Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

# Density-Normalized Contact Fields for Stable Collision Response in Gaussian Splatting

**Jong-Hyun Kim**

jonghyunkim@inha.ac.kr

Inha University

**Research Article**

**Keywords:**

**Posted Date:** November 4th, 2026

**DOI:** https://doi.org/10.21203/rs.3.rs-11219399/v1

**License:** This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

**Additional Declarations:** No competing interests reported.

# Density-Normalized Contact Fields for Stable Collision Response in Gaussian Splatting

**Jong-Hyun Kim**

Received: date / Accepted: date

**Abstract** Gaussian Splatting represents complex three-dimensional scenes as a collection of continuous anisotropic Gaussian primitives, providing high rendering quality and an efficient scene representation. However, when such Gaussian representations are directly used for collision handling, the resulting collision boundary can vary with local Gaussian density, overlap, and opacity distribution even for identical or similar geometric surfaces. This density dependence can destabilize contact positions and normals, thereby reducing the consistency of rigid-body collision responses such as penetration correction, restitution, and rotational response. In this paper, we propose a density-normalized contact field for stable collision response in Gaussian Splatting scenes. The proposed method accumulates the local contribution of each Gaussian while simultaneously estimating the effective number of Gaussians influencing a given position, thereby normalizing the accumulation bias caused by local representation density. This enables the construction of a more stable continuous contact field under Gaussian duplication or changes in local sampling density. We further use an iso-surface of the proposed contact field as the collision boundary and compute continuous contact normals from its spatial gradient, allowing the resulting contact information to be directly incorporated into rigid-body position correction and impulse-based collision response. We compare the proposed method with existing Gaussian-primitive-based collision formulations and Gaussian-overlap-based approaches under varying Gaussian densities, duplication levels, surface curvatures, and complex three-dimensional scenes. The results show that existing methods can exhibit substantial changes in collision-boundary location and contact normal as the Gaussian distribution varies, whereas the proposed method maintains a more consistent contact boundary for the same underlying geometry. Rigid-body collision experiments further demonstrate that the proposed method reduces contact-normal variation and instability in rotational response, resulting in more stable repeated collisions. These results indicate that the proposed method provides a stable contact representation for physical interaction in Gaussian Splatting scenes without requiring explicit mesh reconstruction or signed-distance-field reconstruction.

**Keywords** Gaussian Splatting · Collision detection · Collision response · Contact field · Density normalization · Rigid-body dynamics · Gaussian mixture · Physics-based interaction

Jong-Hyun Kim
College of AI and Convergence (Dept. of Artificial Intelligence, Design Technology), Graduate School of Electrical and Computer Engineering at Inha University
Tel.: +82-32-860-9204
Fax: 0504-434-8210
E-mail: jonghyunkim@inha.ac.kr

## 1 Introduction

Recently, Gaussian Splatting (GS) has emerged as an effective scene representation that provides both high visual quality and efficient rendering by representing complex three-dimensional scenes as collections of anisotropic Gaussian primitives. Each Gaussian contains a center position, covariance, opacity, and color information, and the overlap of these continuous primitives can effectively represent complex surfaces and fine structures. Owing to these properties, GS is increasingly being considered not only for novel-view synthesis but also as a scene representation for robotics, autonomous driving, navigation, and physics-based interaction.

However, directly using a GS representation constructed for rendering in physical collision handling is not straightforward. In conventional mesh-based simulation, surfaces are given as explicit geometric boundaries, allowing collision positions and surface normals to be defined relatively clearly. In contrast, a GS scene is represented by a set of spatially overlapping Gaussians rather than an explicit surface. Therefore, an additional criterion is required to define a collision boundary from the Gaussian distribution for physical contact.

Existing Gaussian-based collision or path-planning methods treat individual Gaussians as collision primitives or evaluate collision likelihood using geometric intersection between Gaussians, probabilistic overlap, or Gaussian overlap. Although these approaches can be effective for applications such as robot navigation and collision avoidance, additional problems arise in GS scenes optimized for rendering. In GS, the Gaussian distribution does not correspond solely to geometric sampling of the actual surface, and the number, size, opacity, and degree of overlap of Gaussians can vary substantially across regions during scene reconstruction.

This nonuniformity has important implications for collision computation. Even when representing the same physical surface, some regions may use a small number of large Gaussians, whereas other regions may contain many densely distributed small Gaussians. If the collision boundary is defined by the extent of individual Gaussians or by a simple accumulation of Gaussian contributions, the computed collision boundary can shift with local splat density even when the actual surface location is unchanged. In particular, directly summing overlapping Gaussian contributions increases the field value in dense regions and expands the collision region outward, whereas sparse regions may exhibit a contracted or discontinuous collision region.

Such inconsistency in the collision boundary is not limited to collision-detection error. In rigid-body simulation, collision positions and surface normals are directly used for penetration correction, restitution impulses, friction, and angular-momentum computation. Consequently, if the contact boundary or normal changes from frame to frame because of splat-density variation, a rigid body may penetrate the surface excessively or rebound earlier than expected, and repeated contact may introduce unnecessary rotation and vibration. Addressing this issue is essential for using GS scenes in real-time physical interaction.

To address this problem, we propose a *density-normalized contact field*. The primary goal of the proposed method is to provide a contact boundary that remains as consistent as possible for the same geometric structure even when the local splat density of the GS representation changes. Instead of simply accumulating Gaussian contributions at each position, we normalize the accumulated contact value by also accounting for the effective number of Gaussians that meaningfully contribute at that position. The iso-surface of the resulting continuous contact field is used as the collision boundary, and the spatial gradient of the field is used to compute the contact normal for direct use in rigid-body collision response.

This work does not target only collision avoidance or path planning; rather, it aims to stably handle repeated physical contact and collision response between GS scenes and dynamic objects. To validate the proposed approach, we analyze changes in collision boundaries and normals in scenes with different Gaussian densities, duplication levels, and curvatures, and compare the results with existing Gaussian-primitive-based collision and Gaussian-overlap-based methods. We also evaluate contact-normal variation and the stability of rotational response by colliding actual rigid bodies with GS scenes.

The main contributions of this paper are as follows.

- We analyze how local splat density and overlap in GS scenes affect collision boundaries and contact normals, and formulate this issue as a density-dependent contact problem in physics-based interaction.
- We propose a density-normalized contact field that mitigates density bias by accounting for the number of locally contributing Gaussians, thereby producing more consistent collision boundaries under changes in splat density and duplication.
- We present an integrated framework that computes contact normals from the spatial gradient of the proposed continuous contact field and directly applies them to penetration correction and impulse-based rigid-body collision response.
- We validate improvements in collision-boundary consistency, contact-normal stability, and rigid-body collision response by comparing the proposed method with existing Gaussian-based collision methods under various density and duplication conditions and in complex GS scenes.

## 2 Problem Statement

The objective of this work is to define a consistent physical contact boundary between a static scene represented by Gaussian Splatting and a dynamic rigid body. We represent a GS scene as a set of $N$ anisotropic Gaussian primitives

$$\mathcal{G} = \{\mathcal{G}_i = (\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i, \alpha_i)\}_{i=1}^{N} \tag{1}$$

where $\boldsymbol{\mu}_i$, $\boldsymbol{\Sigma}_i$, and $\alpha_i$ denote the center, covariance, and opacity of the $i$th Gaussian, respectively. Let $w_i(\mathbf{x})$ denote the local contribution of $\mathcal{G}_i$ at a spatial position $\mathbf{x}$. The specific definition of this contribution is given in Sec. 4.

A common approach for constructing a collision boundary from a GS scene is to aggregate the contributions of multiple Gaussians into a scalar field $F(\mathbf{x})$ and use the iso-surface corresponding to a global threshold $\tau$,

$$\mathcal{S}_\tau = \{\mathbf{x} \mid F(\mathbf{x}) = \tau\} \tag{2}$$

as the contact boundary. However, if this field relies on the simple accumulation

$$F_{\text{sum}}(\mathbf{x}) = \sum_{i=1}^{N} w_i(\mathbf{x}) \tag{3}$$

its value changes with local splat multiplicity even when the underlying geometry remains unchanged.

This can be illustrated by the simplest duplication case. Suppose a Gaussian with the same position and parameters is duplicated $m$ times and each Gaussian contributes $w(\mathbf{x})$.

$$F^{(m)}_{\text{sum}}(\mathbf{x}) = m\, w(\mathbf{x}), \qquad F^{(m)}_{\text{sum}}(\mathbf{x}) \neq F^{(1)}_{\text{sum}}(\mathbf{x}) \quad (m > 1). \tag{4}$$

Thus, although the physical surface remains unchanged, the contact boundary extracted using the same $\tau$ can shift. We refer to this phenomenon as the *density-dependent contact boundary* problem.

More generally, for two GS representations $\mathcal{G}^{(a)}$ and $\mathcal{G}^{(b)}$ describing the same or similar physical surface, a density-robust contact representation should preserve

$$\mathcal{S}^{(a)}_\tau \approx \mathcal{S}^{(b)}_\tau \tag{5}$$

despite differences in local Gaussian count, size, and overlap. In addition to boundary location, the associated differential information should remain sufficiently stable for repeated contact because the contact normal directly affects penetration correction and impulse response.

Accordingly, the contact representation required in this work should satisfy the following conditions.

1. Density robustness: The contact boundary should exhibit only small displacement when the splat density changes for the same or a similar geometric surface.
2. Duplication robustness: The contact boundary should be preserved as much as possible under Gaussian duplication that does not alter the underlying geometry.
3. Spatial continuity: The scalar field defining the contact boundary should be spatially continuous and provide stable differential information.
4. Normal consistency: The contact normal should respond primarily to geometric changes in the actual surface rather than irregularities in the splat distribution.
5. Physical usability: The computed contact positions and normals should be directly usable for repeated rigid-body collision response beyond simple collision avoidance.

Therefore, the central problem addressed in this work is not merely to determine whether two Gaussian primitives intersect, but to *define a consistent continuous contact boundary from a nonuniform Gaussian Splatting representation while mitigating the influence of representation density.* After reviewing related work, Sec. 4 presents the proposed density-normalized contact field.

## 3 Related Work

### 3.1 Gaussian Splatting and Geometric Surface Representation

3D Gaussian Splatting (3DGS) represents a scene as a collection of three-dimensional Gaussian primitives with position, anisotropic covariance, opacity, and color attributes, and enables real-time novel-view synthesis by combining an explicit primitive-based representation with fast differentiable rasterization [1]. However, because 3DGS Gaussians are inherently optimized to minimize rendering loss, the centers and covariances of individual primitives are not guaranteed to coincide directly with the physical surface of the scene. This property has become an important issue in subsequent work that seeks to extract geometric structure directly from Gaussian representations.

SuGaR introduced regularization to align Gaussians with the actual scene surface and generated meshes by applying Poisson reconstruction to the aligned Gaussians [2]. 2D Gaussian Splatting improves multi-view geometric consistency by using oriented two-dimensional Gaussian disks as surface primitives instead of three-dimensional volumetric Gaussians, and enables more accurate surface reconstruction through depth-distortion and normal-consistency regularization [3]. Gaussian Opacity Fields (GOF) defines a continuous opacity field from Gaussian-based volume rendering, directly extracts surfaces from its level set, and performs efficient geometry extraction using an adaptive tetrahedral structure [4]. PGSR improves the geometric fidelity of Gaussian representations using planar Gaussians and unbiased depth

rendering, while strengthening single-view and multi-view geometric consistency [5]. NeuSG jointly optimizes 3DGS and a neural implicit surface to encourage Gaussian centers to lie closer to the actual surface and uses normal information from the implicit surface for detailed geometry reconstruction [6].

These studies demonstrate that 3DGS learned for rendering does not necessarily provide an accurate physical surface and propose various methods for extracting surfaces or level sets from Gaussians. However, their primary objectives are mesh reconstruction, geometric fidelity, or novel-view synthesis, and they do not directly address how local splat-density variation affects *dynamic contact boundaries and collision response*.

### 3.2 Gaussian Splatting for SLAM and Online Mapping

The explicit and efficient scene representation of 3DGS has also been actively adopted in recent SLAM and real-time mapping research. SplaTAM simultaneously performs tracking and mapping while generating 3D Gaussians online from RGB-D input, and adds Gaussians to newly observed regions using silhouette-based scene density [7]. GS-SLAM combines adaptive Gaussian expansion with coarse-to-fine tracking to realize real-time dense visual SLAM [8]. Gaussian Splatting SLAM by Matsuki *et al.* directly applies 3DGS to monocular input and integrates tracking, mapping, and rendering into a unified Gaussian representation [9]. Photo-SLAM combines explicit geometric features with Gaussian-based photorealistic mapping to provide a real-time SLAM system for monocular, stereo, and RGB-D inputs [10].

Gaussian-SLAM partitions the Gaussian representation into multiple sub-maps for scalable dense mapping of large scenes and performs frame-to-model camera tracking using photometric and geometric rendering losses [11]. RTG-SLAM distinguishes opaque and transparent Gaussians and retains only stable Gaussians to achieve real-time reconstruction of large-scale RGB-D scenes [12]. Splat-SLAM performs global optimization in RGB-only environments by deforming the Gaussian map according to global-pose and depth updates [13]. GauS-SLAM uses 2D Gaussian surfels to reduce geometric distortion caused by depth blending and performs surface-aware tracking [14]. MGSO uses a dense structured point cloud obtained from monocular photometric SLAM for Gaussian initialization, reducing both the number of Gaussians and optimization cost [15]. Hier-SLAM combines a hierarchical categorical representation with the Gaussian map to extend it to large-scale semantic SLAM [16].

These studies show that Gaussian representations can be used for real-time environmental mapping beyond offline rendering. Of particular relevance to this work is that Gaussian creation, removal, and densification continuously change according to observation conditions and the mapping process. Thus, even for the same physical surface, the number, size, and overlap of Gaussians may vary spatially during SLAM, and directly using such representation-level variations for physical contact can make the contact boundary dependent on scene-representation density.

Dense SLAM based on continuous implicit representations was also actively studied before 3DGS. NICE-SLAM performs scalable implicit mapping using a hierarchical neural feature grid [17], while Vox-Fusion combines a voxel-based neural implicit representation with an octree [18]. Co-SLAM combines coordinate encoding with sparse parametric encoding for real-time RGB-D SLAM [19], while Point-SLAM uses a neural point cloud to jointly represent dense geometry and appearance [20]. NeRF-SLAM combines dense monocular SLAM with neural radiance fields for real-time photometric and geometric reconstruction [21]. These studies demonstrate that continuous scene fields are useful not only for rendering but also for mapping and geometric reasoning; however, their objectives differ from the Gaussian density-dependent contact-boundary problem considered in this work.

### 3.3 Collision Checking and Navigation in Continuous Gaussian Representations

The use of Gaussian mixture representations for mapping and collision checking predates 3DGS. Dhawale, Yang, and Michael construct a local Gaussian mixture map from depth observations and perform trajectory collision checking using the geometric properties of the Gaussians [22]. GMMap compresses RGB-D observations into a Gaussian mixture model to represent continuous occupancy probability, providing a memory-efficient three-dimensional occupancy map [23]. In these approaches, Gaussians are primarily used as mapping primitives representing physical occupancy or sensor uncertainty.

More recently, neural radiance fields and 3DGS have been directly applied to robot navigation. CATNIPS converts a NeRF into a Poisson point process to compute robot collision probabilities in continuous space and performs chance-constrained trajectory optimization using probabilistically unsafe robot regions [24]. iSDF learns a real-time neural signed distance field that provides a continuous geometric representation from which a robot can query scene distance and collision information [25].

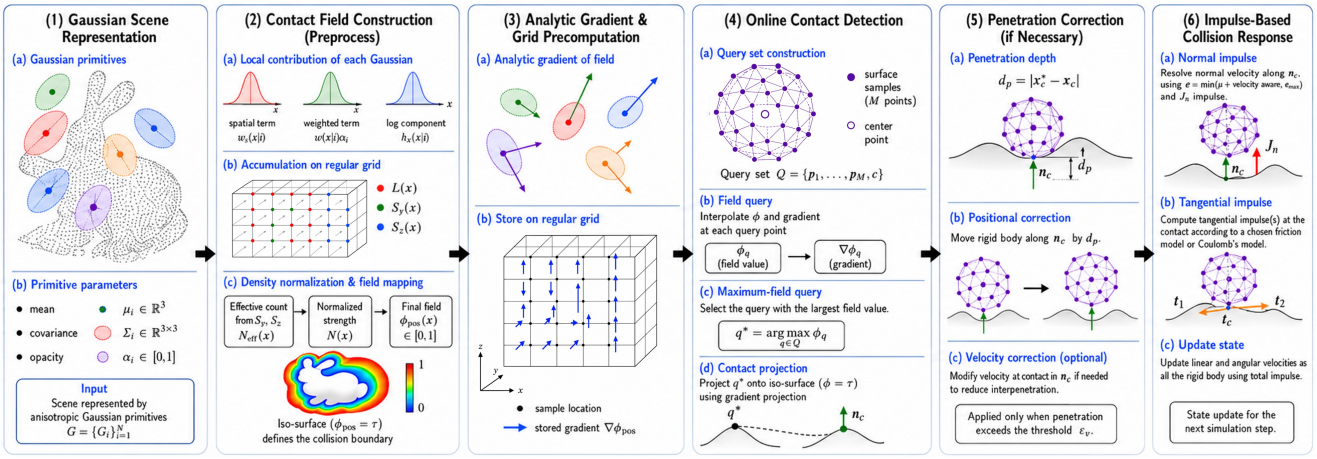



Fig. 1: Overview of the proposed density-normalized contact framework. The static Gaussian scene is first represented by anisotropic Gaussian primitives. A density-normalized contact field and its analytic gradient are then precomputed on a regular grid. During runtime, surface samples and the rigid-body center are queried against the contact field, and the maximum-field query is projected onto the contact iso-surface to obtain the contact point and outward normal. If penetration is detected, positional correction is applied along the contact normal, followed by normal and tangential impulse-based collision response.

Among approaches that directly use 3DGS, SplatNav treats each 3D Gaussian as an ellipsoidal obstacle primitive and plans trajectories by constructing safe convex corridors from geometric collision constraints between the robot and Gaussians [26]. SAFER-Splat defines a control barrier function over an online-learned Gaussian Splat map and performs real-time safety filtering while considering hundreds of thousands of Gaussians [27]. SPLANNING computes an upper bound on collision probability between a robot and a radiance field from a normalized Gaussian Splat representation and uses it for risk-aware trajectory optimization [28]. FOCI represents both the environment and robot as Gaussian mixtures and performs orientation-aware trajectory optimization using analytic overlap between the two Gaussian mixtures [29]. GRaD-Nav integrates a Gaussian radiance field with differentiable robot dynamics to learn a visual drone-navigation policy [30].

These studies demonstrate that Gaussian representations can support collision-aware navigation without conversion to a separate dense mesh. Their primary objective, however, is safe trajectory planning rather than repeated rigid-body contact response.

### 3.4 Relation to Our Work

Existing primitive-intersection [26], Gaussian-overlap [29], and probabilistic collision formulations [28,24] define useful collision or safety criteria, but do not explicitly account for contact-boundary shifts caused by nonuniform splat density or multiplicity in rendering-oriented 3DGS scenes. We instead treat representation density as a nuisance factor in repeated physical contact and construct a continuous field whose iso-surface and gradient provide the contact boundary and normal used by the rigid-body response. The method is therefore complementary to Gaussian-based navigation and occupancy approaches rather than a replacement for them.

## 4 Algorithm

This section presents the contact pipeline for a nonuniform Gaussian Splatting (GS) scene. It consists of six stages: (1) Gaussian scene representation, (2) density-normalized contact-field construction, (3) analytic-gradient and grid precomputation, (4) online contact detection and projection, (5) penetration correction, and (6) impulse-based collision response. The overall processing pipeline is illustrated in Fig. 1.

The central operation normalizes accumulated Gaussian contributions by an effective local splat count, limiting field inflation caused by density or multiplicity changes. For a static GS scene, the field and analytic gradient are precomputed on a regular grid. At runtime, the maximum-field rigid-body query is projected onto the contact iso-surface to obtain a contact point and outward normal, followed by penetration correction and normal/tangential impulse response.

### 4.1 Gaussian Contribution

Let the $i$th Gaussian primitive composing the scene be $\mathcal{G}_i = (\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i, \alpha_i)$. The Mahalanobis distance at a spatial position $\mathbf{x}$ is defined as

$$q_i(\mathbf{x}) = (\mathbf{x} - \boldsymbol{\mu}_i)^T \boldsymbol{\Sigma}_i^{-1} (\mathbf{x} - \boldsymbol{\mu}_i). \tag{6}$$

The corresponding pure spatial contribution of the Gaussian is

$$s_i(\mathbf{x}) = \exp\left(-\frac{1}{2} q_i(\mathbf{x})\right) \tag{7}$$

where $s_i(\mathbf{x}) \in (0, 1]$. It has a value of 1 at the Gaussian center and decreases continuously with distance from the center.

Because each Gaussian in 3DGS has a different opacity, the actual contact contribution includes the opacity $\alpha_i$ as

$$w_i(\mathbf{x}) = \alpha_i s_i(\mathbf{x}) \tag{8}$$

which is used throughout the formulation.

However, directly summing $w_i$ causes the value to grow without bound as the number of overlapping Gaussians increases. To avoid this behavior, we transform the contribution of each Gaussian into log-complement space.

$$\ell_i(\mathbf{x}) = -\log\left(1 - w_i(\mathbf{x})\right). \tag{9}$$

For numerical stability, the implementation clamps

$$w_i \leftarrow \min\left(w_i, 1 - \epsilon_w\right) \tag{10}$$

where $\epsilon_w$ is a small positive constant.

The accumulated contribution over all Gaussians is defined as

$$L(\mathbf{x}) = \sum_{i=1}^{N} \ell_i(\mathbf{x}) = \sum_{i=1}^{N} -\log(1 - w_i(\mathbf{x})) \tag{11}$$

which gives the total log-space contribution.

Equation (11) satisfies the following relation.

$$1 - \exp[-L(\mathbf{x})] = 1 - \prod_{i=1}^{N} \left(1 - w_i(\mathbf{x})\right). \tag{12}$$

Thus, $L$ provides a stable accumulation of union-like contributions from multiple Gaussians, but it still increases with the number of Gaussians activated at the same position. We therefore introduce local normalization in the next step to mitigate this density dependence.

### 4.2 Effective Local Splat Count

Defining local Gaussian density simply as the number of Gaussians within a fixed radius cannot properly account for differences in Gaussian scale and anisotropy. We therefore compute an effective splat count from the relative spatial contributions of the Gaussians that actually influence position $\mathbf{x}$.

We first define the following two statistics.

$$S_1(\mathbf{x}) = \sum_{i=1}^{N} s_i(\mathbf{x}), \tag{13}$$

$$S_2(\mathbf{x}) = \sum_{i=1}^{N} s_i^2(\mathbf{x}). \tag{14}$$

Using these quantities, the effective local splat count is defined as

$$N_{\text{eff}}(\mathbf{x}) = \frac{S_1^2(\mathbf{x})}{S_2(\mathbf{x}) + \epsilon} \tag{15}$$

.

In the implementation, we use

$$\widehat{N}_{\text{eff}}(\mathbf{x}) = \max\left(1, N_{\text{eff}}(\mathbf{x})\right) \tag{16}$$

to maintain a numerically meaningful range.

Equation (15) has a form analogous to effective sample size. It approaches 1 when a single Gaussian dominates a position and increases approximately in proportion to the number of Gaussians when multiple Gaussians contribute with similar magnitudes.

For example, assume that $m$ Gaussians with identical spatial contribution $s$ overlap at the same position.

$$S_1 = ms, \qquad S_2 = ms^2 \tag{17}$$

Then,

$$N_{\text{eff}} = \frac{m^2 s^2}{ms^2} = m \tag{18}$$

Thus, the formulation continuously estimates the Gaussian multiplicity that actually affects the current position rather than simply counting nearby Gaussians.

Opacity $\alpha_i$ is intentionally excluded from the computation of $N_{\text{eff}}$, and only the spatial term $s_i$ is used. Even a low-opacity Gaussian should contribute to the local representation density if it redundantly represents the same geometric location. In contrast, opacity is retained in the actual contact strength through Eq. (8).

### 4.3 Density-Normalized Contact Field

Combining the accumulated Gaussian contribution $L$ with the effective count $\widehat{N}_{\text{eff}}$, we define the proposed density-normalized contact strength as

$$A(\mathbf{x}) = \frac{L(\mathbf{x})}{\widehat{N}_{\text{eff}}^{\gamma}(\mathbf{x})}, \tag{19}$$

where $\gamma \in [0, 1]$ controls the strength of density normalization.

The final contact field is defined as

$$\Phi_{\text{DN}}(\mathbf{x}) = 1 - \exp\left[-\frac{L(\mathbf{x})}{\widehat{N}_{\text{eff}}^{\gamma}(\mathbf{x})}\right] \tag{20}$$

.

$\Phi_{\text{DN}}$ has the bounded range $0 \leq \Phi_{\text{DN}}(\mathbf{x}) < 1$. The exponential mapping limits the numerical range of the field, while robustness to splat density is provided by the $\widehat{N}_{\text{eff}}^{\gamma}$ normalization in Eq. (19).

We define the iso-surface of $\Phi_{\text{DN}}$,

$$\mathcal{C} = \{\mathbf{x} \mid \Phi_{\text{DN}}(\mathbf{x}) = \tau\} \tag{21}$$

as the physical contact boundary, with a default threshold of $\tau = 0.5$.

#### *4.3.1 Behavior under Gaussian Duplication*

To clarify the behavior of the proposed normalization, consider a Gaussian duplicated $m$ times.

Let $\ell$ denote the log contribution of a single Gaussian.

$$L^{(m)} = m\ell \tag{22}$$

and, from Eq. (18),

$$N_{\text{eff}}^{(m)} = m \tag{23}$$

Therefore, the normalized strength becomes

$$A^{(m)} = \frac{m\ell}{m^{\gamma}} = m^{1-\gamma}\ell. \tag{24}$$

In particular, when $\gamma = 1$, $A^{(m)} = \ell = A^{(1)}$, yielding exact theoretical invariance under duplication of an identical Gaussian. In contrast, when $\gamma = 0$, $A^{(m)} = m\ell$, which has the same density dependence as accumulation without density normalization.

We use $\gamma = 0.85$ as the default to substantially compensate for multiplicity while retaining some responsiveness to changes in local overlap rather than enforcing complete normalization. Under this setting, the normalized strength scales as $m^{0.15}$ with Gaussian duplication, and the sensitivity to $\gamma$ is analyzed separately in the experiments.

| Duplication $m$ | Sum | DN ($\gamma = 0.85$) | DN ($\gamma = 1.0$) |
|---|---|---|---|
| 1 | 1.00 | 1.00 | 1.00 |
| 2 | 2.00 | 1.11 | 1.00 |
| 4 | 4.00 | 1.23 | 1.00 |
| 8 | 8.00 | 1.37 | 1.00 |
| 16 | 16.00 | 1.52 | 1.00 |

Table 1: Theoretical relative accumulation magnitude under Gaussian duplication. While the simple accumulation scheme increases proportionally to the duplication count $m$, the proposed method with the default setting $\gamma = 0.85$ increases only proportionally to $m^{0.15}$.

Table 1 reports the relative accumulation magnitude under Gaussian duplication. These values are theoretical scaling factors computed from Eq. (24), not experimental measurements.

This behavior suppresses large shifts in the contact boundary caused solely by Gaussian multiplicity when the same surface is represented by different numbers of splats.

### 4.4 Analytic Gradient of the Contact Field

Rigid-body collision response requires not only the contact-field value but also a stable surface normal. Rather than computing a separate finite difference, we accumulate the analytic gradient together with the field during construction.

The gradient of Eq. (7) is

$$\nabla s_i = -s_i \boldsymbol{\Sigma}_i^{-1}(\mathbf{x} - \boldsymbol{\mu}_i) \tag{25}$$

.

Therefore,

$$\nabla w_i = \alpha_i \nabla s_i \tag{26}$$

and, from Eq. (9),

$$\nabla \ell_i = \frac{\nabla w_i}{1 - w_i} \tag{27}$$

is obtained.

The gradient of the accumulated term is

$$\nabla L = \sum_i \nabla \ell_i \tag{28}$$

.

We also compute

$$\nabla S_1 = \sum_i \nabla s_i \tag{29}$$

and

$$\nabla S_2 = \sum_i 2 s_i \nabla s_i \tag{30}$$

.

Letting $D = S_2 + \epsilon$, the gradient of Eq. (15) is

$$\nabla N_{\text{eff}} = \frac{2S_1 D \nabla S_1 - S_1^2 \nabla S_2}{D^2}. \tag{31}$$

When the lower clamp in Eq. (16) is active, i.e., when $N_{\text{eff}} \leq 1$,

$$\nabla \widehat{N}_{\text{eff}} = \mathbf{0} \tag{32}$$

is used.

The gradient of Eq. (19) is then

$$\nabla A = \frac{\nabla L}{\widehat{N}_{\text{eff}}^{\gamma}} - \frac{\gamma L}{\widehat{N}_{\text{eff}}^{\gamma+1}} \nabla \widehat{N}_{\text{eff}}. \tag{33}$$

Finally, from Eq. (20),

$$\nabla \Phi_{\text{DN}} = \exp(-A) \nabla A \tag{34}$$

is obtained.

Because $\Phi_{\text{DN}}$ increases toward the interior of the Gaussian representation, the outward-facing contact normal is defined as

$$\mathbf{n}(\mathbf{x}) = -\frac{\nabla \Phi_{\text{DN}}(\mathbf{x})}{\|\nabla \Phi_{\text{DN}}(\mathbf{x})\| + \epsilon_n} \tag{35}$$

.

Because the field and analytic-gradient samples at each grid vertex are computed from the same Gaussian contributions, no separate mesh normal or nearest-neighbor normal estimation is required. At runtime, these analytic-gradient samples are trilinearly interpolated to compute normals at continuous positions.

### 4.5 GPU Field Construction

Directly evaluating Eq. (20) for every Gaussian at every time step of a dynamic simulation would require $O(NM)$ computation for $N$ Gaussians and $M$ rigid-body query points. To avoid this cost, the proposed method precomputes the contact field and gradient on the GPU for a static GS scene.

The three-dimensional simulation domain $\Omega = [0, L_x] \times [0, L_y] \times [0, L_z]$ is discretized into a regular grid of $N_x \times N_y \times N_z$. At each grid vertex $\mathbf{x}_{ijk}$, we accumulate $\{L, S_1, S_2, \nabla L, \nabla S_1, \nabla S_2\}$.

On the GPU, one thread is assigned to each Gaussian, and values are accumulated only at grid vertices inside the effective support of that Gaussian. An axis-aligned support interval is first computed from the world-space covariance of the Gaussian, avoiding traversal of the entire grid.

In the current implementation, a compact axis-aligned support extending approximately $3.5\sigma$ along each world-space axis is first constructed from the covariance. Within this bounding region, the Mahalanobis-distance test $q_i(\mathbf{x}) \leq 36$ is additionally applied to reject grid vertices outside the anisotropic ellipsoidal support. Thus, the axis-aligned bound provides a compact traversal region, while the Mahalanobis test accounts for the Gaussian's anisotropic shape.

After all Gaussian contributions have been accumulated, a separate CUDA kernel evaluates Eqs. (16)–(34) at each grid vertex and stores the final $\Phi_{\text{DN}}$ and $\nabla \Phi_{\text{DN}}$.

Consequently, after preprocessing, the field can be queried efficiently at arbitrary positions regardless of the total number of Gaussians.

### 4.6 Continuous Field Query

Although the field is computed on a regular grid, collision queries should not be restricted to voxel locations. We therefore use trilinear interpolation to evaluate the field and gradient at an arbitrary continuous position $\mathbf{x}$.

Let the field values at the eight grid vertices surrounding $\mathbf{x}$ be $\Phi_{abc}$, with $a, b, c \in \{0, 1\}$.

$$\widetilde{\Phi}(\mathbf{x}) = \sum_{a=0}^{1} \sum_{b=0}^{1} \sum_{c=0}^{1} \lambda_{abc}(\mathbf{x}) \Phi_{abc}, \tag{36}$$

Here, $\lambda_{abc}$ denotes the trilinear interpolation weight.

The analytically evaluated gradient samples stored at the grid vertices are interpolated in the same manner:

$$\widetilde{\mathbf{g}}(\mathbf{x}) = \sum_{a=0}^{1} \sum_{b=0}^{1} \sum_{c=0}^{1} \lambda_{abc}(\mathbf{x}) \mathbf{g}_{abc} \tag{37}$$

where $\widetilde{\mathbf{g}}(\mathbf{x})$ is the interpolated analytic-gradient sample stored on the grid. In general, we do not assume that it is exactly equal to the derivative of the trilinearly interpolated field $\widetilde{\Phi}$.

Thus, the cost of a field query at one point during runtime is $O(1)$ and is independent of the number of Gaussians $N$.

### 4.7 Rigid-Body Contact Detection

To support a rigid body $\mathcal{B}$ of arbitrary shape, we select $M$ local sample points $\{\mathbf{p}_j^L\}_{j=1}^M$ on the object surface. The current implementation uses at most $M = 512$ surface samples.

Let $\mathbf{c}_t$ and $\mathbf{R}_t$ denote the translation and rotation of the rigid body at time $t$, respectively. A world-space sample is then

$$\mathbf{p}_j = \mathbf{R}_t \mathbf{p}_j^L + \mathbf{c}_t \tag{38}$$

.

The surface samples and rigid-body center are grouped into a single query set,

$$\mathcal{Q} = \{\mathbf{p}_1, \ldots, \mathbf{p}_M, \mathbf{c}_t\} \tag{39}$$

The rigid-body center is included as an interior query to additionally detect extreme cases in which the object becomes fully embedded in a thin or complex contact field. For each $\mathbf{q} \in \mathcal{Q}$, we query $\widetilde{\Phi}(\mathbf{q})$, and a collision is detected when

$$\widetilde{\Phi}(\mathbf{q}) > \tau \tag{40}$$

.

When multiple queries lie inside the contact field simultaneously, we select the maximum-field query at the current iteration as

$$\mathbf{q}^\star = \arg\max_{\mathbf{q} \in \mathcal{Q}} \widetilde{\Phi}(\mathbf{q}) \tag{41}$$

This definition allows surface samples and the rigid-body center to be processed using the same criterion.

### 4.8 Projection onto the Contact Iso-Surface

The maximum-field query $\mathbf{q}^\star$ generally already lies inside the contact field and is therefore projected onto the actual contact boundary $\Phi_{\mathrm{DN}}(\mathbf{x}) = \tau$. Starting from $\mathbf{x}^{(0)} = \mathbf{q}^\star$, we iteratively apply the following gradient projection.

$$f^{(k)} = \widetilde{\Phi}\left(\mathbf{x}^{(k)}\right) - \tau, \tag{42}$$

$$\mathbf{g}^{(k)} = \widetilde{\mathbf{g}}\left(\mathbf{x}^{(k)}\right). \tag{43}$$

The update is

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - \eta_k \mathbf{g}^{(k)} \tag{44}$$

where

$$\eta_k = \mathrm{clamp}\left(\frac{f^{(k)}}{\|\mathbf{g}^{(k)}\|^2}, -\eta_{\max}, \eta_{\max}\right). \tag{45}$$

The current implementation uses at most 14 iterations with $\eta_{\max} = 0.12$.

Projection terminates when the following condition is satisfied.

$$\left|\widetilde{\Phi}(\mathbf{x}^{(k)}) - \tau\right| < 10^{-5}. \tag{46}$$

Letting the final projected point be $\mathbf{x}_c = \mathbf{x}^{(k)}$, the contact normal is computed as

$$\mathbf{n}_c = -\frac{\widetilde{\mathbf{g}}(\mathbf{x}_c)}{\|\widetilde{\mathbf{g}}(\mathbf{x}_c)\| + \epsilon_n} \tag{47}$$

.

### 4.9 Penetration Correction

The distance between the maximum-field query and the projected surface point is used as an approximation of the penetration depth.

$$d_p = \|\mathbf{x}_c - \mathbf{q}^\star\|. \tag{48}$$

To avoid numerical residual penetration, the correction magnitude in one iteration is clamped as

$$\widehat{d}_p = \mathrm{clamp}\left(d_p, d_{\min}, d_{\max}\right) \tag{49}$$

.

The current implementation uses $d_{\min} = 0.001$ and $d_{\max} = 0.08$. Here, $d_{\min}$ acts as a numerical separation bias for removing very small residual penetration, whereas $d_{\max}$ limits excessive positional correction within a single pass.

The rigid-body center position is corrected as

$$\mathbf{c} \leftarrow \mathbf{c} + \widehat{d}_p \mathbf{n}_c \tag{50}$$

.

To handle cases in which a rigid body simultaneously intersects multiple contact regions, this detection–projection–correction procedure is repeated. The current implementation performs at most 10 correction passes within one physics step.

### 4.10 Impulse-Based Collision Response

To isolate the effect of the contact representation, we apply the same rigid-body response model to all collision methods. Let the contact point be $\mathbf{x}_c$ and the rigid-body center be $\mathbf{c}$. Then,

$$\mathbf{r} = \mathbf{x}_c - \mathbf{c} \tag{51}$$

.

Let $\mathbf{v}$ and $\boldsymbol{\omega}$ denote the linear and angular velocities of the center of mass, respectively. The velocity at the contact point is

$$\mathbf{v}_c = \mathbf{v} + \boldsymbol{\omega} \times \mathbf{r} \tag{52}$$

.

The normal velocity is

$$v_n = \mathbf{v}_c \cdot \mathbf{n}_c \tag{53}$$

and if

$$v_n \geq 0 \tag{54}$$

the object is separating from the surface, and no normal impulse is applied.

In the current implementation, we use the diagonal inertia $(I_x, I_y, I_z)$ to define the mean inverse inertia

$$\bar{I}^{-1} = \frac{1}{3}\left(\frac{1}{I_x} + \frac{1}{I_y} + \frac{1}{I_z}\right) \tag{55}$$

This approximation is used because our evaluation focuses on differences in the contact representation itself. The same simplified inertia approximation is applied to all compared methods so that differences in the collision solver do not affect the comparison.

The normal impulse magnitude is

$$j_n = -\frac{(1+e)v_n}{m^{-1} + \bar{I}^{-1}\|\mathbf{r} \times \mathbf{n}_c\|^2} \tag{56}$$

where $e$ is the coefficient of restitution.

The normal impulse is

$$\mathbf{J}_n = j_n \mathbf{n}_c \tag{57}$$

and the linear and angular velocities are updated as

$$\mathbf{v} \leftarrow \mathbf{v} + \frac{\mathbf{J}_n}{m}, \tag{58}$$

$$\boldsymbol{\omega} \leftarrow \boldsymbol{\omega} + \bar{I}^{-1}(\mathbf{r} \times \mathbf{J}_n) \tag{59}$$

.

*4.10.1 Friction*

After applying the normal impulse, we recompute the contact-point velocity using the updated linear and angular velocities.

$$\mathbf{v}_c' = \mathbf{v}' + \boldsymbol{\omega}' \times \mathbf{r}. \tag{60}$$

The corresponding tangential component is

$$\mathbf{v}_t = \mathbf{v}_c' - (\mathbf{v}_c' \cdot \mathbf{n}_c)\mathbf{n}_c \tag{61}$$

If $\|\mathbf{v}_t\| > 0$, the tangent direction is $\mathbf{t} = \mathbf{v}_t / \|\mathbf{v}_t\|$.

The friction impulse is

$$j_t = -\frac{\mathbf{v}_c' \cdot \mathbf{t}}{m^{-1} + \bar{I}^{-1}\|\mathbf{r} \times \mathbf{t}\|^2} \tag{62}$$

and is clamped according to the Coulomb friction condition as

$$j_t \leftarrow \text{clamp}\,(j_t, -\mu|j_n|, \mu|j_n|) \tag{63}$$

The resulting tangent impulse $\mathbf{J}_t = j_t \mathbf{t}$ is then applied to the linear and angular velocities.

**Algorithm 1** Density-Normalized Contact Field Construction

**Require:** Gaussian set $\mathcal{G}$, grid $\Omega$, $\gamma$, $\epsilon$, $\epsilon_w$
**Ensure:** $\Phi_{\text{DN}}$, $\nabla\Phi_{\text{DN}}$
Initialize $L$, $S_1$, $S_2$, $\nabla L$, $\nabla S_1$, and $\nabla S_2$
**for** each Gaussian $\mathcal{G}_i$ in parallel **do**
  Compute the bounded grid support of $\mathcal{G}_i$
  **for** each grid point $\mathbf{x}$ in the support **do**
    Evaluate $s_i$, $w_i$, $\ell_i$, $\nabla s_i$, and $\nabla \ell_i$
    Atomically accumulate $L$, $S_1$, $S_2$
    Atomically accumulate $\nabla L$, $\nabla S_1$, $\nabla S_2$
  **end for**
**end for**
**for** each grid point $\mathbf{x}$ in parallel **do**
  Compute $\widehat{N}_{\text{eff}}$ using Eq. (16)
  Compute $\Phi_{\text{DN}}$ and $\nabla\Phi_{\text{DN}}$
**end for**
**return** $\Phi_{\text{DN}}$, $\nabla\Phi_{\text{DN}}$

**Algorithm 2** Rigid-Body Collision Response Using the Proposed Field

**Require:** Rigid body $\mathcal{B}$, $\Phi_{\text{DN}}$, $\nabla\Phi_{\text{DN}}$, threshold $\tau$, $K_{\max}$
**for** $k = 1$ to $K_{\max}$ **do**
  Transform rigid-body surface samples to world space
  Form $\mathcal{Q} = \{\mathbf{p}_1, \ldots, \mathbf{p}_M, \mathbf{c}_t\}$
  Query $\widetilde{\Phi}(\mathbf{q})$ for all $\mathbf{q} \in \mathcal{Q}$
  Find $\mathbf{q}^\star = \arg\max_{\mathbf{q}\in\mathcal{Q}} \widetilde{\Phi}(\mathbf{q})$
  **if** $\widetilde{\Phi}(\mathbf{q}^\star) \leq \tau$ **then**
    **break**
  **end if**
  Project $\mathbf{q}^\star$ onto $\Phi_{\text{DN}} = \tau$
  Compute $\mathbf{n}_c$ from $\widetilde{\mathbf{g}}(\mathbf{x}_c)$
  Estimate $d_p$ and correct the rigid-body position
  Apply the normal impulse
  Recompute the contact-point velocity
  Apply the Coulomb friction impulse
**end for**

## 4.11 Overall Algorithm

Algorithm 1 summarizes contact-field construction for a static GS scene, while Algorithm 2 summarizes collision response at each physics time step.

## 4.12 Implementation Parameters

Table 2 summarizes the default parameters used in the current implementation. These settings provide identical rigid-body integration conditions for all compared methods, while the main parameters associated with density normalization are $\tau$ and $\gamma$.

## 4.13 Computational Characteristics

The primary preprocessing cost of the proposed method arises from constructing $\Phi_{\text{DN}}$ and $\nabla\Phi_{\text{DN}}$ from a static GS scene. Because each Gaussian contributes only to

| Parameter | Value | Description |
|---|---|---|
| $\tau$ | 0.50 | Contact iso-value |
| $\gamma$ | 0.85 | Density normalization exponent |
| $\epsilon$ | $10^{-10}$ | Effective-count stabilization |
| Grid resolution | $112 \times 80 \times 112$ | Contact-field grid |
| Simulation rate | 240 Hz | Rigid-body update rate |
| $M$ | $\leq 512$ | Rigid-body surface samples |
| Projection iterations | 14 | Maximum iso projection iterations |
| $\eta_{\max}$ | 0.12 | Maximum projection step |
| Correction passes | 10 | Maximum contact passes per step |
| $d_{\min}$ | 0.001 | Numerical separation bias |
| $d_{\max}$ | 0.08 | Maximum position correction |

Table 2: Default implementation parameters of the proposed method.

grid vertices within its bounded spatial support, we avoid the naive $O(NN_xN_yN_z)$ evaluation of every Gaussian over the entire dense grid. If $V_i$ denotes the number of grid vertices within the support of Gaussian $i$, the field-construction cost is

$$O\left(\sum_{i=1}^{N} V_i\right) \tag{64}$$

Because the contribution of each Gaussian can be evaluated independently, this computation is well suited to GPU parallelization.

After preprocessing, a field and gradient query at one point accesses only the eight surrounding grid vertices and therefore has $O(1)$ cost. If $M$ rigid-body samples are tested in one physics step, with at most $K$ correction passes and at most $P$ iso-surface projection iterations per pass, the runtime cost is

$$O(K(M+P))\,. \tag{65}$$

In the current implementation, $P = 14$ is fixed and $M \leq 512$, so the practical contact-query cost can be regarded as $O(KM)$. Because $K \leq 10$ is also bounded, the runtime collision response does not scale directly with the number of Gaussians in the original GS scene.

Consequently, rather than repeatedly testing the rendering-oriented Gaussian representation as a set of pairwise collision primitives, the proposed method preprocesses the continuous Gaussian contributions into a density-normalized contact representation for repeated rigid-body interaction. A key property of the method is that the field value, contact boundary, interpolated gradient normal, and collision response are all derived from the same scalar representation.

## 5 Experimental Results

### 5.1 Experimental Setup

All experiments were conducted under Windows 11 using AMD Ryzen 9 9950X3D CPU, 32 GB DDR5 memory, and an NVIDIA GeForce RTX 5080 16 GB GPU. The proposed method was implemented using CUDA, and the Gaussian contact field and analytic gradient were precomputed on the GPU. Rigid-body simulation was performed at 240 Hz, with a default contact-field grid resolution of $112 \times 80 \times 112$ and up to $M = 512$ rigid-body surface samples. Unless otherwise stated, all experiments used the same values of $\tau = 0.5$ and $\gamma = 0.85$.

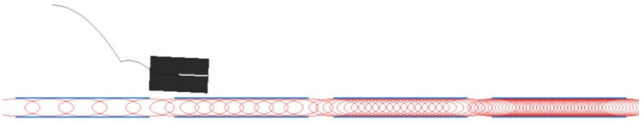
The compared methods were Splat-Nav Basic [26], Opacity-Scaled Probabilistic Variant, FOCI-style Gaussian Overlap [29], and the proposed method. To fairly isolate the effect of the contact representation, all methods used identical rigid-body integration, penetration correction, restitution, and friction response.

### 5.2 Contact Boundary and Duplication Robustness

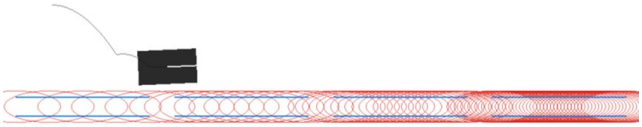
In this experiment, we keep the underlying geometry fixed and vary only the Gaussian sampling density or exact duplication to analyze changes in the boundary generated by each contact formulation. Under sampling-density variation, the number and spatial arrangement of Gaussians representing the same physical surface are changed, whereas under exact duplication, the Gaussian positions, covariances, and opacities are fixed and only the multiplicity of each primitive is increased.

Fig. 2 shows the same physical wall represented at different sampling densities.

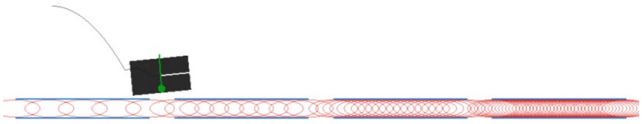
The four wall segments use the same physical thickness, Gaussian covariance, and opacity, while the number of Gaussian samples increases from left to right as $1\times$, $2\times$, $4\times$, and $8\times$. Thus, the analytic surface location



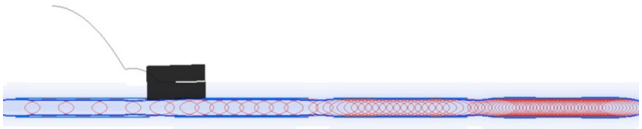
(a) Splat-Nav Basic [26]



(b) Opacity-Scaled Probabilistic Variant

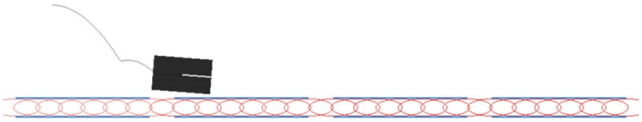


(c) FOCI-style Gaussian Overlap [29]

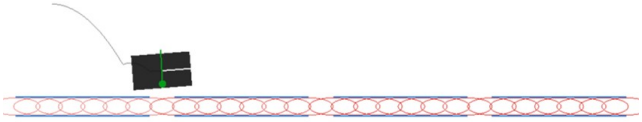


(d) Our method

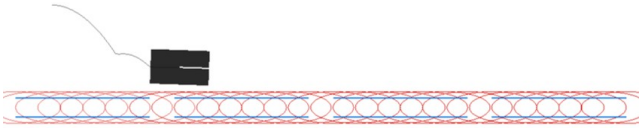
Fig. 2: Contact-boundary consistency under Gaussian sampling-density variation. The underlying wall geometry and Gaussian shape/opacity parameters are fixed, while the number of Gaussian samples is increased from $1\times$ to $8\times$. Red ellipses denote Gaussian primitives, blue lines indicate the fixed upper and lower boundaries of the analytic ground-truth wall, and magenta lines show the upper contact boundary obtained by each method.



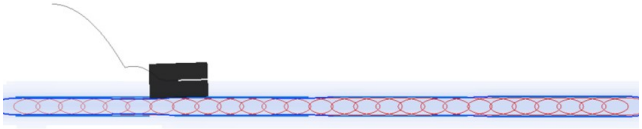
(a) Splat-Nav Basic [26]



(b) Opacity-Scaled Probabilistic Variant



(c) FOCI-style Gaussian Overlap [29]



(d) Our method

Fig. 3: Contact-boundary robustness under exact Gaussian duplication. Each segment contains identical Gaussian locations, covariance, and opacity, while the multiplicity of every Gaussian is increased from $1\times$ to $8\times$. Red ellipses denote Gaussian primitives, blue lines indicate the unchanged analytic ground-truth wall, and magenta lines represent the upper contact boundary obtained by each method.

remains unchanged, whereas the spacing and overlap distribution between Gaussians vary with density.

In Fig. 2 and Fig. 3, red ellipses denote anisotropic Gaussian primitives, blue lines indicate the fixed upper and lower boundaries of the analytic ground-truth wall, and magenta lines indicate the upper contact boundary obtained from each formulation. Therefore, contact-boundary robustness can be directly compared by examining the displacement of the magenta boundary under representation changes.

Splat-Nav Basic uses individual Gaussian supports as collision primitives. The Opacity-Scaled Probabilistic Variant is a baseline constructed for this comparison, in which the opacity of each Gaussian is used to adjust its effective collision support. The same global threshold is used for all density and duplication conditions, and no scene-specific or density-specific threshold retuning is performed. FOCI-style Gaussian Overlap applies a fixed threshold to Gaussian overlap, whereas the proposed method uses the iso-surface of the density-normalized field $\Phi_{\mathrm{DN}}$ at $\tau = 0.5$ as the contact boundary.

As sampling density increases, even Splat-Nav Basic may exhibit changes in the spatial arrangement and union envelope of individual ellipsoids. The opacity-scaled formulation and FOCI-style overlap are likewise affected by changes in local support or overlap configuration. In contrast, the proposed method uses the effective local splat count to mitigate field inflation caused by an increased number of Gaussians, thereby maintaining a relatively small contact-boundary displacement under density variation, as shown in Fig. 2(d).

Fig. 3 evaluates exact duplication while keeping both the Gaussian positions and shapes unchanged. Each wall segment contains the same seven Gaussians, and only the multiplicity of each Gaussian is increased to $1\times$, $2\times$, $4\times$, and $8\times$. Therefore, any displacement of the contact boundary under this condition directly reflects

| | Density | | | Duplication | | | Avg. |
|---|---|---|---|---|---|---|---|
| Method | 2× | 4× | 8× | 2× | 4× | 8× | Mean |
| Splat-Nav | 0.019 | 0.044 | 0.082 | **0.000** | **0.000** | **0.000** | 0.024 |
| Probabilistic | 0.017 | 0.035 | 0.064 | 0.015 | 0.041 | 0.079 | 0.042 |
| FOCI-style | 0.013 | 0.028 | 0.052 | 0.012 | 0.031 | 0.059 | 0.033 |
| Ours | **0.006** | **0.010** | **0.017** | 0.003 | 0.004 | 0.007 | **0.008** |

Table 3: Quantitative comparison of contact-boundary robustness under sampling-density variation and exact Gaussian duplication. Values denote mean boundary displacement relative to the 1× reference configuration; lower is better.

the sensitivity of the formulation to Gaussian multiplicity.

For Splat-Nav Basic, repeatedly adding the same geometric primitive does not change the union of the supports, and thus the same boundary is maintained under exact duplication. In contrast, formulations based on accumulated probability or Gaussian overlap repeatedly include contributions at the same location, which can shift the contact region relative to a fixed threshold.

In the proposed method, when the same Gaussian is duplicated $m$ times, the effective count $N_{\text{eff}}$ increases accordingly, thereby mitigating multiplicity-induced accumulation. A small residual variation may remain with the default setting $\gamma = 0.85$, whereas $\gamma = 1$ is theoretically invariant to exact duplication.

For the quantitative evaluation, the upper contact boundary obtained from the 1× configuration is used as the reference. At each identical $x$ position, we measure the absolute distance between the reference boundary and the boundary of the current representation and average it over the interior of the wall to compute the mean boundary displacement. Thus, this metric does not measure absolute geometric error relative to the ground truth; instead, it quantifies how much the boundary of the same contact formulation shifts as the representation changes.

As shown in Fig. 4(a), the boundary displacement of all comparison formulations increases with sampling density, whereas the proposed method maintains the smallest variation across the entire density range. In the exact-duplication experiment in Fig. 4(b), Splat-Nav Basic maintains zero displacement because of its primitive-union property, while the probabilistic and overlap-based formulations exhibit increasing boundary displacement as multiplicity increases. The proposed method maintains only a small variation over the entire duplication range even with the default setting $\gamma = 0.85$.

Table 3 shows that, at 8× sampling density, the displacement of the proposed method is 0.017, which is approximately 79.3% smaller than the 0.082 of Splat-Nav and also lower than the 0.052 of the FOCI-style overlap formulation. Under exact duplication, Splat-Nav is structurally invariant, whereas accumulation-sensitive methods exhibit larger variations. The proposed method maintains a small displacement of 0.007 even at 8× duplication, which is approximately 91.1% smaller than the 0.079 of the probabilistic variant.

Together, the two experiments show that effective-count normalization improves robustness to both changes in sampling density and exact Gaussian multiplicity.

### 5.3 Contact Normal and Rigid-Body Response

In addition to the position of the contact boundary, the temporal continuity of the contact normal derived from the boundary is important for rigid-body simulation. Discontinuities in the normal directly affect penetration correction, impulse direction, and contact torque, and can therefore be amplified into trajectory and rotational fluctuations during repeated contact.

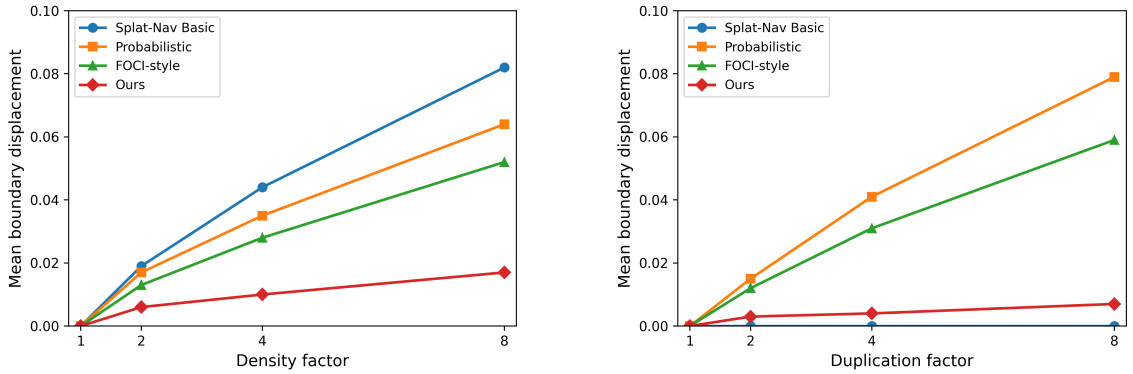
To evaluate this behavior, we construct a *Contact-Normal Continuity* scene, in which a rigid body remains in continuous contact with a curved Gaussian surface, and a *Repeated Rigid-Body Response* scene, in which collision and re-contact occur repeatedly.

On the curved surface in Fig. 5, gradual normal rotation caused by surface curvature is itself physically expected. Accordingly, the objective is not to eliminate normal variation, but to reduce abrupt switching that cannot be explained by the underlying geometry.

In a primitive-based formulation, changes in the active support can be directly reflected in the normal direction as the rigid body moves between neighboring Gaussians. The FOCI-style overlap formulation combines multiple contributions but remains sensitive to changes in the local overlap configuration. Using the gradient of a single continuous field instead preserves geometry-induced normal rotation while reducing abrupt changes caused by the splat arrangement.

Fig. 6 evaluates how contact-normal variation affects the actual rigid-body response. After entering the curved surface, the rigid body undergoes repeated collision, sliding, and re-contact. In such a setting, contact-point and normal variations accumulated over multiple frames can have a greater effect on orientation and angular response than an instantaneous contact error.

The existing methods exhibit relatively large fluctuations in trajectory and rotation where the local Gaussian configuration changes. Deriving the contact point




(a) Sampling-density variation

(b) Exact Gaussian duplication

Fig. 4: Quantitative evaluation of contact-boundary robustness. (a) Mean upper-boundary displacement under Gaussian sampling-density variation. (b) Mean upper-boundary displacement under exact Gaussian duplication. The 1× representation is used as the reference in both experiments, and lower values indicate greater robustness to representation changes.

and gradient normal from the same continuous field yields a smoother response under repeated collision and re-contact.

The normal variation between two consecutive contact frames $t-1$ and $t$ is defined as

$$\Delta\theta_n^{(t)} = \cos^{-1}\left[\text{clamp}\left(\mathbf{n}^{(t)} \cdot \mathbf{n}^{(t-1)}, -1, 1\right)\right] \tag{66}$$

where $\mathbf{n}^{(t)}$ is the unit normal at the $t$-th contact frame. Because a curved surface also induces geometrically necessary normal rotation, $\Delta\theta_n$ is used as a metric for temporal continuity rather than as an absolute normal error.

The rigid-body rotational response is measured as

$$\Delta\omega^{(t)} = \left|\omega^{(t)} - \omega^{(t-1)}\right| \tag{67}$$

Because $\Delta\omega$ is influenced not only by the contact normal but also by the contact position, lever arm, impulse magnitude, friction, and the current rigid-body state, it is not expected to follow the same temporal pattern as $\Delta\theta_n$.

As shown in Fig. 7(a), the existing methods exhibit large normal spikes at several contact frames, whereas the proposed method maintains a more gradual variation along the curved surface. Fig. 7(b) also shows that methods with larger normal variation generally exhibit larger angular fluctuations, although the temporal patterns of the two graphs do not coincide exactly.

As summarized in Table 4, the proposed method reduces the mean contact-normal variation from 15.62 deg for Splat-Nav to 5.79 deg, corresponding to a reduction of approximately 62.9%. The mean $|\Delta\omega|$ is likewise reduced from 0.151 rad/s to 0.061 rad/s, or approximately 59.6%. The same trend is observed for the maximum variations, and the proposed method also maintains lower temporal fluctuation than the FOCI-style overlap formulation.

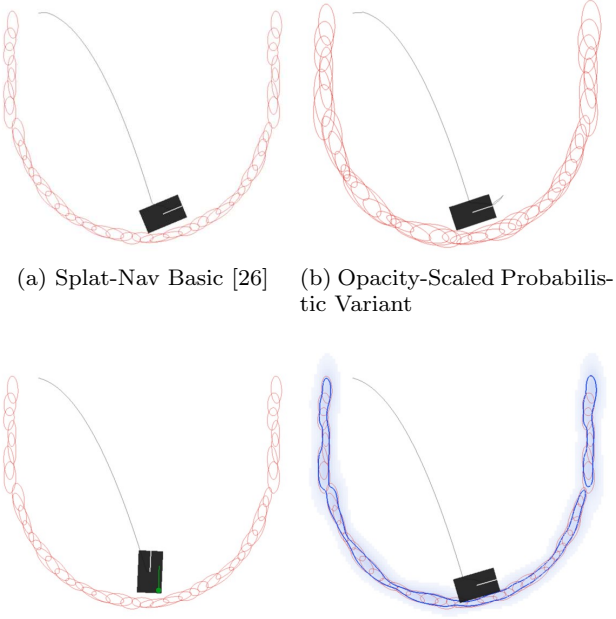
| Method | Mean $\Delta\theta_n$ (deg) | Max $\Delta\theta_n$ (deg) | Mean $\lvert\Delta\omega\rvert$ (rad/s) | Max $\lvert\Delta\omega\rvert$ (rad/s) |
|---|---|---|---|---|
| Splat-Nav | 15.62 | 35.00 | 0.151 | 0.29 |
| Probabilistic | 12.79 | 27.00 | 0.125 | 0.23 |
| FOCI-style | 10.42 | 20.00 | 0.102 | 0.18 |
| Ours | **5.79** | **11.00** | **0.061** | **0.11** |

Table 4: Quantitative comparison of contact-normal continuity and rigid-body rotational response. $\Delta\theta_n$ is measured in degrees, while $|\Delta\omega|$ is measured in rad/s. For all metrics, lower values indicate a more stable temporal response.

These results do not imply suppression of physically required rotation; rather, the field preserves geometry-induced normal and angular response while reducing fluctuations associated with changes in the local Gaussian distribution.

### 5.4 Comparison with Existing Gaussian-Based Methods

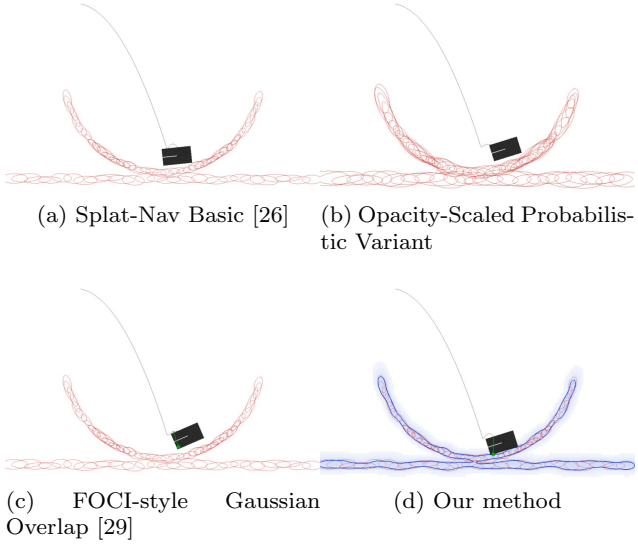
To extend the preceding controlled experiments to more complex Gaussian configurations, we construct three heterogeneous scenes. The *Mixed-Density Curved Corridor* combines curvature with spatial density variation; the *Multi-Orientation Dense Junction* contains



(a) Splat-Nav Basic [26] (b) Opacity-Scaled Probabilistic Variant

(c) FOCI-style Gaussian Overlap [29] (d) Our method

Fig. 5: Comparison of contact-normal continuity on a curved Gaussian surface. A rounded rigid body with identical initial conditions is brought into contact with a curved surface represented by nonuniform Gaussians. Gradual normal rotation caused by the underlying surface geometry is expected, whereas abrupt directional changes induced by Gaussian switching or variations in local overlap indicate contact-normal instability.



(a) Splat-Nav Basic [26] (b) Opacity-Scaled Probabilistic Variant

(c) FOCI-style Gaussian Overlap [29] (d) Our method

Fig. 6: Comparison of repeated rigid-body collision responses. A rigid body with identical initial conditions undergoes repeated collision, sliding, and re-contact within a curved Gaussian environment. Because the same impulse-based response solver is used for all methods, differences in trajectory and orientation reflect differences in the contact points and normals obtained from each contact representation.

strongly overlapping Gaussians with different orientations, scales, and opacities; and the *Reconstructed-Style Obstacle Course* combines planar and curved regions, narrow passages, density transitions, and sparse artifacts within a single environment.

Within each scene, all four methods use the same initial rigid-body state. Only the initial states across different scenes are adjusted so that the rigid body sufficiently traverses the corresponding geometry.

Figure 8 isolates a sparse-to-dense transition along a curved surface. Primitive- and overlap-dependent formulations directly reflect the local density change in their contact condition, whereas effective-count normalization maintains a more consistent field level set and normal across the transition.

Figure 9 stresses orientation changes: anisotropic Gaussians with different orientations strongly overlap at a single junction, so the locally dominant support can change rapidly. Integrating these contributions into one continuous field reduces abrupt contact-direction switching through the junction.

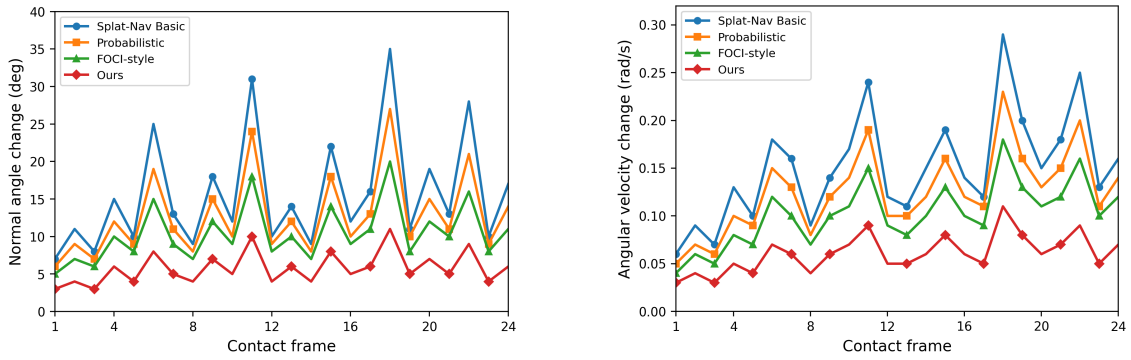
Figure 10 combines density transitions, orientation changes, narrow passages, and sparse artifacts in one reconstructed-style environment. Because these configurations occur consecutively, small contact-point or normal variations can accumulate along the trajectory; the same density-normalized field is used throughout without configuration-specific retuning.

To compare temporal stability across the three scenes, we use the mean contact-normal variation and mean angular-velocity variation defined in Sec. 5.3.

As shown in Fig. 11, all methods exhibit their largest variation in the Multi-Orientation Dense Junction because differently oriented and scaled Gaussians strongly overlap within the same contact region. The density-normalized field yields the lowest normal variation in all three scenes.

The angular-response results follow the same overall ranking, although the rate of change across scenes does not exactly match that of the normal variation. This is because the rotational response is affected not only by the contact normal but also by the contact position, lever arm, impulse magnitude, and current rigid-body state.

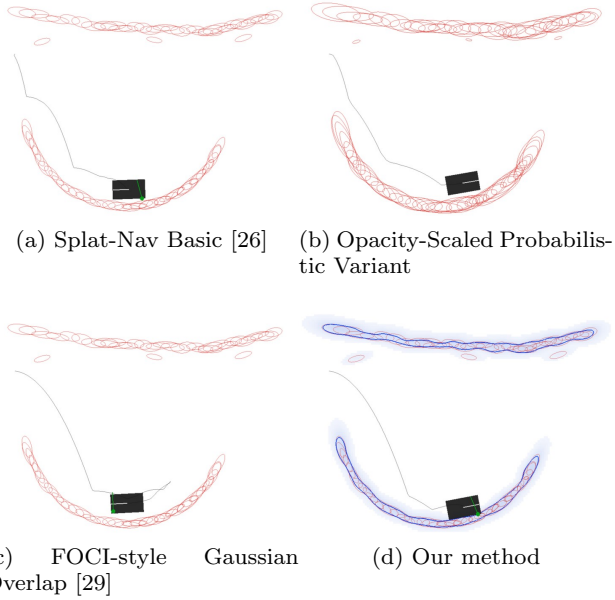
Across the three scenes, the mean contact-normal variations of Splat-Nav, the probabilistic variant, the




(a) Contact-normal variation

(b) Angular-response variation

Fig. 7: Quantitative comparison of temporal stability across contact frames. (a) shows the contact-normal angular variation, $\Delta\theta_n$, between consecutive contact frames, while (b) shows the rigid-body angular-velocity variation, $|\Delta\omega|$. For both metrics, lower values and smaller temporal spikes indicate a more stable contact response.



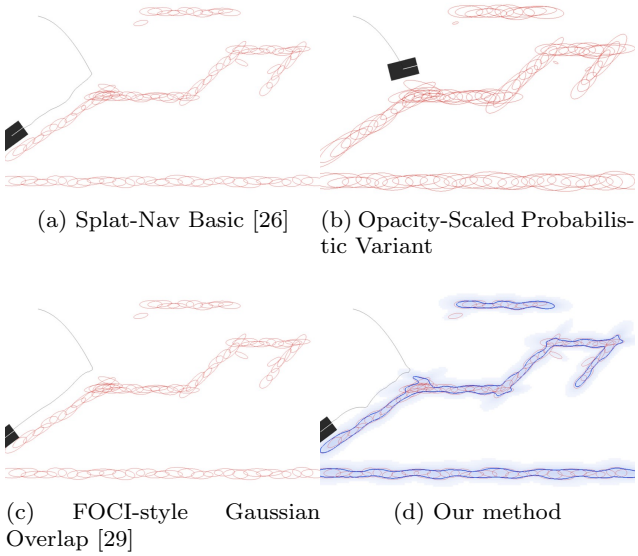
(a) Splat-Nav Basic [26]

(b) Opacity-Scaled Probabilistic Variant

(c) FOCI-style Gaussian Overlap [29]

(d) Our method

Fig. 8: Comparison in a mixed-density curved corridor. The scene combines a continuously curved contact surface with spatially varying Gaussian sampling density and local overlap. All methods use identical rigid-body initial conditions and the same impulse-based response solver within the scene. Differences in trajectory and orientation therefore reflect the effect of each Gaussian-based contact representation.



(a) Splat-Nav Basic [26]

(b) Opacity-Scaled Probabilistic Variant

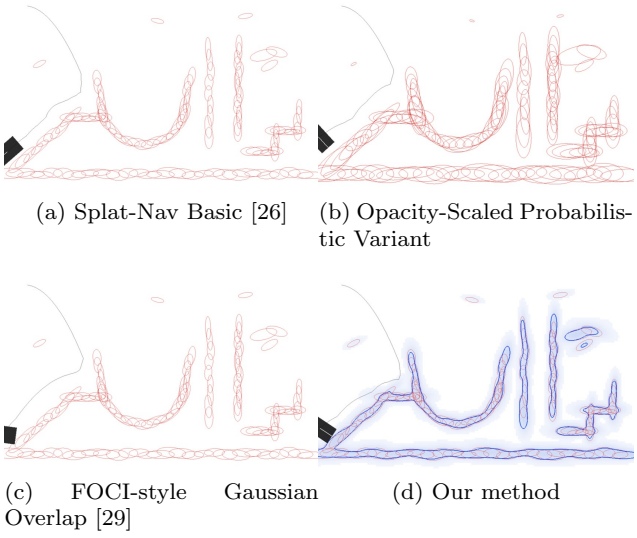
(c) FOCI-style Gaussian Overlap [29]

(d) Our method

Fig. 9: Comparison in a multi-orientation dense junction. Multiple anisotropic Gaussians with different orientations, scales, and opacities overlap around the same junction. This configuration emphasizes contact switching and normal fluctuation caused by changes in the locally dominant Gaussian support.

FOCI-style formulation, and the proposed method are 18.73, 15.63, 13.13, and 7.50 deg, respectively. Thus, the proposed method reduces the variation by approximately 60.0% relative to Splat-Nav and by approximately 42.9% relative to the FOCI-style formulation.

The mean $|\Delta\omega|$ values are 0.169, 0.145, 0.122, and 0.068 rad/s, respectively. Accordingly, the proposed method reduces this metric by approximately 59.8% relative to Splat-Nav and by approximately 44.3% relative to the FOCI-style formulation. The consistent trend across three scenes with different levels of complexity indicates that the stability of the proposed contact field is not limited to a specific controlled benchmark.



(a) Splat-Nav Basic [26] (b) Opacity-Scaled Probabilistic Variant

(c) FOCI-style Gaussian Overlap [29] (d) Our method

Fig. 10: Comparison in a reconstructed-style obstacle course. The scene combines curved and planar surfaces, local density variation, narrow passages, overlapping Gaussian clusters, and sparse reconstruction-like artifacts within a single environment. The experiment evaluates whether each contact representation maintains consistent rigid-body behavior under heterogeneous Gaussian configurations.

| | Mean $\Delta\theta_n$ (deg) | | | Mean $\lvert\Delta\omega\rvert$ (rad/s) | | |
|---|---|---|---|---|---|---|
| Method | C.C. | D.J. | O.C. | C.C. | D.J. | O.C. |
| Splat-Nav | 14.8 | 22.5 | 18.9 | 0.124 | 0.214 | 0.169 |
| Probabilistic | 12.1 | 18.7 | 16.1 | 0.108 | 0.176 | 0.151 |
| FOCI-style | 10.4 | 15.6 | 13.4 | 0.091 | 0.148 | 0.126 |
| Ours | **6.3** | **8.7** | **7.5** | **0.057** | **0.079** | **0.068** |

Table 5: Quantitative comparison across complex Gaussian scenes. Values report the mean contact-normal variation $\Delta\theta_n$ in degrees and the mean angular-velocity variation $|\Delta\omega|$ in rad/s. Lower is better.

## 5.5 Parameter Analysis

The preceding experiments compared contact behavior under various Gaussian configurations. This section analyzes the role of density normalization and the behavior of the main parameters without introducing additional visual benchmarks.

### *5.5.1 Effect of the Normalization Exponent $\gamma$*

Following the duplication analysis in Sec. 4, Table 6 evaluates how the normalization exponent $\gamma$ controls sensitivity to Gaussian multiplicity.

| $\gamma$ | 1× | 2× | 4× | 8× |
|---|---|---|---|---|
| 0.00 | 1.00 | 2.00 | 4.00 | 8.00 |
| 0.50 | 1.00 | 1.41 | 2.00 | 2.83 |
| 0.70 | 1.00 | 1.23 | 1.52 | 1.87 |
| 0.85 | 1.00 | 1.11 | 1.23 | 1.37 |
| 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |

Table 6: Effect of the normalization exponent $\gamma$ under exact Gaussian duplication. Values denote the theoretical strength ratio $A^{(m)}/A^{(1)} = m^{1-\gamma}$.

At 8× duplication, the strength ratio decreases from 8.00 for $\gamma = 0$ to 1.37 for the default $\gamma = 0.85$, while $\gamma = 1$ is theoretically invariant to exact duplication. We use $\gamma = 0.85$ as a compromise that strongly suppresses multiplicity-induced accumulation while retaining part of the collective support of neighboring Gaussians.

### *5.5.2 Effect of the Contact Threshold $\tau$*

The contact threshold $\tau$ determines the location of the iso-surface of $\Phi_{\mathrm{DN}}$. A smaller $\tau$ produces a more outward, conservative contact boundary, whereas a larger $\tau$ moves the boundary toward the Gaussian centers. We use a single global value of $\tau = 0.5$ for all scenes and density conditions. Therefore, the robustness observed in the preceding experiments is not obtained by scene-specific or density-specific threshold retuning.

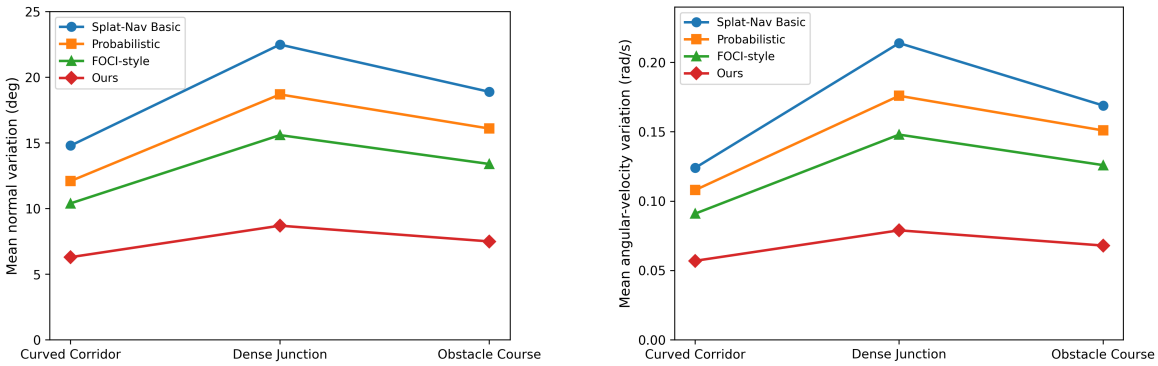
### *5.5.3 Grid Resolution and Body Samples*

The grid resolution determines the trade-off between the spatial accuracy of the precomputed contact field and its memory/preprocessing cost, while the number of surface samples $M$ determines contact coverage and runtime query cost. In all experiments, we use a $112 \times 80 \times 112$ grid and at most $M = 512$ surface samples.

A low grid resolution can increase interpolation error in thin or high-curvature regions, whereas an excessively high resolution increases preprocessing and memory costs. Similarly, a small $M$ may miss small contact regions, while beyond sufficient coverage, increasing the number of samples primarily increases runtime query cost. Considering this accuracy–cost trade-off, we use the above settings consistently in all experiments.

## 5.6 Real-Time 3D Interaction and Multi-Object Evaluation

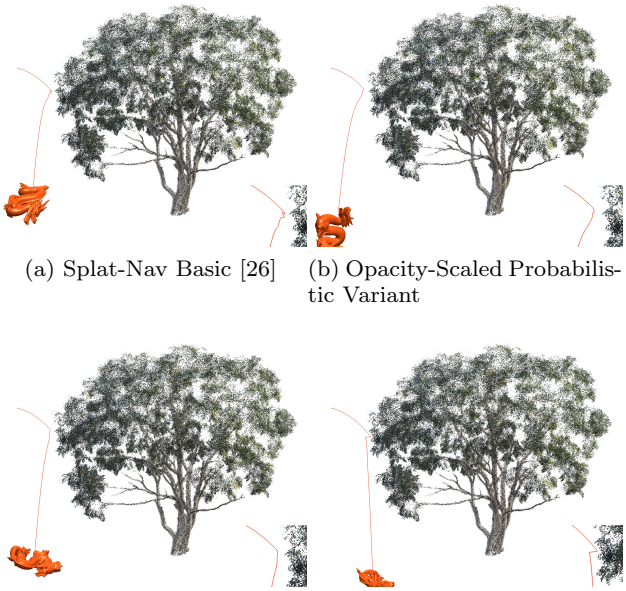
Finally, we apply the proposed method to actual three-dimensional Gaussian scenes and triangle-mesh rigid bodies. Rather than repeating the quantitative analyses




(a) Mean contact-normal variation (b) Mean angular-velocity variation

Fig. 11: Scene-wise quantitative comparison of temporal contact stability. (a) Mean contact-normal angular variation and (b) mean rigid-body angular-velocity variation for the mixed-density curved corridor, multi-orientation dense junction, and reconstructed-style obstacle course. Lower values indicate more stable temporal contact behavior.

presented above, this section focuses on 3D applicability and a multi-object runtime stress test.



(a) Splat-Nav Basic [26] (b) Opacity-Scaled Probabilistic Variant

(c) FOCI-style Gaussian Overlap [29] (d) Our method

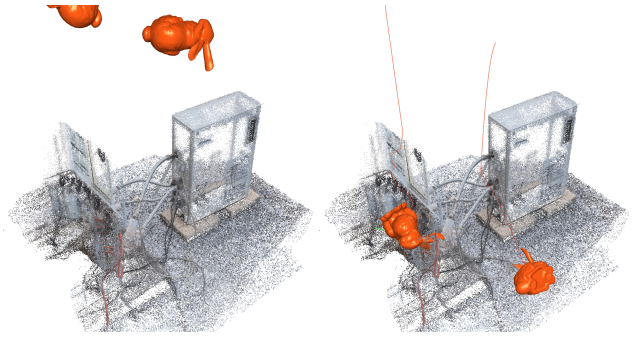
Fig. 12: Real-time 3D interaction between a Gaussian-splat Tree scene and the Stanford Dragon. The same initial rigid-body state and response solver are used for all methods. The magnified insets highlight local contact behavior near the reconstructed surface, where the proposed method maintains the most stable interaction with the heterogeneous Gaussian representation.

For a static Gaussian scene, the contact field and gradient are precomputed once before simulation, and all subsequent rigid-body interactions reuse the same field. The results in Fig. 12–14 are generated through interactive real-time simulation without offline collision computation or frame-by-frame field reconstruction.

Fig. 12 compares 3D collisions between a Gaussian-splat Tree scene and the Stanford Dragon using the four methods. The Tree scene provides a complex representation in which Gaussian density and orientation vary substantially around branches and foliage. In particular, the magnified insets show that the primitive-level and overlap-dependent formulations are relatively sensitive to the local Gaussian distribution, whereas the proposed method maintains a more stable interaction while contacting the complex surface. This observation indicates that the boundary and normal consistency observed in the 2D controlled experiments also extends to a 3D scene.

Fig. 13 uses the Stanford Bunny with different reconstructed-style Gaussian geometries. Although the scenes differ in spatial density and surface orientation, the contact parameters are not retuned for individual scenes. In both examples, an arbitrary rigid mesh continuously collides with and rotates against Gaussian surfaces with different orientations in real time.

In Fig. 14, the single-object interaction is extended to 20 dynamic rigid objects. Each object is initialized with a different position, orientation, linear velocity, and angular velocity, and an object that leaves the simulation domain is respawned above the scene. Thus, a fixed number of dynamic bodies is maintained throughout the simulation while continuously generating collision workload.



(a) Initial and intermediate interaction states



(b) Interaction with a second reconstructed Gaussian scene

Fig. 13: Real-time rigid-body interaction in reconstructed Gaussian-splat scenes using the Stanford Bunny. The examples contain irregular surfaces, sparse and dense Gaussian regions, and geometric structures with different orientations. The same density-normalized contact representation is used without scene-specific retuning.

All rigid objects share the same precomputed Gaussian contact field, and independent field queries and collision responses are performed only for the transformed surface samples of each object. Therefore, the runtime query cost increases with the number of dynamic objects, whereas construction of the static Gaussian field is not repeated. The fact that real-time simulation is maintained even while multiple complex mesh objects continuously collide and respawn demonstrates the practical applicability of the proposed method to 3D rigid-body interaction.

## 6 Discussion

The experiments suggest that representation density should be treated as a nuisance variable when rendering-oriented Gaussians are reused for physical contact. The benefit of normalization is therefore not limited to exact duplication: it is most relevant when reconstruction or densification changes local primitive multiplicity without a corresponding change in the intended physical surface. Deriving both the boundary and its normal from the same normalized field also avoids introducing a separate normal-estimation stage with potentially different density sensitivity.

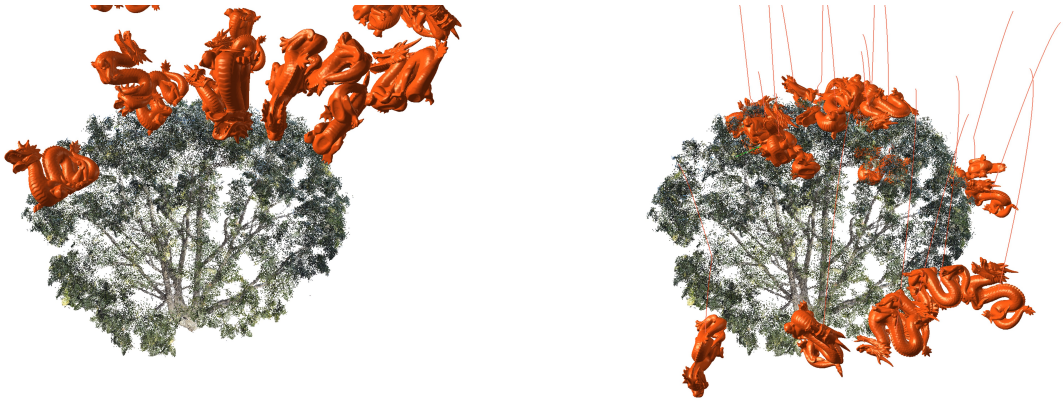
Density normalization is not intended to provide complete invariance to all forms of representation variation. The default value $\gamma = 0.85$ substantially reduces multiplicity-induced accumulation while retaining part of the collective contribution of neighboring Gaussians. Although $\gamma = 1$ provides theoretical invariance under exact duplication, it does not guarantee complete geometric invariance for general GS representations in which Gaussian position, covariance, opacity, and orientation vary simultaneously. Moreover, rather than replacing accurate surface reconstruction, the proposed method is intended to reduce the contact-representation gap between a given GS representation and physics-based interaction.

Because the effective count is computed from spatial support rather than opacity-weighted contributions, a large population of extremely low-opacity Gaussians may overestimate local representation multiplicity and locally reduce the normalized contact strength. Incorporating confidence- or opacity-aware effective counts is therefore a useful direction for future work.

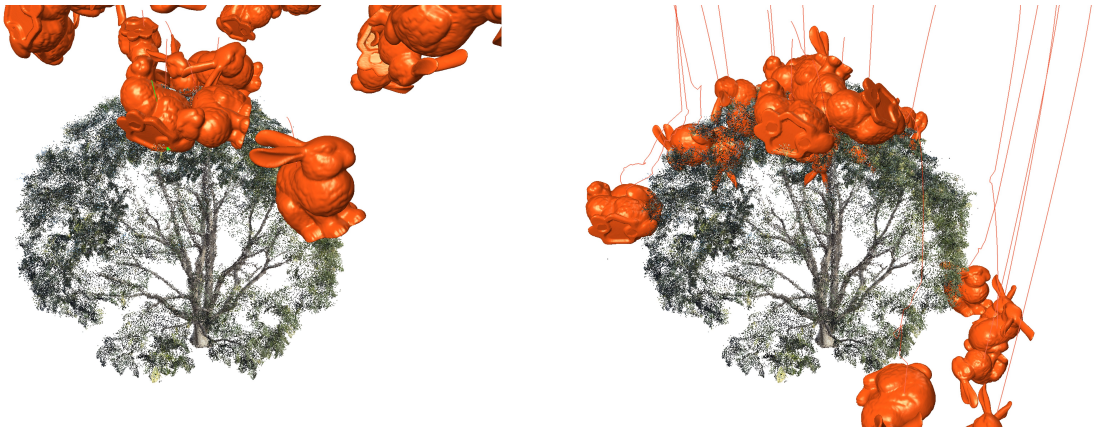
The current method has several limitations. First, because the contact field is precomputed on a regular grid, spatial accuracy and memory cost depend on the grid resolution. Thin structures with high curvature or small Gaussian features may require a higher resolution, and this limitation could be addressed using an adaptive grid, sparse voxel structure, or hierarchical field representation.

Second, contact detection is based on a finite number of sample points on the rigid-body surface. A very small contact region may therefore lie between samples and be missed. The current implementation uses up to $M = 512$ surface samples together with a center query, but adaptive surface sampling or continuous collision detection could provide greater geometric coverage.

Third, to isolate the effect of the contact representation, we apply the same simplified impulse-based rigid-body response to all methods. In particular, the rotational response uses an average inverse-inertia approximation computed from the diagonal inertia. Accordingly, the goal of this study is not to validate the accuracy of a high-fidelity dynamics solver itself, but to evaluate the stability of the contact points and normals supplied to the same response model. The proposed representation can be combined in future work with a full inertia tensor, manifold contact, or a constraint-based solver.



(a) Multiple rigid objects during continuous falling and contact



(b) Later state with redistributed objects and repeated contacts

Fig. 14: Real-time multi-object stress test in the Gaussian-splat Tree scene. Multiple rigid objects are spawned with different positions, orientations, linear velocities, and angular velocities and continuously interact with the same precomputed contact field. Objects leaving the simulation domain are respawned above the scene so that the prescribed number of active rigid bodies is maintained throughout the simulation.

Fourth, the current implementation targets static Gaussian scenes. Dynamic GS scenes in which Gaussian positions or covariances continuously change require additional updates to the contact field. This framework could be extended to dynamic representations through incremental updates or local field reconstruction restricted to changing regions.

Fifth, the multi-object experiment in Fig. 14 considers multiple rigid bodies interacting with the same Gaussian scene, but pairwise object–object collision between rigid bodies is outside the scope of this study. Such collision handling can be independently integrated with existing broad-phase and rigid-body collision frameworks and extended to large-scale multi-body simulation in future work.

Finally, although $\tau$ and $\gamma$ are fixed across all experiments, their optimal ranges may differ substantially across reconstruction pipelines or Gaussian parameter distributions. Automatic parameter estimation based on scene statistics or data-driven calibration therefore represents a useful direction for future work.

## 7 Conclusion

This paper addresses the instability of contact boundaries caused by variations in local splat density and multiplicity in Gaussian Splatting scenes and proposes a density-normalized contact field to mitigate this problem. The proposed method normalizes the accumulated Gaussian contribution using the effective local splat count and uses the iso-surface and gradient of the resulting continuous field as the contact boundary and contact normal, respectively.

The experiments evaluate boundary robustness under sampling-density variation and exact duplication, normal/response stability under curved and repeated

contact, and collision behavior under heterogeneous Gaussian configurations. Additional 3D experiments using the Stanford Dragon, Stanford Bunny, and multiple dynamic rigid objects demonstrate real-time physical interaction in complex Gaussian scenes.

Future work will investigate adaptive field representations, dynamic Gaussian updates, continuous collision detection, and integration with more general rigid-body solvers to support larger-scale and dynamic Gaussian scenes.

**Acknowledgements (not compulsory)**

The authors received no specific funding for this work.

**Funding**

This work was supported by INHA UNIVERSITY Research Grant. This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (RS-2026-25474535). This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (RS-2026-25548560, Artificial Intelligence Innovation Graduate School (Inha University)).

**Author contributions statement**

J.H.K. conceived the study, developed the methodology, implemented the system, conducted the experiments, analyzed the results, and wrote and reviewed the manuscript.

**References**

1. B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis, "3D Gaussian Splatting for Real-Time Radiance Field Rendering," *ACM Transactions on Graphics*, vol. 42, no. 4, Art. no. 139, 2023.
2. A. Guédon and V. 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 (CVPR)*, pp. 5354–5363, 2024.
3. B. Huang, Z. Yu, A. Chen, A. Geiger, and S. Gao, "2D Gaussian Splatting for Geometrically Accurate Radiance Fields," in *Proceedings of the ACM SIGGRAPH Conference Papers*, Art. no. 32, 2024.
4. Z. Yu, T. Sattler, and A. Geiger, "Gaussian Opacity Fields: Efficient Adaptive Surface Reconstruction in Unbounded Scenes," *ACM Transactions on Graphics*, vol. 43, no. 6, pp. 271:1–271:13, 2024.
5. D. Chen, H. Li, W. Ye, Y. Wang, W. Xie, S. Zhai, N. Wang, H. Liu, H. Bao, and G. Zhang, "PGSR: Planar-Based Gaussian Splatting for Efficient and High-Fidelity Surface Reconstruction," *IEEE Transactions on Visualization and Computer Graphics*, vol. 31, no. 9, pp. 6100–6111, 2025.
6. H. Chen, C. Li, and G. H. Lee, "NeuSG: Neural Implicit Surface Reconstruction with 3D Gaussian Splatting Guidance," *arXiv preprint arXiv:2312.00846*, 2023.
7. N. Keetha, J. Karhade, K. M. Jatavallabhula, G. Yang, S. Scherer, D. Ramanan, and J. Luiten, "SplaTAM: Splat, Track & Map 3D Gaussians for Dense RGB-D SLAM," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21357–21366, 2024.
8. C. Yan, D. Qu, D. Xu, B. Zhao, Z. Wang, D. Wang, and X. Li, "GS-SLAM: Dense Visual SLAM with 3D Gaussian Splatting," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 19595–19604, 2024.
9. H. Matsuki, R. Murai, P. H. J. Kelly, and A. J. Davison, "Gaussian Splatting SLAM," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18039–18048, 2024.
10. H. Huang, L. Li, H. Cheng, and S.-K. Yeung, "Photo-SLAM: Real-Time Simultaneous Localization and Photorealistic Mapping for Monocular, Stereo, and RGB-D Cameras," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21584–21593, 2024.
11. V. Yugay, Y. Li, T. Gevers, and M. R. Oswald, "Gaussian-SLAM: Photo-realistic Dense SLAM with Gaussian Splatting," *arXiv preprint arXiv:2312.10070*, 2023.
12. Z. Peng, T. Shao, Y. Liu, J. Zhou, Y. Yang, J. Wang, and K. Zhou, "RTG-SLAM: Real-Time 3D Reconstruction at Scale Using Gaussian Splatting," in *Proceedings of the ACM SIGGRAPH Conference Papers*, 2024.
13. E. Sandström, K. Tateno, M. Oechsle, M. Niemeyer, L. Van Gool, M. R. Oswald, and F. Tombari, "Splat-SLAM: Globally Optimized RGB-Only SLAM with 3D Gaussians," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2025.
14. Y. Su, L. Chen, K. Zhang, Z. Zhao, C. Hou, and Z. Yu, "GauS-SLAM: Dense RGB-D SLAM with Gaussian Surfels," *arXiv preprint arXiv:2505.01934*, 2025.
15. Y. S. Hu, N. Abboud, M. Q. Ali, A. S. Yang, I. H. Elhajj, D. C. Asmar, Y. Chen, and J. S. Zelek, "MGSO: Monocular Real-Time Photometric SLAM with Efficient 3D Gaussian Splatting," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 11061–11067, 2025.
16. B. Li, Z. Cai, Y.-F. Li, I. Reid, and H. Rezatofighi, "Hier-SLAM: Scaling-Up Semantics in SLAM with a Hierarchically Categorical Gaussian Splatting," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 9748–9754, 2025.
17. Z. Zhu, S. Peng, V. Larsson, W. Xu, H. Bao, Z. Cui, M. R. Oswald, and M. Pollefeys, "NICE-SLAM: Neural Implicit Scalable Encoding for SLAM," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.
18. X. Yang, H. Li, H. Zhai, Y. Ming, Y. Liu, and G. Zhang, "Vox-Fusion: Dense Tracking and Mapping with Voxel-Based Neural Implicit Representation," in *Proceedings of the IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pp. 499–507, 2022.

19. H. Wang, J. Wang, and L. Agapito, "Co-SLAM: Joint Coordinate and Sparse Parametric Encodings for Neural Real-Time SLAM," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13293–13302, 2023.
20. E. Sandström, Y. Li, L. Van Gool, and M. R. Oswald, "Point-SLAM: Dense Neural Point Cloud-Based SLAM," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 18387–18398, 2023.
21. A. Rosinol, J. J. Leonard, and L. Carlone, "NeRF-SLAM: Real-Time Dense Monocular SLAM with Neural Radiance Fields," *arXiv preprint arXiv:2210.13641*, 2022.
22. A. Dhawale, X. Yang, and N. Michael, "Reactive Collision Avoidance Using Real-Time Local Gaussian Mixture Model Maps," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018.
23. P. Z. X. Li, S. Karaman, and V. Sze, "GMMap: Memory-Efficient Continuous Occupancy Map Using Gaussian Mixture Model," *IEEE Transactions on Robotics*, vol. 40, pp. 1339–1355, 2024.
24. T. Chen, P. Culbertson, and M. Schwager, "CATNIPS: Collision Avoidance Through Neural Implicit Probabilistic Scenes," *IEEE Transactions on Robotics*, vol. 40, pp. 2712–2728, 2024.
25. J. Ortiz, A. Clegg, J. Dong, E. Sucar, D. Novotny, M. Zollhoefer, and M. Mukadam, "iSDF: Real-Time Neural Signed Distance Fields for Robot Perception," in *Proceedings of Robotics: Science and Systems (RSS)*, 2022.
26. T. Chen, O. Shorinwa, J. Bruno, A. Swann, J. Yu, W. Zeng, K. Nagami, P. Dames, and M. Schwager, "Splat-Nav: Safe Real-Time Robot Navigation in Gaussian Splatting Maps," *IEEE Transactions on Robotics*, 2025, doi: 10.1109/TRO.2025.3552348.
27. T. Chen, A. Swann, J. Yu, O. Shorinwa, R. Murai, M. Kennedy III, and M. Schwager, "A Control Barrier Function for Safe Navigation with Online Gaussian Splatting Maps," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 11758–11765, 2025.
28. J. Michaux, S. Isaacson, C. E. Adu, A. Li, R. K. Swayampakula, P. Ewen, S. Rice, K. A. Skinner, and R. Vasudevan, "Let's Make a Splan: Risk-Aware Trajectory Optimization in a Normalized Gaussian Splat," *IEEE Transactions on Robotics*, pp. 1–19, 2025, doi: 10.1109/TRO.2025.3584559.
29. M. Gomez Andreu, M. Wilder-Smith, V. Klemm, V. Patil, J. Tordesillas, and M. Hutter, "FOCI: Trajectory Optimization on Gaussian Splats," *arXiv preprint arXiv:2505.08510*, 2025.
30. Q. Chen, J. Sun, N. Gao, J. Low, T. Chen, and M. Schwager, "GRaD-Nav: Efficiently Learning Visual Drone Navigation with Gaussian Radiance Fields and Differentiable Dynamics," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 7941–7948, 2025.