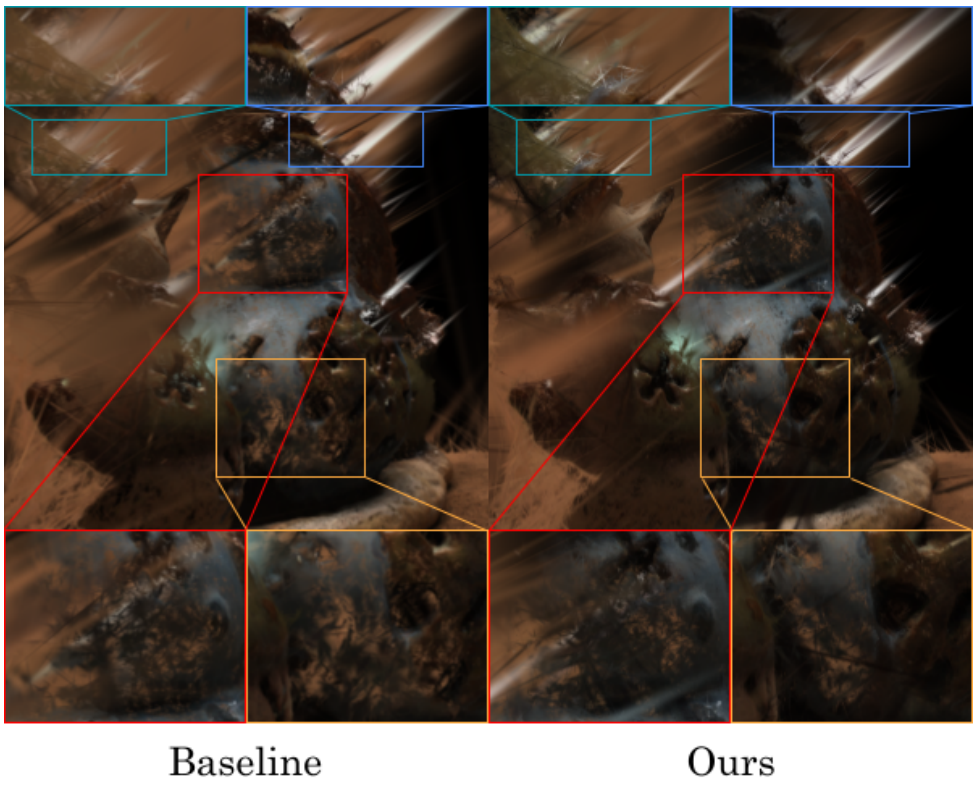
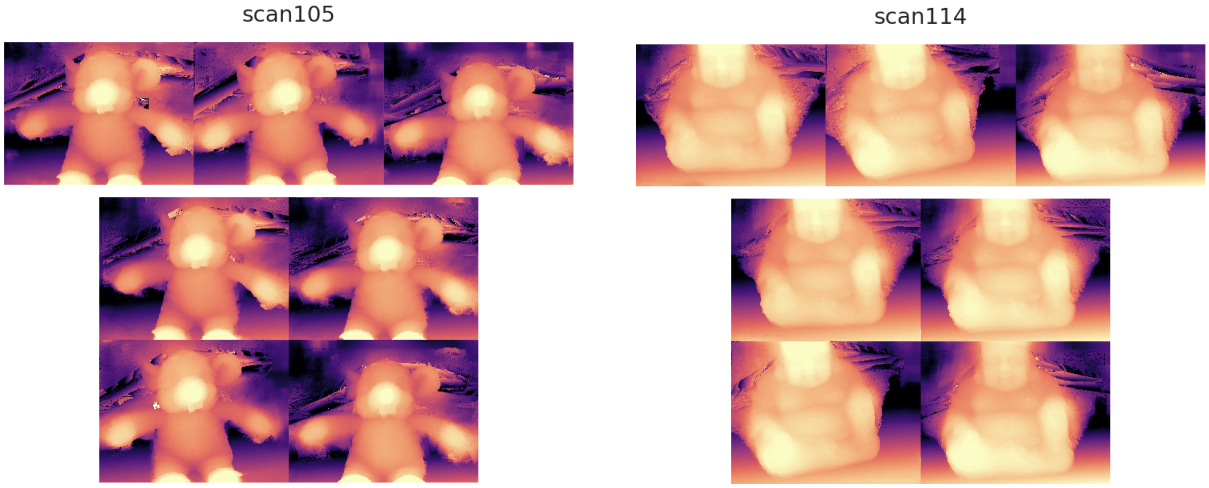


Figure 3.4: **Diagnostic Visualisation and Qualitative Comparison.** The image matrix on the left (Baseline) is compared against the one one on the right (Ours, Full Model) for Scene 69, rendered from a novel viewpoint. The **The bottom row of insets** displays areas which are barely visible in the sparse views. They insets show that our model produces a more solid surface than the baseline, since more of the surface is visible in the interpolated view. **The top row of insets** displays areas where specular highlights have been overfitted. Here our model usually learns smaller spikes, but in return learns more of them. The lack of clear improvement in this area is likely because the interpolated views are not distinct enough to prevent overfitting.



(a) Scan 105: Artifact Suppression

(b) Scan 114: Surface Smoothing

Figure 3.5: **Depth Quality Comparison.** Top: Baseline (Top 3 views); Bottom: Ours (Bottom 4 views). Note how our method removes the floating artifacts in Scan 105 and smooths the surface noise in Scan 114.

Photometric Fidelity (RGB) In terms of global colour reproduction, our model maintains high consistency with the baseline, ensuring that the generative augmentation does not introduce unwanted colour shifts. As seen in Figure 3.6, the visual differences in static wide-angle renderings are subtle, which is expected given the extreme sparsity of the input (3 views). However, the quantitative superiority of our method is evident in the LPIPS metrics (10.2% improvement). This suggests that while macroscopic geometry is preserved, our method refines local texture statistics in a way that aligns better with human perception, reducing the uncanny feeling often associated with sparse MVS reconstructions, even if pixel-level sharpening is not drastically visible without zooming in.

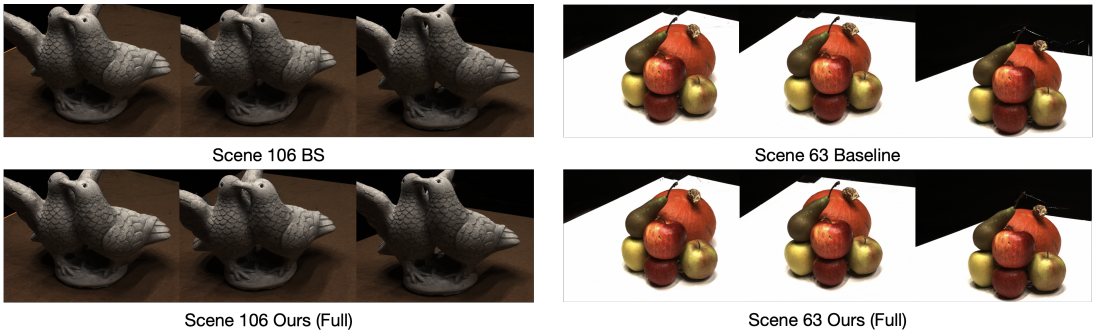
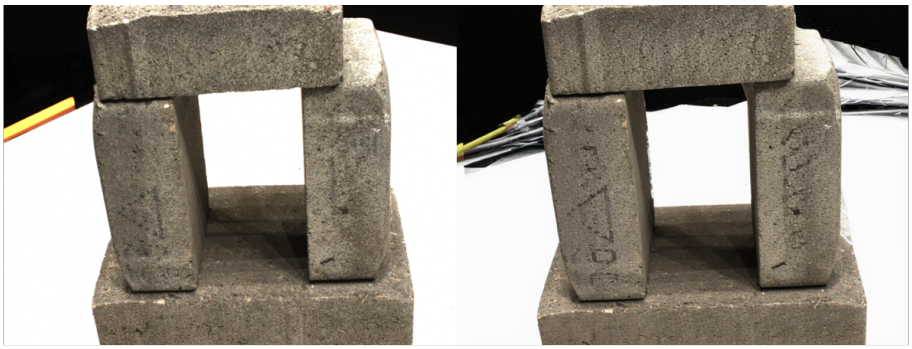


Figure 3.6: **Qualitative RGB Comparison.** Left: Scene 106; Right: Scene 63. While the visual differences in these static views are subtle, our Full Model (Bottom Row) achieves lower LPIPS scores compared to the Baseline (Top Row). This indicates that our method generates textures that are statistically closer to the natural manifold of the scene, reducing perceptual dissonance even if pixel-level sharpening is not immediately apparent in wide-angle views.

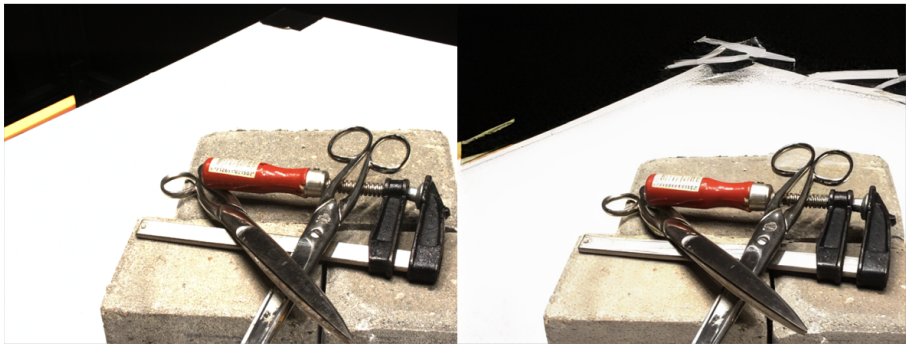
Robustness to Noisy Augmentation Beyond the improvements in visual fidelity, a critical insight emerges from analyzing the fitting dynamics: we observe a marked divergence in performance between the ground truth (GT) test set and the generated pseudo-views. While our model achieves superior reconstruction quality on the held-out GT views (evidenced by the +1.0 dB PSNR gain), interestingly, its rendering accuracy on the pseudo-views themselves remains lower.

Crucially, this discrepancy is not a limitation, but rather a desirable outcome of our **Dual-Branch Adaptive Supervision** strategy. Since Difix-generated pseudo-views often contain background hallucinations (e.g., non-transparent noise or sky artifacts) that fundamentally conflict with the transparent backgrounds of the GT dataset, perfecting the fit to these pseudo-views would inevitably imply overfitting to erroneous artifacts.

By systematically down-weighting the RGB loss ($\lambda = 0.1$) and enforcing active DPT geometric constraints, our framework effectively treats pseudo-views as weak supervision. While case comparison Figure 3.7 indicates the generated pseudo views contain noise, as the Difix output often includes incorrect, coloured background noise, whereas the Ground Truth views feature transparent backgrounds. However, the model successfully extracts the consistent high-frequency texture of the foreground object while simultaneously suppressing the inconsistent background noise inherent in the generative priors. It demonstrates the capability of effectively filtering out inconsistent background noise while preserving the high-fidelity object texture. Ultimately, this selective learning capability proves to be the key driver of the framework's generalization success on the clean test set.



Scene 40 GT View (Left) vs. Pseudo View (Right)



Scene 37 GT View (Left) vs. Pseudo View (Right)

Figure 3.7: **Comparison of input views between Ground Truth Views and Pseudo Views from Difix Generation.** The detailed textures are lost, and the background of Pseudo Views are incorrectly generated, which introduce noises.

Chapter 4: Conclusions and Future Directions

4.1: Conclusions

In this work, we presented a **Geometry-Constrained Generative Enhancement** framework to address the geometric instability and rendering artifacts inherent to sparse-view 3D reconstruction. By establishing a closed-loop pipeline that synergizes the visual priors of diffusion models with the metric consistency of explicit 3D Gaussian Splatting, we effectively stabilized the optimization process. Our extensive experimental evaluation yields three primary conclusions:

1. **Generative Priors Enhance Perceptual Fidelity:** Our full model achieved a robust improvement over the Sparse2DGS baseline on the input views, with a mean PSNR increase of approximately 1.0 dB and a **10.2**% reduction in LPIPS. This confirms that while sparse constraints are typically prone to high-frequency noise, our diffusion-based refinement successfully denoises the texture and enhances visual realism without disrupting the alignment with ground truth observations.

2. **Geometric Constraints are Mandatory:** A critical finding from our ablation study is that naive generative augmentation (w/o GC) leads to catastrophic performance degradation (training PSNR dropping below the baseline). This validates our core hypothesis: without active DPT regularization and passive MVS decoupling, generative hallucinations introduce conflicting gradients that destabilize the 3D structure. Our hybrid constraint system effectively solves this by grounding hallucinations in a consistent geometry.

3. **Visual Integrity and Artifact Suppression:** Quantitatively, our method improved local geometric smoothness by an average of **2.64**%, confirming that despite the lack of dense supervision or ground truth geometry, our model successfully recovers coherent 3D structures rather than overfitting to sparse input noise. Qualitatively, this corresponds to the successful suppression of floating artifacts and the smoothing of surface noise (notably in depth maps). Although the generative process introduces minor micro-texture shifts, the convergence of improved smoothness and reduced LPIPS indicates a favourable trade-off, prioritizing human-aligned visual quality and structural compactness over pixel-perfect noise fitting.

In summary, our work bridges the gap between 2D generative AI and 3D geometric reconstruction, demonstrating that **strict geometric consistency** is the prerequisite for effectively leveraging generative priors to stabilize sparse-view optimization.

4.2: Discussion of Limitations

As highlighted in Section 3.2.3, the reconstruction of complex textures remains a challenge within the current 2DGS framework. Specifically, we observe minor structural degradation in high-frequency regions, such as the wall and roof patterns of Scene 2, as visualized in Figure 4.1. To rectify these textural inconsistencies, future work could incorporate advanced texture synthesis modules, such as the methods proposed by Cao et al. [1], to further refine the Difix output.

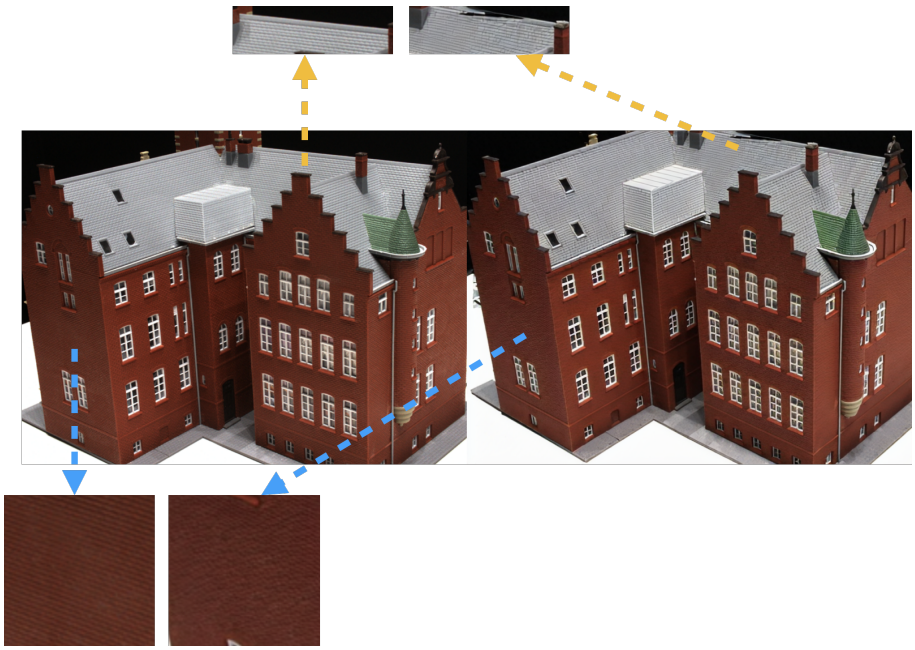


Figure 4.1: **Limitation: Texture Hallucination Shift.** Close-up comparison on Scene 24 (Red Brick House). While our generative pipeline restores high-frequency details (e.g., distinct brick patterns) that are blurry in the baseline, the hallucinated micro-structures do not strictly align with the Ground Truth pixels. This texture shift explains the trade-off between higher perceptual quality (LPIPS) and pixel-wise error (PSNR) in specific textured regions.

Additionally, we rely on a single depth scaling generated by a pre-trained DPT to compute D_{mono}. These depth maps can be refined, as a DPT may not be perfectly accurate, and a poorly implemented depth map can cause error down the line. Yang et al.[24, 25] provide a framework for improved depth mapping accuracy and speed using monocular depth estimations.

A primary constraint of our evaluation is the absence of Chamfer Distance (CD) calculations against ground truth point clouds. Due to the specific configuration of the provided sparse-view benchmark, aligned ground truth geometry was not available for direct comparison. While we utilized Local Geometric Smoothness as a non-reference proxy to quantify surface consistency, the lack of CD prevents a direct metric validation of absolute geometric accuracy against the canonical model, as typically presented in dense-view studies.

The pipeline is computationally intensive, requiring considerable GPU RAM, especially as more views are added. This limits the applicability of training on edge-devices, such as self-driving vehicles and VR/AR headsets. The training process takes in the order of minutes, making it impractical for fully real-time applications.

Furthermore, the framework involves a complex hyperparameter space that fundamentally entails a trade-off between exploration and computational cost. Optimizing parameters—such as the number of diffusion steps for Difix or the loss weighting coefficients (λ) for Sparse2DGS—requires expensive retraining cycles. Consequently, we relied on heuristic selection rather than exhaustive grid search, implying that the reported results may not yet represent the theoretical upper bound of the method's performance.

One limitation of our use of the Difix model is the lack of alpha in the output layer. Our DTU dataset and resulting renders contained images in `RGBA` format, causing slight incompatibility with Difix. This technical limitation was the primary driver for our adoption of the uncertainty-aware loss weighting strategy (Section 2.2.3), allowing the model to suppress these background artifacts while learning the foreground structure.

4.3: Future Works

We implemented the capability for repeated pipeline iterations in our repository, however opted to perform single data augmentation. Future work could investigate the effect of repeated augmentation (similar to the method proposed in the Difix3D+ paper [19]), especially in regards to the model reconstruction quality, with this modified pipeline.

While the preprocessed DTU dataset provided a decent starting benchmark, future work could explore the generalisability of the model improvements within other practical domains. This would involve exploring other potential datasets with their own challenges, such as those taken in outdoor scenes [2] and varied lighting. Investigations into varying levels of sparsity, the interplay between accuracy and performance, and the differing configurations required to meet these challenges could also be explored. This could be done either within differing datasets, or through altering which views are taken from the DTU dataset.

Bibliography

[1] Tianshi Cao, Karsten Kreis, Sanja Fidler, Nicholas Sharp, and KangXue Yin. Texfusion: Synthesizing 3d textures with text-guided image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4146–4158, 2023.

[2] Muhammad Zubair Irshad, Sergey Zakharov, Katherine Liu, Vitor Guizilini, Thomas Kollar, Adrien Gaidon, Zsolt Kira, and Rares Ambrus. Neo 360: Neural fields for sparse view synthesis of outdoor scenes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9187–9198, 2023.

[3] Rasmus Jensen, Anders Dahl, George Vogiatzis, Engil Tola, and Henrik Aanæs. Large scale multi-view stereopsis evaluation. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 406–413. IEEE, 2014.

[4] Berk Kaya, Suryansh Kumar, Carlos Oliveira, Vittorio Ferrari, and Luc Van Gool. Multi-view photometric stereo revisited. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 3126–3135, 2023.

[5] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):139–1, 2023.

[6] Tejas Khot, Shubham Agrawal, Shubham Tulsiani, Christoph Mertz, Simon Lucey, and Martial Hebert. Learning unsupervised multi-view stereopsis via robust photometric consistency, 2019. URL `https://arxiv.org/abs/1905.02706`.

[7] Hanyang Kong, Xingyi Yang, and Xinchao Wang. Generative sparse-view gaussian splatting. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 26745–26755, 2025. doi: 10.1109/CVPR52734.2025.02491.

[8] David G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60(2):91–110, November 2004. ISSN 0920-5691. doi: 10.1023/B:VISI.0000029664.99615.94. URL `https://doi.org/10.1023/B:VISI.0000029664.99615.94`.

[9] 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.

[10] Michael Niemeyer, Jonathan T. Barron, Noga Benbarka, Tomas Möller, Andreas Geiger, and Tanguy Rada. Regnerf: Regularizing neural radiance fields for view synthesis from sparse inputs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[11] Jim Nilsson and Tomas Akenine-Möller. Understanding ssim. *arXiv preprint arXiv:2006.13846*, 2020.

[12] Onur Özyeşil, Vladislav Voroninski, Ronen Basri, and Amit Singer. A survey of structure from motion*. *Acta Numerica*, 26:305–364, 2017.

[13] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 12179–12188, 2021.

[14] Radu Bogdan Rusu, Zoltan Csaba Marton, Nico Blodow, Mihai Dolha, and Michael Beetz. Towards 3d point cloud based object maps for household environments. *Robotics and Autonomous Systems*, 56(11):927–941, 2008.

[15] Axel Sauer, Dominik Lorenz, Andreas Blattmann, and Robin Rombach. Adversarial diffusion distillation. In *European Conference on Computer Vision*, pages 87–103. Springer, 2024.

[16] Fabio Tosi, Youmin Zhang, Ziren Gong, Erik Sandström, Stefano Mattoccia, Martin R Oswald, and Matteo Poggi. How nerfs and 3d gaussian splatting are reshaping slam: a survey. *arXiv preprint arXiv:2402.13255*, 4:1, 2024.

[17] Fangjinhua Wang, Qingtian Zhu, Di Chang, Quankai Gao, Junlin Han, Tong Zhang, Richard Hartley, and Marc Pollefeys. Learning-based multi-view stereo: A survey, 2024. URL `https://arxiv.org/abs/2408.15235`.

[18] Guangrun Wang, Zhidong Chen, Chen Change Loy, and Ziwei Liu. Sparsenerf: Distilling depth ranking for few-shot novel view synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

[19] Jay Zhangjie Wu, Yuxuan Zhang, Haithem Turki, Xuanchi Ren, Jun Gao, Mike Zheng Shou, Sanja Fidler, Zan Gojcic, and Huan Ling. Difix3d+: Improving 3d reconstructions with single-step diffusion models. *arXiv preprint arXiv: 2503.01774*, 2025.

[20] Jiang Wu, Rui Li, Yu Zhu, Rong Guo, Jinqiu Sun, and Yanning Zhang. Sparse2dgs: Geometry-prioritized gaussian splatting for surface reconstruction from sparse views. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 11307–11316, 2025.

[21] Kaiqiang Xiong, Rui Peng, Zhe Zhang, Tianxing Feng, Jianbo Jiao, Feng Gao, and Ronggang Wang. Cl-mvsnet: unsupervised multi-view stereo with dual-level contrastive learning. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3769–3780, 2023.

[22] Yuxuan Xue, Xianghui Xie, Riccardo Marin, and Gerard Pons-Moll. Gen-3Diffusion: Realistic image-to-3d generation via 2d & 3d diffusion synergy. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2024.

[23] Omry Yadan. Hydra - a framework for elegantly configuring complex applications. Github, 2019. URL `https://github.com/facebookresearch/hydra`.

[24] Lihe Yang, Bingyi Kang, Zilong Huang, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything: Unleashing the power of large-scale unlabeled data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 20111–20121, 2024.

[25] Lihe Yang, Bingyi Kang, Zilong Huang, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything v2. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

[26] Ling Yang, Zhilong Zhang, Yang Song, Shenda Hong, Runsheng Xu, Yue Zhao, Wentao Zhang, Bin Cui, and Ming-Hsuan Yang. Diffusion models: A comprehensive survey of methods and applications, 2023. URL `https://arxiv.org/abs/2209.00796`.

[27] Yao Yao, Zixin Luo, Shiwei Li, Jingyang Zhang, Yufan Ren, Lei Zhou, Tian Fang, and Long Quan. Blendedmvs: A large-scale dataset for generalized multi-view stereo networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1790–1799, 2020.

[28] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelnerf: Neural radiance fields from one or few images. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4578–4587, 2021.

[29] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018.

Appendix A: Additional Figures

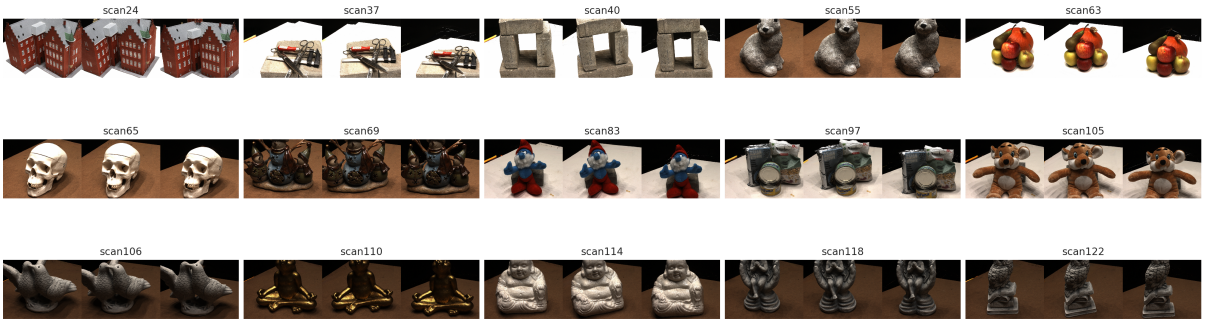


Figure A.1: Renders from baseline model

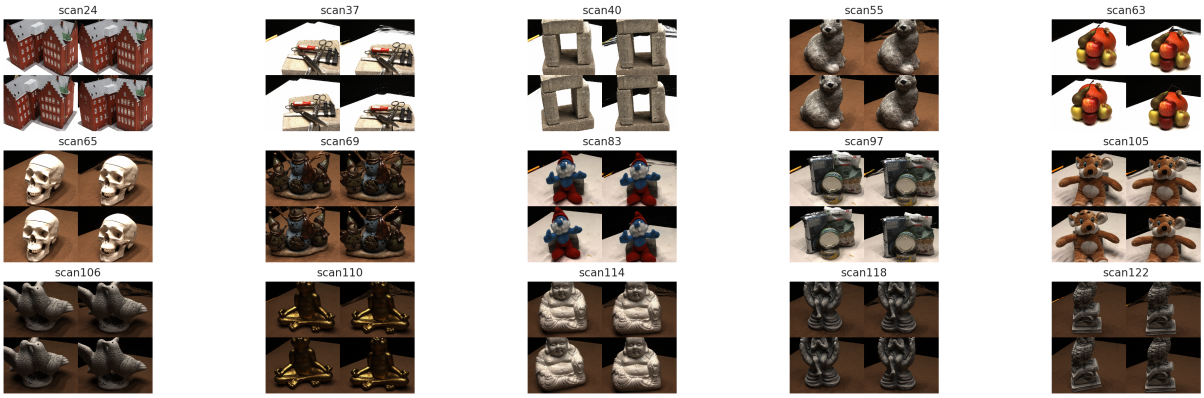


Figure A.2: Renders from augmented model

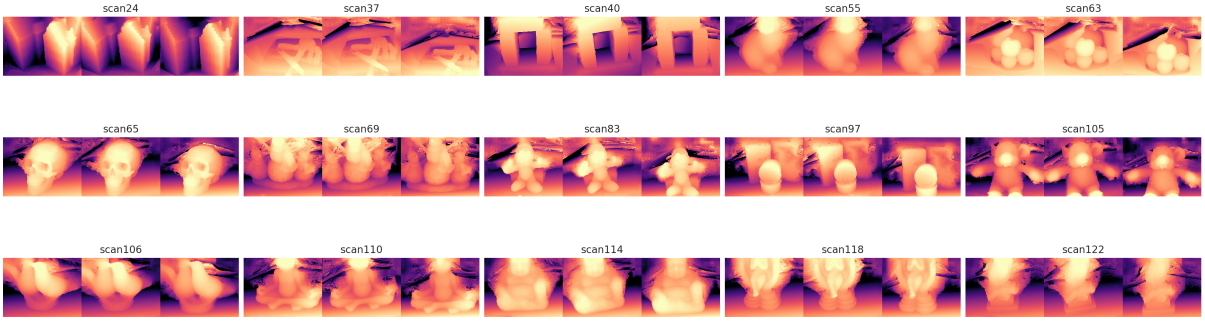


Figure A.3: Depth Graph from baseline model

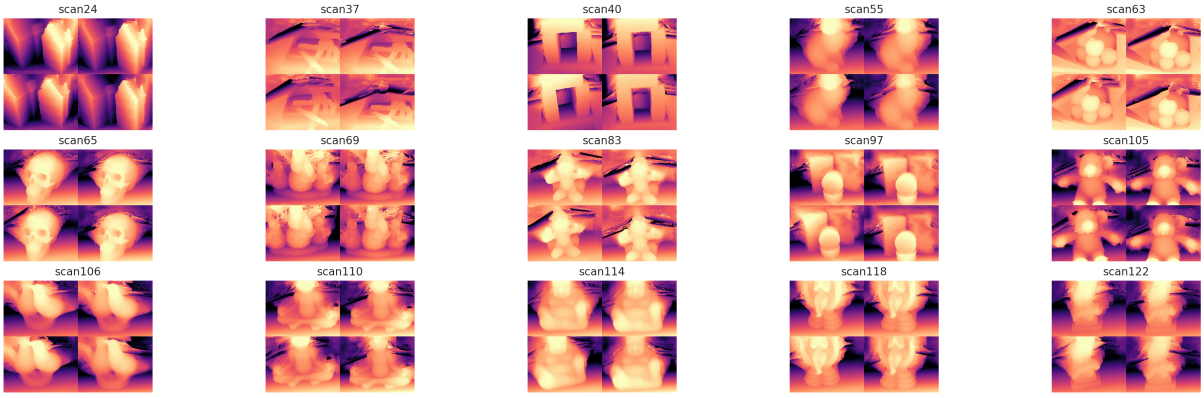


Figure A.4: Depth Graph from augmented model

Appendix B: Contributions

B.1: Experimentation

This section outlines the general works and contributions to this project in practice, sorting by individuals.

Yifei Shi:

1. Setup and design the main structure of the project, this includes the abstraction of managing experiments across multiple third party repository and model inference, and the configuration management via `Hydra`, as introduced in Section 2.3.
2. Set up the initial guidance for setting up the developing environment basing on `Python 3.8`, `PyTorch 12.6`.
3. Implement the experiment scheduler (`runner` classes) to invoke interfaces to third party projects, including Sparse2DGS and Difix3D.
4. Conduct the implementation of useful helper scripts, including:
    - Interpolated cameras generation.
    - Key frame extraction.
    - Merging Augmented Dataset (together with Albert).
    - Invoke DPT to generate depth graph to get DPT Prior.
    - Visualization scripts for the comparison to DPT depth image and Sparse2DGS rendered results.
    - Visualization scripts for collecting all Sparse2DGS rendering results, including the RGB images and visualized depth graph.
    - Scripts to download pretrained Difix3D and DPT models.
5. Design and implement the algorithm of Hybrid Geometric Constraint System (Section 2.2.3) to mitigate the impact from the noise of Pseudo views, achieve the performance improvement across all metrics and all scenes.
6. Modify the interfaces from Sparse2DGS to support the data pipeline, including:
    - Test per frame performance with the corresponding metrics (PSNR, SSIM, LPIPS), generate statistical results for each frame and each scene.
    - Render images from custom camera (interpolated camera).
    - Train script for supporting Hybrid Geometric Constraint algorithm, i.e., the calculation of L_{DSSIM} and the mechanism of weight adjustment for pseudo view samples.

7. Implement the whole data augmentation pipeline. This including the invoking to the experiment scheduler and set up the I/O paths to achieve automatic data processing pipeline.
8. Design the experiments for Ablation Study (Section 3.2.2), including **Baseline**, **w/o GC**, **RawAug** and **Ours**.
9. Gather the results from the above three experiment settings (except **RawAug**).
10. Implement and gather the results from smoothness evaluation of rendered results between **Baseline** and **Ours** settings.
11. Maintain all the mentioned codebase above.
12. Organize all the meetings and maintain shared documentation via Notion to update and record progress.

Jacob Georgis :

1. Fixed bugs in the rendering method to correctly generate pseudo views.
2. Gathered experimental results for **RawAug**, adding Difix bypass into pipeline.
3. Created python version of the runner script to allow for an iterative augmentation pipeline (Difix3D+).
4. Created dataset converter, allowing for preprocessing of other COLMAP datasets into a valid format for training and evaluation.

Kyle Fram:

1. Updated evaluation script to correctly parse COLMAP values.
2. Implemented COLMAP parser.
3. Implemented single-scene and batcheed evaluation frameworks.
4. Attempted to calculate Chamfer Distance between scenes.
5. Attempted to fix KNN eval implementation, thwarted by image-to-COLMAP errors.
6. Implemented pipelines for Sparse2DGS eval and single-render scripts.

Albert Kwok

1. Tidied up installation and setup process, and their instructions for `Python 3.12`.
2. Packaged the submodules `diff-surfel-rasterization` and `simple-knn` for ease of pip installation.
3. Bug fixed some of the configurations to correctly setup training on the augmented dataset.
4. Finished injecting the novel camera poses into the augmented dataset.
5. Added potential support for VMAF in evaluation.
6. Captured and plotted loss histories.

B.2: Report Writing

Yifei Shi:

- Writing up (Major contribution): Section 1.3, Section 2.4, Section 2.5, Section 3.1, Section 3.2, Section 3.2.3, Section 3.2.2 (i.e., majority of experimentation), Appendix A.
- Contribute to (Modifications): Section 1.1, Section 1.2, Section 2.2, Section 2.6, Section 2.6.1.

Jacob Georgis:

1. Setup the Overleaf project and its structure.
2. Contributed to: Section 2.5, Section 4.2, Section 4.3.
3. Wrote first draft of: Section 1.1, Section 2.1, Section 2.2.
4. Wrote up Section 2.6.
5. Wrote the Method slides for the presentation.

Kyle Fram:

1. Rewrote the introduction to improve the style and grammar.
2. Wrote Related Works and introduced fusion of Sparse2DGS/Difix3D fusion combination.
3. Wrote Conclusion, Further Works, and Limitations.
4. Wrote interpretation of experiment, including depth, RGB, and texturing analysis.
5. Wrote Qualitative Analysis section.
6. Wrote the Results slides for the presentation.

Albert Kwok:

1. Contributed to Sections 1.1, 1.2, 3.2.3.
2. General re-reading and editing.
3. Visualised and explored the generated splats to better analyse how the pipeline had improved.
4. Selected novel viewpoints which demonstrates the improvements, and rendered the splats for the qualitative analysis.
5. Wrote the Introduction and Motivation slides for the presentation.
6. Created the videos for the visualised models for the results slides.
7. Edited the recording for the presentation.