

Flash-Refine: Frustum-Guided Local Incremental Learning for Efficient 3D Gaussian Splatting Completion

Nuocheng Ji¹, Hanyang Zhuang^{2*}, Chunxiang Wang¹, and Ming Yang¹

¹ School of Automation and Intelligent Sensing, Shanghai Jiao Tong University, Shanghai, China

² Global College, Shanghai Jiao Tong University, Shanghai, China
zhuanghany11@sjtu.edu.cn

Abstract. While 3D Gaussian Splatting (3DGS) has enabled real-time high-fidelity novel view synthesis, maintaining already-deployed static 3DGS assets remains difficult. When a local region is missing or blurred, retraining the whole scene from scratch is costly, while directly fine-tuning the asset on newly captured repair images causes severe degradation of unobserved views, and geometric stitching introduces boundary and depth conflicts. We propose **Flash-Refine**, a practical in-situ repair pipeline for static 3DGS assets that operates directly on a native PLY model without access to historical training images. Given newly registered repair views, Flash-Refine uses *Point-Wise Consensus Masking* and an *Adaptive Depth-Bounded Prior* to identify a bounded active region, applies *Gradient Locking* to frozen background Gaussians, propagates the mask through native densification, and protects frozen Gaussians from close-up pruning. Under this constrained active set, gradient-gated densification naturally allocates new parameters to the defective region, enabling rapid local completion from peripheral seeds. Extensive experiments on Deep Blending and Mip-NeRF 360 show that Flash-Refine achieves the best Dropped and Kept PSNR among practical repair methods across four scenes, while requiring only 2–4 minutes on an RTX 4090. We further analyze the method’s scope and limitations: it assumes static scenes and preregistered repair views, and although frozen background parameters remain unchanged, newly densified active Gaussians may still induce scene-dependent visibility interactions in unobserved views.

Keywords: Incremental 3DGS · Frustum-Guided Optimization · Local Parameter Allocation · Catastrophic Forgetting · Accuracy-Efficiency Trade-off

1 Introduction

With the rapid development of novel view synthesis, neural scene representations have achieved unprecedented success [5, 20, 22, 32]. In particular, 3D Gaussian

* Corresponding author.

Splatting (3DGS) [12] has emerged as a state-of-the-art framework, delivering high-fidelity visual quality alongside real-time rendering capabilities. Because of its explicit point-based nature and efficiency, 3DGS is increasingly being adopted as the foundational representation for real-world applications such as digital twins, virtual reality (VR), and cultural heritage preservation.

However, capturing large-scale real-world environments is inherently imperfect. Due to physical constraints, complex occlusions, or constrained camera trajectories, the initially captured data often suffers from localized “blind spots” or missing viewing angles. Consequently, the optimized 3DGS models frequently exhibit severe local degradation, characterized by cloudy floaters, geometric collapse [9, 35], or severe structural deficits in unobserved regions. When a user captures a small set of supplementary photographs to repair these localized defects, a critical challenge arises: *How can we efficiently integrate the newly acquired local data into the pre-existing global model?*

Existing pipelines struggle to address this localized maintenance problem efficiently. The standard approach requires *global retraining* from scratch, merging the new photographs with the potentially massive original dataset. This is prohibitively time-consuming and computationally expensive, especially for large-scale scenes where an update might take hours or even days. Alternatively, one could attempt *naive point cloud stitching* by training a small 3DGS model exclusively on the new images and geometrically merging it with the original model. However, this invariably leads to severe Z-fighting, illumination inconsistencies, and visible boundary artifacts at the overlapping regions. Finally, attempting to directly fine-tune the existing model solely on the new images leads to *catastrophic forgetting*—a well-documented phenomenon in continual learning [13, 15, 19] where the neural representation completely overwrites and destroys the previously learned global background to overfit the localized views.

To tackle these critical bottlenecks, we propose **Flash-Refine**, an elegant, highly efficient, and optimization-based in-situ refinement framework tailored for 3DGS local completion. Rather than retraining the global scene or resorting to external generative models, Flash-Refine operates directly on the native, degraded “.ply” asset. The core philosophy of our approach is to treat local completion as a “localized surgical refinement.”

Specifically, we introduce a *Point-Wise Consensus Masking* mechanism. By leveraging the geometric projection of the newly captured camera views and applying an adaptive geometric prior, we compute an exact point-wise binary mask. This mask bifurcates the scene: Gaussians located in the background are subjected to a strict *Masked Gradient Lock*, permanently freezing their positions, colors, opacities, and scales. Meanwhile, Gaussians within the active region are “unlocked” and granted full optimization freedom. Because the background is mathematically isolated from the backpropagation, the complete structural collapse of the global scene—a hallmark of catastrophic forgetting is avoided. Furthermore, by locking the vast majority of the scene, the optimizer is forced to allocate its full representational capacity exclusively to the missing region. This “local parameter allocation” allows the native 3DGS densification process

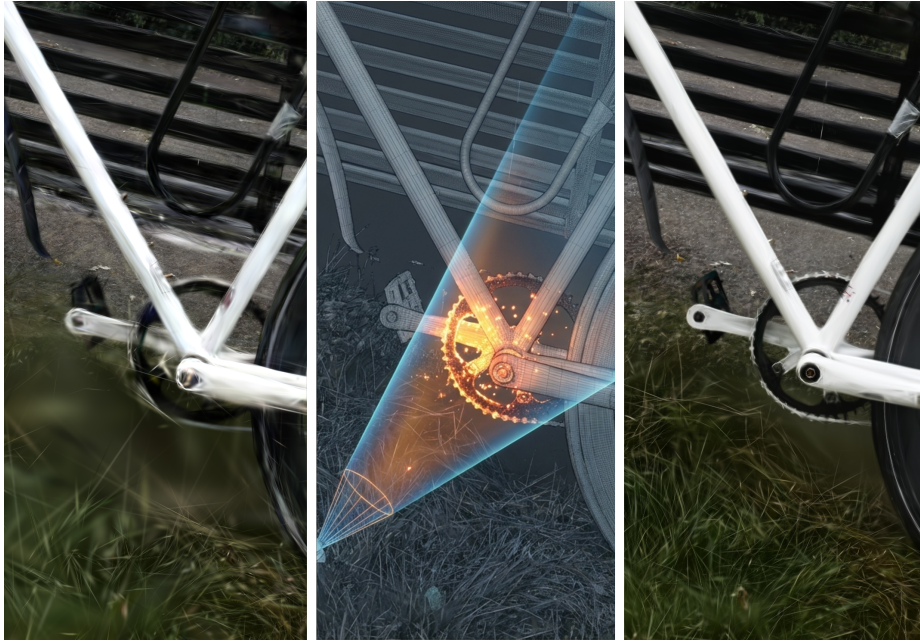


Fig. 1: Completion Quality and Efficiency of Flash-Refine. (Left) The degraded base 3DGS model suffering from geometric collapse due to missing views. (Middle) Our Flash-Refine framework projects the frustums of new images to compute a point-wise mask, strictly locking the background gradients. (Right) The reconstructed model. Our method efficiently recovers local fidelity while strictly preserving the existing parameters of the global background.

to rapidly reconstruct complex, high-frequency geometries from sparse peripheral points.

However, this constrained update is not equivalent to full global retraining. Because the historical Kept views are unavailable during repair, newly split active Gaussians may still create visibility interactions in unobserved views even though the frozen background parameters themselves are unchanged. We refer to this side effect as *Occlusion Bleeding*. Its severity is scene-dependent: it is small when the repair frustums are well separated from the background, but can be more visible when the active region strongly overlaps with kept-view backgrounds. Flash-Refine therefore targets a practical maintenance regime: rapid local repair with explicit protection against the catastrophic parameter overwriting observed in naive fine-tuning.

As shown in Fig. 1, our framework is highly effective even for structurally complex regions. We conduct rigorous evaluations using a localized Drop-Camera protocol on standard benchmarks. The results demonstrate that Flash-Refine requires merely minutes instead of hours, achieving local rendering quality that

approaches the upper bound of full scene retraining, while striking a highly practical balance between efficiency and global structural stability.

In summary, our main contributions are as follows:

- We propose Flash-Refine, an efficient in-situ completion pipeline for static 3D Gaussian Splatting assets that operates directly on pre-trained PLV models without historical Kept images.
- We couple Point-Wise Consensus Masking, an Adaptive Depth-Bounded Prior, Gradient Locking, mask propagation through densification, and pruning protection to maintain a valid bounded active set during native 3DGS repair.
- We analyze Local Parameter Allocation as a consequence of gradient-gated densification under a bounded active set: frozen Gaussians cannot accumulate positional gradients, so cloning and splitting concentrate on the repair region.
- We provide a controlled Drop-Camera benchmark across four scenes, including corrected per-image aggregation, cross-scene practical baselines, and per-scene runtime comparisons on the same RTX 4090.

2 Related Work

2.1 3D Gaussian Splatting and Sparse View Optimization

The advent of 3D Gaussian Splatting (3DGS) [12] has revolutionized the field of novel view synthesis, building upon the foundational success of implicit neural representations [20, 22]. By utilizing anisotropic 3D Gaussians as the explicit underlying representation, 3DGS replaces computationally heavy volumetric ray-marching with an ultra-fast differentiable tile-based rasterizer. However, the optimization of vanilla 3DGS is highly ill-posed and relies heavily on dense, multi-view image coverage. When applied to sparse or locally missing regions, the unconstrained optimization tends to produce severe “floater” artifacts and cloudy geometries.

To mitigate this reliance on dense data, several recent works have incorporated auxiliary structural constraints. Alongside pioneering sparse-view NeRF approaches [10, 23], 3DGS-based methods such as pixelSplat [3], FSGS [37], SparseGS [31], and recent depth-aware frameworks [27] leverage monocular depth priors to regularize the optimization process. SCGaussian [24] utilizes matching priors to maintain 3D structural consistency in few-shot scenarios. Furthermore, recent efforts have also explored altering the native densification strategy to improve convergence efficiency [14]. Despite their remarkable success in sparse reconstruction and optimization efficiency, these methods are intrinsically designed for *offline, global scene training*. When a user captures a small batch of new photos to repair a locally degraded patch, these pipelines still necessitate retraining the entire scene from scratch. In contrast, Flash-Refine introduces a purely incremental paradigm. By directly modifying the pre-trained asset, we bypass the computationally prohibitive cost of global retraining while effectively resolving local degradation.

2.2 Incremental and Continual Learning in Neural Fields

Adapting neural representations to newly arriving data without forgetting previously learned knowledge is a fundamental challenge in continual learning [13, 15, 19]. In neural radiance fields (NeRF) and 3DGS, updating a localized region with new images typically triggers catastrophic forgetting, where the network overfits the new views and completely destroys the unobserved background.

Prior to 3DGS, implicit representations explored continuous updates primarily in RGB-D SLAM settings [28, 38]. In the realm of 3DGS, SplatAM [11] and Gaussian Splatting SLAM [18] pioneered in dense tracking and mapping, achieving incremental mapping on the fly. However, SLAM-based methods prioritize real-time camera tracking and coarse scene reconstruction, which falls short of the strict high-fidelity requirement for offline 3D asset refinement. More recently, CL-Splats [1] explored continual learning specifically for 3DGS. Its setting is dynamic scene evolution, where scene changes are detected, segmented, and consolidated with replay. This assumption differs from our static repair setting: the defective region can be missing or severely underconstrained in the base PLY, and the historical Kept images are unavailable at maintenance time. Similarly, EWC-style regularization [13] depends on historical-data statistics or training-time information that is not provided by a deployed PLY asset. Flash-Refine therefore addresses a different operating point: direct PLY-only repair of a static scene, without historical replay, change detection, or a new global optimizer.

2.3 3D Editing and Inpainting

The explicit, point-based nature of 3D Gaussians makes them inherently more amenable to manipulation than implicit NeRFs, as demonstrated by recent structural enhancements and segmentation frameworks [6, 16, 34]. This has inspired numerous 3D editing frameworks. Methods like GaussianEditor [4] and Instruct-NeRF2NeRF [7] allow users to perform text-driven semantic modifications. Concurrently, manual editing tools like SuperSplat [25] enable explicit spatial manipulation via bounding boxes.

For handling missing regions, 3D inpainting techniques such as Inpaint3603DGS [29] and their implicit counterparts like SPIn-NeRF [21] aim to fill holes by leveraging 2D diffusion models to guess the unobserved background. Crucially, existing inpainting methods rely heavily on *generative hallucination*—synthesizing plausible but essentially fictitious background content. Our task fundamentally differs: we focus on *physically accurate geometric completion*. Rather than hallucinating textures, Flash-Refine ensures absolute fidelity to the physical world by fusing actual, newly captured photographs into the established 3D coordinate system.

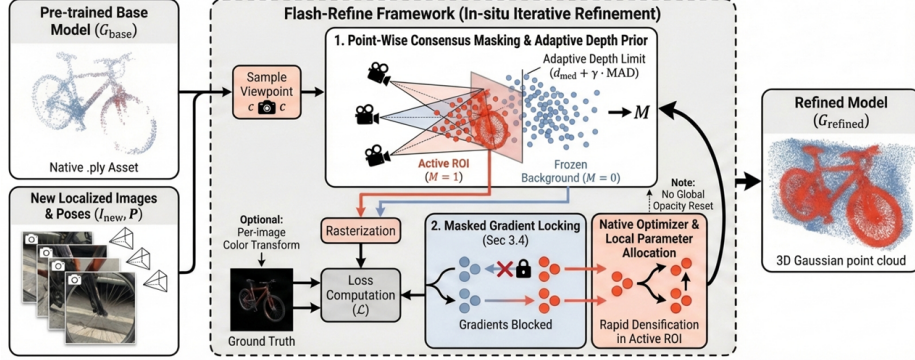


Fig. 2: Overview of the Flash-Refine framework. The system directly loads a pre-trained .ply asset and the new image poses $\mathbf{P}_c$. Before optimization, (1) **Point-Wise Consensus Masking** uses multi-view projection and an Adaptive Depth-Bounded Prior to define an **Active ROI** (red) and a **Frozen Background** (blue). During the incremental iterations, (2) **Masked Gradient Locking** ensures zero gradients flow to background points. This forces the native optimizer to exclusively perform **Local Parameter Allocation**, rapidly densifying geometries within the targeted missing region without structurally destroying the global scene.

3 Methodology

3.1 Problem Formulation and 3DGS Preliminaries

Given a pre-trained 3DGS scene representation, denoted as a base model $G_{base} = \{g_i\}_{i=1}^{N_{base}}$, and a small set of newly captured localized images $I_{new} = \{I_c, \mathbf{P}_c\}_{c=1}^K$ (where $\mathbf{P}_c$ denotes the camera poses) featuring unobserved or missing geometries, our objective is to efficiently update G_{base} without accessing the original training dataset.

In standard 3DGS [12], the scene is parameterized by 3D Gaussians, each characterized by its center μ_i , spherical harmonics C_i , opacity α_i , and covariance Σ_i . Novel views are rendered via α -blending of sorted overlapping Gaussians. Crucially, 3DGS employs an adaptive density control strategy (cloning and splitting) driven directly by the positional gradient norm $\|\nabla_{\mu_i} \mathcal{L}\|$.

A naive approach of directly fine-tuning G_{base} solely on I_{new} alters the optimization objective to $\arg\min_G \sum_c \mathcal{L}(G, I_c)$. Without global constraints, this inevitably leads to catastrophic forgetting, where the network completely destroys the unobserved background to overfit the new localized views. To address this, we propose **Flash-Refine**, an in-situ refinement framework (see Fig. 2). We introduce *Point-Wise Consensus Masking* (Sec. 3.2) and *Masked Gradient Locking* (Sec. 3.3) to strictly isolate the optimization space. This effectively bounds the gradient flow, allowing the native Gaussian densification to rapidly repair missing structures while keeping the existing background parameters strictly invariant.

3.2 Point-Wise Consensus Masking and Depth-Bounded Prior

To prevent the degradation of well-optimized background Gaussians during incremental updates, we must precisely identify which Gaussians belong to the targeted missing region. A heuristic bounding box is often insufficient, as it inevitably encapsulates non-target background surfaces immediately behind the target. Thus, we compute a strict point-wise binary state mask $M(\mu_i) \in \{0, 1\}$ for every point $g_i \in G_{base}$ prior to the training loop, where 1 denotes the active state and 0 denotes the frozen state.

First, we project all points into the camera coordinate systems of I_{new} to evaluate their visibility. A point is considered within the valid view frustum of camera c if its depth $z_{i,c} > 0$ and its projected coordinates (u_i, v_i) fall within the image dimensions. We aggregate the observation count $O(\mu_i)$ and the average projected depth $\bar{z}_i$ across all K new cameras. We enforce a multi-view consensus filter by requiring an active point to be observed by at least τ_{obs} cameras (typically $\tau_{obs} = 5$).

However, distant sky or floor points might still satisfy the consensus filter due to their large geometric scale. To establish an absolute isolation boundary without relying on hardcoded parameter counts, we introduce an *Adaptive Depth-Bounded Prior*. In real-world asset maintenance, repair photos are inherently close-up shots focusing on the defect. Thus, we dynamically calculate the bounding depth based on the camera frustums rather than a fixed subset capacity.

Specifically, we extract the sparse SfM tie points, e.g., initialized via COLMAP [26], associated with the newly captured images I_{new} . While recent monocular depth estimators [33] could also be employed, native SfM tie points provide a robust and scale-consistent anchor without additional forward passes. We compute the median projected depth d_{med} of these tie points relative to the new camera cluster, along with their median absolute deviation (MAD). The active mask is then strictly formulated as:

$$M(\mu_i) = \begin{cases} 1, & \text{if } O(\mu_i) \geq \tau_{obs} \wedge \bar{z}_i \leq d_{med} + \gamma \cdot \text{MAD} \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

where $\bar{z}_i$ is the average projected depth of point g_i , and γ is a conservative margin multiplier (empirically set to 2.0). This purely geometric and adaptive filtering robustly peels the foreground ROI away from the attached background, naturally adapting to varying sizes of missing regions—from small bicycle spokes to entire architectural facades—without manual tuning.

Dynamic State Propagation. Since 3DGS continuously adds and deletes points during optimization, maintaining this mask via static array indices would crash the system. To address this, we inject $M(\mu_i)$ as an inherent attribute tied to each Gaussian point. Whenever the native 3DGS densification mechanism clones or splits an active point, the newly generated offspring dynamically inherit the $M = 1$ state, allowing the active region to grow seamlessly while the background remains permanently locked.

3.3 Masked Gradient Locking and Local Parameter Allocation

Once the precise point-wise mask is initialized, we enforce absolute spatial isolation by intercepting the backpropagation process. After computing the standard rendering loss $\mathcal{L}$ and before the optimizer takes a step, we truncate the gradients of all properties Θ for the Frozen points:

$$\nabla_{\Theta_i} \mathcal{L} \leftarrow \nabla_{\Theta_i} \mathcal{L} \odot M(\mu_i) \quad (2)$$

This operation acts as a “Gradient Lock”, freezing the variables of the background environment because they receive no update signals.

Interaction with Native Densification. A profound yet highly beneficial consequence of our Masked Gradient Lock is its implicit control over the 3DGS densification mechanism. In 3DGS, a Gaussian is cloned or split if its accumulated positional gradient exceeds a specific threshold (i.e., $\|\nabla_{\mu_i} \mathcal{L}\| > \tau_{pos}$). By strictly forcing $\nabla_{\mu_i} \mathcal{L} = 0$ for all points where $M(\mu_i) = 0$, we fundamentally prohibit any densification from occurring in the background.

This yields what we call *Local Parameter Allocation*. It is not a new optimizer; rather, it is a direct consequence of applying gradient-gated native densification under a bounded active set. Since frozen background Gaussians have zero positional gradients, they cannot trigger cloning or splitting. The available densification events therefore concentrate on the unobserved ROI (e.g., the missing bicycle spokes), allowing the native 3DGS mechanism to rapidly reconstruct high-frequency geometry from nearby active seeds.

Seeding from Peripheral Gaussians. A natural question arises: how can native densification populate a completely missing region if gradient computation requires pre-existing points? In practice, the masked ROI is never an absolute vacuum. Because our adaptive depth prior encapsulates a sufficiently deep volumetric shell, it reliably captures pre-existing background surfaces, floor planes, and low-opacity floaters residing immediately behind or around the missing region. These points act as initial “seeds”. When supervised by the new images, they receive massive positional gradients. Driven by our localized parameter allocation, they undergo rapid cloning and splitting along the gradient directions, swiftly migrating forward to fill the spatial void and reconstruct the occluding missing surfaces.

Anti-Deletion Shield and Adaptive Optimization. In standard 3DGS, the learning rate decays exponentially. Our framework deliberately discards the exhausted optimizer states from the “.pth” checkpoint and directly injects the raw G_{base} PLY. By resetting the optimization step counter to zero, we naturally re-initialize the optimizer with the highest initial learning rate, revitalizing the geometry growth in the degraded area.

Furthermore, the native 3DGS pipeline routinely prunes Gaussians that appear exceptionally large in the 2D image space. Since the newly captured I_{new} consists of extreme close-up views, valid frozen background points will inevitably cast massive 2D projections. Without intervention, these essential background elements would be mistakenly flagged as “floaters” and permanently deleted.

To prevent this, we introduce an *Anti-Deletion Shield* by explicitly zeroing the accumulated 2D radii records of all Frozen points prior to the pruning phase.

Finally, we explicitly disable the periodic global opacity reset. While this protects the frozen background from becoming transparent, the standard opacity-based pruning (removing Gaussians where $\alpha < 0.005$) remains active. Combined with a short optimization schedule (e.g., 3,000 steps), this effectively suppresses newly generated floaters in the active ROI.

4 Experiments

4.1 Experimental Setup

Datasets. We comprehensively evaluate Flash-Refine on two widely adopted benchmarks with distinct camera trajectory characteristics: (1) **Mip-NeRF 360 Dataset** [2]: We select two representative scenes (*bicycle* and *counter*). These scenes are object-centric, featuring 360° inward-facing camera trajectories capturing central object clusters with complex backgrounds. (2) **Deep Blending Dataset** [8]: We select two full indoor environments (*Bedroom* and *DrJohnson*). Unlike Mip-NeRF 360, these datasets consist of free-viewpoint, unstructured, and highly chaotic camera poses, providing a rigorous test for real-world unconstrained captures.

Drop-Camera Protocol and Metrics. To simulate the local degradation and in-situ repair process, we design a localized Drop-Camera protocol. For each scene, we intentionally remove a spatially contiguous cluster of images (referred to as **Dropped**) during the initial training, leaving the remaining images (referred to as **Kept**) to train a degraded *Base Model*. The dropped images are then reintroduced strictly as the newly captured repair data I_{new} . For evaluation, we report PSNR, SSIM [30], and LPIPS [36] across three distinct image sets:

- **Dropped Images:** Evaluates the local geometric reconstruction fidelity and the capacity to recover missing structures.
- **Kept Images:** Serves as a monitor for the global scene integrity. A successful refinement should effectively preserve the Base Model’s performance on these unobserved views.
- **All Images:** Reflects the overall global quality of the final 3D asset.

The Dropped/Kept counts are 67/127 for *Bicycle*, 50/190 for *Counter*, 54/144 for *Bedroom*, and 81/183 for *DrJohnson*. In each scene, the Dropped subset is a spatially contiguous camera cluster centered on the target defective region. All-image PSNR/SSIM/LPIPS are computed by merging the same per-image records used for the disjoint Dropped and Kept subsets, rather than by averaging displayed subset scores.

Baselines and Upper Bound. To contextualize the repair setting, we establish a global-retraining **Upper Bound** by training 3DGS from scratch using all (Kept + Dropped) images. We compare Flash-Refine with practical repair baselines across all four scenes: (a) *Naive Fine-Tuning*, which directly optimizes

Table 1: Quantitative Comparison on the *Bedroom* Scene. Flash-Refine approaches the Upper Bound on the Dropped images while effectively preserving the Base Model’s performance on the Kept images, significantly circumventing the devastating catastrophic forgetting observed in Naive Fine-Tuning. Moreover, it requires only a fraction of the training time.

Method	Dropped Images			Kept Images			All Images			Time
	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS	
Upper Bound (Global Retrain)	27.05	0.912	0.110	29.12	0.925	0.134	28.56	0.922	0.127	~40m
Base Model (Degraded)	22.82	0.824	0.203	29.34	0.927	0.130	27.56	0.899	0.150	-
Naive Fine-Tuning	26.08	0.904	0.139	24.30	0.837	0.253	24.79	0.855	0.222	~5m
Geometric Stitching	25.86	0.895	0.140	26.22	0.866	0.214	26.12	0.874	0.194	~10m
SplaTAM	26.42	0.912	0.119	25.09	0.853	0.228	25.45	0.869	0.199	~10m
Inpaint360GS	27.10	0.910	0.117	25.67	0.858	0.221	26.14	0.875	0.190	~30m
Flash-Refine (Ours)	27.17	0.914	0.116	28.71	0.917	0.150	28.29	0.916	0.141	~2m

the base model on new views, and (b) *Geometric Stitching*, which trains a sub-model on new views and merges it with the base PLY. On the *Bedroom* scene, we additionally report SplaTAM [11] and Inpaint360GS [29] for broader reference.

Implementation Details. All experiments are conducted on a single NVIDIA RTX 4090 GPU. We assume the camera poses of the newly captured images $\mathbf{P}_c$ are preregistered into the global coordinate system. In this controlled benchmark, poses come from the dataset-level COLMAP reconstruction so that evaluation isolates 3DGS repair from localization noise. Flash-Refine is applied directly to the pre-trained Base Model PLY without accessing the historical Kept images. The refinement process is executed for only 3,000 iterations, whereas the Upper Bound global retraining requires 30,000 iterations. We use the adaptive depth prior ($\gamma = 2.0$), which dynamically bounds the active points (yielding roughly 60,000 active points on average for the targeted missing regions in our current experiments), with $\tau_{obs} = 5$ across all scenes.

4.2 Completion Quality and Efficiency

Quantitative Results. As demonstrated in Tab. 1 on the *Bedroom* scene, the Base Model exhibits poor performance on the Dropped images due to localized geometric collapse. Naive Fine-Tuning severely degrades the Kept images (dropping to 24.30 dB), confirming catastrophic forgetting under repair-only supervision. Geometric Stitching introduces boundary artifacts and suboptimal global consistency. In contrast, Flash-Refine improves the Dropped PSNR to 27.17 dB while preserving a substantially higher Kept PSNR of 28.71 dB. Tab. 2 summarizes the Base Model, Flash-Refine, and Upper Bound results across all tested scenes, with All-image scores recomputed from merged per-image records. Tab. 3 further compares the three practical repair methods on all four scenes. Flash-Refine obtains the best Dropped and Kept PSNR among these practical methods in every scene, while reducing the repair time to 2–4 minutes.

Table 2: Comprehensive Evaluation across 4 Scenes. We report PSNR values for the Base Model (BM), our Flash-Refine (Ours), and the Global Retraining Upper Bound (UB). All-image scores are aggregated from merged per-image records of the disjoint Dropped and Kept subsets.

Scene	Dropped PSNR $\uparrow$			Kept PSNR $\uparrow$			All PSNR $\uparrow$		
	BM	Ours	UB	BM	Ours	UB	BM	Ours	UB
Bicycle	15.19	16.91	16.75	17.32	16.90	17.24	16.58	16.90	17.07
Counter	22.57	29.66	31.03	31.91	27.47	31.19	29.96	27.93	31.16
Bedroom	22.82	27.17	27.05	29.34	28.71	29.12	27.56	28.29	28.56
DrJohnson	20.14	27.13	28.18	30.07	27.57	29.77	27.02	27.43	29.28

Table 3: Cross-scene comparison with practical repair baselines. We report PSNR on Dropped/Kept subsets and per-scene repair time. Timing measures only the 3DGS repair/refinement stage on the same RTX 4090 and excludes SfM/localization. Bold denotes the best value among the three practical repair methods.

Scene	Method	Dropped	Kept	Time
Bicycle	Naive FT	16.74	16.09	~6m
	Stitching	16.63	16.52	~12m
	Ours	16.91	16.90	~2m
Counter	Naive FT	28.93	20.21	~6m
	Stitching	28.63	26.78	~11m
	Ours	29.66	27.47	~2m
Bedroom	Naive FT	26.08	24.30	~5m
	Stitching	25.86	26.23	~10m
	Ours	27.17	28.71	~2m
DrJohnson	Naive FT	26.88	24.17	~13m
	Stitching	26.65	25.82	~21m
	Ours	27.13	27.57	~4m

Qualitative Results and Local Parameter Allocation. The visual enhancements provided by Flash-Refine are starkly evident in Fig. 3. For the unstructured indoor scenes, the base model of *Bedroom* exhibits cloudy floaters and completely fails to represent the computer desk area. Upon injecting the new frustums, Flash-Refine instantly reallocates the densification quotas, refining the blurry floaters into highly detailed and crisp structural boundaries.

A rigorous stress test is observed in the *DrJohnson* scene, which features two rooms connected by a corridor. During the base model training, we intentionally restricted the viewing trajectories solely to the corridor. Consequently, the base model renders the adjacent room as severely blurred, heavily populated with massive cloudy floaters and geometric artifacts due to the extreme lack of direct observations. Remarkably, upon introducing the newly captured local views, Flash-Refine successfully recovers the crisp geometry of the targeted room. This behavior illustrates *Local Parameter Allocation*: by strictly freezing the well-optimized corridor background, the optimization process is forced to concentrate its densification quotas. The accumulated gradients forcefully seize

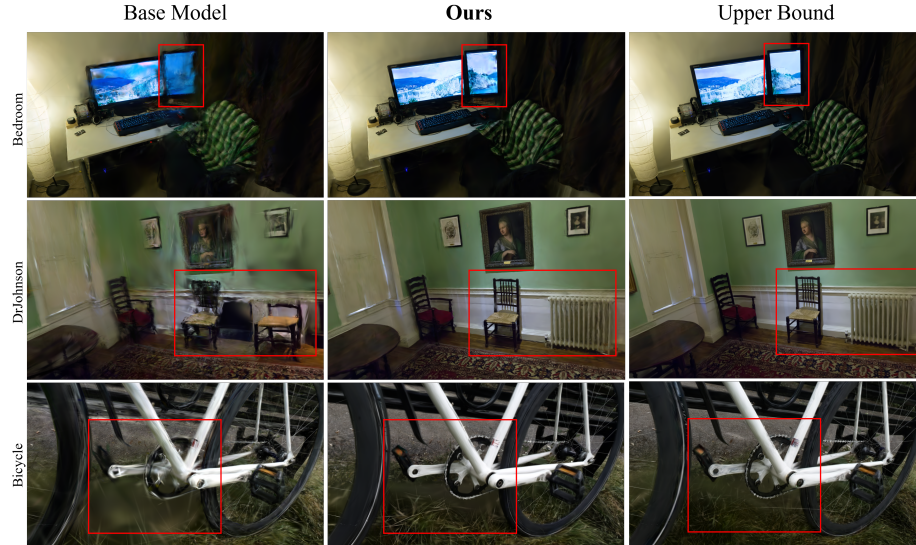


Fig. 3: Qualitative Comparisons on Deep Blending and Mip-NeRF 360. (Row 1: *Bedroom*) The Base Model severely blurs the computer desk due to missing views. Flash-Refine resurrects crisp details on the monitors. (Row 2: *DrJohnson*) We removed all images from an entire room. Remarkably, Flash-Refine reconstructs the entire room from scratch, seeded only by the peripheral corridor points. (Row 3: *Bicycle*) Structural recovery of high-frequency geometries.

the initial low-quality floaters and artifacts residing in the unobserved region. Acting as structural seeding, these points are rapidly split, densified, and re-shaped into accurate, high-frequency room structures without disrupting the global background.

4.3 Ablation Studies

To isolate the contribution of each proposed component, we conduct ablation studies on the *counter* scene. We use *Counter* as a stress case because its repair frustums strongly overlap with kept-view background regions, making visibility interactions more pronounced. The results are summarized in Tab. 4.

Effectiveness of Point-Wise Consensus and Adaptive Depth Prior.

Without the point-wise masking (Variant a), the system degenerates into naive fine-tuning, heavily deteriorating the Kept PSNR. If we only utilize the view frustum intersection without the Depth Prior (Variant b), distant background points (e.g., the wall behind the counter) that coincidentally fall within the new camera frustums are erroneously activated and updated, leading to a measurable drop in background fidelity. The Adaptive Depth Prior ensures an absolute volumetric bound, protecting background structures.

Table 4: Ablation Studies on the *counter* scene. We evaluate the impact of Point-Wise Consensus Masking, Adaptive Depth Prior, and the Anti-Deletion Shield. (Metrics reported on Dropped / Kept PSNR).

Model Variant	Dropped PSNR	Kept PSNR	Active Point Count
(a) w/o Consensus Masking (Naive)	28.91	20.33 (Severe Drop)	All (>2M)
(b) Frustum Masking Only (No Depth Prior)	28.54	24.04 (Slight Drop)	~500K
(c) w/o Anti-Deletion Shield	26.22	20.87	~40K (Deleted)
Flash-Refine (Full)	29.66	27.47	~60K (Bounded)

Necessity of the Anti-Deletion Shield. As discussed in Sec. 3.3, close-up repair shots cause frozen background points to cast excessively large 2D projections. Without the Anti-Deletion Shield (Variant c), the native 3DGS pruning mechanism aggressively flags these essential background structures as anomalies and permanently deletes them, resulting in massive geometric holes in the global scene. Our shield safely bypasses this mechanism, preserving the global structural integrity.

4.4 Real-World Outlook: Large-Scale Aerial Refinement

To demonstrate the scalability and practical industrial value of Flash-Refine, we deploy it on a massive, custom-captured drone dataset of a university campus. In large-scale city digital twins, global retraining can take several days. In our experiment, a specific building exhibited severe blurring due to occlusions in the primary flight path. By providing a short supplementary close-up video of the target building, Flash-Refine perfectly embedded the high-resolution architectural structures back into the massive “ply” base model. As shown in Fig. 4, our paradigm resolves the most critical bottleneck in digital twin maintenance, proving that 3DGS can be treated as an incrementally updatable database rather than a static, one-time artifact.

5 Discussion and Limitations

While Flash-Refine achieves highly efficient local completion, our localized parameter allocation paradigm introduces a few specific challenges that pave the way for future research.

Visibility interactions via Occlusion Bleeding. Masked Gradient Lock guarantees that the parameters of frozen background Gaussians remain unchanged, but it does not fully guarantee unchanged renderings from every unobserved Kept view. Newly split active Gaussians can partially occlude kept-view backgrounds, especially when the repair frustums spatially overlap with background surfaces. The *Counter* scene is such a stress case: the active counter region and the kept-view background are strongly coupled in visibility, so Kept PSNR decreases more than in scenes with better spatial separation. This is a limitation of the current active-region formulation rather than parameter drift



Fig. 4: Outlook: Real-World Campus Digital Twin Maintenance. Flash-Refine deployed on a custom drone-captured campus dataset. When a specific building suffers from texture degradation or missing geometry due to initial flight path limitations, a brief supplementary drone flight allows our method to seamlessly inject the missing high-frequency architectural details directly into the massive city-scale “ply” asset within minutes.

of the frozen background. Even in this stress case, Flash-Refine preserves Kept views much better than Naive Fine-Tuning (27.47 vs. 20.21 dB). Future work could introduce depth, visibility, or volume regularization to explicitly penalize active Gaussians that expand into unobserved background frustums.

Photometric and Illumination Inconsistency. A practical limitation of our in-situ refinement pipeline is the assumption of photometric consistency. When users capture supplementary repair images (I_{new}) at a different time or under varying weather conditions, there are inevitable shifts in lighting, exposure, or white balance compared to the original global capture. Because Flash-Refine rigidly freezes the background parameters (including their Spherical Harmonics), optimizing the active ROI solely on the new images can occasionally produce visible color seams or lighting mismatches at the boundary between the active region and the frozen background. Currently, we assume the inputs are pre-processed with color calibration. For future deployment, integrating an affine color compensation module—similar to the appearance embeddings used in NeRF-W [17]—or optimizing a per-camera exposure affine transform during the refinement step would be highly beneficial to seamlessly blend the newly repaired geometry into the global scene.

Pose, static-scene, and benchmark assumptions. Flash-Refine assumes that repair images are registered into the same coordinate system as the base PLY. In our controlled benchmark, we use dataset-level COLMAP poses to iso-

late local 3DGS repair from localization noise. In practical deployment, pose noise, re-localization error, bundle-adjustment drift, object displacement, and appearance change can all affect repair quality. The method is therefore best understood as a static-scene asset maintenance pipeline, not a dynamic-scene continual learning system.

6 Conclusion

In this paper, we presented Flash-Refine, a highly efficient local incremental completion framework for 3D Gaussian Splatting. By introducing a Point-Wise Consensus Masking mechanism paired with an Adaptive Depth-Bounded Prior, we successfully addressed the prohibitively high computational cost of global retraining and the catastrophic parameter overwriting associated with unconstrained fine-tuning. We revealed that our masked gradient lock induces an advantageous *Local Parameter Allocation*, enabling native densification to optimally reconstruct complex, missing geometries from nearby peripheral seeds. Extensive experiments across object-centric datasets, unconstrained indoor scenes, and large-scale aerial mappings demonstrate that our method achieves a $15\times$ speedup while approaching the localized rendering quality of a full retrain. Flash-Refine offers a robust, optimization-based, and seamlessly integratable solution, paving the way for scalable and low-cost maintenance of large-scale 3D radiance fields in real-world applications.

References

1. Ackermann, J., Kulhanek, J., Cai, S., Xu, H., Pollefeys, M., Wetzstein, G., Guibas, L.J., Peng, S.: Cl-splats: Continual learning of gaussian splatting with local optimization. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (2025)
2. Barron, J.T., Mildenhall, B., Verbin, D., Srinivasan, P.P., Hedman, P.: Mipnerf 360: Unbounded anti-aliased neural radiance fields. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 5470–5479 (2022)
3. Charatan, D., Li, S., Tagliasacchi, A., Sitzmann, V.: pixelsplat: 3d gaussian splats from image pairs for scalable generalizable 3d reconstruction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2024)
4. Chen, Y., et al.: Gaussianeditor: Swift and controllable 3d editing with gaussian splatting. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2024)
5. Fridovich-Keil, S., Yu, A., Tancik, M., Chen, Q., Recht, B., Kanazawa, A.: Plenoxels: Radiance fields without neural networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 5501–5510 (2022)
6. Guédon, A., Lepetit, V.: Sugar: Surface-aligned gaussian splatting for efficient 3d mesh reconstruction and high-quality mesh rendering. In: Proceedings of the

- IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2024)
7. Haque, A., Tancik, M., Efros, A.A., Holynski, A., Kanazawa, A.: Instruct-nerf2nerf: Editing 3d scenes with instructions. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 19740–19750 (2023)
 8. Hedman, P., Philip, J., Price, T., Frahm, J.M., Drettakis, G., Brostow, G.: Deep blending for free-viewpoint image-based rendering. *ACM Transactions on Graphics (ToG)* **37**(6), 1–15 (2018)
 9. Huang, B., Yu, Z., Chen, A., Geiger, A., Gao, S.: 2d gaussian splatting for geometrically accurate radiance fields. In: ACM SIGGRAPH 2024 Conference Papers (2024)
 10. Jain, A., Tancik, M., Abbeel, P.: Putting nerf on a diet: Semantically consistent few-shot view synthesis. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 5885–5894 (2021)
 11. Keetha, N., et al.: Splatam: Splat, track & map 3d gaussians for dense rgb-d slam. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2024)
 12. Kerbl, B., Kopanas, G., Leimkühler, T., Drettakis, G.: 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics (ToG)* **42**(4), 1–14 (2023)
 13. Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A.A., Milan, K., Quan, J., Ramalho, T., Grabska-Barwinska, A., et al.: Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences* **114**(13), 3521–3526 (2017)
 14. Kotovenko, D., Grebenkova, O., Ommer, B.: Edgs: Eliminating densification for efficient convergence of 3dgs. *arXiv preprint arXiv:2504.13204* (2025)
 15. Li, Z., Hoiem, D.: Learning without forgetting. *IEEE transactions on pattern analysis and machine intelligence* **40**(12), 2935–2947 (2017)
 16. Lu, T., Yu, M., Xu, L., Xiangli, Y., Wang, H., Lin, D., Dai, B.: Scaffold-gs: Structured 3d gaussians for view-adaptive rendering. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2024)
 17. Martin-Brualla, R., Radwan, N., Sajjadi, M.S., Barron, J.T., Dosovitskiy, A., Duckworth, D.: Nerf in the wild: Neural radiance fields for unconstrained photo collections. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 7210–7219 (2021)
 18. Matsuki, H., Murai, R., Kelly, P.H., Davison, A.J.: Gaussian Splatting SLAM. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 18039–18048 (2024)
 19. McCloskey, M., Cohen, N.J.: Catastrophic interference in connectionist networks: The sequential learning problem. In: *Psychology of learning and motivation*, vol. 24, pp. 109–165. Elsevier (1989)
 20. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: Nerf: Representing scenes as neural radiance fields for view synthesis. In: *European Conference on Computer Vision (ECCV)*. pp. 405–421. Springer (2020)
 21. Mirzaei, A., Kant, Y., Revsine, J., Fleet, D.J., Brubaker, M.A.: Spin-nerf: Multi-view segmentation and perceptual inpainting with neural radiance fields. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. pp. 20669–20679 (2023)
 22. Müller, T., Evans, A., Schied, C., Keller, A.: Instant neural graphics primitives with a multiresolution hash encoding. *ACM Transactions on Graphics (ToG)* **41**(4), 1–15 (2022)

23. Niemeyer, M., Barron, J.T., Mildenhall, B., Sajjadi, M.S., Geiger, A., Radwan, N.: 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). pp. 5480–5490 (2022)
24. Peng, R., Xu, W., Tang, L., Liao, L., Jiao, J., Wang, R.: Structure consistent gaussian splatting with matching prior for few-shot novel view synthesis. In: Advances in Neural Information Processing Systems (NeurIPS) (2024)
25. PlayCanvas: Supersplat: 3d gaussian splatting editor. <https://supersplat.at/> (2024)
26. Schönberger, J.L., Frahm, J.M.: Structure-from-motion revisited. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016)
27. Shang, H., Chen, M., Feng, K., Li, S., Zhang, Z., Xu, S., Ren, C., Xi, J.: Enhanced 3d gaussian splatting for real-scene reconstruction via depth priors, adaptive denoising, and denoising. *Sensors* **25**(22), 6999 (2025)
28. Sucar, E., Liu, S., Ortiz, J., Davison, A.J.: imap: Implicit mapping and positioning in real-time. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 6229–6238 (2021)
29. Wang, S., Zhang, S., Millerdurai, C., Westermann, R., Stricker, D., Pagani, A.: Inpaint360gs: Efficient object-aware 3d inpainting via gaussian splatting for 360deg *scenes.arXivpreprintarXiv* : 2511.06457(2025)
30. Wang, Z., Bovik, A., Sheikh, H., Simoncelli, E.: Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing* **13**(4), 600–612 (2004). <https://doi.org/10.1109/TIP.2003.819861>
31. Xiong, H., Muttukuru, S., Upadhyay, R., Chari, P., Kadambi, A.: Sparsegs: Real-time 360deg *sparseviewsynthesisusinggaussiansplatting.In : InternationalConferenceon3DVision(3DV)*(2025)
32. Xu, Q., Xu, Z., Philip, J., Bi, S., Shu, Z., Sunkavalli, K., Neumann, U.: Point-nerf: Point-based neural radiance fields. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 5438–5448 (2022)
33. Yang, L., Kang, B., Huang, Z., Zhao, Z., Xu, X., Feng, J., Zhao, H.: Depth anything v2. In: Advances in Neural Information Processing Systems (NeurIPS) (2024)
34. Ye, M., Danelljan, M., Yu, F., Ke, L.: Gaussian grouping: Segment and edit anything in 3d space. In: European Conference on Computer Vision (ECCV) (2024)
35. Yu, Z., Chen, A., Huang, B., Sattler, T., Geiger, A.: Mip-splatting: Alias-free 3d gaussian splatting. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2024)
36. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 586–595 (2018)
37. Zhu, Z., Fan, Z., Jiang, Y., Wang, Z.: Fsgs: Real-time few-shot view synthesis using gaussian splatting. In: European Conference on Computer Vision (ECCV) (2024)
38. Zhu, Z., Peng, S., Larsson, V., Xu, W., Bao, H., Cui, Z., Oswald, M.R., Pollefeys, M.: NICE-SLAM: neural implicit scalable encoding for SLAM. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 12786–12796 (2022)