

VoteSplat: Hough Voting Gaussian Splatting for 3D Scene Understanding

Minchao Jiang^{1*}, Shunyu Jia^{1*}, Jiaming Gu^{1,2}, Xiaoyuan Lu³, Guangming Zhu¹,
Anqi Dong⁴, Liang Zhang^{1†}

¹ School of Computer Science and Technology, Xidian University

² Algorithm R&D Center, Qing Yi (Shanghai)

³ Shanghai Pudong Cryptography Research Institute

⁴ Division of Decision and Control Systems and Department of Mathematics,
KTH Royal Institute of Technology

{jamchaos, syjia_2001}@stu.xidian.edu.cn, jiaming.gu.xidian@outlook.com,
{gmzhu, liangzhang}@xidian.edu.cn, xylu@bnc.org.cn, anqid@kth.se

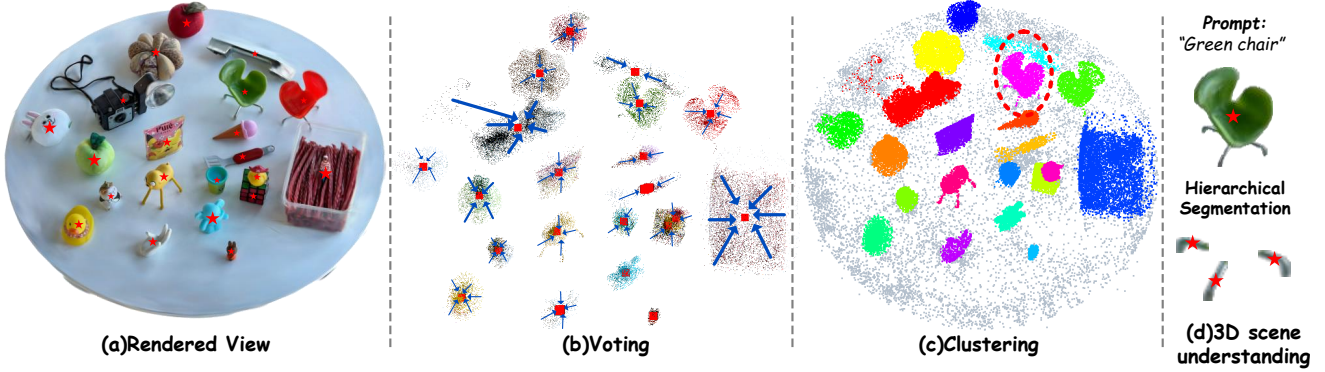


Figure 1. VoteSplat integrates 3DGS and Hough Voting for 3D scene understanding: (a) 3DGS retains its original novel view synthesis capability, (b) each Gaussian primitive encodes an offset vector which votes the point cloud to the instance center, (c) 3D vote clustering enables instance segmentation, and (d) open-vocabulary 3D instance localization and click-based 3D object localization are demonstrated.

Abstract

3D Gaussian Splatting (3DGS) has become horsepower in high-quality, real-time rendering for novel view synthesis of 3D scenes. However, existing methods focus primarily on geometric and appearance modeling, lacking deeper scene understanding while also incurring high training costs that complicate the originally streamlined differentiable rendering pipeline. To this end, we propose VoteSplat, a novel 3D scene understanding framework that integrates Hough voting with 3DGS. Specifically, Segment Anything Model (SAM) is utilized for instance segmentation, extracting objects, and generating 2D vote maps. We then embed spatial offset vectors into Gaussian primitives. These offsets construct 3D spatial votes by associating them with 2D image votes, while depth distortion constraints refine localization

along the depth axis. For open-vocabulary object localization, VoteSplat maps 2D image semantics to 3D point clouds via voting points, reducing training costs associated with high-dimensional CLIP features while preserving semantic unambiguity. Extensive experiments demonstrate VoteSplat’s effectiveness in open-vocabulary 3D instance localization, 3D point cloud understanding, click-based 3D object localization, hierarchical segmentation, and ablation studies. Our code is available at [VoteSplat](#).

1. Introduction

Localization and instance-level semantic understanding of 3D scenes are critical objectives in the computer vision community, especially with the rise of embodied intelligence. Traditional point cloud-based methods for 3D scene understanding have been widely explored, including classification [21], segmentation [22], and detection [23]. Re-

*Contribute equally.

†Corresponding author.

cently, 3D Gaussian Splatting (3DGS) [10] has gained popularity for its ability to achieve real-time, photorealistic novel view synthesis (NVS) at high resolutions. Unlike implicit Neural Radiance Fields (NeRF) [20], which represent scenes as continuous volumetric functions, 3DGS reconstructs scene appearance and geometry using point-clouds-like Gaussian primitives. Building on its success, 3DGS has been extended to various domains, including rendering [6, 18, 39], surface reconstruction [4, 9, 40], generation [37], and scene understanding [24, 41]. Moreover, scene reconstruction and semantic understanding can be seamlessly integrated with 3DGS rendering pipeline, facilitating more advanced intelligent agent interaction and decision-making.

Efforts have aimed to integrate learnable semantic information into 3DGS, enhancing the language-grounded capabilities of 3DGS. Current approaches can be broadly categorized into the following two types:

(i) The first approach [24, 25, 29, 36] directly embeds semantic vectors into Gaussian primitives, project semantic information onto images in the same way as rendering colors. This enables cross-frame association through semantics but has two key limitations [35]: (1) The high dimensionality of CLIP-extracted features leads to excessive training overhead, while dimensionality reduction introduces semantic ambiguity; (2) The object occlusion restricts them to pixel-level segmentation, where embedded semantic vectors are insufficient for point-level scene understanding.

(ii) The second approach [5, 35] adopts a point cloud clustering strategy to improve instance differentiation and reduce ambiguity, following a multi-stage pipeline. 3DGS independently reconstructs the scene and embeds feature vectors into the trained Gaussian primitives, using contrastive learning for instance differentiation. The goal is to address the spatial adjacency of primitives belonging to different instances, with additional features for better separation. However, contrastive learning adds significant computational complexity to the rendering pipeline.

To this end, we propose VoteSplat, that integrates Hough voting with Gaussian splatting for 3D scene understanding. VoteSplat defines a set of 3D Gaussians embedded with additional three-dimensional spatial offset vectors to compute spatial 3D votes. We expect 3D votes to be located at the centroid of the instances so that clustering methods can be directly applied to distinguish each instance without the need for additional learned features for differentiation. Given the Gaussian primitives forming a specific instance and its centroid are unknown, we propose 3D-2D Votes Association Learning as the centroid may exist in multi-view two-dimensional images.

Specifically, for each image, we first apply the Segment Anything Model (SAM) [12] to obtain well-segmented masks and compute their centroids with precise object

boundaries. The pixel coordinates of these centroids serve as 2D ground-truth votes to supervise projection of 3D votes, encouraging convergence toward the instance center while ensuring multiview consistency. Since projection transformations cause depth information loss, relying solely on 2D vote supervision can introduce noise in spatial voting points. To overcome this, we introduce a depth distortion regularization term to improve spatial vote aggregation along the depth dimension.

In a trained VoteSplat scene, each Gaussian primitive surrounding an object has an offset vector pointing to a 3D vote near the instance centroid. Clustering these votes allows us to effectively determine instance IDs to the Gaussian primitives, which in turn establishes correspondences between point clouds and image semantics, enabling robust 3D scene understanding. The contributions can be thus summarized as

1. Hough voting is first considered into 3D Gaussian Splatting (3DGS) to achieve spatial clustering of point clouds belonging to the same instance. This enables point-level segmentation without requiring additional high-dimensional feature vectors, improving training efficiency and scene understanding accuracy.
2. 3D-2D Votes Association Learning is proposed, incorporating a custom depth distortion loss to enhance spatial aggregation of voting points along the depth dimension for denoising.
3. An instance ID-based approach is introduced to associate 2D image semantics with 3D point clouds, enabling the linking of CLIP features to individual 3D instances, and facilitating open-vocabulary scene understanding.

Therefore, VoteSplat enables efficient point-level segmentation without requiring high-dimensional feature embeddings. Additionally, we introduce 3D-2D Votes Association Learning and depth distortion regularization to refine spatial clustering and improve localization accuracy. The rest of the paper is organized as follows: Section 2 reviews related works in 3D Gaussian Splatting and Hough Voting. Section 3 details our methodology, including 3D vote construction and semantic association. Section 4 presents the experimental setup and results across various tasks. Finally, Section 5 concludes the paper.

2. Related Works

3D Gaussian Splatting for Scene Understanding. 3D Gaussian Splatting has emerged as a promising method for real-time scene rendering, offering superior visual quality. Utilize 3DGS to jointly reconstruct the appearance and geometric information of a scene with instance and semantic information, to better support downstream tasks.

NeRF, as an innovative 3D reconstruction method, has inspired numerous works [31, 42] to develop 3D language

fields upon it. LERF[11] first integrated CLIP[3] features into NeRF, constructing language embedded radiance fields to enable open-vocabulary 3D querying. Additionally, DINO features were used to enhance boundary accuracy. However, due to high computational costs, NeRF-based methods face rendering performance bottlenecks, leading to the adoption of 3DGS [10] with semantic information. Building on 3DGS, LangSplat[24] employs an autoencoder to reduce CLIP feature dimensionality, embedding the compressed representations into Gaussian primitives, while incorporating a semantic hierarchy. Shi et al. [29] use VQ-VAE to quantize high-dimensional CLIP features into discrete categories, converting semantic supervision into category-level supervision, thereby reducing computational overhead. Similarly, Shorinwa et al. [30] constructs a 3D language field by mapping semantic features to discrete categories.

These methods achieve multi-view training through cross-frame semantic similarity. To reduce computational complexity, CLIP feature needs to be either dimensionally reduced or classified, which inevitably introduces semantic ambiguity. Other methods [5, 16, 35, 36] rely on learnable features to distinguish instances and ultimately assign complete semantic information. These methods ensure semantic accuracy; however, intra-class and inter-class contrastive learning can be highly time-consuming.

Hough Voting for 3D Point Clouds. The Hough transform (also, Hough voting), originates in late 1950s [8], convert the detection of simple patterns in point samples as peak detection in a parametric space. The Generalized Hough Transform [1] extends this concept to image patches, enabling the identification of complex objects. Hough voting has been widely applied in various tasks, including the implicit shape model[15], plane extraction from 3D point clouds[2], 6D pose estimation [32], and so forth.

Hough voting has been successfully integrated with advanced learning techniques. In 3D object detection, a common approach [13, 14, 33, 34] is to adapt mature 2D detection algorithms, such as Faster R-CNN[27] or YOLO[26], to 3D point clouds by generating proposals at each input point. However, a fundamental challenge arises: 3D sensors capture only surface data, meaning the object center often lies in empty space, far from the available points in the input point cloud. As a result, there are typically no input points near the object center, making it difficult for surface-based networks to extract meaningful contextual information, leading to inaccurate proposals. To overcome this, Qi et al. [23] introduced VoteNet, a Hough voting-based method. VoteNet first samples seed points from the input point cloud and votes for the target’s center, generating voting points near the object center. These voting points are then used to generate bounding box proposals, effec-

tively addressing the issue of inaccurate proposals when the object center is distant from the surface points.

3. Method

We now formally introduce VoteSplat, a 3D scene understanding framework for 3DGS representations, incorporating effective and distinctive voting mechanisms. We first employ SAM’s automatic mask generation module to produce masks for training views of the scene. The resulting multi-level mask information is then used to compute 2D ground-truth votes (detailed in Section. 3.2). Next, we embed offset vectors into 3DGS and train them to generate 3D votes (Section. 3.3). To improve depth consistency across different views, we introduce a depth regularization term, ensuring vote point aggregation during training. Finally, we establish a mapping between Gaussians and semantic information (Section. 3.4). The complete VoteSplat pipeline is illustrated in Figure 2.

3.1. Recap: 3D Gaussian Splatting

Compared to implicit Neural Radiance Fields (NeRF) [20], 3D Gaussian Splatting (3DGS) [39] constructs 3D scenes using explicit 3D Gaussian primitives. These primitives are represented as point clouds with associated attributes and are rendered through a tile-based differentiable rasterizer.

Given a training set of K images $I = \{I_k\}_{k=1}^K$ with associated camera poses, and an image resolution of $H \times W$, the goal is to learn a set of N three-dimensional Gaussian, denoted as $G = \{g_i\}_{i=1}^N$. Each Gaussian g_i is characterized by quintet $g_i := \{\mathbf{p}_i, \mathbf{s}_i, \mathbf{q}_i, o_i, \mathbf{c}_i\}$ of trainable parameters: the center position $\mathbf{p}_i \in \mathbb{R}^3$, the scaling factor $\mathbf{s}_i \in \mathbb{R}^3$, quaternion representing Gaussian’s 3D covariance $\mathbf{q}_i \in \mathbb{R}^4$, opacity value $o_i \in \mathbb{R}$, and $\mathbf{c}_i \in [0, 1]^3$ encodes RGB color using spherical harmonics coefficients.

After projecting 3D Gaussians onto the 2D image space under a given camera pose, 3DGS computes the pixel color \mathbf{C} using its differentiable rasterizer. The color is determined via α -blending across \mathcal{N} depth-ordered points overlapping the pixel, given by

$$\mathbf{C} = \sum_{i \in \mathcal{N}} \mathbf{c}_i \alpha_i T_i, \quad (1)$$

where α_i is determined by evaluating the influence of each projected Gaussian based on its splatted 2D covariance Σ [38], opacity o_i , and distance \mathbf{d} to the pixel that reads

$$\alpha_i = o_i \exp(-\frac{1}{2} \mathbf{d}^T \Sigma^{-1} \mathbf{d}). \quad (2)$$

The transmittance $T_i := \prod_{j=1}^{i-1} (1 - \alpha_j)$, represents the accumulated visibility of i -th Gaussian, accounting for occlusion from previously processed Gaussians in-depth order.

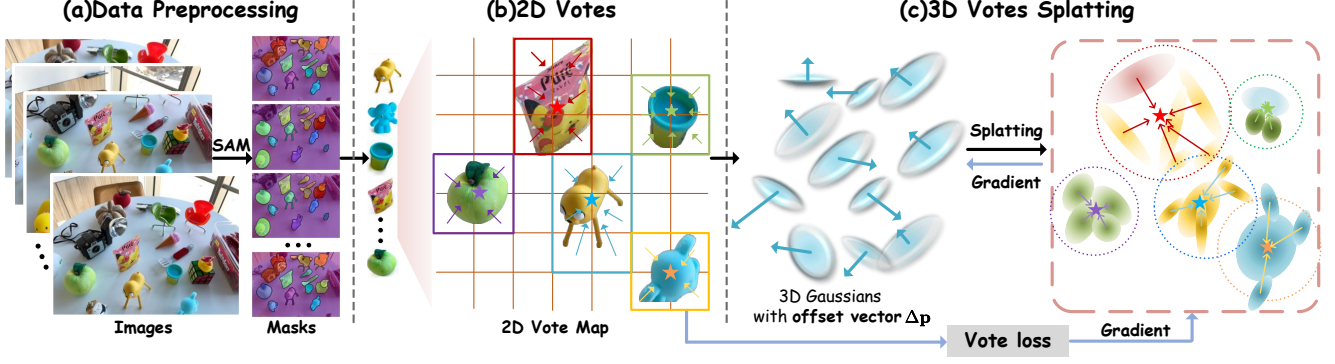


Figure 2. Three main steps in VoteSplat pipeline: (a) We first deploy SAM to automatically generate segmentation masks for all instances independently across different views. (b) For each segmented mask, we compute the instance center to construct the 2D Vote Map. (c) By projecting 3D votes into pixel space through splatting and computing the voting loss together with the previous 2D voting map, each Gaussian primitive forming an instance is ensured to learn an offset vector pointing toward the instance center. For simplicity, we omit the rendering process and density control of other Gaussian parameters, as these are inherited from [10].

3.2. 2D Vote Construction

3D point clouds, such as those obtained from radar or Structure-from-Motion (SfM) [28], typically exhibit a concentration of points on the surface of spatial instances, with sparser distributions toward the instance centers. This property persists during the densification process of 3DGS, wherein the generated point clouds remain concentrated on the instance surfaces. Consequently, directly clustering 3DGS point clouds can introduce boundary ambiguity between instances, leading to clustering errors when instance boundaries are in close proximity. Choi et al. [5], Shi et al. [29] embed additional feature vectors into the Gaussian Splatting framework to distinguish instances to avoid this. While effective, these methods often rely on computationally expensive contrastive learning. In contrast, 2D images inherently preserve structural information about instances and consistently capture object centers (provided the instance lies within the field of view). Exploiting this advantage, we propose a method to infer 3D instance centers from their corresponding 2D instance centers. We then detail the computation of the 2D centers (2D votes).

SAM effectively groups pixels belonging to the same instance and segment images into multiple object masks with well-defined boundaries. Following Lang-Splat, we utilize SAM to obtain precise hierarchical object masks. To improve the accuracy of 2D votes, we further filter out masks whose boundaries extend beyond the field of view (FoV).

For a given instance mask at level l , denoted as M_l , we compute the x -axis instance centroid c_x^l by

$$c_x^l = \text{round} \left(\frac{\sum_{y=0}^{H-1} \sum_{x=0}^{W-1} x \cdot M_l(x, y)}{\sum_{y=0}^{H-1} \sum_{x=0}^{W-1} M_l(x, y)} \right), \quad (3)$$

recall H and W are the resolution of the image and $\text{round}(\cdot)$ is the rounding function and y -axis centroid c_y^l is computed

similarly. the 2D vote \mathbf{V}_i^{2d} is thus defined as the doublet

$$\mathbf{V}_i^{2d}(x, y) := \{c_x^l, c_y^l\}. \quad (4)$$

Each pixel within the mask is then assigned to its corresponding 2D vote \mathbf{V}_i^{2d} , forming the ground-truth vote map, that serves as supervision for subsequent learning stages.

3.3. 3D Vote Construction

3D Vote Splatting. In 3DGS, both the initial and densified point clouds predominantly lie on instance surfaces, making direct clustering-based instance separation challenging. To overcome this, we introduce an offset vector $\Delta \mathbf{p}_i \in \mathbb{R}^3$ for each Gaussian primitive, allowing the point cloud to vote toward instance centers. Consequently, the 3D vote reads

$$\mathbf{V}_i^{3d} = \Delta \mathbf{p}_i + \mathbf{p}_i. \quad (5)$$

The additional vector attributes \mathbf{V}_i^{3d} are typically optimized using the same α -blending approach as color rendering, i.e.,

$$\mathbf{V}^{3d} = \sum_{i \in \mathcal{N}} \mathbf{V}_i^{3d} \alpha_i T_i. \quad (6)$$

While generally effective, this approach faces two key challenges when applied to spatial constraints:

- (i) **Unequal depth-weighting contribution:** In 3D space, all points forming an instance should contribute equally to its center vote. However, the traditional α -weighting mechanism disproportionately reduces the influence of farther points, leading to biased voting.
- (ii) **Incorrect occlusion Handling:** In 3DGS, points behind an instance still affect the final color rendering but should not be involved in the voting process, as they do not belong to the instance.

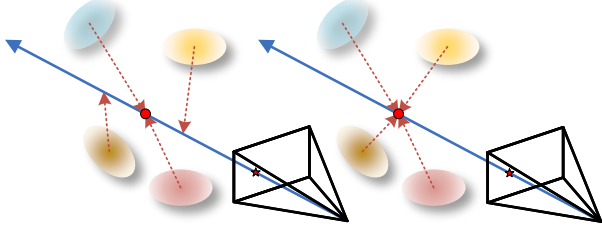


Figure 3. Since projection results in-depth information loss, projected 3D votes may align correctly on the pixel plane but exhibit significant depth deviations, as shown in the left subfigure. The proposed depth regularization enforces spatial proximity along the depth dimension, keeping voting points spatial proximity in 3D space and enhancing overall voting consistency.

We introduce a distinct voting transmittance model to overcome this, applying uniform averaging in α -blending as

$$\mathbf{V}^{3d} := \frac{1}{|\mathcal{M}|} \sum_{i \in \mathcal{M}} \mathbf{V}_i^{3d}, \quad (7)$$

where \mathcal{M} represents set of the depth-ordered points under the voting transmittance \hat{T}_i and $|\mathcal{M}|$ as its cardinality. Next, we project the blended 3D votes \mathbf{V}^{3d} into screen space:

$$\tilde{\mathbf{V}}^{2d} = \mathbf{H}\mathbf{V}^{3d}, \quad (8)$$

where \mathbf{H} is 4-by-4 transformation matrix from world space to screen space. Notably, blending is performed in 3D space before projection, ensuring stable voting. Otherwise, distant votes would experience significant fluctuations, making convergence toward the instance center less reliable.

The vote loss with respect to the precomputed 2D votes \mathbf{V}^{2d} is defined as

$$\mathcal{L}_{vote} := \frac{1}{|\mathcal{P}|} \sum_{i \in \mathcal{P}} |\tilde{\mathbf{V}}_i^{2d} - \mathbf{V}_i^{2d}|, \quad (9)$$

where \mathcal{P} denotes the set of pixels within the mask that contain 2D votes.¹ Enforcing this loss ensures that offset vectors effectively guide Gaussians toward instance centers, while preserving the efficiency of the rasterization pipeline.

Depth Regularization. Projecting 3D votes to 2D votes inherently results in a loss of depth information. While training with multi-view images helps reduce depth uncertainty and encourages the point cloud to collectively vote toward the instance center, disturbances in the voting points may still persist, as shown in Figure 3.

Inspired by depth distortion in 2DGS [9], which promotes the concentration of Gaussian primitives, a parallel strategy is applied to 3D votes to preserve their depth alignment around the instance center. The initial depth distortion

formulation in VoteSplat reads

$$\mathcal{L}_d = \sum_{i,j} \omega_i \omega_j |z_i - z_j|. \quad (10)$$

where z is the \mathbf{V}^{3d} depth in the camera coordinate system with weight $\omega_i := \alpha_i T_i$.

Recall that the votes of Gaussian primitives need to be equally concentrated in-depth (av), the weight should not be reduced, and the formula is modified as

$$\mathcal{L}_d^* = \sum_{i,j} |z_i - z_j|. \quad (11)$$

by eliminating the weights from (10).

Reconstruction-Voting-Depth (RVD) Loss: Lastly, the model optimizes the sparse point cloud, shifting it from its initial positions in the posed images toward a more concentrated target with respect to the RVD loss, i.e.,

$$\mathcal{L} = \mathcal{L}_c + \lambda_{vote} \mathcal{L}_{vote} + \lambda_{L_d} \mathcal{L}_d^*, \quad (12)$$

where $\mathcal{L}_c(\mathcal{L}_1, \text{D-SSIM})$ is the combination of RGB reconstruction loss \mathcal{L}_1 [10, Eq. 7] and D-SSIM term from [10, Eq. 7]. The vote loss (9) and depth regularization loss (11) penalized by their corresponding weights λ_{vote} and λ_{L_d} , respectively, both depends on instances' size in the scene.

3.4. Semantic Construction

By constructing 3D votes, spatially separated voting points are obtained. To facilitate natural, open-vocabulary interactions, effectively associating 3D Gaussians with language features is essential. Thus, we propose an instance-level 3D-2D semantic association method based on voting points and instance IDs. The approach can be detailed as follows.

- **Background Filtering:** Point clouds with an offset vector $\Delta \mathbf{p} = \mathbf{0}$ are removed, as they are considered background and do not contribute to instance construction, receiving no gradient updates.
- **Clustering and Instance Dictionary Construction:** The remaining point clouds are clustered using HDBSCAN [19] based on 3D votes, which performs density-based clustering while filtering out outliers. A dictionary is then built, where instance IDs serve as keys and the corresponding point cloud IDs as values.
- **Rendering Instance ID Maps:** The 3DGS rasterization pipeline is used to render the instance ID map. Combining this with the original RGB images, the pixel regions corresponding to each instance ID are identified.
- **Semantic Association with CLIP Features:** CLIP image features are extracted from the associated pixels, establishing a mapping between instance IDs and CLIP features, with multi-view feature integration incorporated for improved consistency.

¹ Throughout the paper, we adopted the notation $|\cdot|$ for ℓ_1 norm.

Methods	mIoU \uparrow				Mean	mAcc. \uparrow				Mean
	figurines	teatime	ramen	waldo_kitchen		figurines	teatime	ramen	waldo_kitchen	
LangSplat	10.16	11.38	7.92	9.18	9.66	8.93	20.34	11.27	9.09	12.41
OpenGaussian	60.11	65.80	31.01	22.70	44.90	82.14	79.66	42.25	31.92	58.99
VoteSplat	68.62	66.71	39.24	25.84	50.10	85.71	88.14	61.97	33.68	67.38

Table 1. Performance of semantic segmentation on the LeRF dataset compared to LangSplat and OpenGaussian based on text query. Accuracy is measured by mAcc@0.25.



Figure 4. Click-based 3D object selection and scene editing results. VoteSplat enables complete 3D object selection without issues of incompleteness or redundancy. Moreover, after removing the selected Gaussian primitives, the scene can be effectively edited.

Methods	snacks	figurines	teatime	ramen
LS(Level1)	~ 67	~ 116	~ 87	~ 56
OG	~ 114	~ 117	~ 104	~ 55
GG	-	~ 150	~ 122	~ 90
VS	$\sim \mathbf{57}$	$\sim \mathbf{54}$	$\sim \mathbf{43}$	$\sim \mathbf{53}$

Table 2. The training time (in minutes) of LangSplat (LS), OpenGaussian (OG), GaussianGrouping (GG), and VoteSplat (VS).²

4. Experiments

4.1. Open-Vocabulary Object Selection in 3D Space

Experimental Setup. (i) **Objective:** Given an open-vocabulary text query, CLIP extracts textual features, and cosine similarity is computed with each instance ID’s language features. The most relevant instances are selected, and their Gaussian primitives are rendered into multi-view images via the 3DGS rasterization pipeline; (ii) **Baseline:** Our method is compared against LangSplat, Gaussian Grouping, and OpenGaussian. VoteSplat follows the method described in Section 3.4 to associate each instance with a 512-dimensional CLIP feature and select the corresponding Gaussian primitives for rendering. For LangSplat and OpenGaussian, we adhere to their respective procedures. In LangSplat, the 512-dimensional CLIP feature is reconstructed from the low-dimensional language feature of each Gaussian. In OpenGaussian, cosine similarity is computed to select the corresponding Gaussian primitives for rendering. Since Gaussian Grouping does not inherently support semantic queries on Gaussian primitives and is limited to instance segmentation, it is excluded from semantic query comparisons; (iii) **Dataset and Met-**

rics: Experiments are conducted on 3D-OVS[17] and Lerf-OVS[11], with all datasets annotated by LangSplat. Performance is evaluated using average IoU and segmentation accuracy, measuring the alignment between rendered images (from selected 3D Gaussian points) and ground-truth object masks. Additionally, we report training time across different methods and provide feature visualizations of the point clouds.

Result. Table 1 shows VoteSplat outperforms other methods in both mIoU and mAcc. LangSplat, with weaker 3D understanding, struggles to accurately associate 3D Gaussian points with query text. This suggests that 2D image semantics derived from α -blending fail to effectively capture 3D semantics encoded by Gaussian primitives. Table 2 reports the training time for each model, evaluated over 60,000 iterations on NVIDIA RTX 3090 GPU. VoteSplat achieves the shortest training time among all methods, benefiting from its efficient voting mechanism. In contrast, OpenGaussian’s intra-/inter-class contrastive learning significantly increases training time. Gaussian Grouping [36] relies on KNN for feature consistency, leading to high computational and memory complexity during training.

Qualitative results are presented in Figure 5. Given a text query, VoteSplat selects relevant Gaussian points and renders them into multi-view images. Due to the ambiguity of 3D point features, LangSplat struggles to accurately recognize target objects, while OpenGaussian fails to capture fine-grained details as effectively as VoteSplat. For example, with the prompt “old camera,” VoteSplat successfully clusters finer details, such as the rope. Additionally, VoteSplat outperforms other methods in rendering occluded ob-

²The running time is measured in minutes, with seconds omitted, denoted by the symbol \sim .

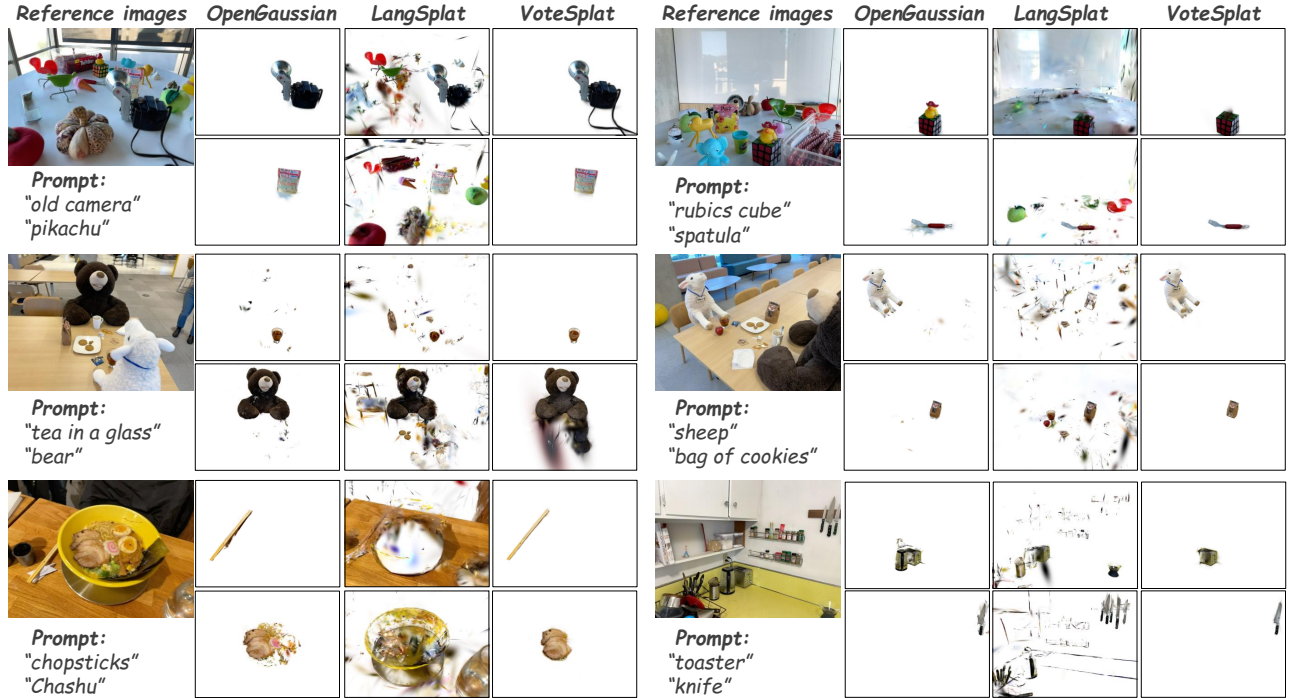


Figure 5. Open-vocabulary 3D object selection on the LERF dataset. VoteSplat outperforms LangSplat and OpenGaussian in accurately identifying 3D objects corresponding to text queries.

jects. In the teatime scene, it reconstructs the bear’s lower body, despite being partially obscured by the table. Moreover, VoteSplat generates images with fewer noise artifacts, enhancing overall rendering quality.

Figure 6 visualizes the point cloud features, where VoteSplat assigns distinct colors to instance categories for clarity. In OpenGaussian, colors are derived by applying PCA to reduce feature dimensions to three, and then mapping them to RGB. In contrast, LangSplat directly uses its three-dimensional features as point cloud colors. The feature visualization for Gaussian Grouping is provided in the supplementary material. The well-segmented instances in VoteSplat demonstrate its superior performance.

4.2. Click-based 3D Object Selection and Editing

Given an image from any viewpoint, clicking on a 2D pixel selects the corresponding 3D Gaussian points. The instance ID associated with the selected Gaussian primitive is then retrieved, enabling click-based object selection. Additionally, removing the entire instance allows for scene editing effects. Figure 4 demonstrates click-based object selection and scene editing on the LERF dataset. The left image highlights instance segmentation under occlusion, while the right image focuses on small object selection.

4.3. Hierarchical Segmentation

In some cases, instance segmentation is sufficient, but certain applications require finer segmentation, such as part segmentation for more precise analysis. VoteSplat supports this functionality by utilizing multi-level masks from SAM to compute hierarchical 2D votes. These votes generate layered 2D voting maps, which then supervise 3D votes, enabling finer-grained segmentation. As shown in Figure 7, hierarchical 3D votes and rendering results are presented on the LLFF [20] and LeRF datasets.

4.4. Instance Segmentation in Complex Scenes

To evaluate VoteSplat in complex scenarios, experiments are conducted on GraspNet dataset [7], where instances are overlapping, adjacent, and contained. Despite instances’ proximity, VoteSplat successfully segments them. As shown in Figure 8(b), 3D votes remain well-separated, demonstrating the method’s effectiveness.

4.5. Ablation Study

As discussed in Section 3.3, projection inherently leads to depth information loss. While the projected points may align with 2D votes on the pixel plane, they can exhibit significant depth discrepancies. This issue is more pronounced in forward-facing data compared to 360-degree captures, as the absence of side-view images prevents voting points from consistently converging toward the instance center. Figure 9

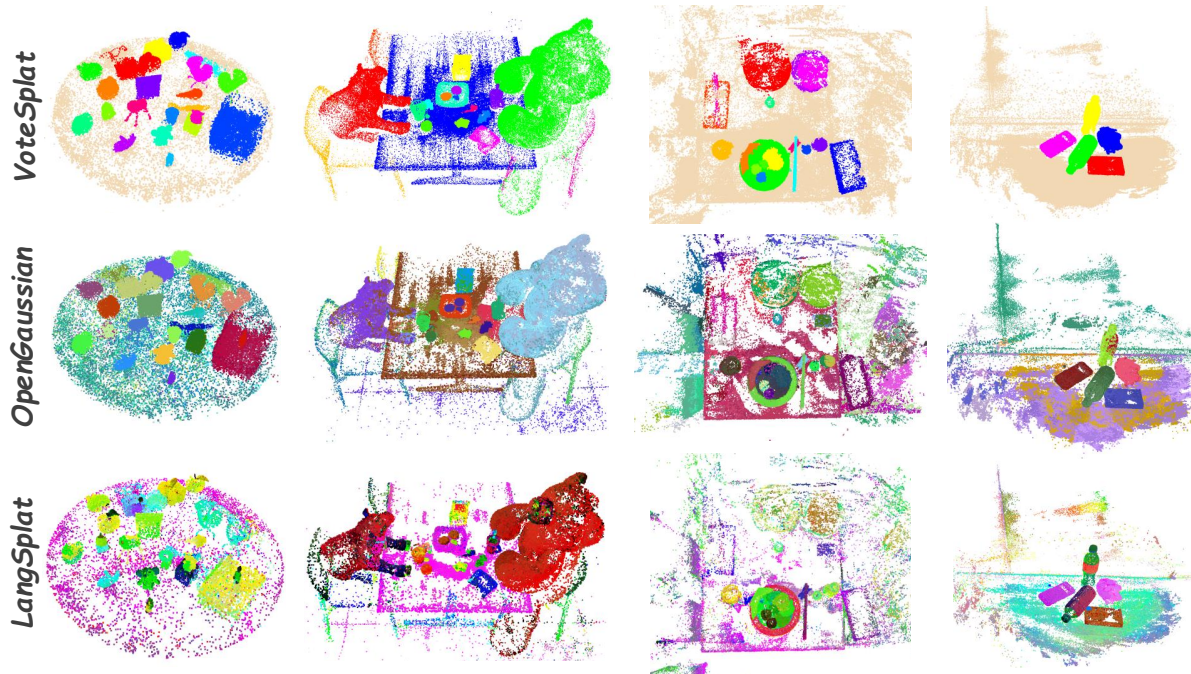


Figure 6. Comparison of point cloud feature visualizations. From left to right, the scenes correspond to *figurines*, *teatime*, *ramen*, and *snacks*. The first three scenes are from LeRF, and the last scene is from 3D-OVS. Our proposed method, VoteSplat, demonstrates superior performance in terms of feature granularity and accuracy.



Figure 7. Using SAM, objects are divided into multiple parts, each assigned a 2D vote. After training, VoteSplat generates a corresponding 3D vote for each component. The rendering results of these components are then visualized.

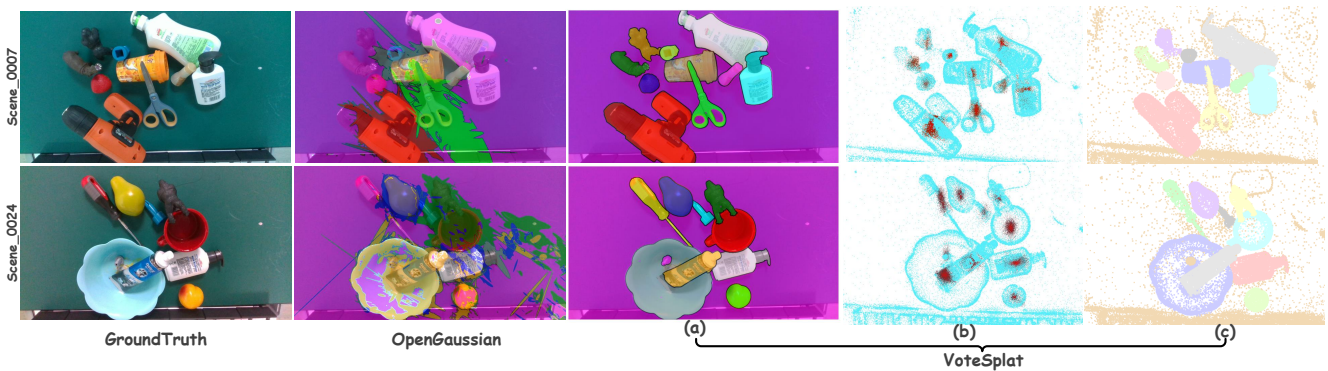


Figure 8. Instance segmentation in complex scenes. Compared with Open-Gaussian, VoteSplat can handle more complex scenarios, such as instance overlap and contain each other.

illustrates the impact of \mathcal{L}_d on the 3D-OVS dataset. Without \mathcal{L}_d , the point cloud appears dispersed along the depth dimension, whereas with \mathcal{L}_d , the points are more tightly

clustered. Similarly, applying projection before accumulation results in 3D vote dispersion. Additionally, when using transmittance T , background points participate in voting,

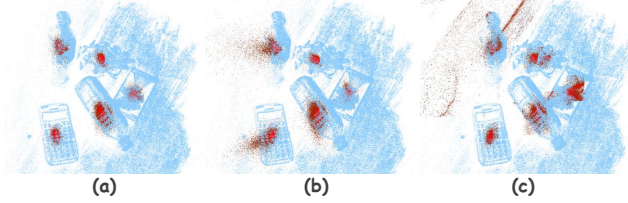


Figure 9. Comparison of ablation experiments. (a) shows 3D votes of VoteSplat are precisely located at the instance center. (b) reflects the effects of $w/o \mathcal{L}_d$ and the consequence of projecting first and then accumulating V_i^{3d} . (c) illustrates the impact of using transmittance T .

Case	Eq. 7	L_d	mIoU(%) \uparrow	mAcc. \uparrow
#1	✓		76.04	0.88
#2		✓	58.72	0.72
#3	✓	✓	85.66	0.96

Table 3. Quantitative ablations result on the snacks scene of the 3D-OVS dataset.

introducing clustering disturbances, as in Figure 9(c). The quantitative results are reported in Table 3. It also demonstrates that each component of our method contributes to segmentation performance. In particular, allowing background Gaussians to participate in the voting process leads to a significant degradation in segmentation accuracy, highlighting the importance of explicitly filtering them out.

5. Concluding Remarks

We introduce VoteSplat, a novel 3D scene understanding method that integrates Hough voting with 3D Gaussian Splatting (3DGS). Utilizing SAM for image instance segmentation, we generate 2D vote maps to supervise 3D votes, which are computed through embedded spatial offset vectors. To further refine clustering, we introduce depth distortion, constraining spatial offsets along the depth dimension, ensuring Gaussian primitives are well-clustered in 3D space. Additionally, by projecting instance IDs, VoteSplat establishes a precise correspondence between Gaussian primitives and 2D image semantics, effectively resolving semantic ambiguities. Experimental results confirm its effectiveness across various tasks. The voting mechanism encounters challenges when instances have projected sizes that significantly exceed the field of view (FoV), leading to inaccuracies in 2D votes. Additionally, it may also struggle with instances enclosed in concave containers, where spatial proximity makes precise separation difficult.

Acknowledgement

The authors sincerely thank the anonymous reviewers for their valuable comments and suggestions. This

work was supported by grants from the Natural Science Foundation of Shanxi Province (2024JCJCQN-66), Science and Technology Commission of Shanghai Municipality (NO.24511106900), and is partially supported by the National Natural Science Foundation of China under grant Nos. 62072358 and 62072352.

References

- [1] Dana H. Ballard. Generalizing the Hough transform to detect arbitrary shapes. *Pattern recognition*, 13(2):111–122, 1981. 3
- [2] Dorit Borrmann, Jan Elseberg, Kai Lingemann, and Andreas Nüchter. The 3D Hough transform for plane detection in point clouds: A review and a new accumulator design. *3D Research*, 2(2):1–13, 2011. 3
- [3] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9650–9660, 2021. 3
- [4] Danpeng Chen, Hai Li, Weicai Ye, Yifan Wang, Weijian Xie, Shangjin Zhai, Nan Wang, Haomin Liu, Hujun Bao, and Guofeng Zhang. PGSR: Planar-based gaussian splatting for efficient and high-fidelity surface reconstruction. *arXiv:2406.06521*, 2024. 2
- [5] Seokhun Choi, Hyeonseop Song, Jaechul Kim, Taehyeong Kim, and Hoseok Do. Click-gaussian: Interactive segmentation to any 3D gaussians. In *European Conference on Computer Vision*, pages 289–305. Springer, 2024. 2, 3, 4
- [6] Yitong Dong, Yijin Li, Zhaoyang Huang, Weikang Bian, Jingbo Liu, Hujun Bao, Zhaopeng Cui, Hongsheng Li, and Guofeng Zhang. A global depth-range-free multi-view stereo transformer network with pose embedding. *arXiv preprint arXiv:2411.01893*, 2024. 2
- [7] Hao-Shu Fang, Chenxi Wang, Minghao Gou, and Cewu Lu. Graspnet-1billion: A large-scale benchmark for general object grasping. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11444–11453, 2020. 7
- [8] Paul V.C. Hough. Machine analysis of bubble chamber pictures. In *International Conference on High Energy Accelerators and Instrumentation, CERN, 1959*, pages 554–556, 1959. 3
- [9] Binbin Huang, Zehao Yu, Anpei Chen, Andreas Geiger, and Shenghua Gao. 2D gaussian splatting for geometrically accurate radiance fields. In *ACM SIGGRAPH 2024 conference papers*, pages 1–11, 2024. 2, 5
- [10] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3D gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):139–1, 2023. 2, 3, 4, 5
- [11] Justin Kerr, Chung Min Kim, Ken Goldberg, Angjoo Kanazawa, and Matthew Tancik. Lrf: Language embedded radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19729–19739, 2023. 3, 6

- [12] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4015–4026, 2023. 2
- [13] Jan Knopp, Mukta Prasad, and Luc Van Gool. Orientation invariant 3D object classification using Hough transform based methods. In *Proceedings of the ACM workshop on 3D object retrieval*, pages 15–20, 2010. 3
- [14] Jan Knopp, Mukta Prasad, and Luc Van Gool. Scene cut: Class-specific object detection and segmentation in 3D scenes. In *2011 International Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission*, pages 180–187. IEEE, 2011. 3
- [15] Bastian Leibe, Aleš Leonardis, and Bernt Schiele. Robust object detection with interleaved categorization and segmentation. *International journal of computer vision*, 77:259–289, 2008. 3
- [16] Haijie Li, Yanmin Wu, Jiarui Meng, Qiankun Gao, Zhiyao Zhang, Ronggang Wang, and Jian Zhang. Instancegaussian: Appearance-semantic joint gaussian representation for 3D instance-level perception. *arXiv:2411.19235*, 2024. 3
- [17] Kunhao Liu, Fangneng Zhan, Jiahui Zhang, Muyu Xu, Yingchen Yu, Abdulmotaleb El Saddik, Christian Theobalt, Eric Xing, and Shijian Lu. Weakly supervised 3D open-vocabulary segmentation. *Advances in Neural Information Processing Systems*, 36:53433–53456, 2023. 6
- [18] Tao Lu, Mulin Yu, Linning Xu, Yuanbo Xiangli, Limin Wang, Dahua Lin, and Bo Dai. Scaffold-GS: Structured 3D gaussians for view-adaptive rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20654–20664, 2024. 2
- [19] Leland McInnes, John Healy, and Steve Astels. hdbscan: Hierarchical density based clustering. *J. Open Source Softw.*, 2(11):205, 2017. 5
- [20] 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. *Communications of the ACM*, 65(1):99–106, 2021. 2, 3, 7
- [21] Charles R. Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. Pointnet: Deep learning on point sets for 3D classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017. 1
- [22] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J. Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30, 2017. 1
- [23] Charles R. Qi, Or Litany, Kaiming He, and Leonidas J. Guibas. Deep Hough voting for 3D object detection in point clouds. In *proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9277–9286, 2019. 1, 3
- [24] Minghan Qin, Wanhua Li, Jiawei Zhou, Haoqian Wang, and Hanspeter Pfister. Langsplat: 3D language gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20051–20060, 2024. 2, 3
- [25] Jiaxiong Qiu, Liu Liu, Zhizhong Su, and Tianwei Lin. GLS: Geometry-aware 3D language gaussian splatting. *arXiv preprint arXiv:2411.18066*, 2024. 2
- [26] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016. 3
- [27] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE transactions on pattern analysis and machine intelligence*, 39(6):1137–1149, 2016. 3
- [28] Johannes L. Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4104–4113, 2016. 4
- [29] Jin-Chuan Shi, Miao Wang, Hao-Bin Duan, and Shao-Hua Guan. Language embedded 3D gaussians for open-vocabulary scene understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5333–5343, 2024. 2, 3, 4
- [30] Ola Shorinwa, Jiankai Sun, and Mac Schwager. Fast-Splat: Fast, ambiguity-free semantics transfer in gaussian splatting. *arXiv:2411.13753*, 2024. 3
- [31] Yawar Siddiqui, Lorenzo Porzi, Samuel Rota Buló, Norman Müller, Matthias Nießner, Angela Dai, and Peter Kotschieder. Panoptic lifting for 3D scene understanding with neural fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9043–9052, 2023. 2
- [32] Min Sun, Gary Bradski, Bing-Xin Xu, and Silvio Savarese. Depth-encoded Hough voting for joint object detection and shape recovery. In *European Conference on Computer Vision*, pages 658–671. Springer, 2010. 3
- [33] Alexander Velizhev, Roman Shapovalov, and Konrad Schindler. Implicit shape models for object detection in 3D point clouds. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 1:179–184, 2012. 3
- [34] Oliver J. Woodford, Minh-Tri Pham, Atsuto Maki, Frank Perbet, and Björn Stenger. Demisting the Hough transform for 3D shape recognition and registration. *International Journal of Computer Vision*, 106:332–341, 2014. 3
- [35] Yanmin Wu, Jiarui Meng, Haijie Li, Chenming Wu, Yahao Shi, Xinhua Cheng, Chen Zhao, Haocheng Feng, Errui Ding, Jingdong Wang, and Jian Zhang. OpenGaussian: Towards point-level 3D gaussian-based open vocabulary understanding. *arXiv:2406.02058*, 2024. 2, 3
- [36] Mingqiao Ye, Martin Danelljan, Fisher Yu, and Lei Ke. Gaussian grouping: Segment and edit anything in 3D scenes. In *European Conference on Computer Vision*, pages 162–179. Springer, 2024. 2, 3, 6
- [37] Taoran Yi, Jiemin Fang, Guanjun Wu, Lingxi Xie, Xiaopeng Zhang, Wenyu Liu, Qi Tian, and Xinggang Wang. GaussianDreamer: Fast generation from text to 3D gaussian splatting with point cloud priors. *arXiv:2310.08529*, 2023. 2

- [38] Wang Yifan, Felice Serena, Shihao Wu, Cengiz Öztireli, and Olga Sorkine-Hornung. Differentiable surface splatting for point-based geometry processing. *ACM Transactions On Graphics (TOG)*, 38(6):1–14, 2019. [3](#)
- [39] Zehao Yu, Anpei Chen, Binbin Huang, Torsten Sattler, and Andreas Geiger. Mip-splatting: Alias-free 3D gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19447–19456, 2024. [2](#), [3](#)
- [40] Zehao Yu, Torsten Sattler, and Andreas Geiger. Gaussian opacity fields: Efficient and compact surface reconstruction in unbounded scenes. *arXiv:2404.10772*, 2024. [2](#)
- [41] Hongjia Zhai, Hai Li, Zhenzhe Li, Xiaokun Pan, Yijia He, and Guofeng Zhang. Panogs: Gaussian-based panoptic segmentation for 3d open vocabulary scene understanding. In *Proceedings of the Computer Vision and Pattern Recognition Conference (CVPR)*, pages 14114–14124, 2025. [2](#)
- [42] Shuaifeng Zhi, Tristan Laidlow, Stefan Leutenegger, and Andrew J. Davison. In-place scene labelling and understanding with implicit scene representation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15838–15847, 2021. [2](#)