Proxy-Free Gaussian Splats Deformation With Splat-Based Surface Estimation

Jaeyeong Kim Seungwoo Yoo Minhyuk Sung
KAIST
jy9394@kaist.ac.kr dreamy1534@kaist.ac.kr mhsung@kaist.ac.kr

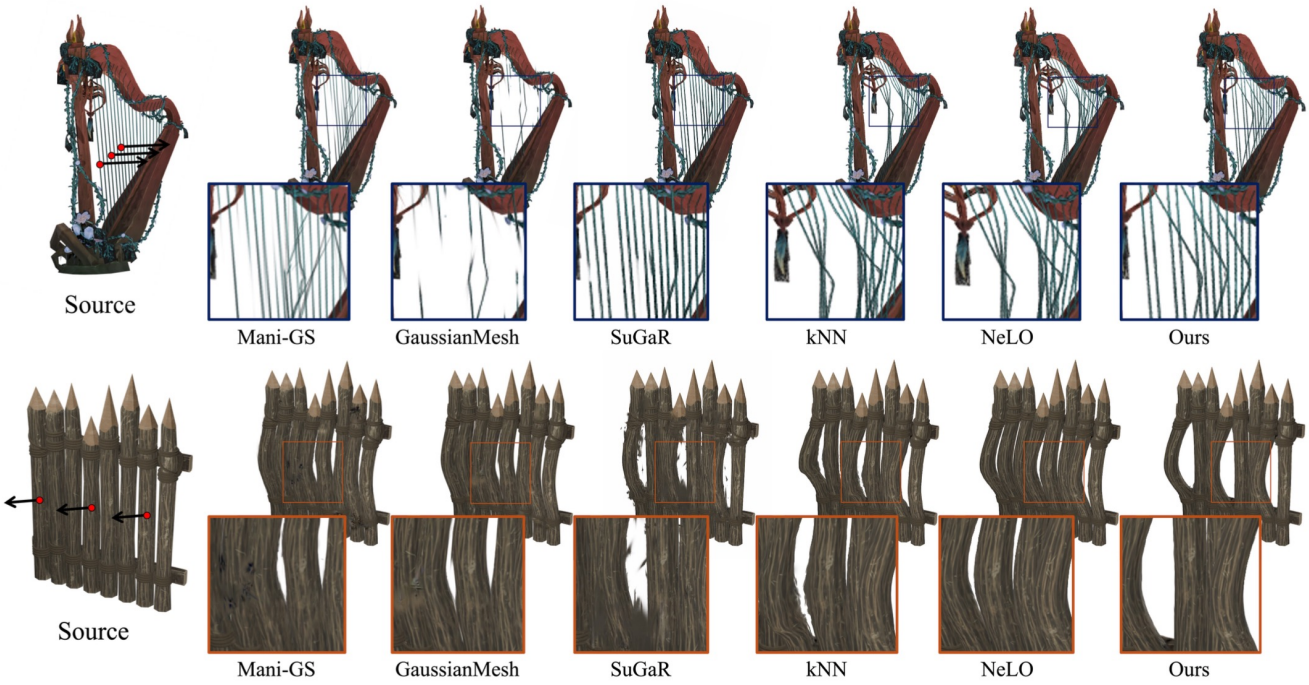

Figure 1. SpLap is proxy-free Laplacian-based deformation framework for Gaussian splats. Our method enables high-quality, large-scale deformation, all without relying on external proxy. We show results for both As-Rigid-As-Possible [1] (top row) and Bounded Biharmonic Weights [2] (bottom row) deformations. The red dots indicate the interaction handles and arrows show the direction of the applied edit.

Abstract

We introduce SpLap, a proxy-free deformation method for Gaussian splats (GS) based on a Laplacian operator computed from our novel surface-aware splat graph. Existing approaches to GS deformation typically rely on deformation proxies such as cages or meshes, but they suffer from dependency on proxy quality and additional computational overhead. An alternative is to directly apply Laplacian-based deformation techniques by treating splats as point clouds. However, this often fail to properly capture surface information due to lack of explicit structure. To address this, we propose a novel method that constructs a surface-aware splat graph, enabling the Laplacian operator derived from it to support more plausible deformations that preserve details and topology. Our key idea is to leverage the spatial arrangement encoded in splats, defining neighboring splats not merely by the distance between their centers, but by their intersections. Furthermore, we introduce a Gaussian kernel adaptation technique that preserves surface structure under deformation, thereby improving rendering quality after deformation. In our experiments, we demonstrate the superior performance of our method compared to both proxy-based and proxy-free baselines, evaluated on 50 challenging objects from the ShapeNet, Objaverse, and Sketchfab datasets, as well as the NeRF-Synthetic dataset. Code is available at https://github.com/kjae0/SpLap.

1. Introduction

Deforming 3D shapes has long been a central problem in geometry processing. For decades, extensive research has provided well-established deformation techniques for 3D representations, most notably polygonal meshes [1–3]. The goal of these techniques is to allow user-driven manipulation while preserving the intrinsic geometry of the shape. As a representative approach, Laplacian-based methods achieve this by minimizing a deformation energy over the geometry to penalize undesirable distortions.

Recently, 3D Gaussian Splatting (3DGS) [4] has shown photorealistic novel view synthesis (NVS) quality at real-time frame rates. Although the explicit point-based nature of 3DGS primitives suggests a potential for direct manipulation, it often leads to suboptimal results due to the lack of connectivity and structural prior. While previous efforts have proposed deformation methods that learn a motion prior from the dynamic scene [5–8], these are not applicable to a general task of editing static objects. Consequently, the prevalent workaround is to introduce a deformable geometric proxy, such as a mesh [9–11] or cage [12–15]. These methods parametrize Gaussian kernels with respect to a proxy such that manipulations of the proxy deform the associated Gaussian kernels.

However, this reliance on a proxy introduces two critical limitations: the final deformation quality is fundamentally limited by the quality of the proxy. In the case of meshes, which are the most widely used as a proxy, geometric defects—such as topological errors or missing parts—propagate directly into the deformation, causing severe visual artifacts and narrowing the applicability of existing deformation techniques. Moreover, this strategy incurs an additional computational cost, which acts as a practical barrier. This approach necessitates a lengthy pipeline of mesh generation, where high-quality surface reconstruction such as NeuS [16] or Neuralangelo [17] takes at least several GPU hours, often followed by laborious post-processing to ensure the mesh is suitable for deformation.

In this paper, to remedy these limitations derived from the dependency on proxy, we propose SpLap, a proxy-free Laplacian-based deformation framework for Gaussian splats. Rather than relying on an external structural prior, our approach leverages an inherent surface geometry encoded within the spatial arrangement of the surface-aligned Gaussian splats [18–21]. Our key observation is that a surface-aligned GS can be interpreted as a discrete, patch-based surface representation, since each primitive is optimized to act as a local chart of the manifold. Motivated by this observation, we leverage this inherent surface alignment of primitives to extract geometric information that is crucial to utilize traditional deformation techniques [1, 2].

Our framework begins with establishing the connectivity from the Gaussian kernels. However, since simple spatial proximity often fails to capture a valid adjacency on the surface manifold, we incorporate intrinsic surface information by leveraging surface-aligned GS. Specifically, we construct a splat intersection graph whose connectivity is defined by the overlap of Gaussian kernels. This intersection-based connectivity is the key component that allows us to estimate the underlying surface without an external geometric proxy. Subsequently, we directly build Laplacian operator and apply Laplacian-based geometric deformation techniques to Gaussian kernels, treating their means as a point cloud. By building a Laplacian operator upon our surface-aware splat intersection graph, we overcome the limitations of existing point cloud Laplacian methods and enable topologically sound deformations even on complex geometries such as adjacent distinct surfaces or thin regions. Finally, we introduce an adaptation method that is crucial for preserving visual fidelity. As the displaced Gaussian kernels no longer reflect the deformed local surface geometry, we adapt each primitive to maintain the original coverage on the surface manifold. Consequently, our framework enables direct geometric deformation for Gaussian splats while preserving photorealistic quality even under a large-scale deformation.

In experiments, we demonstrate the robustness and efficiency of our approach through experiments on a diverse set of objects with intricate geometries. Applying two representative Laplacian-based deformation techniques, As-Rigid-As-Possible (ARAP) [1] and Bounded Biharmonic Weights (BBW) [2], we show that our method achieves a deformation fidelity nearly indistinguishable from the ground truth result, whereas existing methods fail to produce a plausible quality.

2. Related Works

2.1. Gaussian Splats Deformation

Diverse strategies for Gaussian splats deformation have been explored including editable dynamic scenes [5–8], simulation-based methods [12, 22]. However, these methods are not applicable to a general task of direct, user-guided editing for static objects. Consequently, methods that leverage geometric templates such as a mesh or cage as a deformable proxy, have been proposed [9–11, 13, 14, 21, 23]. Although GSDeformer [14] and Cage-GS [15] propose a cage-based deformation approach, these lack fine-grained control. For this reason, mesh-based approaches have become an attractive alternative, providing detailed control over the deformation. Specifically, SuGaR [21] and GaussianFrosting [23] extract a mesh directly from Gaussian primitives via Poisson surface reconstruction [24] and then parametrize the primitives to the mesh. Otherwise, Mani-GS [11] and GaussianMesh [10] generate templates through a separate reconstruction stage using NeuS [16] and

NeuS2 [25] and bind the Gaussian kernels to the mesh faces. However, the effectiveness of these mesh-based methods is fundamentally limited by the quality of the template mesh. Although Mani-GS [11] introduces learnable offsets to compensate for a defective template, this cannot resolve the inaccurate deformations that arise from a flawed geometric proxy. In contrast to these previous works, our approach does not rely on any explicit geometric proxy while enabling direct and point-level deformation.

2.2. Gaussian Splatting with Surface Geometry

Recently, there have been efforts to integrate a surface geometry to Gaussian Splatting [4], while keeping photorealistic rendering [26–29]. For instance, several works directly regularize the Gaussian primitives to be flattened and aligned with the underlying surface [21, 23]. Inspired by earlier surfel-based representations [30–32], 2DGS [18] and G-Surfel [20] even replace 3D primitives with 2D Gaussian kernels with additional geometric regularizations, such as depth distortion loss [33] and normal consistency loss. Built upon 2DGS, SpikingGS [19] utilizes spiking neurons with an adaptive opacity threshold, achieving superior surface alignment capability. These works reveal the potential of Gaussian splats can serve as a patch-based discrete surface representation. In this work, we employ this insight for a geometric deformation of Gaussian splats without relying on any external geometric proxy.

2.3. Laplacian Operator on Point-based Representations

The discrete Laplacian operator is a fundamental tool for geometry processing [34, 35]. Without explicit connectivity, existing methods typically rely on a k-nearest-neighbor (kNN) search for local neighborhood estimation. Based on the kNN graph, various approaches have been proposed including kernel-based methods on triangulation [36–39], mesh Laplacian schemes [40–42], smoothed particle hydrodynamics (SPH) [43] and moving-least-squares (MLS) [44]. More recently, NeLO [45] introduced a learning-based approach, using a graph neural network on the kNN graph. Despite these advances, existing methods inherit the surface-agnostic nature of the initial kNN graph and therefore fail to identify the local surface manifold on intricate geometry. For Gaussian Splatting, LBO-GS [46] utilizes the Mahalanobis distance to employ additional geometric information of Gaussian kernels. However, this approach is critically sensitive to highly anisotropic primitives and is insufficient for robust surface estimation. In contrast, our work addresses this by introducing an intersection-based surface estimation, thereby enabling robust and plausible Laplacian-based deformation.

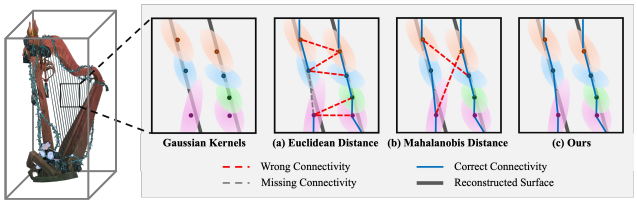

Figure 2. Illustration of different connectivity search methods. Existing metrics, (a) and (b), often produce incorrect connections (red) or fragmentation (gray) due to the lack of structural prior. In contrast, by leveraging geometric constraints of surface-aligned GS, our method avoids this suboptimal situations like (c).

3. Preliminary

Surface-Aligned Gaussian Splatting. 3D Gaussian Splatting [4] represents a 3D scene with a set of 3D Gaussian kernels. Each Gaussian kernel $\mathcal{G}$ is parametrized as:

$$\mathcal{G}(\mathbf{x}) = e^{-\frac{1}{2}(\mathbf{x}-\mathbf{p})^\top \Sigma^{-1}(\mathbf{x}-\mathbf{p})} \tag{1}$$

where $\mathbf{p} \in \mathbb{R}^3$ and $\Sigma \in \mathbb{R}^{3\times3}$ represent position and covariance matrix. The covariance matrix $\Sigma = \mathbf{RSS}^\top\mathbf{R}^\top$ is factorized into a rotation matrix $\mathbf{R} \in \mathbb{R}^{3\times3}$ and diagonal scale matrix $\mathbf{S} \in \mathbb{R}^{3\times3}$. For 2D primitive-based methods [18–20], kernels are simplified into flat 2D Gaussian kernels by fixing the last diagonal element of $\mathbf{S}$ to zero. Typically, these primitives are optimized with photometric losses alongside geometric objectives, such as depth distortion loss [33] and normal consistency loss. These geometric constraints encourage surface alignment by concentrating the weight distribution of kernels on the underlying surface.

4. Method

4.1. Method Overview

The explicit point-based nature of Gaussian Splatting makes the Gaussian kernels amenable to direct manipulation. By treating them as a point cloud, the Laplacian operator can be directly defined over the splats using point cloud Laplacian method such as NonManifold [42]. Although this approach enables Laplacian-based deformation on Gaussian splats, the lack of geometric prior often leads to suboptimal results due to the wrong surface estimation. Since point cloud Laplacian rely on the connectivity of graph over the nodes, the fidelity of its application is determined by the quality of neighbor estimation. To handle this, in this work, we present **SpLap**, a proxy-free deformation framework for Gaussian splats. Specifically, our approach leverages surface-aligned GS to extract geometric information directly from the spatial arrangement of the splats themselves. This splat-based surface estimation obviates the need for a geometric proxy, thereby avoiding the limitations of proxy-based methods. Specifically, our pipeline is

divided into 3 stages:

1. **Surface-Aware Splat Graph Construction (Sec. 4.2):** We first construct a splat graph, where the connectivity is defined by the overlap of splats. Based on this graph, we then estimate the surface geometry through a geodesic distance.
2. **Laplacian-Based Deformation (Sec. 4.3):** We deform the shape represented by the Gaussian splats via Laplacian-based techniques, treating the means of Gaussian kernels as a point cloud. By combining a point cloud Laplacian with our neighborhood estimation, we achieve plausible deformations in a proxy-free manner.
3. **Surface-Preserving Kernel Adaptation (Sec. 4.4):** To preserve visual fidelity during deformation, we introduce a novel Gaussian kernel adaptation method. In this stage, we adapt each kernel to reflect a local geometric change, preserving the underlying surface manifold.

4.2. Surface-Aware Splat Graph Construction

A key challenge of local neighborhood estimation is filtering out spurious neighbors: primitives that are close in Euclidean space but distant along the underlying surface. As illustrated in Fig. 2, existing point-based distance metrics, such as Euclidean and Mahalanobis distance, cannot be reliably applied in challenging conditions. Instead, to establish connectivity exclusively along the surface, we build a graph by defining connectivity as an intersection of Gaussian kernels. The core intuition is that, *only Gaussian kernels that are proximate along the surface manifold will intersect*. First, given a set of 2D images, we reconstruct the scene with a surface-aligned Gaussian Splatting. Specifically, we utilize SpikingGS [19] for the superior surface alignment quality of 2D Gaussian primitives. Since our approach is built upon the surface alignment property, our method can be extended to various surface-aligned GS methods, as demonstrated in Fig. 7.

The reconstructed scene is represented by a set of N Gaussian kernels $\{\mathcal{G}_i\}_{i=1}^N$. Each kernel $\mathcal{G}_i$ is parametrized by position $\mathbf{p}_i$, covariance Σ_i, and opacity α_i. The covariance matrix Σ_i is factorized into a rotation matrix $\mathbf{R}_i$ and a scale matrix $\mathbf{S}_i$, such that $\Sigma_i = \mathbf{R}_i\mathbf{S}_i\mathbf{S}_i^\top\mathbf{R}_i^\top$. To identify the intersections between these kernels, we define an occupancy region Ω_i for each Gaussian kernel $\mathcal{G}_i$ as the set of points in $\mathbb{R}^3$ where the kernel is considered active. The activity of a kernel at point $\mathbf{x}$ is determined by two constraints. The first is the activation cut-off threshold $\bar{V}_i^p$ from a spiking neuron of SpikingGS [19], a learnable parameter that suppresses low-opacity parts as:

$$\hat{\mathcal{G}}_i(\mathbf{x}) = \begin{cases} 0, & \mathcal{G}_i(\mathbf{x}) < \bar{V}_i^p, \\ e^{-\frac{1}{2}(\mathbf{x}-\mathbf{p}_i)^\top\Sigma_i^{-1}(\mathbf{x}-\mathbf{p}_i)}, & \text{otherwise.} \end{cases} \tag{2}$$

The second constraint is that the rendered intensity, $\alpha_i\mathcal{G}_i(\mathbf{x})$, must exceed the minimum contribution threshold c, typically $1/255$. Combining these, we define the occupancy region Ω_i in 3D canonical space as:

$$\Omega_i \;=\; \Big\{\, \mathbf{x}\in\mathbb{R}^3 \;\Big|\; \mathcal{G}_i(\mathbf{x}) > \max\Big\{\; \bar{V}_i^p,\; \frac{c}{\alpha_i}\Big\}\Big\}. \tag{3}$$

Since the covariance Σ_i is rank-2, occupancy region Ω_i is an ellipse. With scale matrix $\mathbf{S}_i = \mathrm{diag}\,(\sigma_1, \sigma_2, 0)$ and rotation matrix $\mathbf{R}_i = [\mathbf{r}_1\ \mathbf{r}_2\ \mathbf{r}_3]$, we can rewrite the occupancy region Ω_i as following:

$$\Omega_i = \big\{\mathbf{p}_i + \sqrt{\lambda_i}\,\rho\,\big(\sigma_1\cos\theta\,\mathbf{r}_1 \,+\, \sigma_2\sin\theta\,\mathbf{r}_2\big) \;\big|\; \rho\in[0,1)\big\} \tag{4}$$

with $\lambda_i = -2\max\big\{\ln|\bar{V}_i^p|,\ \ln(\frac{c}{\alpha_i})\big\}$ and $\theta\in(0,2\pi]$. This elliptic form provides a direct way of estimating the spatial relationships between Gaussian kernels. To determine whether two Gaussian kernels $\mathcal{G}_i$ and $\mathcal{G}_j$ intersect, we compute their normal-wise offset δ_{ij} from $\mathcal{G}_i$ to $\mathcal{G}_j$:

$$\delta_{ij} := \inf\big\{\,|t| \mid \exists\,\mathbf{x}\in\Omega_i:\ \mathbf{x} + t\,\mathbf{r}_{i,3}\in\Omega_j\big\} \tag{5}$$

with $\inf\emptyset = +\infty$. While a zero offset $\delta_{ij} = 0$ denotes a perfect intersection, we consider two Gaussians to be intersecting if their offset δ_{ij} falls below a tolerance ϵ. This tolerance is necessary to account for minor gaps between adjacent Gaussian kernels, which can arise from the joint optimization of photometric and geometric objectives.

4.3. Laplacian-Based Deformation

In this section, we detail the Laplacian-based deformation stage. To achieve proxy-free editing, we directly build Laplacian on the collection of Gaussian kernels, treating them as a point cloud. This enables the direct application of Laplacian-based deformation techniques to the reconstructed scene. Specifically, we construct the Laplacian operator on the set of Gaussian kernels' mean $\{\mathbf{p}_i\}_{i=1}^N$. In this process, we integrate our surface-aware splat graph into off-the-shelf point cloud Laplacian methods. Unlike existing kNN-based methods that only consider spatial proximity, we leverage the geodesic distance on our splat graph to ensure the Laplacian Δ reflect underlying surface faithfully. We use graph geodesic distance rather than raw graph connectivity for robustness against missing edges. For efficiency, we compute the cotangent Laplacian on a local triangulation using NonManifold [42]. With the resulting Laplacian operator, we deform the means of kernels $\{\mathbf{p}_i\}_{i=1}^N$ through Laplacian-based deformation map ϕ_Δ as:

$$\mathcal{G}_{\phi_\Delta}(\mathbf{x}) = e^{-\frac{1}{2}(\mathbf{x}-\phi_\Delta(\mathbf{p}_i))^\top\Sigma_i^{-1}(\mathbf{x}-\phi_\Delta(\mathbf{p}_i))}. \tag{6}$$

Here, ϕ_Δ represents the deformation map obtained by solving a Laplacian-based system such as ARAP [1] or BBW [2] using Laplacian operator Δ. However, simply displacing the positions of the primitives leads to severe visual artifacts, because the covariances of the Gaussian kernels are not adapted properly. We therefore present a subsequent kernel adaptation stage, as we detail in the next section.

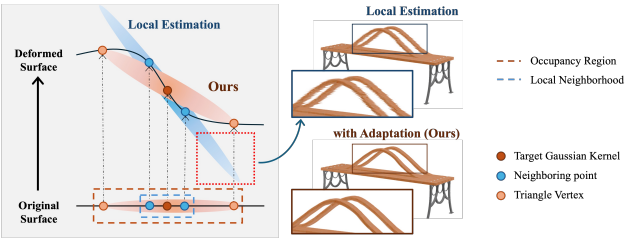

Figure 3. Schematic of kernel adaptation methods. Simply estimating local transformation fails due to a scale mismatch, leading to visual artifacts. Our surface-preserving method avoids this by directly maintaining the original surface coverage.

4.4. Surface-Preserving Kernel Adaptation

Following the deformation stage, we adapt the Gaussian kernels to reflect the local geometric change. However, we observed that a naive approach, such as estimating a local affine transformation from the deformed neighborhood, is insufficient. As illustrated in Fig. 3, when a neighborhood only partially covers a subset of a Gaussian's occupancy region Ω, the local estimation cannot represent the transformation of the entire region. This causes the Gaussian kernel to deviate from the underlying surface, leading to spike-like artifacts. Although prior works have addressed this issue by using geometric proxy such as bounding scales [10], envelope [12] or splitting [14], these are not applicable in our proxy-free scenario. To resolve this, we propose an novel adaptation method that aims to preserve the underlying surface manifold. Our key idea is based on the foundation of 2D Gaussian Splatting [18], which encourages each kernel to act as a local patch that covers the underlying surface. Instead of estimating a transformation, our goal is to directly adapt each deformed Gaussian kernel $\mathcal{G}_{\phi_\Delta,i}$ to preserve its original surface coverage on the newly deformed surface by tracking its occupancy region. However, tracking the entire boundary is computationally intractable. We therefore employ a simpler but representative proxy: the inscribed triangle of maximum area $\mathbf{T}_i = \{\mathbf{t}_i^k\}_{k=1}^3$, defined as:

$$\mathbf{t}_i^1 = \mathbf{p}_i + \mathbf{v}_{i,1}, \quad \mathbf{t}_i^{2,3} = \mathbf{p}_i - \tfrac{1}{2}\,\mathbf{v}_{i,1} \pm \tfrac{\sqrt{3}}{2}\,\mathbf{v}_{i,2} \tag{7}$$

where $\mathbf{p}_i$ is the mean of Gaussian kernel $\mathcal{G}_i$ and $\mathbf{v}_{i,1}$, $\mathbf{v}_{i,2}$ represent the major and minor axis of Ω_i in Eqn. 4. The adapted Gaussian kernel and corresponding parameters can then be uniquely recovered from triangle's vertices by computing Steiner circumellipse [47]. However, as the deformation map ϕ_Δ is defined only on the means of the Gaussian kernels $\{\mathbf{p}_i\}_{i=1}^N$, we transfer the displacement $\phi_\Delta(\mathbf{p}_i) - \mathbf{p}_i$ to each triangle vertex $\mathbf{t}_i^k$. Since the displacement on the surface is continuous, we compute the displacement of each vertex $\mathbf{t}_i^k$ using the inverse distance weighting [48] from a set of the adjacent Gaussians, indexed by $\mathcal{M}(\mathbf{t}_i^k)$. The displaced triangle vertex $\mathbf{t'}_i^k$ is then computed as follows:

$$\mathbf{t'}_i^k = \mathbf{t}_i^k + \frac{\sum_{j\in\mathcal{M}(\mathbf{t}_i^k)} \big(w_{ij}\big(\phi_\Delta(\mathbf{p}_j) - \mathbf{p}_j\big)\big)}{\sum_{j\in\mathcal{M}(\mathbf{t}_i^k)} w_{ij}} \tag{8}$$

where w_{ij} is $\|\mathbf{t}_i^k - \mathbf{p}_j\|_2^{-1}$. The adjacent primitives $\mathbf{p}_j \in \mathcal{M}(\mathbf{t}_i^k)$ are chosen based on graph distance on our splat graph to ensure adjacency of the surface. This adaptation is crucial to ensure that the deformed Gaussian kernels accurately cover their corresponding coverage on the deformed surface, preserving geometric structure and visual fidelity as shown in Fig. 6.

5. Experiments

We evaluate our framework by applying two representative Laplacian-based deformation techniques, As-Rigid-As-Possible (ARAP) [1] and Bounded Biharmonic Weights (BBW) [2], to diverse objects with intricate geometries. We provide the implementation details and further analysis in the supplementary materials.

5.1. Experiment Setup.

Dataset & Benchmark. To the best of our knowledge, no suitable dataset exists for evaluating geometric deformations of scenes reconstructed from 2D images. We therefore introduce a new benchmark, specifically curated to assess the fidelity and robustness of deformation. Our benchmark comprises 50 textured synthetic objects across 10 categories, sourced from ShapeNet [50], Objaverse [51], and Sketchfab [52] datasets. The number of objects and the breadth of categories in this benchmark are comparable to standard 3D reconstruction benchmarks [49, 53], providing sufficient coverage for evaluating deformation.

For each object, we provide a ground-truth mesh, 100 multi-view rendered images, and a set of interaction handles. To test the robustness efficiently, the handles are annotated in geometrically ambiguous regions manually. An average of 14.8 handles are provided per object, for a total of 740 deformation handles across the benchmark. In addition, we also conduct a evaluation on NeRF-Synthetic dataset [49]. The handles for NeRF-Synthetic dataset are sampled by farthest point sampling. Further details of the benchmark are available in the supplementary materials.

Baselines. We evaluate our framework against two distinct categories of approaches: proxy-based methods and proxy-free methods. For the first category, we compare with the leading methods, Mani-GS [11], GaussianMesh [10], and SuGaR [21]. For our proxy-free baselines, we compare against existing point-based Laplacian methods including standard kNN-based NonManifold [42], LBO-GS [46] and NeLO [45]. For a fair comparison, we use the optimal hyperparameter for the number of neighbors k in kNN search.

Method	Bench		Bed		Headphone		Basket		Coil		Birdcage		Windmill		Ferris Wheel		Harp		Fence		Average		NS [49]	
Proxy-Based Methods																								
Mani-GS [11]	0.961	0.990	0.808	0.927	0.806	0.850	0.703	0.898	0.605	0.816	0.733	0.850	0.822	0.891	0.842	0.920	0.785	0.817	0.934	0.948	0.800	0.891	0.801	0.825
GaussianMesh [10]	0.962	0.988	0.811	0.921	0.800	0.857	0.705	0.882	0.598	0.821	0.730	0.856	0.813	0.904	0.829	0.932	0.741	0.871	0.935	0.948	0.792	0.898	0.799	0.816
SuGaR [21]	0.862	0.650	0.870	0.869	0.898	0.710	0.935	0.878	0.975	0.831	0.954	0.789	0.903	0.873	0.871	0.692	0.807	0.642	0.858	0.640	0.893	0.758	0.882	0.863
Proxy-Free Methods																								
kNN (k=10) [42]	0.647	0.789	0.980	0.978	0.952	0.981	**0.993**	0.969	0.895	0.930	0.969	0.975	0.974	0.973	0.963	0.973	0.919	0.954	0.979	0.960	0.927	0.948	0.915	0.910
kNN† (k=30) [42]	0.599	0.753	0.946	0.943	0.951	0.978	0.982	0.960	0.779	0.808	0.964	0.955	0.955	0.977	0.951	0.953	0.718	0.813	0.915	0.905	0.876	0.904	0.913	0.908
Mahalanobis (k=10) [46]	0.640	0.836	0.981	**0.993**	0.761	0.928	0.977	0.977	0.866	0.927	0.938	0.984	0.912	0.975	0.943	0.975	0.940	0.969	0.972	0.987	0.893	0.955	0.862	0.911
Mahalanobis† (k=30) [46]	0.606	0.791	0.960	0.991	0.751	0.939	0.966	0.973	0.847	0.882	0.952	0.977	0.916	0.980	0.942	0.964	0.939	0.959	0.961	0.965	0.884	0.942	0.875	0.917
NeLO [45]	0.671	0.785	0.968	0.936	0.876	0.879	0.968	0.885	0.909	0.931	0.952	0.952	0.932	0.870	0.942	0.921	0.884	0.910	0.950	0.939	0.905	0.901	0.874	0.858
Ablation Result																								
Splat Graph	0.996	0.994	**0.993**	0.986	0.945	0.976	0.987	**0.983**	**0.993**	0.980	0.974	0.991	0.967	0.972	**0.980**	0.980	0.903	0.956	0.995	0.993	0.973	0.981	0.929	0.922
Ours (Geodesic distance)	**0.997**	**0.998**	**0.993**	**0.993**	**0.953**	**0.982**	**0.993**	0.977	0.983	**0.996**	**0.996**	**0.997**	**0.980**	**0.987**	0.965	**0.983**	**0.982**	**0.980**	**0.999**	**1.000**	**0.985**	**0.988**	**0.931**	**0.924**

Table 1. Quantitative comparison of deformation quality. We report 3DPCK on our benchmark and NeRF-synthetic [49] at a threshold of 0.075 (higher is better). The score for each category is the average performance across all handles defined for objects in that category. Gray columns correspond to ARAP [1], while white columns show BBW [2] Best scores are in bold. † indicates default parameter settings.

We set $k = 10$, which we found to be a robust value that avoids graph fragmentation while maintaining locality. Additionally, our kernel adaptation stage (Sec. 4.4) is applied to all proxy-free methods for the qualitative evaluation.

Evaluation Method. To assess deformation quality, we apply identical transformations to both the ground truth mesh and Gaussian splats. For a standardized evaluation, we assign a translation of repulsive direction for each handle using a local PCA of its neighbors, selecting the eigenvector with the smallest eigenvalue. Unfortunately, mesh processing steps such as watertighting [54] and remesh [55] required for Laplacian construction discard UV coordinates and textures, As a result, reliable ground truth renderings are no longer available, making standard image-based metrics like PSNR or LPIPS inapplicable.

For this reason, we evaluate the results from two complementary perspectives: 1) topological consistency and 2) visual fidelity. To measure topological consistency in 3D space, we utilize 3D Percentage of Correct Keypoints (3DPCK) [56], a standard metric in keypoint tracking [57–60]. Keypoints are sampled by farthest point sampling around each handle on the ground truth meshes and paired to nearest Gaussian kernels. A deformed keypoint is counted as correct if distance to the mean of paired Gaussian kernel $\mathbf{p}_i$ lies within a threshold. We employ 100 keypoints for computing 3DPCK. We select 3DPCK instead of metrics for point clouds such as Chamfer Distance for fair comparison, since 3DPCK does not penalize the transparent, but misplaced primitives common in mesh-based methods. For visual fidelity, we assess the rendered results qualitatively. We provide more details about the evaluation and deformation setting in the supplementary materials.

5.2. Quantitative Results

Tab. 1 summarizes the 3DPCK results of the baselines and our method. Our approach achieves superior performance across all categories, nearly matching the ground-truth. Proxy-based approaches [10, 11, 21] are fundamentally limited by surrogate quality; topological errors in the mesh generation lead to significant degradation in deformation. Other proxy-free baselines [42, 45, 46] also show suboptimal performance, particularly on categories with complex topology such as bench and harp, due to their inaccurate neighborhood estimation. Our method also shows superior performance on the NeRF-Synthetic dataset [49] (last column). We provide further result on different PCK thresholds in supplementary materials.

5.3. Qualitative Results

We present our qualitative results in Fig. 4. Our method consistently produces high-fidelity deformations, even on objects with intricate geometries. In contrast, proxy-based methods are commonly limited by the quality of their proxy, failing to produce plausible deformations for shapes with intricate geometries such as the harp shown in the second row of Fig. 4. Furthermore, Mani-GS [11], which lacks regularization on kernel scales, produces severe artifacts regardless of template quality. GaussianMesh [10] cannot compensate for template defects, leading to missing geometry in the reconstruction. While other proxy-free baselines circumvent this dependency, their inaccurate neighborhood estimation causes distortions in unintended regions.

5.4. Ablation Study

We perform ablation studies to validate the impact of three key components: graph construction, surface-preserving adaptation, and the choice of the underlying reconstruction method.

Graph Construction. Our splat graph-based approach consistently outperforms other point-based methods. However, as illustrated in Fig. 5, a graph defined solely by splat intersection can be sensitive to missing edges, particularly in challenging cases like thin regions. As validated in

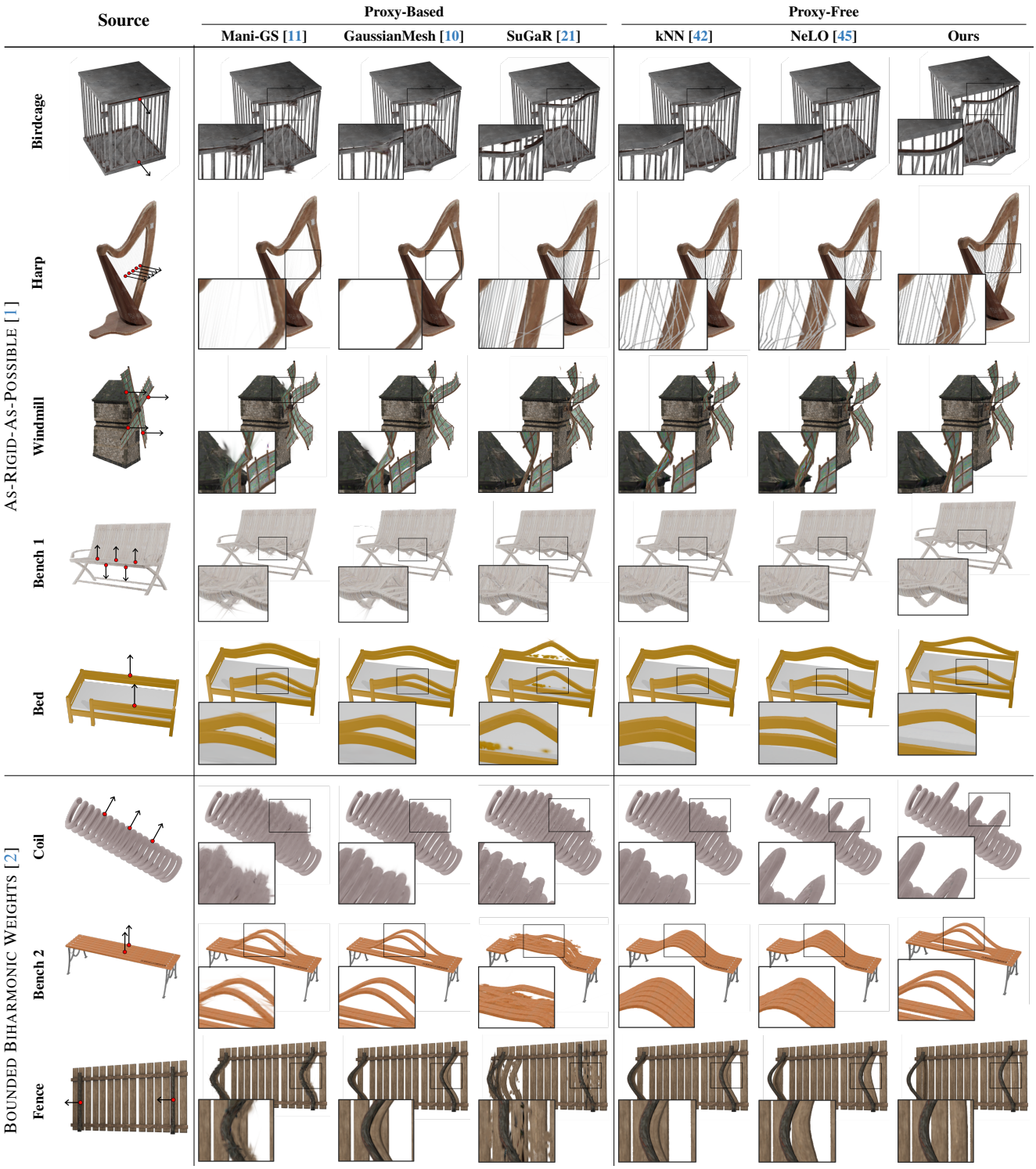

Figure 4. Qualitative comparison of deformation results. The red dots denote the interaction handles, with arrows indicating the edit direction. The top five rows show results using the ARAP [1], while the bottom three rows use BBW [2]. Our method demonstrates superior visual fidelity and geometric consistency compared to baselines, especially in preserving fine details and preventing artifacts across both deformation techniques.

Figure 5. Comparison of neighborhood estimations. Graph distance based method provides improved robustness than raw graph connectivity on challenging geometries, such as thin structures.

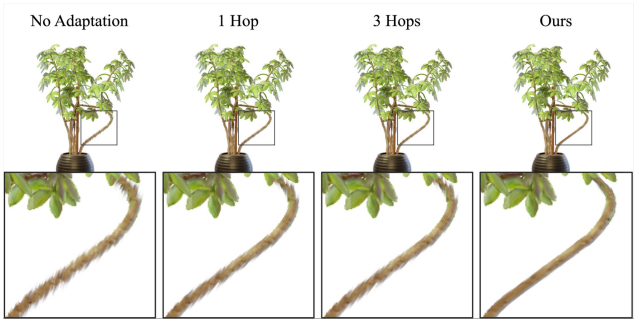

Figure 6. Comparison of kernel adaptations. N Hop(s) denotes the local transformation estimation based on neighbors within N edges. While simple local transformations fail even with wider neighborhoods (3 hops), our method effectively adapts each kernel to maintain high visual quality.

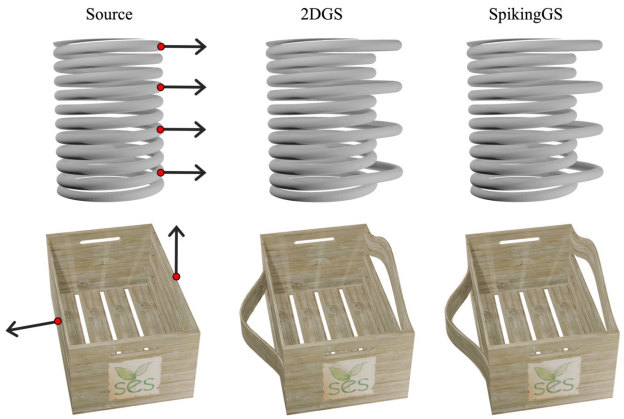

Figure 7. Application to other surface-aligned GS methods. Our framework successfully deforms scenes reconstructed with other surface-aligned GS methods.

Tab. 1, using graph distance with neighborhood estimation yields a consistent enhancement over the raw splat graph, showing its improved robustness.

Surface-Preserving Adaptation. We validate the impact of the adaptation stage introduced in Sec. 4.4. As shown in Fig. 6, noticeable spike artifacts appear since a naive local transformation estimate cannot capture the full extent of elliptical support. Simply expanding the neighborhood (*e.g.,* by increasing the number of hops) does not resolve this fundamental scale mismatch. Our method explicitly re-aligns each kernel with its deformed surface patch, a step that is crucial for preserving visual fidelity.

Generalization to Other Reconstruction Methods. To validate the robustness of our framework to the underlying reconstruction method, we apply our deformation framework to scenes generated by 2DGS [18], a representative 2D primitive-based method. For this experiment, we define the occupancy regions by omitting the spiking threshold $\bar{V}^p$ constraint, which is not present in 2DGS. The qualitative results in Fig. 7 confirm that our approach successfully achieves high-quality geometric deformations, proving its independence from a specific reconstruction method.

6. Conclusion and Limitations

In this work, we present SpLap, a proxy-free framework for the high-fidelity geometric deformation of Gaussian splats. Our approach leverages the geometric information inherent in surface-aligned GS, re-interpreting it as a discrete surface representation. Our experiments demonstrate that this approach can achieve a deformation quality that is on par with applying the same edits to the ground truth meshes, thereby circumventing the dependency on geometric proxies. This finding positions Gaussian Splatting not just as a technique for inverse rendering, but as a viable deformable geometric representation in its own right.

Limitations. Although SpLap achieves high-quality deformation without relying on an external proxy, our approach requires a well-aligned reconstruction. Since our framework does not explicitly refine the reconstruction itself, it may fail to infer intrinsic distances when the underlying Gaussian Splatting (GS) quality is poor. This reliance on the assumption that Gaussian kernels are well-aligned with the surface may limit the applicability of our method to general, unconstrained 3D Gaussian scenes. We leave the extension toward more general and robust deformation handling for future work.

Acknowledgement

This work was supported by the NRF of Korea grant (RS-2023-00209723); IITP grants (RS-2022-II220594, RS-2023-00227592, RS-2024-00399817, RS-2025-25441313, RS-2025-25443318); and the Industrial Technology Innovation Program (RS-2025-02317326), all funded by the Korean government (MSIT and MOTIE), as well as grants from the DRB-KAIST SketchTheFuture Research Center.

References

[1] Olga Sorkine and Marc Alexa. As-rigid-as-possible surface modeling. In *Proceedings of the Fifth Eurographics Symposium on Geometry Processing*, SGP '07, page 109–116, Goslar, DEU, 2007. Eurographics Association. 1, 2, 4, 5, 6, 7

[2] Alec Jacobson, Ilya Baran, Jovan Popović, and Olga Sorkine. Bounded biharmonic weights for real-time deformation. *ACM Trans. Graph.*, 30(4), July 2011. 1, 2, 4, 5, 6, 7

[3] Seungwoo Yoo, Kunho Kim, Vladimir G. Kim, and Minhyuk Sung. As-plausible-as-possible: Plausibility-aware mesh deformation using 2d diffusion priors, 2024. 2

[4] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4), July 2023. 2, 3

[5] Yi-Hua Huang, Yang-Tian Sun, Ziyi Yang, Xiaoyang Lyu, Yan-Pei Cao, and Xiaojuan Qi. Sc-gs: Sparse-controlled gaussian splatting for editable dynamic scenes. *arXiv preprint arXiv:2312.14937*, 2023. 2

[6] Joanna Waczyńska, Piotr Borycki, Joanna Kaleta, Sławomir Tadeja, and Przemysław Spurek. D-miso: Editing dynamic 3d scenes using multi-gaussians soup. 2024.

[7] Yuxin Yao, Zhi Deng, and Junhui Hou. Riggs: Rigging of 3d gaussians for modeling articulated objects in videos. In *CVPR*, 2025.

[8] Ziyi Yang, Xinyu Gao, Wen Zhou, Shaohui Jiao, Yuqing Zhang, and Xiaogang Jin. Deformable 3d gaussians for high-fidelity monocular dynamic scene reconstruction. *arXiv preprint arXiv:2309.13101*, 2023. 2

[9] Joanna Waczyńska, Piotr Borycki, Sławomir Tadeja, Jacek Tabor, and Przemysław Spurek. Games: Mesh-based adapting and modification of gaussian splatting. 2024. 2

[10] Lin Gao, Jie Yang, Botao Zhang, Jiamu Sun, Yujie Yuan, Hongbo Fu, and Yu-Kun Lai. Real-time large-scale deformation of gaussian splatting. *ACM Transactions on Graphics (SIGGRAPH Asia 2024)*, 2024. 2, 5, 6, 7

[11] Xiangjun Gao, Xiaoyu Li, Yiyu Zhuang, Qi Zhang, Wenbo Hu, Chaopeng Zhang, Yao Yao, Ying Shan, and Long Quan. Mani-gs: Gaussian splatting manipulation with triangular mesh. *arXiv preprint arXiv:2405.17811*, 2024. 2, 3, 5, 6, 7

[12] Ying Jiang, Chang Yu, Tianyi Xie, Xuan Li, Yutao Feng, Huamin Wang, Minchen Li, Henry Lau, Feng Gao, Yin Yang, and Chenfanfu Jiang. Vr-gs: A physical dynamics-aware interactive gaussian splatting system in virtual reality. *arXiv preprint arXiv:2401.16663*, 2024. 2, 5

[13] Wojciech Zielonka, Timur Bagautdinov, Shunsuke Saito, Michael Zollhöfer, Justus Thies, and Javier Romero. Drivable 3d gaussian avatars. In *I3DV*, March 2025. 2

[14] Jiajun Huang, Shuolin Xu, Hongchuan Yu, and Tong-Yee Lee. Gsdeformer: Direct, real-time and extensible cage-based deformation for 3d gaussian splatting, 2024. 2, 5

[15] Xiao Han Dingyao Liu Fenggen Yu Yan Zhang Yifei Tong, Runze Tian. Cage-gs: High-fidelity cage based 3d gaussian splatting deformation. *arXiv preprint arXiv:2504.12800*, 2025. 2

[16] Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. In *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, volume 34, pages 27171–27183, 2021. 2

[17] Zhaoshuo Li, Thomas Müller, Alex Evans, Russell H Taylor, Mathias Unberath, Ming-Yu Liu, and Chen-Hsuan Lin. Neuralangelo: High-fidelity neural surface reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8456–8465, 2023. 2

[18] Binbin Huang, Zehao Yu, Anpei Chen, Andreas Geiger, and Shenghua Gao. 2d gaussian splatting for geometrically accurate radiance fields. In *SIGGRAPH 2024 Conference Papers*. Association for Computing Machinery, 2024. 2, 3, 5, 8

[19] Weixing Zhang, Zongrui Li, De Ma, Huajin Tang, Xudong Jiang, Qian Zheng, and Gang Pan. Spiking gs: Towards high-accuracy and low-cost surface reconstruction via spiking neuron-based gaussian splatting. *arXiv preprint arXiv:2410.07266*, 2024. 3, 4

[20] Pinxuan Dai, Jiamin Xu, Wenxiang Xie, Xinguo Liu, Huamin Wang, and Weiwei Xu. High-quality surface reconstruction using gaussian surfels. In *ACM SIGGRAPH 2024 Conference Papers*. Association for Computing Machinery, 2024. 3

[21] Antoine Guédon and Vincent Lepetit. Sugar: Surface-aligned gaussian splatting for efficient 3d mesh reconstruction and high-quality mesh rendering. *CVPR*, 2024. 2, 3, 5, 6, 7

[22] Tianyi Xie, Zeshun Zong, Yuxing Qiu, Xuan Li, Yutao Feng, Yin Yang, and Chenfanfu Jiang. Physgaussian: Physics-integrated 3d gaussians for generative dynamics. *arXiv preprint arXiv:2311.12198*, 2023. 2

[23] Antoine Guédon and Vincent Lepetit. Gaussian frosting: Editable complex radiance fields with real-time rendering. *ECCV*, 2024. 2, 3

[24] Michael Kazhdan, Matthew Bolitho, and Hugues Hoppe. Poisson surface reconstruction. In *Proceedings of the fourth Eurographics symposium on Geometry processing*, volume 7, 2006. 2

[25] Yiming Wang, Qin Han, Marc Habermann, Kostas Daniilidis, Christian Theobalt, and Lingjie Liu. Neus2: Fast learning of neural implicit surfaces for multi-view reconstruction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023. 3

[26] Mulin Yu, Tao Lu, Linning Xu, Lihan Jiang, Yuanbo Xiangli, and Bo Dai. Gsdf: 3dgs meets sdf for improved rendering and reconstruction. *arXiv preprint arXiv:2403.16964*, 2024. 3

[27] Yi-Hua Huang Xiuzhe Wu Ziyi Yang Yilun Chen Jiangmiao Pang Xiaojuan Qi Xiaoyang Lyu, Yang-Tian Sun. 3dgsr: Implicit surface reconstruction with 3d gaussian splatting. *arXiv preprint arXiv:2404.00409*, 2024.

[28] Zehao Yu, Torsten Sattler, and Andreas Geiger. Gaussian opacity fields: Efficient adaptive surface reconstruction in unbounded scenes. *ACM Transactions on Graphics*, 2024.

[29] Hanlin Chen, Chen Li, Yunsong Wang, and Gim Hee Lee. Neusg: Neural implicit surface reconstruction with 3d gaussian splatting guidance, 2025. 3

[30] Hanspeter Pfister, Matthias Zwicker, Jeroen van Baar, and Markus Gross. Surfels: surface elements as rendering primitives. In *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '00, page 335–342, USA, 2000. ACM Press/Addison-Wesley Publishing Co. 3

[31] Matthias Zwicker, Hanspeter Pfister, Jeroen van Baar, and Markus Gross. Surface splatting. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '01, page 371–378, New York, NY, USA, 2001. Association for Computing Machinery.

[32] M. Zwicker, H. Pfister, J. van Baar, and M. Gross. Ewa volume splatting. In *Proceedings Visualization, 2001. VIS '01.*, pages 29–538, 2001. 3

[33] Jonathan T. Barron, Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. *CVPR*, 2022. 3

[34] Mario Botsch, Leif Kobbelt, Mark Pauly, Pierre Alliez, and Bruno Lévy. *Polygon Mesh Processing*. CRC Press, Boca Raton, FL, 2010. 3

[35] Jian Sun, Maks Ovsjanikov, and Leonidas Guibas. A concise and provably informative multi-scale signature based on heat diffusion. In *Proceedings of the Symposium on Geometry Processing*, SGP '09, page 1383–1392, Goslar, DEU, 2009. Eurographics Association. 3

[36] Mikhail Belkin, Jian Sun, and Yusu Wang. Constructing laplace operator from point clouds in rd. In *Proceedings of the Twentieth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '09, page 1031–1040, USA, 2009. Society for Industrial and Applied Mathematics. 3

[37] Yang Liu, Balakrishnan Prabhakaran, and Xiaohu Guo. Point-based manifold harmonics. *IEEE Transactions on Visualization and Computer Graphics*, 18(10):1693–1703, 2012.

[38] Chuanjiang Luo, Issam Safa, and Yusu Wang. Approximating Gradients for Meshes and Point Clouds via Diffusion Metric. *Computer Graphics Forum*, 2009.

[39] Hongxing Qin, Yi Chen, Yunhai Wang, Xiaoyang Hong, Kangkang Yin, and Hui Huang. Laplace–beltrami operator on point clouds based on anisotropic voronoi diagram. *Computer Graphics Forum*, 37(6):106–117, 2018. 3

[40] U. Clarenz, M. Rumpf, and A. Telea. Finite Elements on Point Based Surfaces. In Markus Gross, Hanspeter Pfister, Marc Alexa, and Szymon Rusinkiewicz, editors, *SPBG'04 Symposium on Point - Based Graphics 2004*. The Eurographics Association, 2004. 3

[41] Junjie Cao, Andrea Tagliasacchi, Matt Olson, Hao Zhang, and Zhinxun Su. Point cloud skeletons via laplacian based contraction. In *2010 Shape Modeling International Conference*, pages 187–197, 2010.

[42] Nicholas Sharp and Keenan Crane. A Laplacian for Nonmanifold Triangle Meshes. *Computer Graphics Forum (SGP)*, 39(5), 2020. 3, 4, 5, 6, 7

[43] F. Petronetto, A. Paiva, E. S. Helou, D. E. Stewart, and L. G. Nonato. Mesh-Free Discrete Laplace–Beltrami Operator. *Computer Graphics Forum*, 2013. 3

[44] Jian Liang, Rongjie Lai, Tsz Wai Wong, and Hongkai Zhao. Geometric understanding of point clouds using laplace-beltrami operator. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 214–221, 2012. 3

[45] Bo Pang, Zhongtian Zheng, Yilong Li, Guoping Wang, and Peng-Shuai Wang. Neural laplacian operator for 3d point clouds. *ACM Transactions on Graphics (SIGGRAPH Asia)*, 2024. 3, 5, 6, 7

[46] Hongyu Zhou and Zorah Lähner. Laplace-beltrami operator for gaussian splatting, 2025. 3, 5, 6

[47] John R. Silvester. Extremal area ellipses of a convex quadrilateral. *The Mathematical Gazette*, 101(550):11–26, 2017. 5

[48] Donald Shepard. A two-dimensional interpolation function for irregularly-spaced data. In *Proceedings of the 1968 23rd ACM national conference*, pages 517–524, 1968. 5

[49] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020. 5, 6

[50] Angel X. Chang, Thomas A. Funkhouser, Leonidas J. Guibas, Pat Hanrahan, Qi-Xing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, Li Yi, and Fisher Yu. Shapenet: An information-rich 3d model repository. *CoRR*, abs/1512.03012, 2015. 5

[51] Matt Deitke, Dustin Schwenk, Jordi Salvador, Luca Weihs, Oscar Michel, Eli VanderBilt, Ludwig Schmidt, Kiana Ehsani, Aniruddha Kembhavi, and Ali Farhadi. Objaverse: A universe of annotated 3d objects, 2022. 5

[52] Cédric Pinson. Sketchfab – the best 3d viewer on the web. https://sketchfab.com/, 2011. Accessed: 2025-06-09. 5

[53] Rasmus Jensen, Anders Dahl, George Vogiatzis, Engin Tola, and Henrik Aanaes. Large scale multi-view stereopsis evaluation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014. 5

[54] Jingwei Huang, Hao Su, and Leonidas Guibas. Robust watertight manifold surface generation method for shapenet models. *arXiv preprint arXiv:1802.01698*, 2018. 6

[55] Hugues Hoppe, Tony DeRose, Tom Duchamp, John McDonald, and Werner Stuetzle. Mesh optimization. In *Proceedings of the 20th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '93, page 19–26, New York, NY, USA, 1993. Association for Computing Machinery. 6

[56] Yi Yang and Deva Ramanan. Articulated human detection with flexible mixtures of parts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(12):2878–2890, 2013. 6

[57] Gyeongsik Moon, Ju Yong Chang, and Kyoung Mu Lee. Camera distance-aware top-down approach for 3d multi-person pose estimation from a single rgb image. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10133–10142, 2019. 6

[58] Wen Jiang, Nikos Kolotouros, Georgios Pavlakos, Xiaowei Zhou, and Kostas Daniilidis. Coherent reconstruction of multiple humans from a single image. In *Proceedings of*

the IEEE/CVF conference on computer vision and pattern recognition, pages 5579–5588, 2020.

[59] Can Wang, Jiefeng Li, Wentao Liu, Chen Qian, and Cewu Lu. Hmor: Hierarchical multi-person ordinal relations for monocular multi-person 3d pose estimation. In *European Conference on Computer Vision*, pages 242–259. Springer, 2020.

[60] Yu Cheng, Bo Wang, Bo Yang, and Robby T Tan. Graph and temporal convolutional networks for 3d multi-person pose estimation in monocular videos. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 1157–1165, 2021. 6