Blur Issue Matters for Thermal Novel View Synthesis: A Floating Gaussian Suppression Approach

Mingyuan Xie[1] **Sheng Wan**[2] **Jin Xie**[3]
Hang Yang[1] **Liangyun Sun**[1] **Chen Gong**[4*]
[1]School of Computer Science and Engineer, Nanjing University of Science and Technology
[2]College of Artificial Intelligence, Nanjing Agricultural University
[3]School of Intelligence Science and Technology, Nanjing University
[4]School of Automation and Intelligent Sensing, Shanghai Jiao Tong University

Abstract

Novel View Synthesis (NVS) for thermal infrared scenarios has become increasingly important in practical applications. Recent advances in 3D Gaussian Splatting (3DGS) have demonstrated its strong performance for NVS, and a handful of preliminary efforts have explored its extension to thermal NVS. However, existing 3DGS-based methods for thermal NVS often suffer from severe local blur in rendered images. Theoretical and empirical analyses reveal that the single-channel nature of thermal data allows intensity discrepancy to be compensated through opacity adjustment, which can erroneously amplify the opacity of floating Gaussians, especially when the background behind them is insufficiently represented. High-opacity occlusion by floating Gaussians further suppresses the optimization of background Gaussians, preventing their high-frequency surface textures from being faithfully reconstructed and ultimately leading to local blur. To address this issue, we propose a method termed **F**loating **G**aussian **S**uppression (FGS) for thermal NVS. Specifically, we leverage an over-saturation of luminance to introduce an implicit gradient penalty, which guides the optimizer to suppress the opacity of floaters to match the ground truth. By effectively reducing the opacity of these floaters, this mechanism alleviates the occlusion of distant details, thereby mitigating local blur. Moreover, to preserve rendering fidelity, the luminance parameters of Gaussians are optimized concurrently. Experimental results demonstrate that our method effectively mitigates local blur and produces sharper and more faithful thermal reconstruction compared with the baseline methods.

Project page: `https://xmy7857.github.io/FGS`

1 Introduction

Recently, Novel View Synthesis (NVS) for thermal infrared scenarios has become increasingly important in various practical applications, which include night-time surveillance [17], industrial equipment monitoring [15], and low-light autonomous navigation [9]. However, most existing NVS methods focus on the visible-light domain. Different from visible-light imaging, thermal imaging visualizes the surface temperatures by mapping the emitted infrared radiation of objects onto a detector array. These unique characteristics often render visible-light NVS methods suboptimal in thermal scenarios. Specifically, thermal conduction and heat diffusion may reduce temperature discontinuities across object boundaries, which makes edges less distinguishable in thermal observations [2]. In addition, temporal temperature variations make static thermal representations insufficient for modeling

*Corresponding author: Chen Gong <chen.gong@sjtu.edu.cn>

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

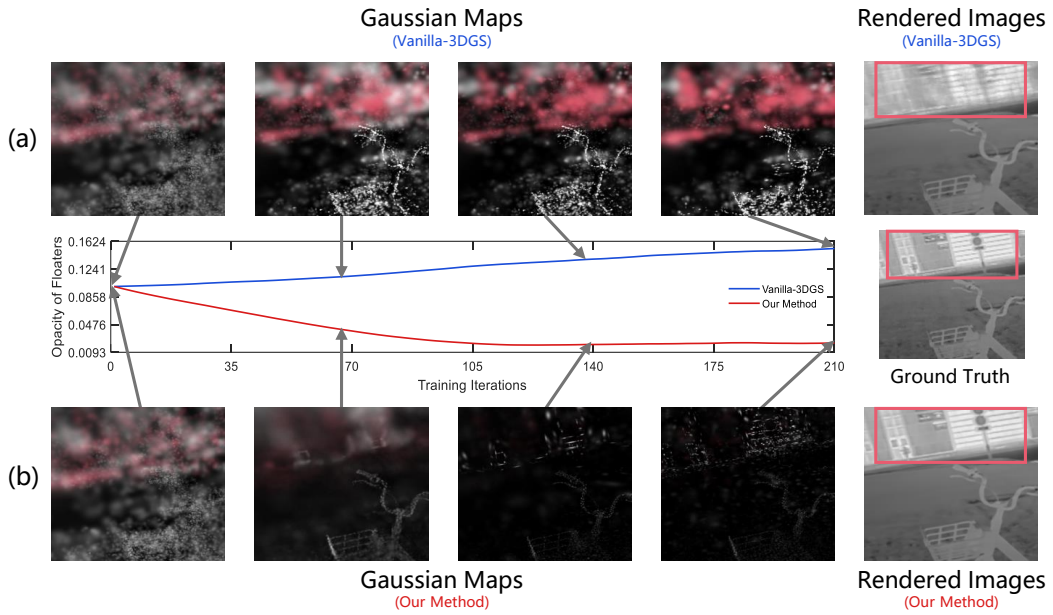

Figure 1: Comparison of Gaussian evolution between (a) vanilla-3DGS and (b) our method. Red points indicate floating Gaussians incorrectly positioned at near-camera depths. In the Gaussian maps, brighter regions indicate higher opacity. In (a), the opacity of floating Gaussians increases over time, which results in local blur in the final rendered image (see the red box). Conversely, in (b), these floating Gaussians are effectively suppressed by reducing the opacity, leading to clearer and more accurate reconstruction. The curve reports the opacity evolution of floating Gaussians during the first 210 training iterations.

time-dependent changes of thermal intensity [27]. Therefore, bridging the gap between visible-light and thermal infrared domains is of great significance, and developing NVS algorithms for thermal infrared scenarios has become a critical task in the computer vision community.

3D Gaussian Splatting (3DGS) [8] has recently shown remarkable capabilities in NVS, which offers high visual fidelity, rapid convergence, and real-time rendering capabilities. Considering these advantages, extending 3DGS to the thermal domain has emerged as a promising direction. For instance, Thermal3D-GS [2] introduces atmospheric transmission and thermal conduction effects to alleviate indistinct edges and discrepancies of thermal intensity when vanilla-3DGS is directly applied to infrared scenes. Besides, NTR-Gaussian [27] enhances temporal rendering fidelity by explicitly modeling night-time heat dissipation via thermodynamic principles. However, as shown in the rendered image of Figure 1(a), these 3DGS-based methods may suffer from severe local blur in rendered images.

Different from visible-light datasets, thermal infrared images usually provide only single-channel intensity information. This limited appearance constraint makes opacity and intensity more ambiguous during optimization. When the rendered intensity is lower than the ground truth, the optimizer may reduce the reconstruction loss by increasing Gaussian opacity, even for near-camera floating Gaussians that are not aligned with actual object surfaces. We also observe that such excessive amplification is more likely to arise in regions with sparse background Gaussians. When high-opacity floaters persistently occlude the background, the resulting reduction in transmittance weakens the gradients received by background Gaussians [8], thereby impeding their optimization and adaptive densification [29]. Consequently, high-frequency surface details may not be faithfully reconstructed [30]. The resulting loss of high-frequency texture appears as local blur in rendered images [24]. As shown in Figure 1(a), the opacity of these floating Gaussians progressively increases with training, resulting in local blur in the final rendered image (see the red box). In contrast, visible-light images provide multi-channel color constraints. For a floating Gaussian to become opaque without increasing reconstruction error, its color must align with the ground truth across three channels simultaneously.

However, this condition is difficult to satisfy in RGB scenes due to the rich chromatic variations. Therefore, the multi-channel nature of visible-light images suppresses the formation of over-opaque floaters, whereas the lack of channel diversity incurs opaque floaters and local blur in thermal scenes.

To address this issue, we propose a strategy termed **F**loating **G**aussian **S**uppression (FGS) for thermal NVS. The key idea is to induce different opacity evolution for floating and valid Gaussians through two-stage optimization. During early training, FGS deliberately creates an over-bright rendering state by fixing Gaussian luminance to the maximum normalized value. This makes near-camera floating Gaussians receive stronger opacity-decreasing updates, while valid background Gaussians aligned with actual surfaces are less affected due to their weaker gradients under occlusion. Afterwards, FGS switches to standard joint optimization. As the opacity of floaters has been reduced, valid background Gaussians receive higher transmittance and lead to improved reconstruction, while the opacity growth of floaters remains suppressed. In this way, FGS suppresses over-opaque floaters without discarding valid Gaussians. To preserve appearance fidelity during the early stage, FGS also performs luminance optimization in parallel by updating the SH coefficients. As shown in Figure 1(b), our method suppresses the opacity of floating Gaussians over training, which effectively alleviates local blur and preserves sharp rendering details.

While some existing works have addressed the issue of "floaters" in the visible-light domain, they typically rely on depth-based supervision [19, 26] or explicit Gaussian pruning strategies [11, 28], which are not well-suited for thermal scenarios. Concretely, depth priors, especially those estimated from images, can be unreliable for thermal observations with weak textures and low contrast. Meanwhile, explicit pruning will introduce extra computational cost and may mistakenly remove valid Gaussians. Other methods mitigate artifacts by adjusting Gaussian parameters or optimization dynamics [18, 32], but they are not specifically designed to suppress opacity-induced floaters in single-channel thermal scenes. Differently, by manipulating the rendered intensity, we guide the floating Gaussians to reduce their opacity adaptively. As a result, opacity-induced local blur can be suppressed without relying on external depth priors or explicit pruning operations.

In summary, our contributions are as follows:

- Theoretical analysis reveals that local blur in thermal NVS is closely related to the excessively high opacity of floating Gaussians.
- We propose a decoupled optimization framework that modulates rendered intensity to reduce the opacity of near-camera floating Gaussians, thereby alleviating opacity-induced local blur in thermal NVS.
- Extensive experiments have demonstrated that our method effectively mitigates local blur and achieves better reconstruction quality than various baseline methods.

2 Related Work

In this section, we review some representative works on 3DGS, as they are closely related to this article.

3DGS represents a 3D scene as a set of anisotropic Gaussian primitives and renders novel views through differentiable splatting and alpha blending [8]. Recent works have improved both the rendering quality and efficiency of 3DGS, and expanded its applications to various downstream tasks. To enhance rendering fidelity, EGGS [31] introduces a hybrid representation that adaptively alternates between 2D and 3D Gaussians. To improve efficiency, Wang et al. [20] optimize scene representations with faster convergence and fewer primitives. To avoid the heavy computational cost of per-scene optimization, ZPressor [21] and FreeSplat [22] employ pre-trained models to directly generate 3D representations for novel scenes from input images. Beyond view synthesis, 3DGS has also shown strong potential in complex downstream tasks. For instance, 3DOT [1] exploits 3DGS to reduce multi-view inconsistencies in 3D object editing, while RAD [5] builds a 3DGS-based environment for reinforcement learning in autonomous driving.

The success of 3DGS has further motivated its extension to thermal infrared reconstruction and novel view synthesis. Methods such as ThermalGaussian [10], TGA-GS [33], and MMOne [6] introduce multimodal regularization to achieve reliable reconstruction. Due to the difficulty in acquiring high-quality RGB data in extremely low-light conditions, Wen et al. [25] introduce a

decoupled framework that reconstructs scenes directly from infrared images. Similarly, Thermal3D-GS [2] models thermal characteristics with neural networks to alleviate artifacts and indistinct edges. On this basis, Nam et al. [14] devise a deformation field to further improve the rendering quality. However, most of these methods focus on static thermal reconstruction and do not explicitly model temporal temperature variations. NTR-Gaussian [27] and ETGS [23] address this issue by learning 4D spatiotemporal temperature representations. Despite these advances, these methods still pay limited attention to opacity-induced floaters, which can cause severe local blur in rendered thermal images.

Existing methods for mitigating floating artifacts in visible-light 3DGS can be broadly grouped into three categories, namely parameter control, depth regularization, and Gaussian pruning. Parameter control methods adjust Gaussian parameters or gradients to suppress unstable primitives [18, 32], but they do not explicitly address the excessive opacity growth of floaters in thermal scenes. Depth regularization methods improve structural consistency through external depth priors [19, 26], yet such priors can be unreliable for thermal images due to weak textures. Gaussian pruning methods explicitly remove suspicious Gaussians [11, 16, 28], but their effectiveness depends heavily on pruning criteria and may incur extra computational cost or erroneous removal of valid primitives. Differently, our method suppresses floater opacity within the optimization process, without relying on external depth priors or specialized pruning operations.

3 Preliminaries

NVS aims to generate photorealistic renderings of a 3D scene from arbitrary viewpoints given a set of training images with known camera poses [3]. Formally, let the training set be $\mathcal{I}^{\text{train}} = \{(\mathbf{I}_k, \mathbf{T}_k)\}_{k=1}^{K}$, where $\mathbf{I}_k$ is a ground-truth image and K is the total number of viewpoints. Here, $\mathbf{T}_k = [\mathbf{R}_k|\mathbf{t}_k]$ denotes the extrinsic parameters with the rotation matrix $\mathbf{R}_k \in \mathbb{R}^{3\times3}$ and the translation vector $\mathbf{t}_k \in \mathbb{R}^3$. Scene representation parameters are optimized to minimize the reconstruction error between rendered and ground-truth images.

3DGS implements this idea by representing a scene with a collection of 3D Gaussian primitives $\mathcal{G} = \{g_i\}_{i=1}^{N^{\text{gau}}}$. Each Gaussian g_i is parameterized by its mean position $\mathbf{x}_i \in \mathbb{R}^3$, covariance matrix $\mathbf{\Sigma}_i \in \mathbb{R}^{3\times3}$, opacity $\sigma_i \in (0, 1)$, and spherical harmonic (SH) coefficients $\mathbf{A}_i \in \mathbb{R}^{(r+1)^2\times l}$ for view-dependent color, where r is the degree of SH and l is the number of color channels. To ensure $\mathbf{\Sigma}_i$ remains positive semi-definite, it is decomposed into a scaling vector $\mathbf{s}_i \in \mathbb{R}^3$ and a rotation quaternion $\mathbf{q}_i \in \mathbb{R}^4$ [8]. Hence, each Gaussian can be written as $g_i = \{\mathbf{x}_i, \sigma_i, \mathbf{A}_i, \mathbf{s}_i, \mathbf{q}_i\}$. The rendering pipeline of 3DGS is formulated as $\hat{\mathbf{I}}_k = \mathcal{R}^{\text{gau}}(\mathcal{G}, \mathcal{C}, \mathbf{T}_k)$, where $\mathcal{C} = \{\mathbf{c}_i\}_{i=1}^{N^{\text{gau}}}$ denotes the set of view-dependent colors, and $\mathbf{c}_i = \text{eval}(\mathbf{A}_i, \mathbf{T}_k) \in [0, 1]^l$ is obtained by evaluating the SH coefficients $\mathbf{A}_i$ under the viewing direction determined by the camera pose $\mathbf{T}_k$. At the pixel level, Gaussians are first projected onto the 2D screen space via local affine approximation, which yields the projected mean $\mathbf{x}'_i$ and covariance $\mathbf{\Sigma}'_i$. For a pixel u with coordinate $\mathbf{u}$, its rendered color $\widehat{\mathbf{C}}^u$ is then computed by alpha blending:

$$\widehat{\mathbf{C}}^u = \sum_{i=1}^{N^{\text{cov}}} \mathbf{c}_i \alpha_i^u \prod_{j=1}^{i-1} (1 - \alpha_j^u). \tag{1}$$

Here, $\alpha_i^u = \sigma_i G_i^u$ is the blending factor of Gaussian g_i at pixel u and N^{cov} is the number of Gaussians covering this pixel. The 2D Gaussian response G_i^u is defined as

$$G_i^u = \exp\left(-\frac{1}{2}(\mathbf{u} - \mathbf{x}'_i)^\top {\mathbf{\Sigma}'_i}^{-1}(\mathbf{u} - \mathbf{x}'_i)\right). \tag{2}$$

4 Analysis on Local Blur in Thermal Scenes

In thermal 3DGS, we observe that local blur often appears in regions where near-camera floating Gaussians occlude distant scene details. These floaters are misaligned with actual object surfaces. In single-channel thermal scenes, when the background behind them is sparsely represented by Gaussians, their opacity is more prone to excessive amplification. As a result, these over-opaque floaters can occlude distant textures and produce local blur in rendered images.

To analyze this phenomenon, we consider the rendering loss at the current viewpoint:

$$\mathcal{L} = \sum_{u \in \mathcal{U}} \left\| \widehat{\mathbf{C}}^u - \mathbf{C}^u \right\|^2, \tag{3}$$

where $\mathcal{U}$ is the set of pixels covered by Gaussian g_i, $\widehat{\mathbf{C}}^u$ is the rendered color or intensity at pixel u, and $\mathbf{C}^u$ is the corresponding ground-truth value. The gradient of the loss with respect to the opacity σ_i of Gaussian g_i can be written as

$$\frac{\partial \mathcal{L}}{\partial \sigma_i} = \sum_{u \in \mathcal{U}} \left(\frac{\partial \mathcal{L}}{\partial \widehat{\mathbf{C}}^u} \right)^\top (\mathbf{c}_i - \widehat{\mathbf{C}}_i^u) \cdot \prod_{j=1}^{i-1} (1 - \alpha_j^u) \cdot G_i^u, \tag{4}$$

where $\mathbf{c}_i$ is the view-dependent color or intensity of Gaussian g_i, and $\widehat{\mathbf{C}}_i^u$ denotes the accumulated contribution of the Gaussians behind g_i at pixel u. According to Equation (3), we have $\frac{\partial \mathcal{L}}{\partial \widehat{\mathbf{C}}^u} = 2(\widehat{\mathbf{C}}^u - \mathbf{C}^u)$. Therefore,

$$\frac{\partial \mathcal{L}}{\partial \sigma_i} = \sum_{u \in \mathcal{U}} 2(\widehat{\mathbf{C}}^u - \mathbf{C}^u)^\top (\mathbf{c}_i - \widehat{\mathbf{C}}_i^u) \cdot \prod_{j=1}^{i-1} (1 - \alpha_j^u) \cdot G_i^u. \tag{5}$$

As indicated by Equation (5), the sign of the opacity-gradient contribution from each pixel u is determined by $(\widehat{\mathbf{C}}^u - \mathbf{C}^u)^\top (\mathbf{c}_i - \widehat{\mathbf{C}}_i^u)$, since the remaining terms are non-negative. When the negative contributions dominate over the pixels covered by g_i, gradient descent tends to increase the opacity σ_i, otherwise the opacity tends to decrease. If the Gaussians behind a floating Gaussian g_i are sparse, $\widehat{\mathbf{C}}_i^u$ becomes small. Therefore, the above condition for the increase of σ_i can be simplified as $(\widehat{\mathbf{C}}^u - \mathbf{C}^u)^\top \mathbf{c}_i < 0$.

In multi-channel images, opacity amplification of misaligned floaters is less likely to occur. When the background contribution is small, increasing the opacity of a floater can reduce the reconstruction loss only if its color is sufficiently consistent with the ground-truth appearance across multiple channels and multiple pixels covered by the floater. Due to rich chromatic variations in visible-light scenes, this condition is difficult to satisfy for floating Gaussians that are not aligned with actual object surfaces. Therefore, visible-light scenes are less susceptible to opacity-driven local blur.

However, the excessive opacity growth of floating Gaussians occurs more frequently in thermal scenarios. Due to the single-channel nature of thermal images, the vector inner product reduces to scalar multiplication. Therefore, if the inequality $(\widehat{\mathbf{C}}^u - \mathbf{C}^u)\mathbf{c}_i < 0$ holds, the opacity σ_i tends to increase. Since the view-dependent luminance $\mathbf{c}_i$ is non-negative, once $\widehat{\mathbf{C}}^u - \mathbf{C}^u < 0$, namely the rendered intensity is lower than the ground truth, the floating Gaussian g_i tends to receive a positive opacity update. Furthermore, thermal images usually exhibit low local contrast, so this condition is likely to hold across neighboring pixels covered by the same floater. The accumulated gradient can therefore amplify the opacity of near-camera floating Gaussians, which causes them to occlude distant textures and produce local blur. The experimental validation of the above theoretical analysis is provided in Section C.3 of the appendix.

5 Method

This section details the proposed FGS (see Figure 2). In our FGS, a two-stage optimization strategy is adopted to suppress opacity-induced floating Gaussians in thermal NVS. During the first N iterations, FGS decouples the optimization into geometry optimization and luminance optimization. Geometry optimization fixes the view-dependent luminance of all Gaussians to the maximum normalized value and updates only the geometric parameters, which encourages near-camera floaters to reduce their opacity. In parallel, luminance optimization updates only the SH coefficients with the normal view-dependent luminance to preserve appearance fidelity. Afterwards, FGS performs joint optimization, where all Gaussian parameters are updated together for high-fidelity reconstruction. During this stage, valid Gaussians can rapidly increase their opacity to reconstruct actual object surfaces, while the previously suppressed floaters are less likely to become over-opaque, thereby mitigating local blur. In the following, we detail the critical steps of our FGS by introducing the geometry optimization and luminance optimization, and explaining the joint optimization of the framework.

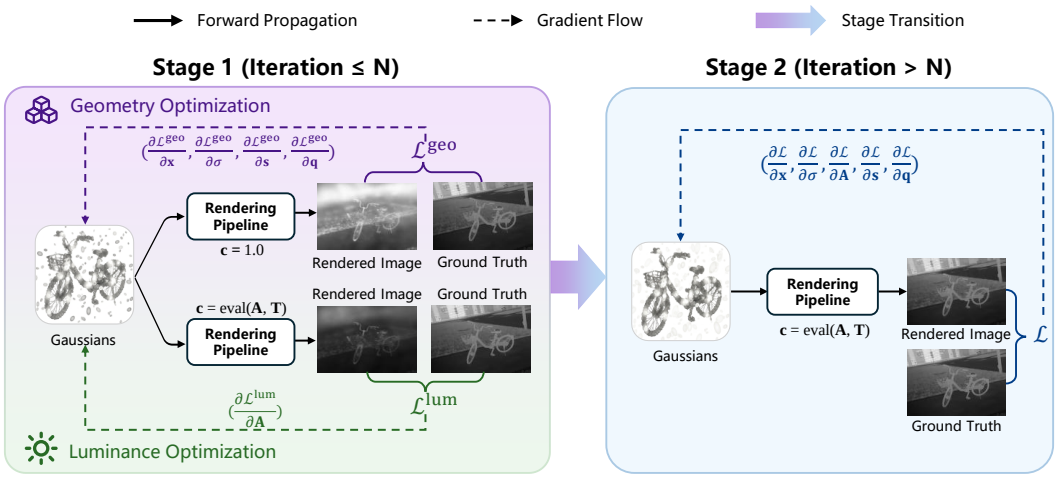

Figure 2: Overview of the proposed FGS method. During the first N iterations (*i.e.*, Stage 1), FGS decouples the optimization into geometry optimization and luminance optimization. In geometry optimization, the view-dependent luminance of all Gaussians $\mathbf{c}$ is fixed to the maximum normalized value 1.0, and the geometry loss $\mathcal{L}^{\text{geo}}$ is used to update only the geometric parameters, including the 3D position $\mathbf{x}$, scaling factor $\mathbf{s}$, rotation quaternion $\mathbf{q}$, and opacity σ, thereby suppressing over-opaque floaters. Meanwhile, in luminance optimization, we compute the luminance loss $\mathcal{L}^{\text{lum}}$ with $\mathbf{c}$ derived from the SH coefficients $\mathbf{A}$ and camera pose $\mathbf{T}$, and update only $\mathbf{A}$ while keeping the geometric parameters fixed. After the N-th iteration, FGS switches to Stage 2, where the standard rendering loss $\mathcal{L}$ is used to jointly optimize all Gaussian parameters for high-fidelity reconstruction.

5.1 Geometry Optimization

The goal of geometry optimization is to suppress the excessive opacity of near-camera floating Gaussians during the early training stage. To this end, we render the scene by fixing the view-dependent luminance of all Gaussians to the maximum normalized value, *i.e.*, 1.0. The rendering process is formulated as

$$\hat{\mathbf{I}}_k^{\text{geo}} = \mathcal{R}^{\text{gau}}(\mathcal{G}, \mathcal{C}^{\text{geo}}, \mathbf{T}_k), \tag{6}$$

where $\mathcal{C}^{\text{geo}} = \{\mathbf{c}_i \mid \mathbf{c}_i = 1.0\}_{i=1}^{N^{\text{gau}}}$ denotes the fixed luminance set used in geometry optimization. The corresponding rendering loss is defined as

$$\mathcal{L}^{\text{geo}} = \|\hat{\mathbf{I}}_k^{\text{geo}} - \mathbf{I}_k\|^2. \tag{7}$$

During back-propagation, the gradients of the SH coefficients $\{\mathbf{A}_i\}_{i=1}^{N^{\text{gau}}}$ are stopped, while the geometric parameters $\{\mathbf{x}_i, \sigma_i, \mathbf{s}_i, \mathbf{q}_i\}_{i=1}^{N^{\text{gau}}}$ are updated. In addition, Adaptive Density Control (ADC), including densification and pruning, is performed only in geometry optimization, so that the scene structure is refined according to the geometry-related gradients.

This design is motivated by the opacity-gradient analysis in Section 4. When the luminance of Gaussians is fixed to 1.0, the rendered intensity of pixels affected by frontal floaters tends to become higher than the ground-truth intensity. In this case, $\widehat{\mathbf{C}}^u - \mathbf{C}^u > 0$ is more likely to hold. Since the fixed luminance $\mathbf{c}_i$ is positive, the opacity gradient of a floating Gaussian tends to be positive, and gradient descent therefore decreases its opacity. As a result, near-camera floaters are encouraged to become transparent rather than increasingly opaque. Note that although valid Gaussians aligned with actual object surfaces may also receive decreasing opacity updates, their gradients are usually much weaker. This is because these background Gaussians are partially occluded by frontal floaters, leading to a smaller transmittance term $\prod_{j=1}^{i-1}(1-\alpha_j^u)$ in their opacity gradients. Consequently, during the first N iterations, geometry optimization suppresses the opacity of floating Gaussians strongly, while largely preserving the opacity of valid background Gaussians.

5.2 Luminance Optimization

While geometry optimization suppresses over-opaque floaters by fixing Gaussian luminance, the SH coefficients are not updated. This may limit the ability of the model to learn accurate view-dependent thermal appearance during the early training stage. To preserve rendering fidelity, we perform luminance optimization in parallel, where only the SH coefficients are updated while the geometric parameters are kept fixed.

The rendering process for luminance optimization is defined as

$$\hat{\mathbf{I}}_k^{\text{lum}} = \mathcal{R}^{\text{gau}}(\mathcal{G}, \mathcal{C}^{\text{lum}}, \mathbf{T}_k). \tag{8}$$

Here, the view-dependent luminance $\mathbf{c}_i$ is obtained from the SH coefficients $\mathbf{A}_i$ and the camera pose $\mathbf{T}_k$, and the luminance set is defined as

$$\mathcal{C}^{\text{lum}} = \{\mathbf{c}_i \mid \mathbf{c}_i = \text{eval}(\mathbf{A}_i, \mathbf{T}_k)\}_{i=1}^{N^{\text{gau}}}. \tag{9}$$

The corresponding rendering loss is

$$\mathcal{L}^{\text{lum}} = \|\hat{\mathbf{I}}_k^{\text{lum}} - \mathbf{I}_k\|^2. \tag{10}$$

During back-propagation, the gradients of all geometric parameters $\{\mathbf{x}_i, \sigma_i, \mathbf{s}_i, \mathbf{q}_i\}_{i=1}^{N^{\text{gau}}}$ are stopped, and only the SH coefficients $\{\mathbf{A}_i\}_{i=1}^{N^{\text{gau}}}$ are updated. In this way, the model can improve image fidelity while suppressing floating Gaussians during early training.

5.3 Joint Optimization

The decoupled optimization suppresses over-opaque floating Gaussians while maintaining the learning of view-dependent luminance. However, since geometry and luminance are optimized separately, joint optimization is still required to recover high-fidelity reconstruction. Therefore, after the first N iterations, FGS switches to joint optimization, where all Gaussian parameters are updated together under the standard rendering process.

In this stage, the rendered image is computed as

$$\hat{\mathbf{I}}_k = \mathcal{R}^{\text{gau}}(\mathcal{G}, \mathcal{C}, \mathbf{T}_k). \tag{11}$$

Here, $\mathcal{C} = \{\mathbf{c}_i\}_{i=1}^{N^{\text{gau}}}$ is the standard view-dependent luminance set. The rendering loss is computed as

$$\mathcal{L} = \|\hat{\mathbf{I}}_k - \mathbf{I}_k\|^2. \tag{12}$$

All Gaussian parameters, including the geometric parameters $\{\mathbf{x}_i, \sigma_i, \mathbf{s}_i, \mathbf{q}_i\}_{i=1}^{N^{\text{gau}}}$ and the SH coefficients $\{\mathbf{A}_i\}_{i=1}^{N^{\text{gau}}}$, are then jointly optimized until convergence.

After switching to joint optimization, the artificially saturated luminance used in geometry optimization is removed. The rendered intensities may therefore become lower than the ground-truth intensities in some regions, causing valid Gaussians to increase their opacity to better reconstruct the actual scene surfaces. Since the opacity of frontal floaters has already been reduced during decoupled optimization, the transmittance to background Gaussians becomes higher. This allows valid background Gaussians behind the floaters to receive stronger gradients and contribute more effectively to reconstruction. As a result, the accumulated background intensity $\widehat{\mathbf{C}}_i^u$ for frontal floaters also increases. According to Equation (5), when $\widehat{\mathbf{C}}^u - \mathbf{C}^u < 0$, a larger $\widehat{\mathbf{C}}_i^u$ reduces $\mathbf{c}_i - \widehat{\mathbf{C}}_i^u$ and weakens the negative gradient that increases floater opacity. Therefore, although the opacity of some floaters may increase again, excessive floater-opacity growth is suppressed as valid background Gaussians contribute more strongly to the rendered intensity. We provide experimental evidence for this suppression effect in Section C.4 of the appendix.

In summary, FGS suppresses floating Gaussians through different opacity updates across the two stages. During decoupled optimization, the opacity of floating Gaussians decreases more strongly than that of valid Gaussians. During joint optimization, valid Gaussians recover their opacity more effectively, while the opacity growth of floating Gaussians remains suppressed. This allows FGS to reduce over-opaque floaters without weakening valid Gaussians, thereby mitigating opacity-induced local blur in thermal NVS.

Table 1: Quantitative comparison on TI-NSD and ThermoScenes datasets, where "AVG" denotes the average performance over all the scenes across the two datasets. Best results for each metric are highlighted in **bold**. "↑" indicates that higher values correspond to better performance, while "↓" indicates that lower values correspond to better performance.

	TI-NSD			**ThermoScenes**			**AVG**		
Method	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
InstantNGP-Base [13]	24.620	0.811	0.332	27.610	0.918	0.148	25.618	0.847	0.271
InstantNGP-Big [13]	24.910	0.812	0.323	28.099	0.920	0.146	25.973	0.848	0.264
Plenoxel [4]	23.280	0.805	0.390	25.070	0.906	0.164	23.877	0.839	0.315
3DGS [8]	32.852	0.943	0.194	30.937	0.965	0.073	32.144	0.950	0.156
PixelGS [32]	34.216	0.954	0.182	32.407	0.975	0.064	33.613	0.961	0.142
AD-GS [16]	33.714	0.951	0.199	33.427	0.975	0.062	33.618	0.959	0.153
NTR-Gaussian [27]	27.887	0.902	0.291	25.640	0.834	0.085	27.138	0.882	0.222
Thermal3D-GS [2]	35.040	0.955	0.187	32.261	0.973	0.066	34.114	0.961	0.147
Veta-GS [14]	35.703	0.957	0.177	34.511	0.979	0.056	35.305	0.964	0.137
FGS (Ours)	**36.330**	**0.960**	**0.173**	**35.908**	**0.981**	**0.051**	**36.189**	**0.967**	**0.132**

6 Experiments

To demonstrate the effectiveness of our method, we conduct experiments on multiple thermal benchmarks. Both quantitative and qualitative results are provided for comprehensive evaluation.

6.1 Experimental Setup

Datasets and Metrics. We evaluate the proposed FGS on TI-NSD [2] and ThermoScenes [7] datasets. TI-NSD contains 20 scenes covering outdoor, indoor, and UAV-captured scenarios, while ThermoScenes consists of 10 scenes capturing both indoors and outdoors. Following established protocols [12], we adopt PSNR, SSIM, and LPIPS as evaluation metrics.

Baseline Methods. For comparison, we evaluate FGS against representative NVS methods from both thermal and visible-light domains, including vanilla-3DGS [8], PixelGS [32], AD-GS [16], Thermal3D-GS [2], NTR-Gaussian [27], and Veta-GS [14]. Among them, Thermal3D-GS, NTR-Gaussian, and Veta-GS are 3DGS methods designed for thermal scenes, with Veta-GS serving as a strong recent baseline method for thermal NVS. Besides, PixelGS and AD-GS are included as representative visible-light 3DGS methods for mitigating floater artifacts. We also compare our FGS with two representative NeRF-based methods, *i.e.* Instant-NGP [13] and Plenoxel [4]. For Instant-NGP, we evaluate both the base model and the enlarged variant, denoted as "InstantNGP-Base" and "InstantNGP-Big", respectively.

Implementation Details. FGS is implemented on top of Veta-GS [14]. For most scenes, the switch point is set to the 500th iteration, *i.e.*, $N = 500$. Detailed parameter settings and sensitivity analysis are provided in the appendix. All experiments are conducted on NVIDIA RTX 3090 GPUs. Concretely, NTR-Gaussian uses two GPUs following its original setting, while the other methods are evaluated on a single GPU.

6.2 Quantitative and Qualitative Results

The quantitative results in Table 1 demonstrate that FGS achieves the best overall performance on both TI-NSD and ThermoScenes datasets compared with all baseline methods. Specifically, FGS obtains the highest average PSNR, SSIM, and lowest average LPIPS, which indicates superior reconstruction accuracy and perceptual quality. Compared with recent 3DGS methods designed for thermal scenes, such as Thermal3D-GS and Veta-GS, FGS achieves better performance, demonstrating its effectiveness in thermal NVS. In addition, FGS also outperforms PixelGS and AD-GS, which are designed to mitigate floater artifacts in visible-light 3DGS. This suggests that methods developed for visible-light scenes may not fully address the opacity-induced floaters that appear in thermal scenarios.

As shown in Figure 3, our FGS method significantly reduces local blur and improves overall rendering quality on both datasets. The boxed areas in the images rendered by baseline methods exhibit noticeable texture degradation and blurred edges. Differently, the corresponding areas in the images rendered by FGS display sharper textures and clearer edges. Overall, these results demonstrate

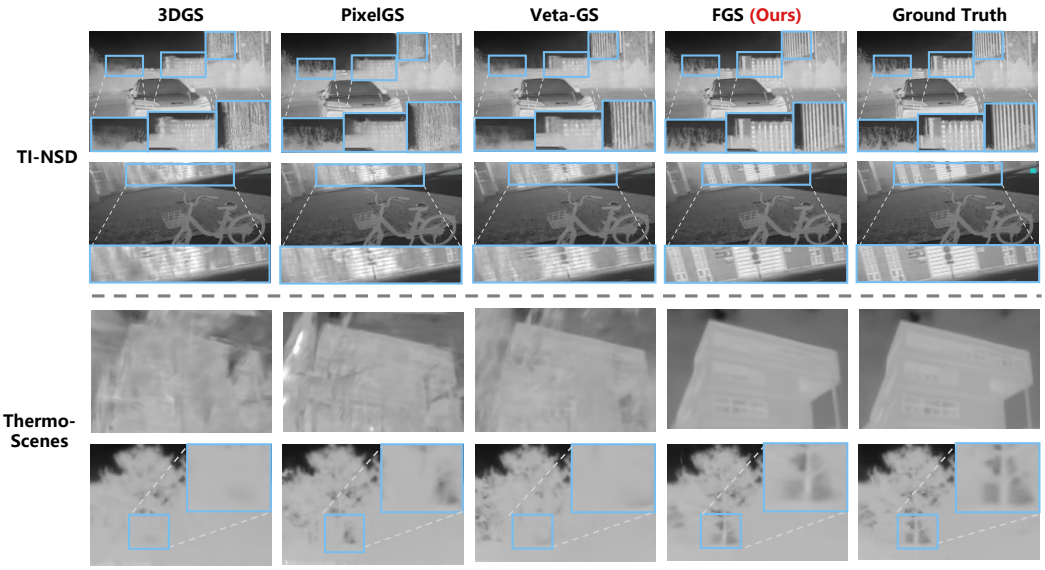

Figure 3: Visual comparison of 3DGS, PixelGS, Veta-GS, and FGS on TI-NSD and ThermoScenes datasets. The blue-boxed regions indicate enlarged local areas for detailed comparison.

Table 2: Ablation study of the proposed components, including Geometry Optimization (GO) and Luminance Optimization (LO). Average PSNR, SSIM, and LPIPS on both TI-NSD and ThermoScenes datasets are reported.

	TI-NSD			**ThermoScenes**		
Method	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
FGS w/o GO&LO	35.703	0.957	0.177	34.511	0.979	0.056
FGS w/o GO	35.751	0.958	0.175	33.002	0.953	0.060
FGS w/o LO	35.976	0.959	0.172	35.521	0.978	0.054
FGS	36.330	0.960	0.173	35.908	0.981	0.051

the effectiveness of our FGS in alleviating local blur and improving rendering quality. Detailed per-scene metrics and additional visual results are provided in the appendix.

6.3 Ablation Study

We conduct an ablation study on the TI-NSD and ThermoScenes datasets to evaluate the contributions of Geometry Optimization (GO) and Luminance Optimization (LO). The results are summarized in Table 2. When both geometry optimization and luminance optimization are removed, FGS degenerates to the base Veta-GS model, denoted as "FGS w/o GO&LO". By removing geometry optimization and keeping only luminance optimization, the model cannot explicitly suppress the opacity of floating Gaussians during the early stage. As a result, "FGS w/o GO" performs worse than the complete FGS, which confirms that geometry optimization is essential for mitigating opacity-induced local blur. In contrast, when geometry optimization is preserved and luminance optimization is removed, *i.e.*, "FGS w/o LO", the model already achieves clear improvements over the baseline. This demonstrates that geometry optimization plays the main role in suppressing over-opaque floaters. Overall, the statistics in Table 2 demonstrate that geometry optimization and luminance optimization play complementary roles in FGS.

7 Conclusion

In this work, we investigate the local blur problem in thermal NVS, which has been previously overlooked, and reveal its connection to over-opaque floating Gaussians. Through gradient-based analysis, we show that the single-channel nature of thermal observations makes near-camera floaters more prone to erroneous opacity amplification, especially when the background is sparsely represented by Gaussians. Based on this insight, we propose a simple yet effective optimization strategy termed FGS that suppresses the opacity of floating Gaussians during early training while preserving view-dependent luminance learning. Extensive experiments on thermal NVS benchmarks demonstrate that FGS improves reconstruction quality, produces sharper local details, and effectively alleviates opacity-induced blur.

Acknowledgements

This work was supported by the National Natural Science Foundation of China under Grant Nos. 62336003, 12371510, 62506171, and 62276144; the Shanghai Municipal Science and Technology Major Project under Grant No. 2025SHZDZX025G12; the Natural Science Foundation of Jiangsu Province under Grant No. BK20241469; and the Fundamental Research Funds for the Central Universities under Grant No. YDZX2026052.

References

[1] X. Cao, B. Lin, B. Wang, Z. Huang, and R. Tan. 3DOT: Texture Transfer for 3DGS Objects from a Single Reference Image. In *Advances in Neural Information Processing Systems*, pages 173944–173964, 2025.

[2] Q. Chen, S. Shu, and X. Bai. Thermal3D-GS: Physics-Induced 3D Gaussians for Thermal Infrared Novel-View Synthesis. In *European Conference on Computer Vision*, pages 253–269, 2024.

[3] Z. Chen, X. Chen, C. Ye, S. Wu, and X. Wu. Neural Radiance Fields Assisted by Image Features for UAV Scene Reconstruction. *Scientific Reports*, 15:30608, 2025.

[4] S. Fridovich-Keil, A. Yu, M. Tancik, Q. Chen, B. Recht, and A. Kanazawa. Plenoxels: Radiance Fields without Neural Networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5491–5500, 2022.

[5] H. Gao, S. Chen, B. Jiang, B. Liao, Y. Shi, X. Guo, Y. Pu, H. Yin, X. Li, X. Zhang, Y. Zhang, W. Liu, Q. Zhang, and X. Wang. RAD: Training an End-to-End Driving Policy via Large-Scale 3DGS-based Reinforcement Learning. In *Advances in Neural Information Processing Systems*, pages 36642–36667, 2025.

[6] Z. Gu and B. Wang. MMOne: Representing Multiple Modalities in One Scene. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1088–1098, October 2025.

[7] M. Hassan, F. Forest, O. Fink, and M. Mielle. ThermoNeRF: A Multimodal Neural Radiance Field for Joint RGB-Thermal Novel View Synthesis of Building Facades. *Advanced Engineering Informatics*, 65:103345, 2025.

[8] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis. 3D Gaussian Splatting for Real-Time Radiance Field Rendering. *ACM Transactions on Graphics*, 42(4):139:1–139:14, 2023.

[9] Y. Li, J. Moreau, and J. Ibanez-Guzman. Emergent Visual Sensors for Autonomous Vehicles. *IEEE Transactions on Intelligent Transportation Systems*, 24(5):4716–4737, 2023.

[10] R. Lu, H. Chen, Z. Zhu, Y. Qin, M. Lu, L. Zhang, C. Yan, and A. Xue. ThermalGaussian: Thermal 3D Gaussian splatting. In *International Conference on Learning Representations*, 2025.

[11] Z. Lu, H. Gu, and B. Huang. FFGS: Preemptive Filtering for Robust 3D Gaussian Splatting. In *Proceedings of International Conference on Computer Information Science and Artificial Intelligence*, pages 1665–1675, 2025.

[12] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng. NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis. In *European Conference on Computer Vision*, pages 405–421, 2020.

[13] T. Müller, A. Evans, C. Schied, and A. Keller. Instant Neural Graphics Primitives with A Multiresolution Hash Encoding. *ACM Transactions on Graphics*, 41(4):1–15, 2022.

[14] M. Nam, W. Park, M. Kim, H. Hur, and S. Lee. Veta-GS: View-Dependent Deformable 3D Gaussian Splatting for Thermal Infrared Novel-View Synthesis. In *IEEE International Conference on Image Processing*, pages 965–970, 2025.

[15] A. S. Nazmul Huda and S. Taib. Application of Infrared Thermography for Predictive/Preventive Maintenance of Thermal Defect in Electrical Equipment. *Applied Thermal Engineering*, 61(2):220–227, 2013.

[16] G. Patle, N. Girgaonkar, N. Somraj, and R. Soundararajan. AD-GS: Alternating Densification for Sparse-Input 3D Gaussian Splatting. In *Proceedings of the SIGGRAPH Asia 2025 Conference Papers*, pages 1–11, 2025.

[17] P. Shyam and H. Yoo. Lightweight Thermal Super-Resolution and Object Detection for Robust Perception in Adverse Weather Conditions. In *Winter Conference on Applications of Computer Vision*, pages 7471–7482, 2024.

[18] J. Wang, P. Zhou, C. Li, R. Quan, and J. Qin. Low-Frequency First: Eliminating Floating Artifacts in 3D Gaussian Splatting. In *International Conference on Cyberworlds*, pages 48–55, 2025.

[19] L. Wang, Q. Ren, K. Liao, H. Wang, Z. Chen, and Y. Tang. StableGS: A Floater-Free Framework for 3D Gaussian Splatting. *arXiv preprint arXiv: 2503.18458*, 2025.

[20] T. Wang, M. Li, G. Zeng, C. Meng, and Q. Zhang. Gaussian Herding across Pens: An Optimal Transport Perspective on Global Gaussian Reduction for 3DGS. In *Advances in Neural Information Processing Systems*, pages 175354–175379, 2025.

[21] W. Wang, D. Y. Chen, Z. Zhang, D. Shi, A. Liu, and B. Zhuang. ZPressor: Bottleneck-Aware Compression for Scalable Feed-Forward 3DGS. In *Advances in Neural Information Processing Systems*, pages 125793–125822, 2025.

[22] Y. Wang, T. Huang, H. Chen, and G. H. Lee. FreeSplat: Generalizable 3D Gaussian Splatting towards Free View Synthesis of Indoor Scenes. In *Advances in Neural Information Processing Systems*, pages 107326–107349, 2024.

[23] Z. Wang, H. Ling, W. Zhang, Y. Sun, and Q. Sun. ETGS: Explicit Thermodynamics Gaussian Splatting for Dynamic Thermal Reconstruction. In *International Conference on Learning Representations*, 2026.

[24] A. B. Watson and A. J. Ahumada. Blur Clarified: A Review and Synthesis of Blur Discrimination. *Journal of Vision*, 11(5):1–23, 2011.

[25] X. Wen, G. Fang, B. Yang, and B. Wang. Geometry Aware 3D Multiview Thermal Reconstruction with Emissive Residual Decomposition Gaussian Splatting. In *Proceedings of the ACM International Workshop on Thermal Sensing and Computing*, pages 7–12, 2025.

[26] H. Xiong, S. Muttukuru, H. Xiao, R. Upadhyay, P. Chari, Y. Zhao, and A. Kadambi. SparseGS: Sparse View Synthesis Using 3D Gaussian Splatting. In *International Conference on 3D Vision*, pages 1032–1041, 2025.

[27] K. Yang, Y. Liu, Z. Cui, Y. Liu, M. Zhang, S. Yan, and Q. Wang. NTR-Gaussian: Nighttime Dynamic Thermal Reconstruction with 4D Gaussian Splatting Based on Thermodynamics. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 691–700, 2025.

[28] S. Yang, C. Im, J. W. Lee, and J. B. Choi. TIDI-GS: Floater Suppression in 3D Gaussian Splatting for Enhanced Indoor Scene Fidelity. *arXiv preprint arXiv: 2601.09291*, 2026.

[29] Z. Ye, W. Li, S. Liu, P. Qiao, and Y. Dou. AbsGS: Recovering Fine Details in 3D Gaussian Splatting. In *ACM International Conference on Multimedia*, pages 1053–1061, 2024.

[30] Z. Zeng, Y. Wang, L. Ju, and T. Guan. Frequency-Aware Density Control via Reparameterization for High-Quality Rendering of 3D Gaussian Splatting. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(9):9833–9841, 2025.

[31] Y. Zhang, G. Sun, and C. Chen. EGGS: Exchangeable 2D/3D Gaussian Splatting for Geometry-Appearance Balanced Novel View Synthesis. In *Advances in Neural Information Processing Systems*, pages 163927–163953, 2025.

[32] Z. Zhang, W. Hu, Y. Lao, T. He, and H. Zhao. Pixel-GS: Density Control with Pixel-aware Gradient for 3D Gaussian Splatting. In *European Conference on Computer Vision*, pages 326–342, 2024.

[33] C. Zou, Q. Ma, J. Wang, R. Lu, M. Lu, and Z. Qu. TGA-GS: Thermal Geometrically Accurate Gaussian Splatting. *Applied Sciences*, 15(9):4666, 2025.

Appendix

A Detailed Derivation

The derivation of the loss gradient with respect to opacity (*i.e.*, Equation (4)) is crucial for the analysis in Section 4. Therefore, the detailed derivation of Equation (4) is provided below.

For a Gaussian g_i, we denote $\mathcal{U}$ as the set of pixels covered by it. Differentiating the per-view loss $\mathcal{L}$ with respect to the opacity σ_i by the chain rule yields

$$\frac{\partial \mathcal{L}}{\partial \sigma_i} = \sum_{u \in \mathcal{U}} \left(\frac{\partial \mathcal{L}}{\partial \widehat{\mathbf{C}}^u} \right)^\top \frac{\partial \widehat{\mathbf{C}}^u}{\partial \alpha_i^u} \cdot \frac{\partial \alpha_i^u}{\partial \sigma_i}, \tag{13}$$

where

$$\frac{\partial \widehat{\mathbf{C}}^u}{\partial \alpha_i^u} = (\mathbf{c}_i - \widehat{\mathbf{C}}_i^u) \prod_{j=1}^{i-1} \left(1 - \alpha_j^u\right), \tag{14}$$

and

$$\frac{\partial \alpha_i^u}{\partial \sigma_i} = G_i^u. \tag{15}$$

In Equations (13)-(15), $\alpha_i^u = \sigma_i G_i^u$ denotes the blending factor of g_i, $\mathbf{c}_i$ is the view-dependent color or intensity of g_i, and G_i^u is the Gaussian function value of Gaussian g_i at the pixel u. Besides, $\widehat{\mathbf{C}}^u$ denotes the rendered color of the pixel u, which can be expressed as

$$\widehat{\mathbf{C}}^u = \sum_{i=1}^{N^{\text{cov}}} \mathbf{c}_i \alpha_i^u \prod_{j=1}^{i-1} (1 - \alpha_j^u). \tag{16}$$

Here, N^{cov} is the number of the Gaussians that cover the pixel u. In Equation (14), $\widehat{\mathbf{C}}_i^u$ is the color rendered by the Gaussians farther from the camera than g_i. Assuming that the Gaussians have been arranged in ascending order of their distance to the camera, $\widehat{\mathbf{C}}_i^u$ can be expressed as

$$\widehat{\mathbf{C}}_i^u = \sum_{k=i+1}^{N^{\text{cov}}} \mathbf{c}_k \alpha_k^u \prod_{j=i+1}^{k-1} (1 - \alpha_j^u). \tag{17}$$

Substituting Equations (14) and (15) into Equation (13), we obtain

$$\frac{\partial \mathcal{L}}{\partial \sigma_i} = \sum_{u \in \mathcal{U}} \left(\frac{\partial \mathcal{L}}{\partial \widehat{\mathbf{C}}^u} \right)^\top (\mathbf{c}_i - \widehat{\mathbf{C}}_i^u) \prod_{j=1}^{i-1} \left(1 - \alpha_j^u\right) \cdot G_i^u. \tag{18}$$

At this point, we have obtained the form of Equation (4).

B Training Process of FGS

To further illustrate the training process of the proposed method, the detailed pseudocode is presented in Algorithm 1.

C Additional Experiments

This section provides additional quantitative and qualitative results on both TI-NSD and ThermoScenes datasets.

C.1 Parameter Configuration

The key hyperparameter of FGS is the switch point N, which controls the duration of opacity suppression during the decoupled optimization stage. A larger N applies opacity suppression for more iterations and is therefore suitable for scenes with severe floating Gaussian artifacts. For all

Algorithm 1 Training Pipeline of FGS

Require: Training set $\mathcal{I}^{\text{train}} = \{(\mathbf{I}_k, \mathbf{T}_k)\}_{k=1}^{K}$, initial Gaussians $\mathcal{G} = \{g_i = \{\mathbf{x}_i, \sigma_i, \mathbf{A}_i, \mathbf{s}_i, \mathbf{q}_i\}\}_{i=1}^{N^{\text{gau}}}$, switch point N, and total iterations T.
Ensure: Optimized Gaussians $\mathcal{G}$.

// **Stage 1: Decoupled Optimization**
for iteration $t = 1 : N$ **do**
  Sample a training view $(\mathbf{I}_k, \mathbf{T}_k)$ from $\mathcal{I}^{\text{train}}$
  // **Geometry Optimization**
  $\mathcal{C}^{\text{geo}} = \{\mathbf{c}_i \big| \mathbf{c}_i = 1.0\}$ ▷ *the maximum luminance value is 1.0*
  $\hat{\mathbf{I}}_k^{\text{geo}} = \mathcal{R}^{\text{gau}}(\mathcal{G}, \mathcal{C}^{\text{geo}}, \mathbf{T}_k)$
  $\mathcal{L}^{\text{geo}} = \|\hat{\mathbf{I}}_k^{\text{geo}} - \mathbf{I}_k\|^2$
  Update geometric parameters $\{\mathbf{x}_i, \sigma_i, \mathbf{s}_i, \mathbf{q}_i\}_{i=1}^{N^{\text{gau}}}$ while stopping gradients of $\{\mathbf{A}_i\}_{i=1}^{N^{\text{gau}}}$
  $\mathcal{C}^{\text{lum}} = \{\mathbf{c}_i \big| \mathbf{c}_i = \text{eval}(\mathbf{A}_i, \mathbf{T}_k)\}$
  $\hat{\mathbf{I}}_k^{\text{lum}} = \mathcal{R}^{\text{gau}}(\mathcal{G}, \mathcal{C}^{\text{lum}}, \mathbf{T}_k)$
  $\mathcal{L}^{\text{lum}} = \|\hat{\mathbf{I}}_k^{\text{lum}} - \mathbf{I}_k\|^2$
  Update SH coefficients $\{\mathbf{A}_i\}_{i=1}^{N^{\text{gau}}}$ while stopping gradients of $\{\mathbf{x}_i, \sigma_i, \mathbf{s}_i, \mathbf{q}_i\}_{i=1}^{N^{\text{gau}}}$
end for
// **Stage 2: Joint Optimization**
for iteration $t = N + 1 : T$ **do**
  Sample a training view $(\mathbf{I}_k, \mathbf{T}_k)$ from $\mathcal{I}^{\text{train}}$
  $\mathcal{C} = \{\mathbf{c}_i \big| \mathbf{c}_i = \text{eval}(\mathbf{A}_i, \mathbf{T}_k)\}$
  $\hat{\mathbf{I}}_k = \mathcal{R}^{\text{gau}}(\mathcal{G}, \mathcal{C}, \mathbf{T}_k)$
  $\mathcal{L} = \|\hat{\mathbf{I}}_k - \mathbf{I}_k\|^2$
  Jointly update all parameters of $\mathcal{G}$
end for
return $\mathcal{G}$

Table A1: Detailed quantitative results of different methods on outdoor, indoor, and UAV-captured scenes of TI-NSD dataset. Best results for each metric are highlighted in **bold**. "↑" indicates that higher values correspond to better performance, while "↓" indicates that lower values correspond to better performance.

	Outdoor			**Indoor**			**UAV-Captured**		
Method	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
InstantNGP-Base [13]	26.000	0.837	0.302	26.990	0.916	0.291	20.860	0.681	0.404
InstantNGP-Big [13]	26.450	0.839	0.292	27.460	0.918	0.289	20.820	0.680	0.387
Plenoxel [4]	22.150	0.768	0.433	22.130	0.867	0.385	25.560	0.780	0.351
3DGS [8]	30.667	0.923	0.206	32.954	0.951	0.260	34.936	0.954	0.116
PixelGS [32]	32.720	0.943	0.175	34.481	0.959	0.250	35.650	0.960	0.109
AD-GS [16]	32.140	0.939	0.199	34.511	0.960	0.260	34.621	0.956	0.127
NTR-Gaussian [27]	27.456	0.882	0.302	27.537	0.931	0.308	28.797	0.890	0.257
Thermal3D-GS [2]	32.600	0.942	0.187	36.010	0.962	0.252	36.740	0.962	0.112
Veta-GS [14]	33.895	0.949	0.169	36.804	0.963	0.243	**36.897**	**0.963**	**0.103**
FGS (Ours)	**34.724**	**0.952**	**0.165**	**37.483**	**0.965**	**0.242**	36.858	**0.963**	0.104

scenes in the TI-NSD dataset, we set $N = 500$. For the ThermoScenes dataset, we set $N = 500$ for "Building (Spring)", "Building (Winter)", and "Trees", $N = 1000$ for "Heated Water Kettle" and "Freezing Ice Cup", $N = 2000$ for "Melting Ice Cup" and "Raspberry Pi", and $N = 3000$ for "Heated Water Cup", "Double Robot", and "Exhibition Building". All other hyperparameters are kept the same as those in the baseline method Veta-GS.

C.2 Comparative Experiments

Table A1 presents the performance metrics of all baseline methods and the proposed FGS method across three scene categories of TI-NSD dataset [2], *i.e.*, outdoor scenes (including 7 scenes), indoor scenes (including 7 scenes), and UAV-captured scenes (including 6 scenes). In addition, Table A2

Table A2: Quantitative comparison between the baseline methods and our FGS method across the ten scenes on ThermoScenes dataset. Best results for each metric are highlighted in **bold**. "↑" indicates that higher values correspond to better performance, while "↓" indicates that lower values correspond to better performance.

Metric	Method	Heated Water Cup	Heated Water Kettle	Freezing Ice Cup	Melting Ice Cup	Building (Spring)	Building (Winter)	Double Robot	Raspberry Pi	Exhibition Building	Trees
PSNR ↑	InstantNGP-Base [13]	16.540	29.890	15.240	31.700	27.290	27.350	28.220	36.150	32.270	31.490
	InstantNGP-Big [13]	16.828	30.410	15.505	32.304	27.765	27.826	28.711	36.779	32.832	32.038
	Plenoxel [4]	22.320	33.450	27.720	19.610	21.470	27.290	20.160	24.260	32.510	21.930
	3DGS [8]	33.122	34.979	29.885	29.254	26.179	34.430	27.679	28.664	30.722	34.453
	PixelGS [32]	31.778	34.404	29.554	31.711	27.754	34.202	31.328	35.199	33.690	34.450
	AD-GS [16]	33.640	35.489	31.995	33.192	28.966	32.597	30.782	37.214	36.186	34.214
	NTR-Gaussian [27]	22.759	22.245	18.237	16.599	26.562	**35.785**	17.730	24.748	**36.965**	**34.768**
	Thermal3D-GS [2]	33.020	35.762	26.533	31.301	27.862	33.896	35.028	28.664	33.219	32.824
	Veta-GS [14]	37.050	37.987	34.161	33.351	28.759	31.812	35.815	36.884	35.839	33.450
	FGS (Ours)	**38.918**	**38.219**	**37.087**	**34.073**	**29.829**	34.225	**36.905**	**39.134**	36.834	33.853
SSIM ↑	InstantNGP-Base [13]	0.743	0.941	0.901	0.983	0.933	0.875	0.925	0.968	0.957	0.957
	InstantNGP-Big [13]	0.745	0.941	0.906	0.984	0.931	0.880	0.923	0.971	0.959	0.958
	Plenoxel [4]	0.807	0.958	0.981	0.935	0.904	0.865	0.842	0.914	0.959	0.898
	3DGS [8]	0.954	0.968	0.990	0.982	0.966	0.978	0.937	0.952	0.952	0.974
	PixelGS [32]	0.946	0.968	0.989	0.984	0.974	0.980	0.971	0.983	0.977	0.974
	AD-GS [16]	0.966	**0.976**	0.993	0.988	**0.980**	0.978	0.930	0.987	0.984	0.974
	NTR-Gaussian [27]	0.947	0.298	0.962	0.958	0.977	**0.986**	0.388	0.867	**0.987**	**0.976**
	Thermal3D-GS [2]	0.948	0.964	0.992	0.985	0.973	0.978	0.976	0.952	0.972	0.971
	Veta-GS [14]	**0.973**	0.973	0.993	0.989	0.972	0.974	0.976	0.987	0.984	0.971
	FGS (Ours)	**0.973**	**0.976**	**0.996**	**0.990**	0.979	0.978	**0.978**	**0.988**	0.985	0.972
LPIPS ↓	InstantNGP-Base [13]	0.265	0.096	0.153	0.034	0.175	0.332	0.147	0.068	0.120	0.084
	InstantNGP-Big [13]	0.263	0.091	0.152	0.034	0.178	0.328	0.145	0.063	0.120	0.085
	Plenoxel [4]	0.157	0.096	0.037	0.160	0.213	0.278	0.271	0.145	0.088	0.199
	3DGS [8]	0.022	0.030	0.019	0.048	0.140	0.123	0.065	0.058	0.133	0.096
	PixelGS [32]	0.024	0.038	0.021	0.042	0.123	**0.108**	0.055	0.034	0.101	0.091
	AD-GS [16]	0.019	0.026	0.015	0.037	0.113	0.146	0.060	0.030	0.073	0.102
	NTR-Gaussian [27]	0.037	0.057	0.084	0.095	0.145	0.119	0.084	0.052	0.071	0.106
	Thermal3D-GS [2]	0.020	0.019	0.024	0.042	0.120	0.122	0.038	0.058	0.112	0.112
	Veta-GS [14]	**0.012**	**0.013**	0.013	0.027	0.120	0.153	0.037	0.027	0.076	**0.088**
	FGS (Ours)	0.013	**0.013**	**0.010**	**0.025**	**0.112**	0.128	**0.030**	**0.024**	**0.066**	0.090

reports the quantitative metrics for all the baseline methods and FGS across the ten scenes of ThermoScenes [7] dataset.

As shown in Table A1, our FGS method demonstrates promising results, which achieves the best performance on outdoor and indoor scenes. These results validate the robustness of the proposed framework in handling diverse and complex environments. On UAV-captured scenes, FGS achieves performance comparable to Veta-GS, with only marginal differences across metrics. This is because UAV imagery typically captures large-scale temperature variations, which can provide more detailed structures and fewer floating Gaussians. Since our method primarily targets the local blur caused by opaque floating Gaussians, the benefit is less prominent on UAV-captured scenes.

As shown in Table A2, FGS achieves strong overall performance on ThermoScenes and obtains competitive or superior results on most scenes, which demonstrates its power in complex thermal environments. Notably, FGS achieves significant gains in PSNR for several scenes, such as the "Heated Water Cup" (38.918 dB in PSNR) and "Raspberry Pi" (39.134 dB in PSNR), where the baseline methods exhibit relatively lower reconstruction quality. Overall, these results validate the effectiveness of the proposed approach in preserving local structures and reducing blur.

Figure A1 and Figure A2 present the visual comparison on TI-NSD and ThermoScenes datasets, respectively. As is exhibited, the baseline methods produce noticeable blurry edges, floating artifacts, or structural distortions in the enlarged areas. Differently, FGS preserves sharper boundaries and clearer local details. These visual results further demonstrate the effectiveness of FGS in suppressing floating Gaussians and alleviating local blur.

C.3 Validation of Opacity-Increase Condition

To validate the analysis in Section 4 and demonstrate the effectiveness of FGS, we visualize the opacity evolution of Gaussians in Figure A3. In the upper row of each method, white pixels indicate regions where the rendered intensity is lower than the ground truth, *i.e.*, $\widehat{\mathbf{C}}^u - \mathbf{C}^u < 0$, while black pixels indicate $\widehat{\mathbf{C}}^u - \mathbf{C}^u \geq 0$. In the lower row, red points denote floating Gaussians, white points denote valid Gaussians, and brighter points correspond to higher opacity. For vanilla-3DGS, the blue-boxed region contains many white pixels during training, which indicates that the rendered intensity remains lower than the ground truth in this region, thereby encouraging the opacity of

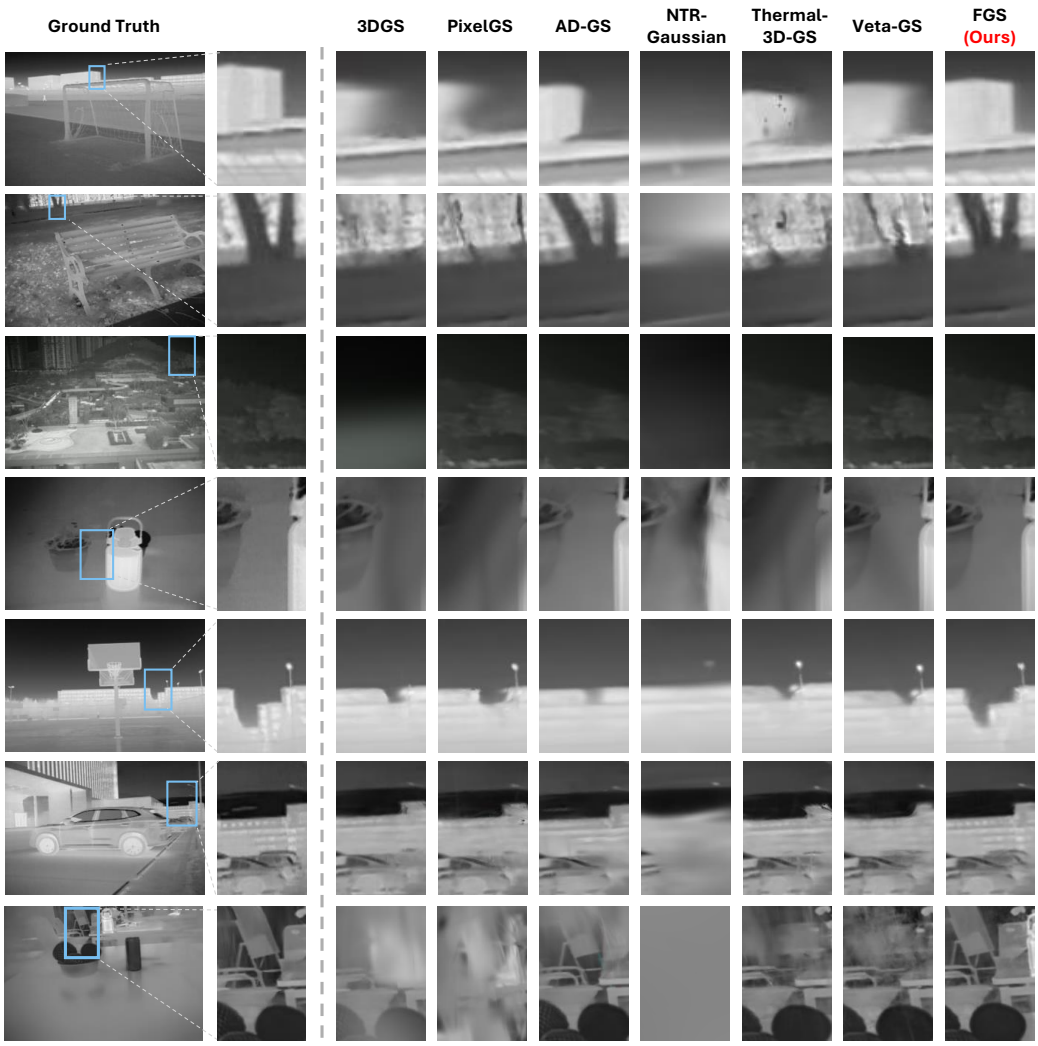

Figure A1: Visual comparison of the baseline methods and FGS on TI-NSD dataset. The blue boxes highlight local regions enlarged for detailed comparison.

floating Gaussians to increase. Hence, the corresponding Gaussian maps show that these floaters gradually become more opaque, which eventually causes local blur in the rendered image. Differently, FGS deliberately creates over-bright rendering during the early stage, so more pixels in the blue-boxed region satisfy $\hat{\mathbf{C}}^u - \mathbf{C}^u \geq 0$. This changes the opacity update direction of the associated floating Gaussians and reduces their opacity. As a result, the final rendered image preserves clearer local textures and avoids the local blur observed in vanilla-3DGS.

These results support the analysis in Section 4, which shows that floating Gaussians tend to increase their opacity when the background contribution is weak and the rendered intensity is lower than the ground truth. They also show that increasing the rendered intensity during early training can suppress excessive floater opacity by changing the sign of the opacity-related gradient.

C.4 Validation of Floating Gaussian Suppression

To further evaluate the efficacy of the proposed FGS method, we visualize the opacity evolution of floating and valid Gaussians during training. As shown in Figure A4, the blue curve represents the opacity of floating Gaussians, the red curve represents the opacity of valid Gaussians, and the black triangle denotes the switch point from decoupled optimization to joint optimization.

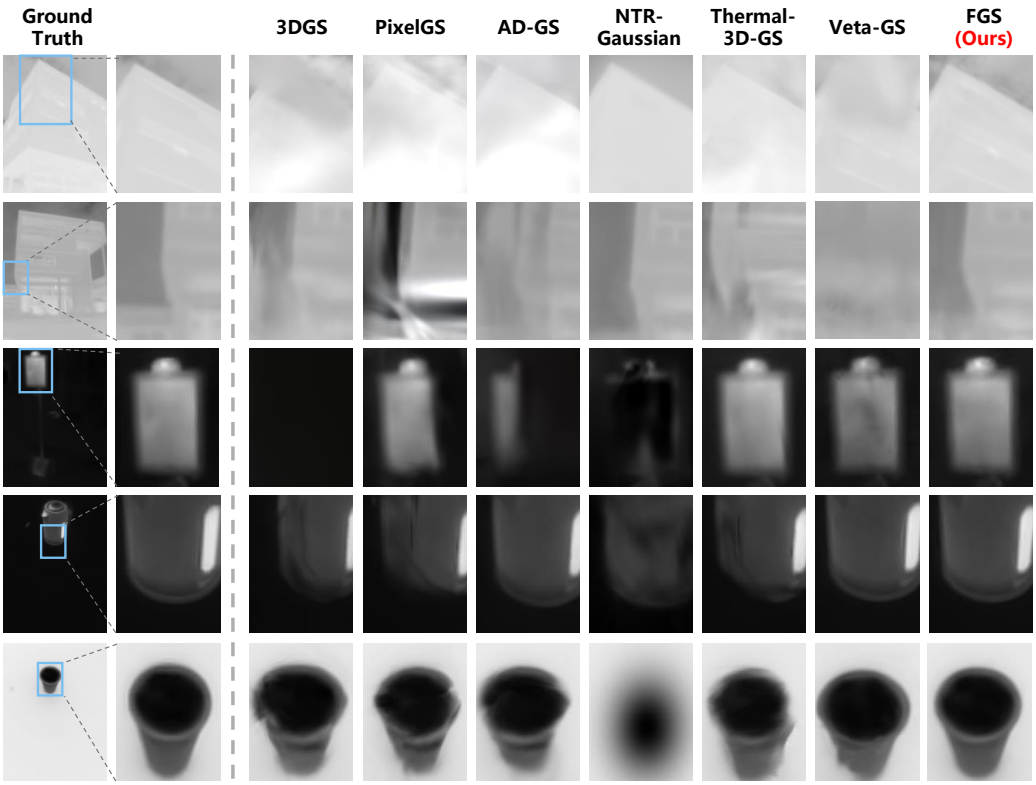

Figure A2: Visual comparison of the baseline methods and FGS on ThermoScenes dataset. The blue boxes highlight local regions enlarged for detailed comparison.

Before the switch point, the opacity of floating Gaussians decreases sharply. Although valid Gaussians also show a slight opacity decrease, their reduction is much smaller than that of floating Gaussians. This discrepancy arises because valid Gaussians are partially occluded by floating Gaussians, thereby resulting in low transmittance. As discussed in Section 5.1, low transmittance yields smaller opacity gradients, which limits the opacity decrease of valid Gaussians. After the switch point, the opacity of both floating and valid Gaussians begins to increase. However, valid Gaussians recover their opacity much more strongly than floating Gaussians. This phenomenon is attributed to two factors. First, with the opacity of floating Gaussians already reduced, the previously occluded valid Gaussians achieve higher transmittance, which in turn amplifies their opacity growth rate. Second, the increasing contribution of valid Gaussians raises the background rendered intensity for the floaters. As analyzed in Section 5.3, the increased background intensity effectively suppresses the growth of floating Gaussians during the joint optimization.

During decoupled optimization, floating Gaussians undergo a stronger opacity decrease than valid Gaussians. During joint optimization, valid Gaussians recover their opacity more effectively, while the opacity growth of floaters remains suppressed. This allows FGS to suppress over-opaque floaters while preserving valid Gaussians for high-quality reconstruction.

C.5 Validation of Thermal Sensitivity to Local Blur

To further investigate why opacity-induced local blur is more pronounced in thermal scenes, we compare Gaussian evolution under visible-light and thermal observations of the same scene. As shown in Figure A5(a), vanilla-3DGS produces few visible floating Gaussians in the visible-light scene, and the final rendered image preserves clear local structures. In contrast, as shown in Figure A5(b), floating Gaussians in the thermal scene gradually become opaque during training, especially in regions where background Gaussians are sparse. The rendered image consequently exhibits noticeable edge blur in the highlighted region.

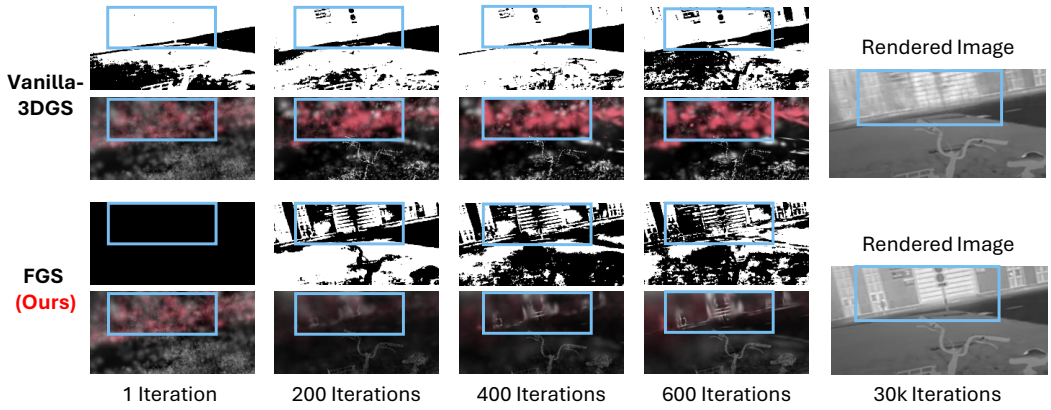

Figure A3: Visualization of the opacity-increase condition during training for vanilla-3DGS and FGS. For each method, the upper row visualizes whether the rendered intensity is lower than the ground-truth intensity. White pixels indicate $\widehat{\mathbf{C}}^u - \mathbf{C}^u < 0$, while black pixels indicate $\widehat{\mathbf{C}}^u - \mathbf{C}^u \geq 0$. The lower sub-row shows the corresponding Gaussian map, where red points denote floating Gaussians and white points denote valid Gaussians. Here, brighter points indicate higher Gaussian opacity and the blue boxes highlight the local regions used for comparison.

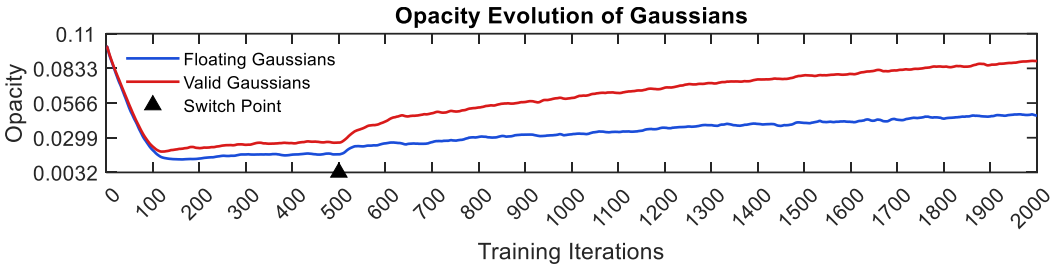

Figure A4: Opacity evolution of floating and valid Gaussians during training. The blue curve shows the opacity of floating Gaussians, while the red curve represents the opacity of valid Gaussians. The black triangle marks the switch point from decoupled optimization to joint optimization.

This difference is consistent with the analysis in Section 4. When the background contribution is weak, the opacity of a floating Gaussian in visible-light scenes can be excessively amplified only if its color is sufficiently consistent with the ground-truth appearance across multiple channels and multiple covered pixels. Due to rich chromatic variations in RGB images, this condition is difficult to satisfy for floating Gaussians that are not aligned with actual surfaces. However, in thermal scenes, the single-channel intensity constraint makes this condition easier to satisfy. When the rendered intensity at pixels covered by a floater is lower than the ground-truth intensity, the floater tends to receive opacity-increasing updates. Therefore, thermal scenes are more susceptible to opacity-induced local blur.

As shown in Figure A5(c), FGS suppresses the opacity growth of floating Gaussians in the thermal scene. Compared with vanilla-3DGS, FGS produces fewer over-opaque floaters and preserves clearer local structures in the final rendered image. These results demonstrate the strong tendency of thermal scenes to suffer from opacity-induced local blur and validate the effectiveness of FGS in alleviating this problem.

C.6 Parametric Sensitivity

The switch point N is the key hyperparameter of FGS, which determines the duration of the decoupled optimization stage. To evaluate the influence of the switch point, we vary N from 500 to

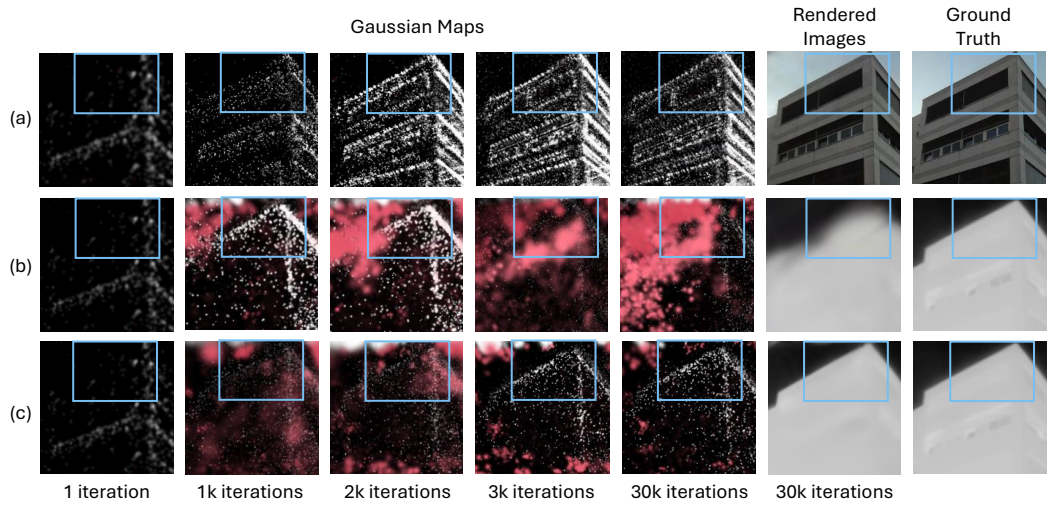

Figure A5: Gaussian evolution in visible-light and thermal scenes. (a) Gaussian evolution of vanilla-3DGS in the visible-light scene. (b) Gaussian evolution of vanilla-3DGS in the thermal scene. (c) Gaussian evolution of FGS in the thermal scene. In the Gaussian maps, red points represent floating Gaussians, while valid Gaussians are displayed in white. Brighter points indicate Gaussians with higher opacity. The blue boxes highlight the local regions enlarged for detailed comparison.

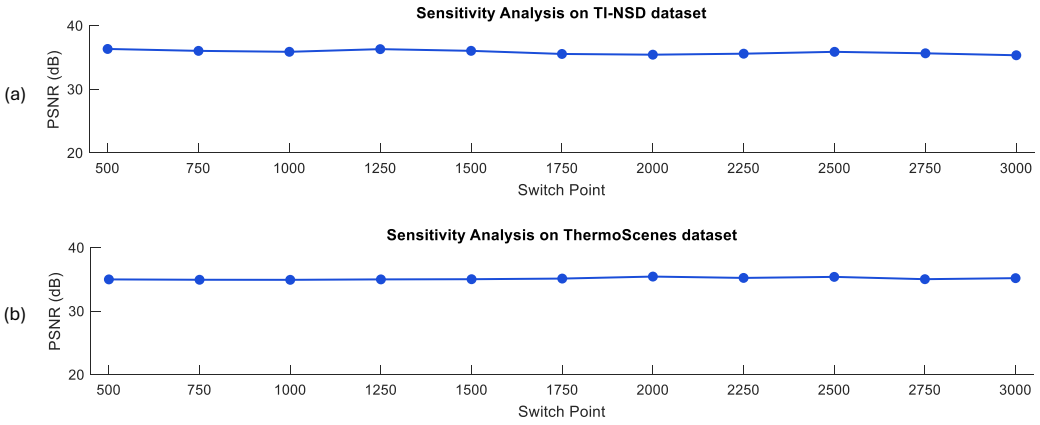

Figure A6: Sensitivity analysis of the switch point N. (a) Results on the TI-NSD dataset. (b) Results on the ThermoScenes dataset.

3000 on both the TI-NSD and ThermoScenes datasets. As shown in Figure A6, the PSNR curves show only minor fluctuations across different switch points. This indicates that FGS is relatively insensitive to the choice of N within the range, thereby requiring little hyperparameter tuning in practice.

D SfM Initialization in Thermal Scenes and Its Impact on Floating Gaussians

Following the standard 3DGS pipeline, we use COLMAP to estimate camera poses and reconstruct an initial sparse point cloud. However, the low contrast and limited texture of thermal images make reliable feature detection and matching more challenging than in visible-light images. Consequently, the initial point cloud may sparsely cover actual object surfaces and may contain occasional mislocalized 3D points. This can leave surface geometry insufficiently initialized and may introduce or exacerbate floating Gaussians during subsequent optimization.

Instead of refining the mislocalized SfM points, our analysis indicates that high-quality reconstruction can still be achieved despite imperfect SfM initialization when the excessive opacity growth of floating Gaussians is suppressed. FGS addresses this downstream effect. Its decoupled optimization suppresses floater opacity during early training, thereby mitigating the impact of imperfect initialization on rendered image quality.

E Limitations

FGS is designed for thermal NVS and relies on the single-channel intensity property of thermal images. Therefore, directly applying the same fixed-luminance strategy to visible-light scenes may not be suitable, since RGB images impose richer color constraints and fixing Gaussian colors may introduce large appearance errors. Converting the RGB colors to YUV or YCbCr and fixing the Y-channel luminance may appear to mimic the fixed-luminance strategy in thermal scenes. However, gradient contributions from the two unfixed chrominance channels can still drive opacity updates away from the intended suppression direction. Thus, adjusting luminance alone does not guarantee the same opacity suppression as in single-channel thermal images.

In addition, although FGS is relatively stable under different switch points in our experiments, an adaptive strategy for choosing the switch point could further improve its robustness across different scenes. Like other optimization-based 3DGS methods, the effectiveness of FGS also depends on geometry initialization, camera pose accuracy, and the quality of thermal observations. These issues will be explored in future work.

F Broader Impacts

The proposed FGS method for thermal NVS may benefit practical applications, such as night-time monitoring, industrial equipment inspection, and low-light autonomous navigation, where clearer thermal reconstructions can improve scene understanding, support predictive maintenance, and enhance safety. The method focuses on improving reconstruction quality from thermal observations and does not introduce new data acquisition capabilities or additional sensing modalities. Therefore, its societal impact mainly depends on the application context in which thermal NVS systems are deployed. To the best of our knowledge, the proposed method has no significant negative societal impacts.

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes], [No], or [N/A].
- [N/A] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for [N/A]).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will also be asked to include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While [Yes] is generally preferable to [No], it is perfectly acceptable to answer [No] provided a proper justification is given (e.g., error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering [No] or [N/A] is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS Paper Checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We clearly explain the contribution of our paper in the abstract and introduction.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Please refer to the limitations section of the appendix.

Guidelines:

- The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: We provide the full set of assumptions and a complete proof for our theoretical results.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We provide all the information (*e.g.*, hyperparameters) in the appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.

- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: The datasets and baseline methods in this paper are publicly available. Our source code and the will be openly accessible after the paper is accepted.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).

- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: We specify all details in the experiment section and appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: We follow the experimental settings of previous works to conduct experiments. Therefore, the comparison with previous works is fair.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
   - If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We provide the computation resources in the experiment section.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.

- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We have carefully reviewed the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We discuss the positive and negative impact of our method in the broader impacts section of the appendix.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: Our work poses no such risks for misuse of released data or models.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All datasets are public datasets.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: Our paper does not release new assets.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [N/A]

Justification: We does not involve crowdsourcing.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A]

Justification: We does not involve crowdsourcing.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: We use LLMs only for writing.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.