# INJECTING INDUCTIVE BIAS TO 3D GAUSSIAN SPLATTING FOR GEOMETRICALLY ACCURATE **RADIANCE FIELDS**

## **Anonymous authors**

Paper under double-blind review



Figure 1: Our approach enhances geometric accuracy of 3D Gaussian Splatting by parameterizing covariance, the key determinant of surface normal, with distribution of adjacent Gaussians. Zoom in for detailed comparison.

## ABSTRACT

3D Gaussian Splatting (3DGS) has significantly advanced high-fidelity, real-time novel view synthesis. However, its discrete nature limits the accurate reconstruction of geometry. To address this issue, recent methods have introduced rendering and regularization of depth and normal maps from 3D Gaussians, leading to plausible results. In this paper, we argue that computing normals from independently trainable Gaussian covariances contradicts the strict definition of normals, which should instead be derived from the distribution of neighboring densities. To address this, we introduce an inductive bias into 3DGS by explicitly parameterizing covariances of Gaussians using principal axes and variances of distribution computed from neighboring Gaussians. These axes and variances are then regularized to ensure local surface smoothness. Our approach achieves state-of-the-art performance on multiple datasets.

043 044 045

046

#### INTRODUCTION 1

047 Along with the advancements made by Neural Radiance Fields (NeRF) (Mildenhall et al., 2021), 048 3D Gaussian Splatting (3DGS) (Kerbl et al., 2023) has significantly influenced novel view syn-049 thesis research, enabling high-fidelity and real-time rendering from a collection of posed 2D images. Progress in NeRF research naturally led to advances such as NeuS (Wang et al.) and Neu-050 ralangelo (Li et al., 2023) for accurate geometry representation via implicit Signed Distance Fields 051 (SDF). Similarly, the evolution of geometrically accurate Gaussian Splatting is ongoing by recent 052 works such as 2D Gaussian Splatting (Huang et al., 2024), Gaussian Opacity Fields (GOF) (Yu et al., 2024b), and RaDe-GS (Zhang et al., 2024).

027

028

029

031 032

033

034

038

039

040

041

042

000

069



Figure 2: Renderings on scenes trained with IBGS on MipNeRF-360 (top) and DTU (bottom) dataset. IBGS is a model for geometrically accurate representation of 3D Gaussians.

071 The core insight of the recent 3DGS-based geometry reconstruction methods is to determine 072 the depth and normal of a Gaussian given its co-073 variance. For instance, 2DGS defines a surfel 074 disk, so that a normal can be defined as the vec-075 tor perpendicular to the disk. Meanwhile, GOF 076 and RaDe-GS calculate the intersection plane 077 between 3D Gaussian and a camera ray, so that 078 normal of the plane can be regarded as the nor-079 mal of the Gaussian.

However, all these methods calculate normal 081 from a covariance matrix independently train-082 able for each Gaussian. In fact, this schemat-083 ics is counter-intuitive to the common under-084 standing of normal estimation in computational 085 graphics (Shirley et al., 2009), where normal is instead defined by the gradient of adjacent density distribution. Consider 3DGS in Figure 3. 880 From the distribution of the two adjacent Gaussians located left and right, a geometrically rea-089 sonable surface normal is the vector perpendic-090 ular to the local surface formed by these Gaus-091



Figure 3: Normal determined by fully trainable covariance in 3DGS does not necessarily correspond to normal of local surface determined by neighboring Gaussians. However, covariance in IBGS is directly parameterized using the neighboring Gaussians, so that the two normal vectors are aligned. Also, updating covariance leads to the relocation of adjacent Gaussians in IBGS.

sians. However, a fully trainable covariance in 3DGS can be oriented arbitrarily, creating a dis-092 crepancy between the normal calculated from the covariance and the normal calculated from the underlying surface formed by adjacent Gaussians. In other words, the orientation of covariance has 094 room to fall into local minima while satisfying the photometric loss, which is ill-posed for accurate geometry reconstruction. 096

In this paper, we tackle this issue by directly injecting Inductive Bias to 3D Gaussian Splatting (**IBGS**). Specifically, we parameterize the covariance matrix of a Gaussian using the distribution 098 of adjacent Gaussians as illustrated in the bottom left figure of Figure 3. Our design enforces a Gaussian covariance to be oriented by the locations of the adjacent Gaussians via Singular Value 100 Decomposition (SVD). Thus, updating covariance requires back-propagating gradients to adjacent 101 Gaussian locations, as illustrated in the bottom right figure of Figure 3. As these adjacent points also 102 form covariances using their own neighbors, the scene eventually forms a set of globally chained 103 Gaussians by satisfying our inductive bias. To form a coherent and smooth local surface of this 104 graph-like structure, we also propose geometry regularization methods to (i) induce the neighboring 105 Gaussians to be distributed along a local plane and (ii) assimilate normals of adjacent local planes to form a smooth surface. As a result, our method yields high-quality rendering of scene geometry 106 such as depth and normal reported in Figure 2, enabling high-quality surface reconstruction results 107 compared to state-of-the-arts as shown in Figure 1.

- In summary, our contribution is three-fold:
  - We propose a novel variant of a 3DGS model, which parameterizes a covariance of a Gaussian with neighboring Gaussians, to reflect local geometry structure to the Gaussian covariance.
  - We propose regularization losses for smooth and coherent surface reconstruction.
  - Our method achieves competitive quantitative and qualitative results in surface reconstruction among 3DGS-based methods, while maintaining faster training time compared to implicit neural representation-based methods.
- 117 118 119

110

111

112

113 114

115

116

2 RELATED WORKS

121 Novel View Synthesis Given a set of posed images, NeRF (Mildenhall et al., 2021) employs a 122 multi-layer perceptron (MLP) to model the density and view-dependent color of a scene. Image 123 is rendered from the MLP via the volume rendering frameworks (Drebin et al., 1988; Kajiya & 124 Von Herzen, 1984), and optimized with a photometric loss for training. Following the broad impact 125 of NeRF on the research community, advancements have been made to enhance its efficiency and 126 effectiveness. Notably, several approaches have accelerated NeRF's training and rendering process 127 through the integration of octree (Yu et al., 2021), sparse voxel-grid (Fridovich-Keil et al., 2022), 128 multi-resolution hash-grid (Fridovich-Keil et al., 2022), and factorized tensors (Chen et al., 2022). Other lines of work successfully mitigate aliasing effect (Barron et al., 2021; 2023; Hu et al., 2023), 129 or model unbounded scenes (Zhang et al., 2020; Barron et al., 2022). More recently, 3DGS (Kerbl 130 et al., 2023) has emerged as a powerful technique for achieving real-time and high-fidelity rendering. 131 Representing scenes as a set of 3D Gaussians, the method employed EWA volume splatting (Zwicker 132 et al., 2001) to rasterize the 3D Gaussians into screen space. Subsequent research has refined this 133 approach by improving rendering quality through anti-aliasing (Yu et al., 2024a) or extending to 134 dynamic scene modeling (Wu et al., 2024; Yang et al., 2024). 135

136

Surface Reconstruction Traditionally, Multi-View Stereo (MVS) techniques (Schönberger et al., 2016; Yao et al., 2018; Yu & Gao, 2020) addressed the 3D reconstruction problem from multi-view images. These methods involve a series of steps, including feature matching, depth map estimation, and the fusion of these maps into a point cloud. The point cloud is then used to reconstruct a surface, often via approaches like screened Poisson surface reconstruction (Kazhdan & Hoppe, 2013).

142 Meanwhile, NeRF-based methods have been employed for accurate surface reconstruction, as novel view synthesis from implicit scene representation is closely related to underlying 3D reconstruction. 143 For instance, pioneering methods such as UNISURF (Oechsle et al., 2021), NeuS (Wang et al.), 144 and VolSDF (Yariv et al., 2021) propose an implicit SDF for surface representation, followed by its 145 volume rendering for supervision with posed images. Triangle meshes can then be straightforwardly 146 extracted using techniques like Marching Cubes (Lorensen & Cline, 1998). Noticeable progress in 147 this field has been driven by Neuralangelo (Li et al., 2023), which proposed a regularization method 148 with numerical gradient and coarse-to-fine representation for SDFs. Other lines of work integrated 149 monocular priors (Yu et al., 2022) or sensor depth (Azinović et al., 2022) to integrate geometric cues 150 for accurate surface representations. 151

Although 3DGS has recently gained traction as a robust method for novel view synthesis, extract-152 ing 3D surfaces from Gaussian splats presents significant challenges. To address this issue, meth-153 ods such as SuGaR (Guédon & Lepetit, 2024) and NeuGS (Chen et al., 2023) have introduced the 154 concept of flat Gaussians, which are designed to better conform to object surfaces, thereby improv-155 ing surface alignment and reconstruction quality. 2DGS (Huang et al., 2024) similarly proposes 156 2D Gaussian surfel, yet formulates the rasterization process into 2D-to-2D transformation, making 157 the splatting process more perspectively accurate. Recent works such as GOF (Yu et al., 2024b) 158 and RaDe-GS (Zhang et al., 2024) computes ray-Gaussian intersection to render depth and normal. 159 Meanwhile, a line of works improves geometry by regularizing Gaussians' scale. Hyung et al. (2024) regularizes effective rank of Gaussian scales, while Hwang et al. (2024) minimizes scale along the 160 surface normal estimated from monocular prior. However, none of them questions the importance 161 of the relationship between the neighboring Gaussians for geometrically accurate representation.

#### 162 3 METHOD 163

#### 3.1 PRELIMINARIES

166 **3D Gaussian Splatting** 3DGS proposes scene representation and rasterization method from a col-167 lection of 3D Gaussian primitives. A 3D Gaussian primitive  $\mathcal{G}$  contains a set of trainable parameters: the mean position  $\mu \in \mathbb{R}^{3 \times 1}$ , a covariance matrix  $\Sigma \in \mathbb{R}^{3 \times 3}$ , opacity  $\alpha \in [0, 1]$ , and a color c ex-168 169 pressed in spherical harmonics to model view dependency. The covariance matrix  $\Sigma$  is positive 170 semi-definite and is constructed using a rotation matrix  $\mathbf{R} \in \mathbb{R}^{3 \times 3}$ , which is parameterized by a quaternion, and a diagonal scale matrix S. This Gaussian primitive in 3D space can be described as 171

172 173

164

165

174 175

187

188

189 190 191

192

193

194

195

 $\mathcal{G}(\mathbf{x}) = e^{-\frac{1}{2}(\mathbf{x}-\mu)^T \mathbf{\Sigma}^{-1}(\mathbf{x}-\mu)}.$ (1)

The primitives are then projected onto a 2D plane (Zwicker et al., 2001). In this projection, the 176 covariance matrix is transformed into screen space as  $\Sigma' = \mathbf{J}\mathbf{W}\Sigma\mathbf{W}^{\top}\mathbf{J}^{\top}$ , where W is the world-to-177 camera transformation matrix and J is the Jacobian of the affine projection matrix approximation. 178  $\sum_{m=1}^{M} c_m \alpha_m \mathcal{G}_m^{2D}(\mathbf{u}) \prod_{j=1}^{m-1} (1 - \alpha_j \mathcal{G}_j^{2D}(\mathbf{u})), \text{ where } \mathcal{G}^{2D} \text{ is the Gaussian projected to screen-space and } \mathbf{u} \text{ refers to the screen coordinate.}$ The final image is then produced by alpha-blending these Gaussians based on their depth as  $\mathbf{c}(\mathbf{u}) =$ 179 180 181

182 Given that the Gaussians are initially derived from sparse Structure-from-Motion (SfM) points, the 183 Adaptive Density Control (ADC) technique is applied to refine and densify the scene during optimization. ADC clone and splits Gaussians (Yu et al., 2024b) when the densification signal of a 185 Gaussian,  $\frac{\partial L'}{\partial \mathbf{u}}$ , is greater than a predefined threshold  $\tau$  as following: 186

 $\frac{\partial L'}{\partial \mathbf{x}} := \sum_{i \in \mathcal{P}} \left\| \frac{\partial L}{\partial \mathbf{p}_i} \frac{\partial \mathbf{p}_i}{\partial \mathbf{x}} \right\|_2 > \tau,$ (2)

where  $\mathbf{x}, \mathcal{P}$ , and  $\mathbf{p}_i$  denote the center of projected Gaussians, the list of pixel indices that the Gaussian contributed to, and the following pixel, respectively. This condition ensures that regions with significant positional gradients, which indicate incomplete reconstruction, are effectively densified by adding more Gaussians, thereby improving the expressibility.

196 **Rasterization of Depth and Normal** Given a Gaussian primitive, RaDe-GS (Zhang et al., 2024) 197 proposes a method to rasterize depth and normal for a camera ray. Specifically, depth is calculated by finding the intersection point to a Gaussian primitive. Normal can then be calculated by finding 199 a plane formed by the intersection point and the center of the Gaussian. Subsequently, depth d and 200 normal n can simply be rasterized to pixel  $\mathbf{u} = (u, v)$  as  $d = z_c + p^{\top} \Delta \mathbf{u}_c$ ,  $n = -\mathbf{J}^{\top} (q \ 1)^{\top}$ , where  $z_c$  is the depth of the Gaussian center,  $\Delta u_c = (u_c - u, v_c - v)^{\top}$ ,  $u_c$  and  $v_c$  are Gaussian centers projected to screen space. Here, q and p are vectors that hold the first two elements of  $\hat{q}$  and 202  $\hat{p}$  respectively, which can be calculated using Gaussian center and covariance as 203

204 205 206

201

$$\hat{q} = \frac{\mathbf{v}' \boldsymbol{\Sigma}'^{-1}}{\mathbf{v}'^{\top} \boldsymbol{\Sigma}'^{-1} \mathbf{v}'}, \quad \hat{p} = \frac{z_c}{t_c} \hat{q}, \tag{3}$$

207 208

where  $\mathbf{v}' = (0, 0, 1)^{\top}$ ,  $z_c$  and  $t_c$  are depth of Gaussian center in Euclidean space and ray space, 209 respectively. Readers may refer to the original paper for detailed derivation. 210

211 In fact, 2DGS (Huang et al., 2024) and GOF (Yu et al., 2024b) propose similar methods of locating 212 ray-Gaussian intersection, where covariance is the key determinant of depth and normal rasteriza-213 tion, as exemplified in Equation 3. However, an independently trainable covariance can be oriented in any directions and thus can significantly differ from the distribution of adjacent Gaussians, as 214 illustrated in Figure 3. As a result, rasterizing depth and normal from such covariance does not 215 necessarily reflect local geometry.

# 216 3.2 COVARIANCE PARAMETERIZATION WITH ADJACENT GAUSSIANS

Instead of defining a fully trainable covariance matrix for every Gaussian primitive, our method
 defines the covariance as a function of adjacently located Gaussians, so that a Gaussian covariance
 can reflect the local geometry formed by its neighbors.

**Parameterization with Singular Value Decomposition** For a given Gaussian primitive  $\mathcal{G}$ , consider the set of positions  $P = [\mu; \mu^1; \cdots \mu^K]$ , where  $\mu^k$  is the mean of  $\mathcal{G}^k$ , the *k*th-nearest neighbor of  $\mathcal{G}$ . We first calculate the covariance of these points as

$$\Gamma = \operatorname{Var}(P) = \frac{1}{K} (P - \bar{\mu}) (P - \bar{\mu})^{\top},$$
(4)

where  $\bar{\mu} = \frac{1}{K+1} (\mu + \sum_{k \in K} \mu^k)$ . A naive approach is to simply use  $\Gamma$  as a covariance of the Gaussian. However, we instead decompose  $\Gamma$  to retrieve the orientation of local plane that these neighboring Gaussians form. Specifically, we decompose  $\Gamma$  into principal components using Singular Value Decomposition (SVD) (Klema & Laub, 1980) as

$$\Gamma = U\Lambda V^{\top},\tag{5}$$

where  $V = [\mathbf{v}_1; \mathbf{v}_2; \mathbf{v}_3]$  are orthonormal eigenvectors and  $\Lambda = diag(\lambda_1, \lambda_2, \lambda_3)$  are eigenvalues. These eigenvalues indicate the variance of the point distribution along each eigenvector ordered from largest to smallest, or  $\lambda_1 \ge \lambda_2 \ge \lambda_3 \ge 0$ . The first two eigenvectors lie on a local plane formed by the neighboring Gaussians, with the third eigenvector oriented perpendicular to this plane. Therefore, we can define the covariance matrix of a Gaussian using these eigenvectors and learnable scale S as  $\Sigma = VSS^{\top}V^{\top}$ . Gradients will be back-propagated to P to update  $\Sigma$ , so that the means of neighboring Gaussians are jointly optimized to form a desired  $\Sigma$ .

However, defining the rotation factor of  $\Sigma$  solely from K Gaussian neighbors may limit the expressiveness of the scene. To address this, we introduce a residual rotation term,  $\Delta R$ . This term is initialized as an identity matrix and is learned by integrating it into V, leading to the updated transformation  $V' = \Delta RV$ . Our Gaussian covariance is then computed as follows:

$$\mathbf{\Sigma} = V' S S^{\top} V'^{\top}.$$
(6)

It is crucial to regularize the residual rotation to preserve the inductive bias we have established. To achieve this, we introduce the following regularization term:

$$\mathcal{L}_r = |\Delta r - r_{\text{identity}}|,\tag{7}$$

where  $\Delta r$  represents  $\Delta R$  in quaternion form, and  $r_{\text{identity}}$  is the identity quaternion. Note that we do not train  $\Delta R$  during the earlier optimization stages, where most geometry is determined, to prioritize inductive bias over expressibility. Details and analysis on the residual rotation regularization can be found in Appendix A.3.

## 260 3.3 EIGENSYSTEM REGULARIZATION FOR LOCALLY SMOOTH PLANE ASSUMPTION

In this section, we introduce methods to regularize the eigensystem, the set of the eigenvectors paired with their eigenvalues calculated from Equation 5, in order to generate smooth and coherent local surfaces. We first regularize the smallest eigenvalue to align the neighboring Gaussians along a local plane. Furthermore, we enhance surface smoothness by assimilating the normals of adjacent planes proportional to the surface planarity.

265 266 267

268

221

222

233

234 235

247 248 249

250

251

253 254

255

256

257

258 259

261

262

264

### 3.3.1 EIGENVALUE REGULARIZATION

269 Since photometric loss by itself is geometrically ill-posed, there can be cases when a set of neighboring Gaussians do not form a plane (*i.e.* distributed uniformly toward all directions). Thus, positions

of these neighboring Gaussians need to be optimized toward a local plane to make the set of locally 271 distributed Gaussians to behave as a surface. 272

Meanwhile, an eigenvalue reflects the amount of variance along its corresponding eigenvector. Thus, 273 smaller  $\lambda_3$  indicates that the Gaussians are distributed closer to the local plane whose normal is 274 defined by the third eigenvector  $\mathbf{v}_3$ . To optimize our model as such, we simply minimize  $\lambda_3$  using 275 the following loss: 276

$$\mathcal{L}_{\text{eigval}} = \lambda_3. \tag{8}$$

### 3.3.2 EIGENVECTOR REGULARIZATION

280 In order to form locally smooth surfaces, we propose a regularization loss to assimilate the nor-281 mals of neighboring local planes. Specifically, we minimize the cosine distance between the third eigenvectors of adjacent Gaussians using the following loss: 282

$$\mathcal{L}_{\text{eigvec}} = \frac{1}{K} \sum_{k \in K} \gamma^2 (1 - \mathbf{v}_3 \cdot \mathbf{v}_3^k), \tag{9}$$

where  $\mathbf{v}_3^k$  is the third eigenvector of covariance calculated with K nearest neighbors of the k-th neighboring Gaussian using Equation 5. Note that cosine distance is computed proportional to local planarity  $\gamma$ , which we elaborate in the next paragraph.

Local Planarity Estimation with Eigenvalues When Gaussians express complex surface geometry, adjacent local planes do not necessarily share similar normal vectors. To quantify such cases, we define the local planarity formed by the neighboring Gaussians by leveraging the eigenvalues calculated from Sec. 3.2. Formally,  $\gamma$  is defined as follows:

$$\gamma = 1 - \frac{3\lambda_3}{\sum_{j=\{1,2,3\}} \lambda_j}.$$
 (10)

Note that  $0 \le \gamma \le 1$  since eigenvalues are positive and descending. A value of  $\gamma = 1$  indicates a completely planar surface, while lower values suggest otherwise. Also,  $\gamma$  is detached from the optimization graph, as minimizing  $\gamma$  can increase  $\lambda_1$  and  $\lambda_2$ , which is irrelevant to accurate geometry reconstruction.

#### SPARSITY-AWARE ADAPTIVE DENSITY CONTROL 3.4

304 ADC densifies Gaussians with a high densification signal as in Equation 2. However, for homoge-305 neously colored regions with low view-dependency (i.e., extremely shadowed region as Figure 7), 306 Gaussians yield low densification signals even when they are sparsely distributed, as a few number 307 of points are enough to express the color of this relatively simple regions. However, such a case is problematic for accurate geometry reconstructions, as planes formed by these sparse neighbors 308 are not local anymore, which may cause over-smoothing. We propose an additional densification 309 algorithm to prevent such cases by selectively augmenting the number of Gaussians in regions with 310 low densification signals and high sparsity. We term this strategy Sparsity-aware Adaptive Density 311 Control (SADC) and use this along with the existing ADC (Kerbl et al., 2023). 312

313 Specifically, Gaussians with densification signal lower than  $\tau$  are first selected. Then, we calculate 314 the mean distance with its K-nearest Gaussians. The region these K neighbors cover is considered sparse if the distance is greater than  $\eta_{min} \cdot \tau_s$ . We borrow the split threshold  $\tau_s$  to measure sparsity 315 (Gaussian is split into two if its scale is larger than  $\tau_s$  (Kerbl et al., 2023)) as  $\tau_s$  can be regarded as 316 the maximal expectation of a Gaussian size. We also set the upper bound of sparsity  $\eta_{max} \cdot \tau_s$  to 317 reject background points.  $\eta_{min} = 3.0$  and  $\eta_{max} = 10.0$  are hyper-parameters. 318

319 Once a Gaussian is considered low-signaled and sparse, we interpolate N additional Gaussians from 320 each neighbor. Specifically, we sample the center of the interpolated Gaussian from a Gaussian distribution, where its mean is interpolated between the center of the original Gaussian and its neighbor 321 and the covariance is inherited from the original Gaussian. All other properties, such as opacity  $\alpha$ 322 and SH color c, are copied from the original Gaussian, except for S, which is scaled by  $\frac{1}{N+1}$  to 323 reflect the distance between adjacent Gaussians. The detailed algorithm is provided in Algorithm 1.

270

277 278

279

287

288

289 290

291

292

293

298

299

300

301 302

Alg	orithm 1 Sparsity-aware Adaptive Density Control (SADC)	
1:	$\mathcal{G}, \mathcal{G}' = \{\}, K, N$	
2:	if $\frac{\partial L'}{\partial \mathbf{u}} < \tau$ then	▷ Low densification signal
3:	$d = rac{1}{K} \sum_{k \in K} \left\  oldsymbol{\mu} - oldsymbol{\mu}^k  ight\ _2$	
4:	if $d > \eta_{\min} \cdot \tau_s$ and $d < \eta_{\max} \cdot \tau_s$ then	▷ High sparsity signal
5:	for $k \leftarrow 1$ to $K$ do	
6:	for $n \leftarrow 1$ to N do	
7:	$oldsymbol{\mu}' \sim \mathcal{N}(oldsymbol{\mu} + rac{n}{N+1}(oldsymbol{\mu}^k - oldsymbol{\mu}), rac{1}{N+1}oldsymbol{\Sigma})$	▷ Interpolative sampling
8:	$oldsymbol{lpha}' \leftarrow oldsymbol{lpha}, \mathbf{c}' \leftarrow \mathbf{c}, \mathbf{S}' \leftarrow rac{1}{N+1} \mathbf{S}$	
9:	$\mathcal{G}'.append(\{\mu', \alpha', \mathbf{c}', \mathbf{S}'\})$	
10:	return $\mathcal{G}'$	

339

340

341

#### 3.5 Optimization

We supervise our model with RGB loss  $\mathcal{L}_c = (1 - \beta)\mathcal{L}_1 + \beta\mathcal{L}_{D-SSIM}$  following Kerbl et al. (2023). In addition, we render depth and normal maps to screen space using rasterizer from Zhang et al. (2024), which are then supervised in screen space using depth-distortion and depth-normal consistency loss (Huang et al., 2024):

$$\mathcal{L}_d = \sum_{i,j} \omega_i \omega_j |z_i - z_j|, \quad \mathcal{L}_n = \sum_i \omega_i (1 - n_i^\top \tilde{n}), \tag{11}$$

where  $\omega_i = \alpha_i \mathcal{G}_i^{2D}(\mathbf{u}) \prod_{j=1}^{i-1} (1 - \alpha_j \mathcal{G}_j^{2D}(\mathbf{u}))$  is the blending weight of the *i*-th intersection,  $z_i$  is the depth of intersection points,  $\tilde{n}$  is the normal map estimated from gradient of depth map, and  $n_i$ is per-Gaussian normal rasterized with Equation 3. We empirically learned that employing  $\mathcal{L}_d, \mathcal{L}_n$ along with  $\mathcal{L}_{\text{eigvec}}$  and  $\mathcal{L}_{\text{eigvel}}$  yield the best results. We also discuss in Appendix. A.1 on distinct behaviors of  $\mathcal{L}_n$  and  $\mathcal{L}_{\text{eigvec}}$ , which may look similar. Finally, our loss can be described as:

351 352

353

354

355

356 357 358

359

365

366

367

368

$$\mathcal{L} = \mathcal{L}_c + w_{\text{eigval}} \mathcal{L}_{\text{eigval}} + w_{\text{eigvec}} \mathcal{L}_{\text{eigvec}} + w_r \mathcal{L}_r + w_d \mathcal{L}_d + w_n \mathcal{L}_n, \tag{12}$$

where we set  $w_{\text{eigval}} = 10$ ,  $w_{\text{eigvec}} = 0.4$ ,  $w_r = 0.01$ . We follow GOF (Yu et al., 2024b) and RaDe-GS (Zhang et al., 2024) and set  $w_d = 100$ , and  $w_n = 0.05$  for all experiments. We report detailed training strategy in Appendix A.2.

### 4 EXPERIMENT

We compare our methods with SuGaR (Guédon & Lepetit, 2024), 2DGS (Huang et al., 2024), GOF (Yu et al., 2024b), and RaDe-GS (Zhang et al., 2024), which are state-of-the-art surface reconstruction methods based on 3DGS (Kerbl et al., 2023). Implicit representation-based methods such as NeRF (Mildenhall et al., 2021), VolSDF (Yariv et al., 2021), NeuS (Wang et al.), and Neuralangelo (Li et al., 2023) are also compared.

Mesh Extraction Depth maps are rendered from all training views to construct Truncated Signed Distance Fields (TSDF) (Curless & Levoy, 1996). From TSDF, Marching Cube (Lorensen & Cline, 1998) is used to construct a mesh.

**Datasets and Evaluation** Since our objective is an accurate surface reconstruction, we experimented on DTU (Jensen et al., 2014) benchmark to compare Chamfer Distance (CD) on surface reconstructions. To measure the accuracy of surface normal as another aspect of geometry evaluation, we experimented on Synthetic NeRF (Mildenhall et al., 2021), which includes ground-truth normal maps. We measured Normal Similarity Score (NSS) between rendered normal and groundtruth normal maps as

NSS := 
$$\frac{1}{\|M\|} \sum_{i} M_{i} \cdot \frac{n_{i}^{\top} n_{i}^{\text{gr}}}{\|n_{i}\| \|n_{i}^{\text{gr}}\|},$$
 (13)

375 376 377

where M is an optional GT mask,  $n_i$  and  $n_i^{\text{gt}}$  are rendered normal and GT normal at *i*-th pixel.

Table 1: **Quantitative comparison on the DTU Dataset**. We report CD and average optimization time on different methods. Best, second best, and third best within explicit/implicit methods are marked as red, orange, and yellow, respectively. *Underlined italic bold* denotes the best among all.

		24	37	40	55	63	65	69	83	97	105	106	110	114	118	122	Mean	Time
+	NeRF	1.90	1.60	1.85	0.58	2.28	1.27	1.47	1.67	2.05	1.07	0.88	2.53	1.06	1.15	0.96	1.49	>12h
ici	VolSDF	1.14	1.26	0.81	0.49	1.25	0.70	0.72	1.29	1.18	0.70	0.66	1.08	0.42	0.61	0.55	0.86	>12h
[di	NeuS	1.00	1.37	0.93	0.43	1.10	0.65	0.57	1.48	1.09	0.83	0.52	1.20	0.35	0.49	0.54	0.84	>12h
Ξ.	Neuralangelo	<u>0.37</u>	0.72	0.35	<u>0.35</u>	0.87	<u>0.54</u>	<u>0.53</u>	1.29	<u>0.97</u>	0.73	<u>0.47</u>	<u>0.74</u>	<u>0.32</u>	<u>0.41</u>	<u>0.43</u>	<u>0.61</u>	>12h
	3DGS	2.14	1.53	2.08	1.68	3.49	2.21	1.43	2.07	2.22	1.75	1.79	2.55	1.53	1.52	1.50	1.96	13.8m
-	SuGaR	1.47	1.33	1.13	0.61	2.25	1.71	1.15	1.63	1.62	1.07	0.79	2.45	0.98	0.88	0.79	1.33	82.1m
:C	2DGS	0.46	0.80	0.33	0.37	0.95	0.86	0.80	1.25	1.25	0.73	0.67	1.24	0.39	0.65	0.47	0.75	14.1m
ſď	GOF	0.45	0.76	0.37	0.37	0.95	0.84	0.74	1.18	1.30	0.70	0.81	0.80	0.40	0.73	0.48	0.72	71.0m
G	RaDe-GS	0.46	0.74	0.34	0.38	0.81	0.76	0.76	1.21	1.22	0.66	0.70	0.86	0.36	0.68	0.47	0.69	16.2m
	IBGS (Ours)	0.49	<u>0.68</u>	0.37	0.38	<u>0.78</u>	0.73	0.73	<u>1.12</u>	1.24	<u>0.61</u>	0.61	0.90	0.37	0.65	0.46	0.67	91.2m

### 4.1 COMPARISON TO STATE-OF-THE-ARTS

We compare CD on DTU, and report the results in Table 1. Our method achieves the best performance among 3DGS-based methods. Compared to Neuralangelo, our method takes faster training time, while outperforming on a few scenes within dataset. This means that our method is a more computationally efficient choice while having a potential to outperform on some scenarios. We also report qualitative comparisons on DTU in Figure 1 and Figure 4. As can be observed, our method is robust against broken surface geometry reconstruction. We also observed that our method successfully reconstruct surface with curvature such as round foot instep of the doll, as observed in second row result in Fig 4.

401 We also compare NSS on NeRF-synthetic and 402 report the results in Table 2. We report the 403 complete list of NSS for each scene in Ap-404 pendix A.6. Our method achieves the best NSS 405 among SOTA 3DGS-based methods. Such out-406 performance can also be observed from quali-407 tative comparison on surface reconstructions in Figure 6. Our method yields the most complete 408 and smooth surface reconstructions. 409

410 Although the Mip-NeRF360 dataset does not 411 include a quantitative evaluation of geome-412 try, we have provided a qualitative comparison of surface reconstruction and normal ren-413 dering in Fig.5. In particular, the surface recon-414 struction results displayed in Fig.5-(a) highlight 415 the ability of our method to produce a highly 416 accurate and plausible surface reconstruction. 417 Another notable observation is the preserva-418 tion of complex geometrical features, such as 419 the intricate and bumpy texture of cloth sur-420 faces, which remains intact under our regular-421 ization. Our method can thus be optimized to-422 ward smooth surfaces without compromising

Table 2: Quantitative comparison on Synthetic NeRF dataset. Our method yields the best Normal Similarity Score (NSS) among state-of-theart 3DGS-based methods.

	NSS $\uparrow$	PSNR↑	SSIM↑	LPIPS↓
2DGS	0.695	33.65	0.969	0.0317
GOF	0.697	33.71	0.969	0.0308
RaDe-GS	0.700	33.68	0.970	0.0304
IBGS (Ours)	0.703	33.70	0.971	0.0321

Table 3: Quantitative comparison on Mip-NeRF 360 Dataset. Our method yields comparable novel view synthesis result especially on indoor scenes.

	0	utdoor Sco	ene	]	Indoor sce	ne
	PSNR $\uparrow$	SSIM $\uparrow$	LPIPS $\downarrow$	PSNR $\uparrow$	SSIM $\uparrow$	LPIPS $\downarrow$
NeRF	21.46	0.458	0.515	26.84	0.790	0.370
Deep Blending	21.54	0.524	0.364	26.40	0.844	0.261
Instant NGP	22.90	0.566	0.371	29.15	0.880	0.216
MERF	23.19	0.616	0.343	27.80	0.855	0.271
MipNeRF360	24.47	0.691	0.283	31.72	0.917	0.180
3D GS	24.54	0.731	0.234	30.41	0.920	0.189
Mip-Splatting	24.65	0.729	0.245	30.90	0.921	0.194
2D GS	24.34	0.717	0.246	30.40	0.916	0.195
GOF	24.82	0.750	0.202	30.79	0.924	0.184
RaDe-GS	25.17	0.764	0.199	30.74	0.928	0.165
IBGS (Ours)	24.64	0.723	0.213	30.69	0.932	0.161

high-fidelity reconstructions of complex surfaces. Such balance is crucial for generating high-quality
 surface representations across various geometrical complexities.

Our method demonstrates competitive performance in novel view synthesis on the MipNeRF-360 dataset, particularly for indoor scenes, as shown in Table 3. However, as observed from Figure 5-(b), PSNR does not reflect the quality of the underlying geometry. While GOF and RaDe-GS produce visually appealing RGB renderings on the floor, the underlying geometries observed from the normal renderings indicate inaccurate geometric structure with holes. In contrast, our approach delivers comparable visual quality and maintains a more accurate and consistent geometric structure. Such distinction between RGB and geometry rendering underscores the importance of evaluating visual fidelity and geometric accuracy when assessing the model as a scene reconstruction method.

391 392

378



Figure 4: Qualitative comparisons on DTU (Jensen et al., 2014) dataset. IBGS yields the most plausible results among 3DGS-based methods.



Figure 5: Qualitative comparisons on MipNeRF360 (Barron et al., 2022) dataset over (a) surface reconstruction, (b) novel view synthesis (*top*) and rendered normal maps (*bottom*).

472 473 4.2 Ablation Study

448

449

470

471

We conducted an ablation study for our proposed methods and reported the results in Table 4. Specifically, we ablate over *param.*,  $\mathcal{L}_{eigvec}$ , where *param.* refers to our covariance parameterization with eigenvectors of neighboring Gaussians. Notably, full regularization without our parameterization yields the worst result. For regularizations, we can empirically conclude that the absence of either  $\mathcal{L}_{eigval}$  or  $\mathcal{L}_{eigvec}$  results in a bad result. Such observation corresponds to the design philosophy of our method, as eigenvector assimilation is significant only when it can behave as local plane normal, which can be achieved by minimizing the third eigenvalue.

Furthermore, SADC exhibits improvements in both quantitative and qualitative outcomes from Table 4 and Figure 7 respectively. When SADC is not applied, certain regions, particularly the shadowed area between the arms, exhibit excessive smoothing, leading to the loss of geometric details.
In contrast, SADC effectively addresses this issue by ameliorating the over-smoothed surfaces. Such
observation highlights the role of SADC in successfully densifying the sparse regions suitable for our inductive bias to assume planarity in local space.



Figure 6: Qualitative comparisons on NeRF-Synthetic (Mildenhall et al., 2021) dataset. IBGS yields the most high-quality surface reconstruction results.

	$CD\downarrow$
Ours w.o/ param.	0.702
Ours w.o/ $\mathcal{L}_{eigval}$	0.695
Ours w.o/ $\mathcal{L}_{eigvec}$	0.692
Ours w.o/ SADC	0.682
Ours	0.674

Table 4: **Ablation study** of our proposed methods on DTU. Our parameterization method significantly improves the performance, followed by the regularization techniques and SADC.



Figure 7: **Qualitative ablation results on SADC**. Without SADC, unilluminated regions tend to be over-smoothed.

## 5 CONCLUSION

528 In this work, we presented IBGS, a novel 3D Gaussian model incorporating geometrically accurate 529 parameterization and regularization. Recognizing the critical role of covariance in determining sur-530 face normals, we argued that it should be parameterized based on the distribution of neighboring 531 Gaussians for precise geometric representation rather than being entirely trainable. To achieve this, 532 we applied SVD to derive the principal axes and corresponding variances of neighboring Gaussians, 533 or the eigenvectors and eigenvalues. The model's covariance is then defined by combining these 534 eigenvectors with a learnable scaling matrix. Additionally, we introduced regularization techniques 535 that minimize the third eigenvalue, encouraging neighboring Gaussians to align on a local plane and 536 align the third eigenvectors, which approximate local plane normals, for smoother surface represen-537 tations. We also proposed SADC to densify regions with low densification signal. Our experiments on state-of-the-art benchmarks demonstrate that IBGS provides the most geometrically accurate rep-538 resentation among 3D Gaussian-based methods while requiring significantly less training time than implicit neural representations.

509 510 511

520

521

522

523

524 525 526

527

# 540 REPRODUCIBILITY STATEMENT

To ensure the reproducibility of our method, we provided all the hyperparametes we used in Section 3.4 and Section 3.5. We also provide experimental details in Section 4, pseudocode on SADC in Algorithm 1 and details on training schedules in Appendix A.2.

### 546 547 REFERENCES

542

543

544

545

548

549

550

566

567

568

572

578

579

580

581

585

586

587 588

589

590

- Dejan Azinović, Ricardo Martin-Brualla, Dan B Goldman, Matthias Nießner, and Justus Thies. Neural rgb-d surface reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6290–6301, 2022.
- Jonathan T Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and
   Pratul P Srinivasan. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 5855–5864, 2021.
- Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Mip-nerf
   360: Unbounded anti-aliased neural radiance fields. In *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*, pp. 5470–5479, 2022.
- Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Zip-nerf: Anti-aliased grid-based neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 19697–19705, 2023.
- Anpei Chen, Zexiang Xu, Andreas Geiger, Jingyi Yu, and Hao Su. Tensorf: Tensorial radiance
   fields. In *Proceedings of the European Conference on Computer Vision*, pp. 333–350. Springer,
   2022.
  - Hanlin Chen, Chen Li, and Gim Hee Lee. Neusg: Neural implicit surface reconstruction with 3d gaussian splatting guidance. *arXiv preprint arXiv:2312.00846*, 2023.
- Brian Curless and Marc Levoy. A volumetric method for building complex models from range
   images. In *Proceedings of the 23rd annual Conference on Computer Graphics and Interactive Techniques*, pp. 303–312, 1996.
- Robert A Drebin, Loren Carpenter, and Pat Hanrahan. Volume rendering. ACM SIGGRAPH Computer Graphics, 22(4):65–74, 1988.
- Sara Fridovich-Keil, Alex Yu, Matthew Tancik, Qinhong Chen, Benjamin Recht, and Angjoo Kanazawa. Plenoxels: Radiance fields without neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Pecognition*, pp. 5501–5510, 2022.
  - Antoine Guédon and Vincent Lepetit. Sugar: Surface-aligned gaussian splatting for efficient 3d mesh reconstruction and high-quality mesh rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5354–5363, 2024.
- Wenbo Hu, Yuling Wang, Lin Ma, Bangbang Yang, Lin Gao, Xiao Liu, and Yuewen Ma. Tri-miprf: Tri-mip representation for efficient anti-aliasing neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 19774–19783, 2023.
  - Binbin Huang, Zehao Yu, Anpei Chen, Andreas Geiger, and Shenghua Gao. 2d gaussian splatting for geometrically accurate radiance fields. In *ACM SIGGRAPH*, pp. 1–11, 2024.
  - Sungwon Hwang, Min-Jung Kim, Taewoong Kang, Jayeon Kang, and Jaegul Choo. Vegs: View extrapolation of urban scenes in 3d gaussian splatting using learned priors. *arXiv preprint arXiv:2407.02945*, 2024.
- Junha Hyung, Susung Hong, Sungwon Hwang, Jaeseong Lee, Jaegul Choo, and Jin-Hwa Kim.
   Effective rank analysis and regularization for enhanced 3d gaussian splatting. *arXiv preprint* arXiv:2406.11672, 2024.

594 595 596	Rasmus Jensen, Anders Dahl, George Vogiatzis, Engil Tola, and Henrik Aanæs. Large scale multi- view stereopsis evaluation. In 2014 IEEE Conference on Computer Vision and Pattern Recogni- tion, pp. 406–413. IEEE, 2014.
597 598	Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with GPUs. <i>IEEE</i> Transactions on Big Data 7(3):535–547, 2019
599 600 601	James T Kajiya and Brian P Von Herzen. Ray tracing volume densities. <i>ACM SIGGRAPH Computer</i> Graphics 18(3):165–174, 1984
602 603	Michael Kazhdan and Hugues Hoppe. Screened poisson surface reconstruction. ACM Transactions on Graphics, 32(3):1–13, 2013
604 605 606	<ul> <li>Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. ACM Transactions on Graphics, 42(4):139–1, 2023.</li> </ul>
607 608	Virginia Klema and Alan Laub. The singular value decomposition: Its computation and some appli- cations. <i>IEEE Transactions on Automatic Control</i> , 25(2):164–176, 1980.
609 610 611 612	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 <i>Proceedings of</i> <i>the IEEE/CVF Conference on Computer Vision and Pattern Recognition</i> , pp. 8456–8465, 2023.
613 614 615	William E Lorensen and Harvey E Cline. Marching cubes: A high resolution 3d surface construction algorithm. In <i>Seminal graphics: pioneering efforts that shaped the field</i> , pp. 347–353. 1998.
616 617 618	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. <i>Communications of the ACM</i> , 65(1):99–106, 2021.
619 620 621	Michael Oechsle, Songyou Peng, and Andreas Geiger. Unisurf: Unifying neural implicit surfaces and radiance fields for multi-view reconstruction. In <i>Proceedings of the IEEE/CVF International Conference on Computer Vision</i> , pp. 5589–5599, 2021.
622 623 624 625	Johannes L Schönberger, Enliang Zheng, Jan-Michael Frahm, and Marc Pollefeys. Pixelwise view selection for unstructured multi-view stereo. In <i>Proceedings of the European Conference on Computer Vision</i> , pp. 501–518. Springer, 2016.
626 627	Peter Shirley, Michael Ashikhmin, and Steve Marschner. <i>Fundamentals of computer graphics</i> . AK Peters/CRC Press, 2009.
628 629 630 631	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 <i>Advances</i> <i>in Neural Information Processing Systems</i> .
632 633 634 635	Guanjun Wu, Taoran Yi, Jiemin Fang, Lingxi Xie, Xiaopeng Zhang, Wei Wei, Wenyu Liu, Qi Tian, and Xinggang Wang. 4d gaussian splatting for real-time dynamic scene rendering. In <i>Proceedings</i> of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 20310–20320, 2024.
636 637 638 639	Ziyi Yang, Xinyu Gao, Wen Zhou, Shaohui Jiao, Yuqing Zhang, and Xiaogang Jin. Deformable 3d gaussians for high-fidelity monocular dynamic scene reconstruction. In <i>Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition</i> , pp. 20331–20341, 2024.
640 641 642	Yao Yao, Zixin Luo, Shiwei Li, Tian Fang, and Long Quan. Mvsnet: Depth inference for unstruc- tured multi-view stereo. In <i>Proceedings of the European Conference on Computer Vision</i> , pp. 767–783, 2018.
643 644 645	Lior Yariv, Jiatao Gu, Yoni Kasten, and Yaron Lipman. Volume rendering of neural implicit surfaces. <i>Advances in Neural Information Processing Systems</i> , 34:4805–4815, 2021.
646 647	Alex Yu, Ruilong Li, Matthew Tancik, Hao Li, Ren Ng, and Angjoo Kanazawa. Plenoctrees for real-time rendering of neural radiance fields. In <i>Proceedings of the IEEE/CVF International Conference on Computer Vision</i> , pp. 5752–5761, 2021.

648 649 650	Zehao Yu and Shenghua Gao. Fast-mvsnet: Sparse-to-dense multi-view stereo with learned propa- gation and gauss-newton refinement. In <i>Proceedings of the IEEE/CVF Conference on Computer</i> <i>Vision and Pattern Recognition</i> , pp. 1949–1958, 2020.
651 652 653 654	Zehao Yu, Songyou Peng, Michael Niemeyer, Torsten Sattler, and Andreas Geiger. Monosdf: Exploring monocular geometric cues for neural implicit surface reconstruction. <i>Advances in Neural Information Processing Systems</i> , 35:25018–25032, 2022.
655 656 657	Zehao Yu, Anpei Chen, Binbin Huang, Torsten Sattler, and Andreas Geiger. Mip-splatting: Alias- free 3d gaussian splatting. In <i>Proceedings of the IEEE/CVF Conference on Computer Vision and</i> <i>Pattern Recognition</i> , pp. 19447–19456, 2024a.
658 659 660	Zehao Yu, Torsten Sattler, and Andreas Geiger. Gaussian opacity fields: Efficient adaptive surface reconstruction in unbounded scenes. <i>ACM Transactions on Graphics</i> , 2024b.
661 662	Baowen Zhang, Chuan Fang, Rakesh Shrestha, Yixun Liang, Xiaoxiao Long, and Ping Tan. Rade- gs: Rasterizing depth in gaussian splatting. <i>arXiv preprint arXiv:2406.01467</i> , 2024.
663 664 665	Kai Zhang, Gernot Riegler, Noah Snavely, and Vladlen Koltun. Nerf++: Analyzing and improving neural radiance fields. <i>arXiv preprint arXiv:2010.07492</i> , 2020.
666 667	Matthias Zwicker, Hanspeter Pfister, Jeroen Van Baar, and Markus Gross. Ewa volume splatting. In <i>Proceedings Visualization, 2001. VIS'01.</i> , pp. 29–538. IEEE, 2001.
000	
670	
671	
672	
673	
674	
675	
676	
677	
678	
679	
680	
681	
682	
683	
684	
685	
686	
687	
688	
689	
690	
691	
692	
693	
694	
695	
696	
697	
698	
699	
700	
701	

#### APPENDIX А

702

703 704

705 706

708

715 716

## COMPARING $\mathcal{L}_n$ with $\mathcal{L}_{\text{Eigvec}}$



717 Figure 8: Illustrations on (*Left*) a case when vanilla 3DGS satisfy  $\mathcal{L}_n$  yet cannot converge toward the 718 underlying smooth surface, and (*Right*) how  $\mathcal{L}_{eigvec}$  induce local planes to reside on a single smooth surface by having to adjust neighboring Gaussians (K = 2) to assimilate the third covariance axis. 719 720

721 In this section, we would like to clarify a potential confusion on the working mechanism between 722  $\mathcal{L}_{eigvec}$  under our parameterization versus depth-normal consistency loss  $\mathcal{L}_n$ , a geometry regulariza-723 tion method in screen space.

724 Primarily,  $\mathcal{L}_n$  is NOT EXPLICITLY designed for smooth surfaces, but is designed to *make an* 725 agreement between rendered normal and depth. In other words, if one or all of depth/normal 726 maps are not smooth, none of the rendered normal and depth will likely be optimized to form a 727 smooth surface. For example, consider the illustration in Figure 8, where green/blue Gaussians 728 are our optimization targets, and orange/pink Gaussians are other adjacent Gaussians. Figure 8 729 shows an example of vanilla 3DGS optimized with  $\mathcal{L}_n$ . In this example, normal and depth-driven 730 normal completely agree with each other (depth-driven normal is calculated with a plane formed by 731 two closest Gaussians), but this does not guarantee the Gaussians to be converged toward a smooth underlying surface, yielding non-planar distribution of adjacent Gaussians. 732

733 On the other hand,  $\mathcal{L}_{eigvec}$  assimilates the neighboring third covariance axes (blue and green dotted 734 arrows in the right-side figure of Fig 8), which requires back-propagation of gradients to adjacent 735 Gaussians that form these covariances. Aligning covariances then require the neighboring Gaussians 736 to be aligned correspondingly such that the third covariance axes become parallel. As a result, the neighboring Gaussians are forced to form a locally smooth plane under our inductive bias. Due 737 to such working mechanism,  $\mathcal{L}_{eigvec}$  works only under our parameterization, as was empirically 738 validated in Table 4, where  $\mathcal{L}_{eigvec}$  and  $\mathcal{L}_{eigval}$  without our inductive bias performs worse. 739

740 Also, note that  $\mathcal{L}_{eigvec}$  is conducted in 3D space whereas  $\mathcal{L}_n$  regularizes rendered maps in screen 741 space. Thus, they are not strictly comparable.

742

#### 743 A.2 DETAILED TRAINING STRATEGY 744

745 Given total optimization steps I, we divide the process into multiple phases, which can be summa-746 rized in Equation 14. We learned that imposing  $\mathcal{L}_{eigval}$  and  $\mathcal{L}_{eigvec}$  from the start brings unstable 747 training. Thus, we restore coarse geometry based on our inductive bias only, and apply our regular-748 ization term afterwards. Also, when depth distortion and depth normal consistency loss is applied, we disable our parameterization by initializing a trainable rotation matrix using the rotation matrix 749 determined at step  $\frac{1}{2}$ . 750

- 751
- 752 753

$$\mathcal{L} = \begin{cases} \mathcal{L}_{c} & i \leq \frac{I}{6} \\ \mathcal{L}_{c} + \beta_{\text{eigval}} \mathcal{L}_{\text{eigval}} + \beta_{\text{eigvec}} \mathcal{L}_{\text{eigvec}} & \frac{I}{6} < i \leq \frac{I}{3} \\ \mathcal{L}_{c} + \beta_{reg} \mathcal{L}_{r} & \frac{I}{3} < i \leq \frac{I}{2} \\ \mathcal{L}_{c} + \beta_{d} \mathcal{L}_{d} + \beta_{n} \mathcal{L}_{n} & \frac{I}{2} < i \leq I \end{cases}$$
(14)

 $i < \frac{I}{2}$ 

# A.3 BAG OF REGULARIZATION TECHNIQUES FOR RESIDUAL ROTATION

It is important to keep the balance between our parameterization and expressibility when introducing residual rotation,  $\Delta R$ . To prioritize inductive bias, V is used instead of V' until step  $\frac{I}{3}$  given a total iteration step I in order to force our inductive bias in earlier stages of training, where we empirically learned that most geometry is determined. Then, we introduce  $\Delta R$  from  $\frac{I}{3}$  along with the proposed regularization term for residual rotation,  $\mathcal{L}_r$ . We may refer to these as a bag of regularization techniques for residual rotation (BRR).

To demonstrate that BRR leads to faithful reflection of inductive bias, we visualize the angles of residual rotation when trained with and without BRR in Figure 9, where without BRR, rotation residual is trained throughout all iterations without  $\mathcal{L}_r$ . As can be observed, most of the residual rotations is very close to 0° under our regularization, meaning that IBGS reasonably follows the inductive bias our parameterization forms. Also note that without BRR, chamfer distance results in 0.692 in DTU benchmark, where with BRR result in 0.674. This also implies that sticking to our parameterization leads to more accurate geometry.



Figure 9: Angles of residual rotation histograms with and without BRR. BRR effectively regularizes residual rotation, which means that IBGS well retains the inductive bias formed with our parameter-ization with eigenvectors.

786

787

788

789

790 791

792 793

794

796

801

802 803

804

780

781

# A.4 K-NEAREST NEIGHBOR CALCULATION

We used Faiss-GPU (Johnson et al., 2019) library for fast computation of K nearest neighbors. Ideally, one may calculate the kNN graph for every iteration, which also causes longer training time. Instead, we update KNN graph for every 100-th iteration to balance between computational efficiency and accuracy.

A.5 ABLATION ON K

In this section, we present the ablation study results for the parameter K on the DTU Dataset, which were conducted to determine the optimal value of K for achieving the best performance. Based on the experimental findings, we observed that setting K = 5 produced the most favorable outcomes in terms of the Chamfer Distance (CD).

K	2	3	4	5	6	7	8	9	10
CD	0.701	0.709	0.684	0.674	0.678	0.686	0.684	0.688	0.692

Table 5: Ablation results of K on DTU Dtaset Jensen et al. (2014)

# A.6 ADDITIONAL QUALITATIVE AND QUANTITATIVE RESULTS

We report additional qualitative results on DTU and NeRF-synthetic dataset in Figure 10 and Figure 11, respectively. The results demonstrate high-fidelity surface reconstructions. It is also notable that our parameterization and geometry regularization does not disturb expressing fine geometric details.

805 806 807

808

A.7 APPLICATION OF OUR METHOD TO OTHER 3DGS-BASED METHODS

In addition, we also experimened how our proposed method can improve surface reconstruction quality with other 3DGS-based methods. For that, we have applied our method on 2DGS and GOF,

0.1.0	
810	2DGS Huang et a
811	GOF Yu et al. (20
812	IBGS (Ours)
813	Table 6. Norr
814	
815	
816	2DGS Huang et al. (20 GOF Yu et al. (2024b)
817	Table 7: Quan
818	Table /: Quan
819	
820	and experimer
821	ments regardle
822	ments regardic
823	
824	
825	
820	
021	
020 820	
830	
831	
832	
833	
834	
835	
836	
837	
838	
839	
840	
841	
842	
843	
844	
845	
846	
847	
848	
849	
850	
851	
852	
853	
854	
855	
055	
050	
050	
009	
00U 961	
001	
002	

		Chair	Dr	ums	Ficus	s l	Hotdog		Lego	Ma	terials	N	/lic	Ship		Mean
2DGS Huang et al. (2024)		0.670	0.	667	0.64	3	0.795		0.685	0	724	0.	612	0.765	5	0.695
GOF Yu et al. (2024b)	(	0.673	0.	666	0.653	3	0.793		0.691	0.	722	0.	611	0.769	)	0.697
RaDe-GS Zhang et al. (2024)	(	0.686	0.	662	0.658	8	0.792		0.679	0.	.738	0.	609	0.778	3	0.700
IBGS (Ours)	(	0.689	0.	670	0.659	9	0.797		0.686	0.	.734	0.	613	0.778	3	0.703
able 6: Normal Sim	ilari	ty So	core	(NSS	) on	all sc	enes	of N	VeRF	-Syn	thetic	e Mi	ldenl	nall e	t al.	(202
2DGS Huang et al. (2024) + Ours	24	37	40	55	63	65	69	83	97	105	106	110	114	118	122	Mea
GOF Yu et al. $(2024) + Ours$	0.45	0.73	0.34	0.37	0.92	0.79	0.73	1.14	1.25	0.62	0.02	0.81	0.39	0.00	0.47	0.7
ble 7: Quantitative i	resul	lts or J Dat	app aset.	lying . As r	our	metho ted in	od usi n Tabl	ing e. 7	rende , our	ering meth	equa	tion ring:	from	n 2DC sonab	3S a	nd G nprc
ble 7: Quantitative r d experimented on l ents regardless of th	resul DTU e rei	lts or J Dat nderi	app aset. ng e	lying As r quatio	epor on.	metho ted in	od usi 1 Tabl	ing e. 7	rende , our	ering meth	equa od b	tion	from s reas	n 2DC sonab	3S a	nd C
able 7: Quantitative r ad experimented on l ents regardless of th	resul DTU e rei	lts or J Dat nderi	app aset. ng e	lying . As r quatio	epor on.	metho	od usi 1 Tabl	ing :	rende	ering meth	equa	tion	from s reas	n 2DC	3S an	nd C
able 7: Quantitative r ad experimented on l ents regardless of th	resul DTU e rei	lts or J Dat nderi	app aset. ng e	lying . As r quatio	eporton.	methe	od usi 1 Tabl	ing e. 7	rende	ering meth	equa	tion	from s reas	n 2DC	3S a	nd C
able 7: Quantitative n ad experimented on 1 ents regardless of th	nesu DTU e rei	lts or J Dat nderi	app aset. ng e	lying As r quatio	eporton.	methe	od usi 1 Tabl	ing :	rende	meth	equa	tion	from s reas	n 2DC	3S a	nd C
able 7: Quantitative r ad experimented on l ents regardless of th	DTU e rei	lts or J Dat nderi	app aset. ng e	lying As r quati	epor on.	metho	od usi 1 Tabl	ing :	rende	meth	equa	tion	from s reas	n 2DC	3S an	nd G

