SceneSplat++: A Large Dataset and Comprehensive Benchmark for Language Gaussian Splatting

*Mengjiao Ma[1,2], *Qi Ma[1,3], *Yue Li[4], Jiahuan Cheng[5], Runyi Yang[1], †Bin Ren[1,6,7], Nikola Popovic[1]
Mingqiang Wei[2], Nicu Sebe[7], Luc Van Gool[1], Theo Gevers[4], Martin R. Oswald[4], Danda Pani Paudel[1]

[1]INSAIT, Sofia University "St. Kliment Ohridski" [2]Nanjing University of Aeronautics and Astronautics [3]ETH Zürich
[4]University of Amsterdam [5]Johns Hopkins University [6]University of Pisa [7]University of Trento

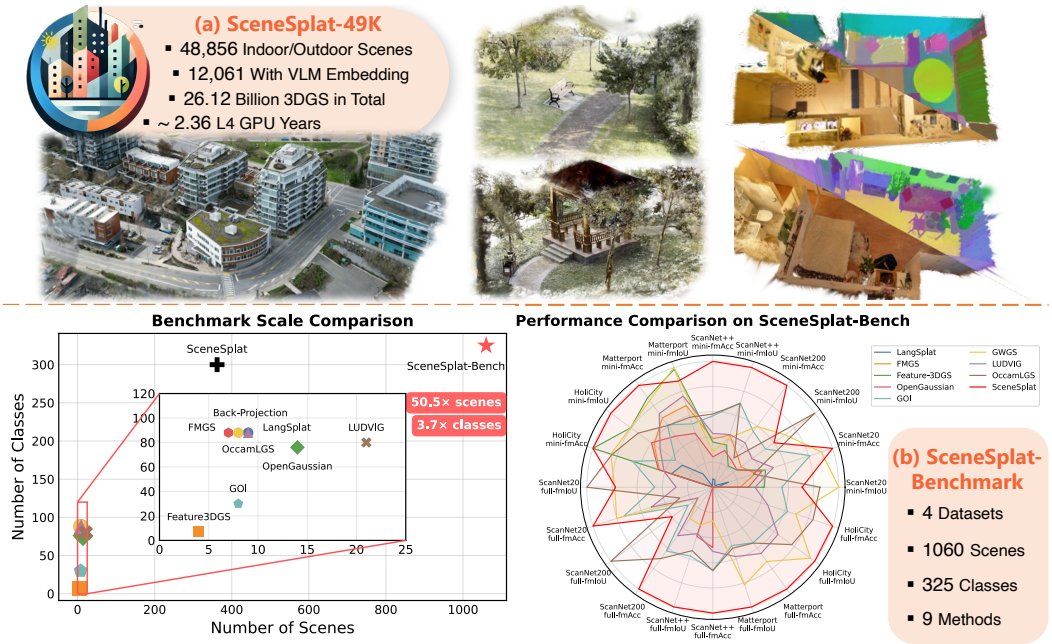


Figure 1: **(a) SceneSplat-49K Dataset**. We introduce SceneSplat-49K, a large-scale 3DGS dataset comprising approximately 49 K diverse indoor and outdoor scenes. **(b) SceneSplat-Bench**. We introduce a comprehensive benchmark for evaluating *Language Gaussian Splatting* (LGS) methods at scale. **Benchmark Scale Comparison** shows SceneSplat-Bench significantly exceeds existing evaluation protocols with 3.7× more semantic classes and 50.5× more scenes. **Performance Comparison on SceneSplat-Bench** using 4 datasets across 9 methods demonstrates that the generalizable approach(SceneSplat) consistently outperforms the per-scene methods in most of the benchmark variants. Our codes, benchmark, and datasets are released at SceneSplat-49K and SceneSplat-Bench.

Abstract

3D Gaussian Splatting (3DGS) serves as a highly performant and efficient encoding of scene geometry, appearance, and semantics. Moreover, grounding language in 3D scenes has proven to be an effective strategy for 3D scene understanding. Current *Language Gaussian Splatting* line of work falls into three main groups: (i) per-scene optimization-based, (ii) per-scene optimization-free, and (iii) generalizable approach. However, most of them are evaluated only on rendered 2D views of a handful of scenes and viewpoints close to the training views, limiting their ability and insight into holistic 3D understanding. To address this gap, we propose the first large-scale benchmark that systematically assesses these three groups of methods directly in 3D space, evaluating on 1060 scenes across three indoor datasets and

∗ indicates equal contribution. † indicates the corresponding author: Bin Ren <bin.ren@insait.ai>.

one outdoor dataset. SceneSplat-Bench results demonstrate a clear advantage of the generalizable paradigm, particularly in relaxing the scene-specific limitation, enabling fast feed-forward inference on novel scenes, and achieving superior segmentation performance. We further introduce SceneSplat-49K, a carefully curated 3DGS dataset comprising around 49K diverse indoor and outdoor scenes obtained from multiple sources, with which we demonstrate the generalizable approach could harness strong data priors. Our codes, benchmark, and datasets are available at SceneSplat-Bench and SceneSplat-49K.

1 Introduction

The field of 3D computer vision is highly focused on capturing the geometry [74, 21, 18, 47, 65, 50, 68, 59] and the visual appearance [14, 27, 13, 43, 22] of a scene, as well as understanding its content [6, 76, 51, 95, 64]. Recently, 3D Gaussian Splatting (3DGS) [22] has emerged as the most desirable 3D representation due to its unique ability to jointly encode geometry, appearance, and scene understanding properties [84, 67, 19] in a compact form that can be effectively optimized from 2D posed images [85]. Furthermore, vision-language reasoning represents the most promising direction for 3D scene understanding [63, 11, 82, 41, 19, 79, 16], as it bridges the visual and geometric attributes of a scene with the language we use to define, describe, and reason about concepts. Therefore, this work focuses exclusively on vision-language scene understanding with 3DGS.

The most relevant methods for language Gaussian Splatting (LGS) can be categorized into three groups. The first two groups begin by extracting 2D features from all training images using vision-language foundation models (e.g., CLIP [55]). The first group then performs gradient-based per-scene optimization [52, 97, 93, 78, 53], assigning feature vectors to each 3D Gaussian primitive and optimizing them such that their renderings align with the corresponding 2D feature maps. The second group also operates on a per-scene basis but adopts an optimization-free approach [20, 42, 9]. Instead of iterative refinement, these methods directly lift 2D features to 3D primitives via weighted feature aggregation schemes. Lastly, we consider the generalizable approach. To date, only a single method has been trained on 2K indoor scenes [31], learning to directly predict a vision-language feature vector for each 3D Gaussian primitive. Although 2D vision-language features are required during training, inference relies solely on 3D Gaussians, eliminating the need to prepare additional 2D feature maps. Furthermore, all features are produced in a single forward pass.

While these methods represent important advances in integrating vision-language reasoning with 3DGS, their evaluation protocols still suffer from several critical limitations. First, most methods are evaluated on only a small number of selected scenes [52, 97, 93, 78, 53, 20, 9]. As a result, such evaluations carry a high risk of producing scene-specific rather than generalizable conclusions. Moreover, the absence of a standardized benchmark further limits systematic and comparable evaluation. Second, most methods are evaluated close to the training views [52, 42, 78]. Consequently, there is no guarantee that the results generalize well to novel viewpoints. Finally, most evaluations are conducted in 2D rather than directly in 3D space [52, 97, 93, 78, 53, 20, 42, 9]. However, *3D scene understanding fundamentally concerns performance in 3D space, which cannot be fully inferred from 2D projections*. Therefore, we introduce SceneSplat-Bench, an evaluation benchmark for 3DGS vision-language scene understanding that includes 1060 scenes spanning 325 unique semantic classes. It evaluates performance in 3D for each segment of the scene, and we use it to assess representative methods from all three groups, as presented in Tab. 1.

Among the evaluated approaches, the generalizable 3D scene understanding paradigm demonstrates the strongest potential. It eliminates the need for substantial per-scene computation at inference and enables a single forward pass per 3D scene. In addition, this approach enables the application of 3D foundation models for scene understanding without the need for task- or scene-specific retraining. This concept aligns with established practices in 2D computer vision, where pretrained foundation models have been widely adopted [17, 12, 48, 58, 55, 34]. Foundation models tailored to 3DGS have already emerged across key areas, including reconstruction [5, 7, 89, 96], scene understanding [31], and generation [80, 62]. Lastly, benchmark results demonstrate the state-of-the-art performance of the generalizable approach, highlighting its capacity to leverage data priors and extract meaningful vision-language features—effectively mitigating the noise inherent in weak supervision.

Methods	Generalizable	Optimization-free	Runtime / Scene	Benchmark Dataset	Scene Number	Class Number
Per-Scene Optimization Methods						
LangSplat [52]	✗	✗	621 min	LERF [24], 3DOVS [35]	9	88
FMGS [97]	✗	✗	76 min	LERF [24], 3DOVS [35]	11	123
Feature3DGS [93]	✗	✗	235 min	Replica [69]	4	7
OpenGaussian [78]	✗	✗	297 min	LERF [24], ScanNet [10](10scenes)	14	76
GOI [53]	✗	✗	252min	Mip-NeRF360 [2], Replica [69]	8	30
Per-Scene Optimization-Free Methods						
Gradient-Weighted 3DGS [20]	✗	✓	4 min	LERF [24], 3DOVS [35]	9	88
LUDVIG [42]	✗	✓	5.6 min	LERF [24], SPIn-NeRF [45], NVOS [60]	21	80
OccamLGS [9]	✗	✓	3.2 min	LERF [24], 3DOVS [35]	9	88
Generalizable Method						
SceneSplat [31]	✓	✓	0.24 min	ScanNet [10], ScanNet++ [86],Matterport3D [4]	732	321
SceneSplat-Bench	–	–	–	**ScanNet** [10], **ScanNet++** [86], **Matterport3D** [4], **HoliCity** [94],	**1060**	**325**

Table 1: **Language Gaussian Splatting (LGS) Benchmark Overview. Left:** Grouped methods and their properties. **Right**: Benchmark characteristics. Our proposed SceneSplat-Bench benchmark evaluates the LGS methods at scale, across three indoor datasets and one outdoor dataset.

To facilitate the development of generalizable 3DGS scene understanding, we introduce SceneSplat-49K, a carefully curated 3DGS dataset comprising diverse indoor and outdoor scenes collected from multiple sources. Existing large-scale 3DGS datasets primarily focus on single objects [39, 36, 73], whereas scene-level datasets [31] remain limited in scale. To the best of our knowledge, SceneSplat-49K represents the most extensive open-source dataset of complex, high-quality 3DGS reconstructions at the full-scene level. The preparation of the dataset required approximately 891 GPU days and considerable human involvement. Furthermore, we demonstrate that training a generalizable 3DGS scene understanding method on a larger subset from SceneSplat-49K leads to improved performance, achieving state-of-the-art results on the benchmark.

Our contributions can be summarized as follows:

- We introduce SceneSplat-Bench, the first *benchmark* for systematically evaluating vision-language scene understanding methods in the 3DGS domain.
- We release SceneSplat-49K, a high-quality, carefully curated 3DGS *dataset* comprising 49K indoor and outdoor scenes.
- We scale up the training and evaluation of a generalizable VLM for 3DGS-based scene understanding across both indoor and outdoor environments, offering new insights.

2 Related Works

Gaussian Splatting Datasets. Most existing 3DGS datasets remain object-centric, synthetic, or limited in scale and scope. Moreover, recent works [5, 62, 96, 8] showcase 3DGS generalization, yet do not fully release their datasets, thus limiting reproducibility and scalability. Among publicly available datasets, ShapeSplat [39] transforms 65K object-level CAD models into 3DGS. 3DAffordSplat [73] focuses on functionality, providing 23K object instances labeled with affordances across 21 categories. uCO3D [36] expands the object-centric landscape with over 1,000 annotated categories and 360° coverage. Furthermore, scene-level 3DGS remains underexplored, with SceneSplat [31] being the only available dataset providing 6.8K indoor scenes to support open-vocabulary 3D understanding, though its scope is limited to the indoor domain. In summary, existing datasets fall short of enabling generalizable, multi-modal learning across diverse scenes. To address this, we introduce SceneSplat-49K, a large-scale dataset of 49K 3DGS scenes spanning indoor and outdoor environments, enriched with vision-language embeddings to support scene-level understanding.

Open-Vocabulary Scene Understanding. Recent advancements in open-vocabulary models have significantly expanded the capabilities of vision-language understanding. Foundation models [88, 48, 26, 57, 54, 87, 71] enabled visual feature extraction at scale, effectively supporting tasks like detection, segmentation and open-vocabulary understanding. Recent works have also introduced 2D foundation models into 3D by utilizing NeRF [44] or 3DGS [23]. Existing 3DGS approaches fall into three categories: per-scene optimization-based [93, 91, 52, 15, 53, 97, 51, 78], per-scene optimization-free [20, 42, 9] and generalizable [31]. Per-scene optimization-based LGS emerged first, with LERF [24, 56] and LangSplat [52] integrating 2D foundation model features into NeRF- and GS-based models, a design adopted by many later methods. Per-scene optimization-free LGS, such as OccamLGS [9], LUDVIG [42] and Gradient Weighted 3DGS [20] further optimized this process

Property	SceneSplat-7K [31]	DL3DV-10K [33]	HoliCity [94]	Aria-ASE [1]	Crowdsourced	Total
Data Statistics						
Raw Scenes	9114	10K	6 253	25K	603	48 856 ∼49K
GS Scenes	7 916	6 376	5 940	25K	603	45 835 ∼46K
Data Type	Indoor	Indoor/Outdoor	Outdoor	Indoor	Indoor/Outdoor	Indoor/Outdoor
RGB Frames	4.72M	3.09M	48K	14.79M	N/A	22.65M
Avg. GS / Scene	1.42M	1.48M	800K	1.27M	1.02M	1.26M
3DGS GPU Time [h]	3.59K	2.1K	0.79K	6.25K	N/A	12.73K
Scenes w/ VLM Embedding	6 121	-	5 940	-	-	12 061
Embedding GPU Time [h]	8 161	-	495	-	-	8 656
Storage [TB]	2.76	2.91	0.26	2.2	0.23	8.36
Total 3DGS	11.27B	9.42B	4.75B	3.18B	615M	29.24B
Evaluation Metrics (Avg.)						
PSNR [dB] ↑	29.64	25.70	29.85	27.32	N/A	27.83
Depth Loss [m] ↓	0.035	-	0.010	0.082	N/A	0.061
SSIM ↑	0.897	0.845	0.921	0.906	N/A	0.898
LPIPS ↓	0.212	0.203	0.122	0.230	N/A	0.209

Table 2: **SceneSplat-49K Statistics and Quality Metrics.** Our proposed dataset includes outdoor and indoor scenes drawn from established datasets and publicly available sources, including our own collected data. In total, it comprises approximately $49\,\mathrm{K}$ raw scenes, $46\,\mathrm{K}$ curated 3D Gaussian Splatting (3DGS) scenes, and $29.24\,\mathrm{B}$ Gaussian splats. Generating all 3DGS required $\sim$530 GPU days on an NVIDIA L4 GPU. Appearance and geometry reconstruction quality are evaluated using four metrics: PSNR, SSIM, LPIPS and depth ℓ_1 error. Additionally, $12\,\mathrm{K}$ of the 3DGS scenes are annotated with per-primitive vision–language embeddings to support the training of generalizable scene understanding methods. Computing these embeddings further required $\sim$361 GPU days.

by employing feature lifting, directly projecting all 2D features into 3DGS, reducing computation time from hours to seconds. SceneSplat [31] is the first generalizable LGS method, trained and evaluated directly on 3D data, achieving significant gains in open-vocabulary scene understanding with substantially faster inference on novel scenes. However, most of these methods have been evaluated in constrained 2D settings, with a limited number of scenes and semantic classes. We therefore introduce a comprehensive benchmark and use it to systematically evaluate recent language Gaussian Splatting methods.

3D Generalizability. Large-scale training and generalization in 2D have become increasingly mature [55, 71, 37, 38, 57, 58, 92]. In contrast, achieving semantic and language generalization in 3D remains challenging and relatively underexplored [8, 49, 90, 77, 76, 28, 75]. One major obstacle is the limited quantity and fragmented nature of 3D datasets, as no single dataset provides the needed combination of (i) large-scale coverage, (ii) multi-scale diversity (from object-centric to indoor and outdoor scenes), and (iii) rich multimodal information (geometry, appearance, and semantics). As a result, even state-of-the-art 3D models struggle to generalize broadly. Mosaic3D-5.6M compiles over 30K RGB-D scenes with 5.6 million mask-text pairs by leveraging 2D vision-language models, yielding an significant resource for open-vocabulary 3D segmentation [28]. The SceneSplat-7K dataset supports open-vocabulary training but with limited scale [31]. These efforts represent early steps toward 3D foundation models, aiming to enable robust, optimization-free, and generalizable 3D reasoning across vision-language tasks. Therefore, we introduce SceneSplat-49K, a large and comprehensive dataset designed to unlock scaling laws in generalizable 3DGS tasks.

3 Dataset

We present SceneSplat-49K, a large-scale 3D Gaussian Splatting dataset comprising approximately $49\,\mathrm{K}$ raw scenes and $46\,\mathrm{K}$ curated 3DGS scenes, aggregated from multiple established sources, including SceneSplat-7K [31], DL3DV-10K [33], HoliCity [94], Aria Synthetic Environments [1], and our own crowdsourced data. This diverse dataset consists of both indoor and outdoor environments, spanning from rooms, apartments to streets. To support 3DGS scene understanding model training, $12\,\mathrm{K}$ scenes are further enriched with per-primitive vision-language embeddings extracted with state-of-the-art vision-language models [71]. The comprehensive statistics of the introduced dataset are presented in Tab. 2.

Fig. 2 visualizes the distribution of evaluation metrics of our SceneSplat-49K dataset. Across $\sim 49\,\mathrm{K}$ 3DGS scenes, the mean photometric quality is $27.8\,\mathrm{dB}$ PSNR and 0.90 SSIM, with a perceptual LPIPS of 0.20—values that are of high quality renders. Importantly, the geometry reconstruction is

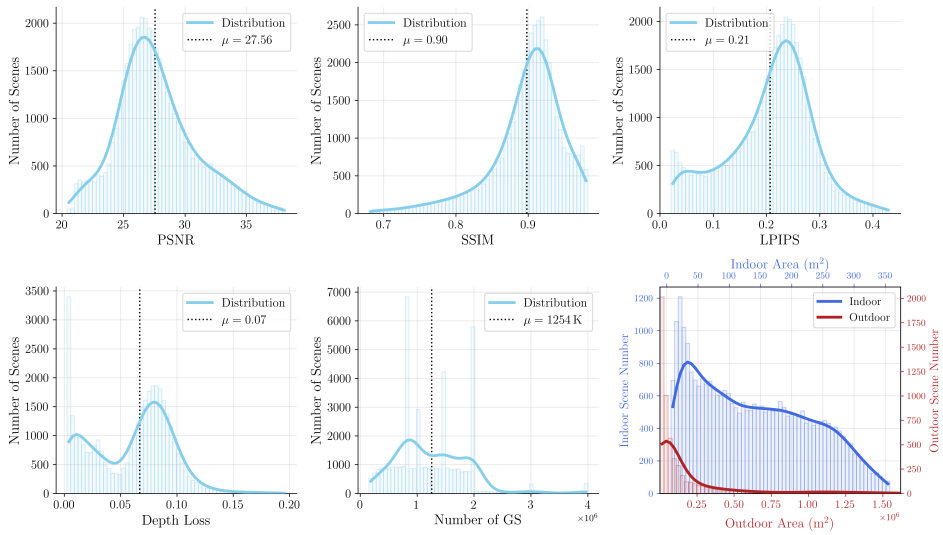


Figure 2: **Appearance, Geometry, and Scale Statistics of the SceneSplat-49K Dataset.** Distributions of photometric (PSNR, SSIM, LPIPS) and geometric (depth ℓ_1) reconstruction errors show consistently high-quality renders across scenes, while the wide spread in total Gaussian number and indoor/outdoor scene floor area demonstrates the dataset's diversity. The curves are convolved from the bucket values and vertical dotted lines mark the mean of each metric.

equally reliable, with a mean depth ℓ_1 error of $0.061\,\mathrm{m}$. The distribution of the per-scene Gaussian number spans two orders of magnitude, indicating the dataset's complexity. The scene footprint areas also show clear diversity: indoor environments cluster all across $25\,\mathrm{m}^2$ to $250\,\mathrm{m}^2$, whereas outdoor scenes extend over a square kilometer, with a long-tailed distribution. This diversity in spatial extent, together with high-quality appearance and geometry, underpins the suitability of the dataset for training a generalizable 3DGS scene understanding model.

3.1 Data Collection and Processing

To ensure high-quality 3DGS scenes, we implement several quality control measures throughout the optimization pipeline. Starting with the training views, we select scenes with at least 400 frames to ensure sufficient multi-view coverage. When depth information is available, we initialize the Gaussians at fused point cloud locations and apply depth supervision to maximize geometry quality. We use gsplat [85] for 3DGS optimization. We filter blurry frames using the variance of the Laplacian as a sharpness metric. To efficiently compress the 3DGS scene, we employ the Markov Chain Monte Carlo strategy [25] and add opacity and scale regularization. Once optimized, the dataset supports filtering 3DGS scenes based on PSNR and depth quality before using them as inputs for model training. We introduce the processing of each data source in the following.

SceneSplat-7K contains 7,916 curated 3DGS indoor scenes derived from ScanNet [10], ScanNet++ [86], Replica [69], Hypersim [61], 3RScan [72], ARKitScenes [3], and Matterport3D [4].We further enrich over 6,000 of these scenes with vision–language embeddings to support open-vocabulary scene queries.

Aria Synthetic Environments (ASE) is a large-scale synthetic dataset of procedurally-generated multi-room indoor scenes, each populated with 3D object models drawn from a digital asset library. We select the first $25\,\mathrm{K}$ scene subset of it. For our 3DGS optimization, we undistort the fisheye image and depth frames and apply the devignetting mask for brightness correction. The sensor depths are fused and transformed to yield point clouds for the 3DGS initialization.

DL3DV-10K dataset covers 65 everyday environments, featuring both indoor and outdoor locations. It includes 10,510 high-quality videos captured by mobile devices and drones, along with sparse COLMAP reconstructions. For 3DGS training, we use an evenly sampled set of every 10 views for novel view evaluation, while the remaining views are used for training.

HoliCity is a city-scale 3D dataset comprising 6,300 real-world panoramas, each accurately aligned to a downtown London CAD model covering over $20\,\mathrm{km}^2$, providing depth frames and semantic

labels. We used eight perspective views from each panorama generated by evenly sampling the yaw angles at 45° intervals to optimize the 3DGS.

Crowdsourced Data. To diversify our dataset, we collected high-quality 3DGS scenes from various sources, including Polycam and Sketchfab. We have also distributed our questionnaire to the community to collect data.

3.2 Vision-Language Embedding Collection

Vision-language embeddings are extracted from the frames used to optimize the 3DGS. Unlike existing pretraining methods that align 3D primitives with text embeddings from vision-language models, we align each Gaussian directly in the image-embedding space, thereby preserving richer latent semantics. This approach avoids the information loss inherent in textual descriptions.

We follow the fusion strategy in [31] and employ a dynamic weighting mechanism that adaptively combines three distinct features: global context from the entire frame, local features with background, and masked features without background. This mechanism automatically balances contributions based on each segment's contextual relationship—emphasizing background-inclusive features for integrated objects (e.g., keyboard with monitor) and object-specific features for isolated objects. This adaptive approach provides more nuanced semantic understanding than fixed weighting strategies.

4 Benchmark

We formulate the assigning of Gaussian language feature as follows. A scene is represented by a set of N Gaussian primitives $\mathcal{G} = \{G_j\}_{j=1}^{N}$, where each $G_j \in \mathbb{R}^{59}$ encodes its position, covariance, color, and opacity. For every primitive we compute a language embedding $E_j^G \in \mathbb{R}^d$, obtained by per-scene processing or by inference with a learned encoder. The user provides a set of free-form text queries $C = \{c_i\}_{i=1}^{n}$. A frozen text encoder P_T (e.g., CLIP) maps each query to an embedding $E_i^T = P_T(c_i)$. Each Gaussian is labeled with the query whose embedding is most similar to its own:

$$\text{label}(G_j) \;=\; \arg\max_{i} \; \cos\big(E_j^G,\, E_i^T\big). \tag{1}$$

This label field assigns every primitive to the most relevant query.

4.1 Benchmark settings

Evaluation Protocols. We evaluate 9 methods from 3 categories on our SceneSplat-Bench benchmark. We report two key metrics in 3D: foreground mean Intersection over Union (f-mIoU) and foreground mean accuracy (f-mAcc), which address object size imbalances and reduce viewpoint dependency compared with traditional 2D-only evaluation. We unified the evaluation pipeline by first converting the language fields into a unified format $\mathcal{E}_G = \{E_j\}_{j=1}^{N}$, then computing the cosine similarity between each language feature and the queried text embeddings. Each Gaussian primitive is assigned the semantic label corresponding to the most similar text embedding. For every ground-truth 3D point used for evaluation, we identify its K-nearest Gaussian neighbors (K=25) and assign the majority-voted label as the semantic prediction for that point. We exclude floor, wall, and ceiling classes as background for indoor scenes and the sky class for outdoor scenes. To make our evaluation more complete, we additionally evaluate semantics on 2D renders, following the protocol of LERF [24]. The 2D evaluation is conducted on subsets of the benchmark datasets with two key 2D metrics: mean Intersection over Union (mIoU) and mean accuracy (mAcc) on all classes (background and foreground classes). All runtime measurements were obtained on NVIDIA RTX A6000 GPUs, except for FMGS, whose training requires 50–80 GB of GPU memory; its runtime was therefore measured on NVIDIA H200 GPUs.

Benchmark Dataset. We evaluate the methods' performance on indoor scenes using ScanNet (with 20 and 200 classes), ScanNet++ (100 classes), and Matterport3D (21 classes). Queries cover a wide range of object granularity—from small items like a mouse to larger objects such as a bed—reflecting the method's capability to varying scales understanding. HoliCity (4 classes) is employed to evaluate the performance on outdoor scenes, emphasizing large structures under complex surrounding environment. Performance results are reported on randomly sampled 10-scenes subsets

Method	ScanNet20 (10 scenes)		ScanNet20 (312 scenes)		ScanNet200 (10 scenes)		ScanNet200 (312 scenes)	
	f–mIoU	f–mAcc	f–mIoU	f–mAcc	f–mIoU	f–mAcc	f–mIoU	f–mAcc
Per-Scene Optimization Methods								
LangSplat [52]	0.0153	0.0741	-	-	0.0057	0.0106	-	-
OpenGaussian [78]	0.1186	0.2340	-	-	0.0545	0.1086	-	-
Feature-3DGS [93]	0.1711	0.2512	-	-	0.0541	0.0876	-	-
FMGS [97]	0.0916	0.2028	0.1169	0.2684	0.0641	0.1374	0.0579	0.1443
GOI [53]	0.2295	0.4638	0.2115	0.4286	0.1193	0.2297	0.1001	0.2240
Per-Scene Optimization-Free Methods								
Gradient-Weighted 3DGS [20]	**0.4184**	0.5174	0.3573	0.4638	0.1638	0.2502	0.0601	0.1100
LUDVIG [42]	0.1406	0.2885	0.1337	0.2969	0.0887	0.1833	0.0602	0.1701
OccamLGS [9]	0.3559	0.4587	0.2308	0.4148	0.1936	0.2458	0.1204	0.2503
SceneSplat [31] (Pseudo Label)	0.3521	0.5210	0.3500	0.5550	**0.2520**	0.3038	**0.2280**	**0.3590**
Generalizable Method								
SceneSplat [31]	0.2968	**0.5711**	**0.3540**	**0.5780**	0.1387	**0.4198**	0.1648	0.3573

Table 3: **Zero-shot 3D Semantic Segmentation Experiments on ScanNet20 [10] (20 classes) and ScanNet200 [10] (200 classes)**. All methods are evaluated on a 10-scene mini-validation set, with the 312-scene full validation set run only for selected methods due to runtime limitations (Tab. 1).

for each of the four benchmarks across all methods. For methods demonstrating strong efficiency, we conduct a full evaluation on the complete validation set.

Baseline Methods. We categorize the methods based on their optimization paradigm and generalizability to novel scenes. ***(i)*** **Per-Scene Optimization-based Methods** assign learnable attributes to Gaussian splats and optimize them via backpropagation. However, rendering additional features with alpha compositing is computationally expensive and requires per-scene optimization. ***(ii)*** **Per-Scene Optimization-Free methods** use a training-free approach, lifting 2D features into 3D space as a weighted sum of 2D representations, requiring only a single forward rendering pass. ***(iii)*** **Generalizable Method** SceneSplat trains a feed-forward 3DGS encoder, learning to predict representations aligned with image embeddings. We further extend its training based on our dataset. SceneSplat (Pseudo Label) denotes the VLM embeddings used for its vision-language pretraining. We refer to the supplement for details on running each baseline.

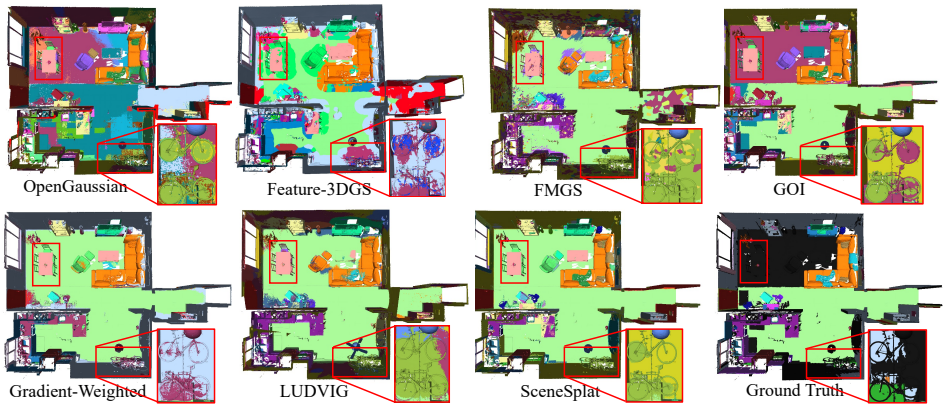


Figure 3: **Qualitative Results of Zero-Shot 3D Semantic Segmentation.** The semantic classes "bicycle" and "kitchen table" are highlighted, which are not labeled in Ground Truth.

4.2 Key Insights

The primary experimental results from our SceneSplat-Bench benchmark across all selected methods are presented in Tabs. 3 to 5 for indoor scenes and in Tab. 6 for outdoor scenes. The corresponding runtime statistics for individual methods can be found in Tab. 1. SceneSplat – representing the generalizable paradigm – stands out as the clear winner in terms of both performance and efficiency (see Fig. 3 for qualitative examples). Interestingly, in 75% of the experiments, it even outperforms its own pseudo labels used for pretraining (SceneSplat (Pseudo Label)). This result highlights its

Method	ScanNet++ (10 scenes)		ScanNet++ (50 scenes)		Matterport3D (10 scenes)		Matterport3D (370 scenes)	
	f–mIoU	f–mAcc	f–mIoU	f–mAcc	f–mIoU	f–mAcc	f–mIoU	f–mAcc
Per-Scene Optimization Methods								
LangSplat [52]	0.0131	0.0332	-	-	0.0219	0.1257	-	-
OpenGaussian [78]	0.0695	0.1186	-	-	0.1387	0.2698	-	-
Feature-3DGS [93]	0.0798	0.1765	-	-	0.3043	0.4732	-	-
FMGS [97]	0.0984	0.1892	0.1004	0.2300	0.2105	0.3664	-	-
GOI [53]	0.1593	0.2790	0.1631	0.3296	0.1631	0.3296	0.1652	0.3130
Per-Scene Optimization-Free Methods								
Gradient-Weighted 3DGS [20]	0.0925	0.1391	0.0900	0.1328	**0.3083**	0.4586	0.2762	0.4187
LUDVIG [42]	0.0995	0.2099	0.1099	0.2547	0.2348	0.3888	0.1960	0.3738
OccamLGS [9]	0.1580	0.2861	0.1502	0.3312	0.1803	0.3263	0.1725	0.3151
SceneSplat [31] (Pseudo Label)	**0.2340**	0.3709	0.2243	0.4667	0.2392	0.4368	0.2748	0.4384
Generalizable Method								
SceneSplat [31]	0.2263	**0.4916**	**0.2836**	**0.4992**	0.2732	**0.5379**	**0.3384**	**0.5745**

Table 4: **Zero-Shot 3D Semantic Segmentation Experiments On ScanNet++ [86] and Matterport3D [4].** All methods are evaluated on a 10-scene mini-validation set, with the full validation set evaluated only for selected methods due to runtime limitations (Tab. 1).

Method	ScanNet20 (10 scenes)		ScanNet200 (10 scenes)		ScanNet++ (10 scenes)	
	mIoU	mAcc	mIoU	mAcc	mIoU	mAcc
Per-Scene Optimization Methods						
LangSplat [52]	0.0193	0.0700	0.0133	0.0272	0.0111	0.0272
OpenGaussian [78]	0.1958	0.3294	0.0740	0.1273	0.1290	0.2133
Feature-3DGS [93]	**0.5054**	0.6474	0.1867	0.2798	0.1605	0.2368
FMGS [97]	0.1413	0.2629	0.0848	0.1825	0.1134	0.2244
GOI [53]	0.2866	0.4536	0.1134	0.1806	0.1635	0.2767
Per-Scene Optimization-Free Methods						
Gradient-Weighted 3DGS [20]	0.4533	0.5785	0.1639	0.2367	0.1210	0.1760
LUDVIG [42]	0.1673	0.3058	0.0958	0.1828	0.1239	0.2588
OccamLGS [9]	0.2937	0.4645	0.1300	0.2146	0.1670	0.2866
Generalizable Method						
SceneSplat [31]	0.4458	**0.6727**	**0.2599**	**0.4049**	**0.3165**	**0.5094**

Table 5: **Zero-Shot 2D Semantic Segmentation Experiments on ScanNet20 [10] (20 classes), ScanNet200 [10] (200 classes) and ScanNet++ [86].** All methods are evaluated on a 10-scene validation set.

ability to generalize by leveraging large-scale data priors and learning to predict meaningful vision-language features, effectively mitigating the noise inherent in weak supervision. In addition, Tab. 5 shows that the performance on the 2D metric is aligned with the zero-shot performance measured with the 3D metric, with a generalizable method leading the performance. As illustrated in Fig. 4, visualizations of text query results using enhanced NeRFView tools [70, 83, 31] further demonstrate that the generalizable method outperforms the others. Moreover, unlike other methods that rely on extensive feature extraction, the generalizable approach requires only 3DGS as input at inference time and predicts per-primitive vision-language features in a single forward pass.

Method	HoliCity (10 scenes)		Holicity (328 scenes)	
	f–mIoU	f–mAcc	f–mIoU	f–mAcc
Per-Scene Optimization Methods				
LangSplat [52]	0.0918	0.1950	-	-
OpenGaussian [78]	0.1853	0.2567	-	-
Feature-3DGS [93]	0.2479	0.5646	-	-
FMGS [97]	0.1774	0.2513	-	-
GOI [53]	0.1126	0.3773	0.1587	0.3772
Per-Scene Optimization-Free Methods				
Gradient-Weighted 3DGS [20]	0.2484	0.4808	0.2762	0.4187
LUDVIG [42]	0.1593	0.2296	0.1843	0.2775
OccamLGS [9]	0.2255	0.4754	0.2234	0.5056
SceneSplat [31] (Pseudo Label)	0.2464	0.4780	0.2522	0.5159
Generalizable Method				
SceneSplat [31]	**0.3081**	**0.5647**	**0.2880**	**0.6045**

Table 6: **Zero-Shot 3D Outdoor Segmentation On HoliCity [94] Dataset.** All methods are evaluated on mini-validation sets,with the full validation set evaluated only for selected methods due to runtime limitations (Tab. 1).

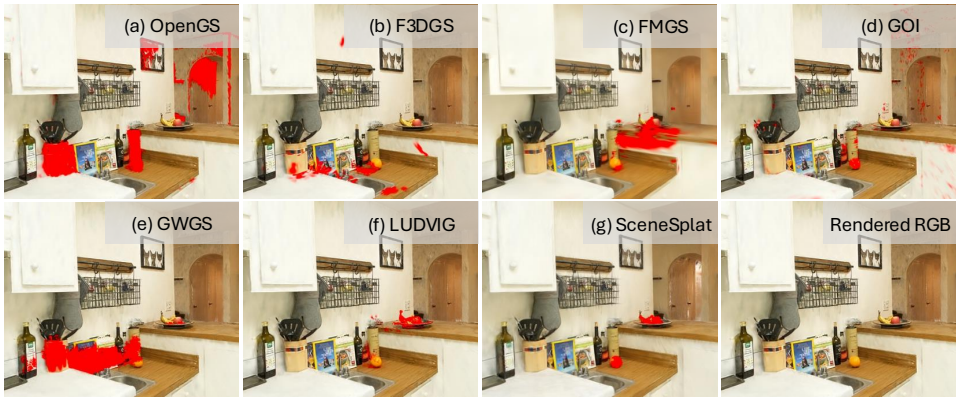


Figure 4: **Text-Based Scene Query.** Given the prompt "These are fruits" to different LGS methods, the queried parts are highlighted in red.

Method	ScanNet (10 scenes)		ScanNet++ (10 scenes)	
	Accuracy (BBox)	Accuracy (Seg)	Accuracy (BBox)	Accuracy (Seg)
Per-Scene Optimization Methods				
LangSplat [52]	0.0663	0.0415	0.0655	0.0337
OpenGaussian [78]	0.4565	0.3483	0.5138	0.3394
Feature-3DGS [93]	0.5861	0.4759	0.5046	0.2783
FMGS [97]	0.6082	0.4639	0.5494	0.3303
GOI [53]	0.5269	0.4414	0.5107	0.3568
Per-Scene Optimization-Free Methods				
Gradient-Weighted 3DGS [20]	0.5486	0.4669	0.4322	0.2661
LUDVIG [42]	0.6750	0.5481	0.4801	0.2875
OccamLGS [9]	0.2074	0.1599	0.1896	0.1182
Generalizable Method				
SceneSplat [31]	**0.7701**	**0.6928**	**0.7339**	**0.5464**

Table 7: **3D Object Localization Experiments on ScanNet [10] and ScanNet++ [86]**. Accuracy is reported with bounding box–based and segmentation–based evaluation.

Training Source	#Scene	ScanNet++		ScanNet200		Matterport3D		HoliCity	
		f–mIoU	f–mAcc	f–mIoU	f–mAcc	f–mIoU	f–mAcc	mIoU	mAcc
Indoor Training-Data Scaling									
ScanNet++ (v1)	280	0.1683	0.3230	0.1081	0.1650	0.1010	0.2108	0.1888	0.3003
ScanNet++ (v2)	906	0.2383	0.4324	0.1180	0.1920	0.1491	0.2756	0.2339	0.3524
ScanNet++ (v2) + ScanNet	2107	0.2779	**0.5055**	0.1427	0.3434	0.2289	0.3803	0.2633	0.3504
ScanNet++ (v2) + ScanNet + M3D	3503	**0.2836**	0.4992	**0.1648**	**0.3573**	**0.3384**	**0.5745**	0.2026	0.3600
Matterport3D only	1396	—	—	—	—	0.3244	0.5354	0.2240	0.3889
Outdoor-Domain Training									
HoliCity only	3000	—	—	—	—	—	—	**0.2880**	**0.6045**

Table 8: **Impact of Training-Data Scaling on Indoor Benchmarks and Cross-Domain Generalization to HoliCity [94].** Results show that more training data consistently improves indoor performance. Furthermore, models trained on indoor data only surprisingly transfer to outdoor scenes.

Among the per-scene methods, optimization-free approaches clearly outperform optimization-based ones, both in terms of segmentation accuracy, as shown in Tabs. 3 to 6, and in runtime efficiency, as reported in Tab. 1. This is likely due to the fact that the objective function in optimization-based methods is typically designed to perform as well as possible on the training views, which may not be ideal to novel viewpoints. Taking this into account, along with the strong performance of the generalizable approach, we conclude that per-scene optimization is not necessary for effective vision-language reasoning in 3DGS. However, within both the optimization-based and optimization-free groups, no single method consistently dominates across all datasets, indicating that performance remains sensitive to dataset-specific characteristics.

A key requirement for reliable benchmarking is having a sufficient number of scenes and appropriately challenging evaluation settings. As shown in Tabs. 3, 4 and 6, the performance of various methods varies noticeably when scaling from a small subset of 10 scenes to the full validation set, highlighting the importance of the benchmark scale in assessing performance. Furthermore, Tab. 3 demonstrates

a clear drop in accuracy when the task complexity increases from 20 to 200 semantic classes, emphasizing the need for demanding benchmarks to reveal the limitations of competing approaches.

To further diversify the open-vocabulary evaluation, we include a 3D object localization task that assesses how well each method retrieves objects in 3D space using its constructed 3DGS field. Following the protocol of LangSplat [52], we directly select the point with the highest relevance score for each query. Our evaluation is conducted on 2,895 objects from ScanNet and ScanNet++ validation scenes, which is over 12× more objects compared to LERF [24], offering a broader and more realistic assessment. This evaluation is presented in Tab. 7 and shows similar conclusion as previously discussed evaluations. The generalizable paradigm shows dominant performance, followed by the per-scene optimization-free methods, which outperform per-scene optimization method group.

Tab. 8 presents our scaling study, which offers two key takeaways. First, training-data scaling consistently improves performance on indoor benchmarks: expanding the training set from 280 to 3503 scenes lifts ScanNet++ f-mIoU from 0.168 → 0.284 and ScanNet200 from 0.108 → 0.165. A similar effect appears when the model jointly trained on three indoor datasets outperforms that trained on Matterport3D alone. Second—and more surprisingly—an indoor-only model transfers to outdoor scenes, but domain-specific data is still required to close the gap. Without ever seeing outdoor data, the largest indoor model reaches 0.263 mIoU on HoliCity, an improvement of 7.4% over the 280-scene baseline. This suggests that large-scale indoor training induces representations that partially generalize across the indoor–outdoor boundary, see Fig. 5 for zero-shot examples. Nevertheless, a model trained only on HoliCity still leads (0.288 mIoU), indicating that cross-domain capability ultimately needs outdoor supervision. These findings underscore the value of our large-scale indoor–outdoor 3DGS dataset as a powerful training resource.

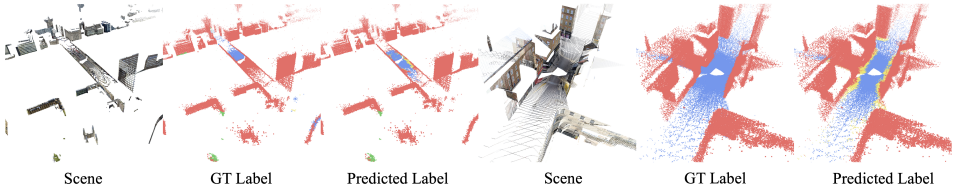


Figure 5: **Zero-Shot Predictions of Indoor-Trained SceneSplat on Outdoor Scenes.** The results highlight the cross-domain capability. Color palette denotes buildings, roads, terrains, and trees.

5 Conclusion and Limitations

This work introduces the first comprehensive benchmark for 3DGS-based scene understanding methods, along with a large-scale 3DGS dataset of diverse indoor and outdoor scenes. Our evaluation demonstrates that generalizable approach consistently outperforms per-scene methods, establishing a new paradigm for scalable scene understanding through pretrained models. Despite these contributions, limitations remain. Our dataset scale, while substantial, could be further expanded. The outdoor benchmark is constrained by limited semantic classes, and not all 3DGS scenes include precomputed vision-language embeddings. We leave these as important directions for future work and hope our benchmark and dataset will be helpful to advance language-grounded 3DGS scene understanding.

Acknowledgments and Disclosure of Funding

Yue Li is financially supported by TomTom, the University of Amsterdam and the allowance of Top consortia for Knowledge and Innovation (TKIs) from the Netherlands Ministry of Economic Affairs and Climate Policy. This work used Dutch national e-infrastructure with the support of the SURF Cooperative under grant no. NWO-2024.035. This work was partially supported by INSAIT, Sofia University "St. Kliment Ohridski" (Partially funded by the Ministry of Education and Science of Bulgaria's support for INSAIT as part of the Bulgarian National Roadmap for Research Infrastructure), EU Horizon projects ELIAS (No.101120237), and ELLIOT (No.101214398). Some computational resources were provided by the Google Cloud Platform (GCP).

References

[1] Armen Avetisyan, Christopher Xie, Henry Howard-Jenkins, Tsun-Yi Yang, Samir Aroudj, Suvam Patra, Fuyang Zhang, Duncan Frost, Luke Holland, Campbell Orme, et al. Scenescript: Reconstructing scenes with an autoregressive structured language model. In *European Conference on Computer Vision*, pages 247–263. Springer, 2024. 4

[2] Jonathan T. Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P. Srinivasan. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields, 2021. 3

[3] Gilad Baruch, Zhuoyuan Chen, Afshin Dehghan, Tal Dimry, Yuri Feigin, Peter Fu, Thomas Gebauer, Brandon Joffe, Daniel Kurz, Arik Schwartz, et al. Arkitscenes: A diverse real-world dataset for 3d indoor scene understanding using mobile rgb-d data. *arXiv preprint arXiv:2111.08897*, 2021. 5

[4] Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niessner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3d: Learning from rgb-d data in indoor environments. *arXiv preprint arXiv:1709.06158*, 2017. 3, 5, 8

[5] David Charatan, Sizhe Li, Andrea Tagliasacchi, and Vincent Sitzmann. pixelsplat: 3d gaussian splats from image pairs for scalable generalizable 3d reconstruction. In *CVPR*, 2024. 2, 3

[6] R. Qi Charles, Hao Su, Mo Kaichun, and Leonidas J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 77–85, 2017. 2

[7] Yuedong Chen, Haofei Xu, Chuanxia Zheng, Bohan Zhuang, Marc Pollefeys, Andreas Geiger, Tat-Jen Cham, and Jianfei Cai. Mvsplat: Efficient 3d gaussian splatting from sparse multi-view images. *arXiv preprint arXiv:2403.14627*, 2024. 2

[8] Yutong Chen, Marko Mihajlovic, Xiyi Chen, Yiming Wang, Sergey Prokudin, and Siyu Tang. Splatformer: Point transformer for robust 3d gaussian splatting, 2025. 3, 4

[9] Jiahuan Cheng, Jan-Nico Zaech, Luc Van Gool, and Danda Pani Paudel. Occam's lgs: A simple approach for language gaussian splatting. *arXiv preprint arXiv:2412.01807*, 2024. 2, 3, 7, 8, 9, 6

[10] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5828–5839, 2017. 3, 5, 7, 8, 9

[11] Runyu Ding, Jihan Yang, Chuhui Xue, Wenqing Zhang, Song Bai, and Xiaojuan Qi. Pla: Language-driven open-vocabulary 3d scene understanding. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7010–7019, 2023. 2

[12] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *ICLR*, 2021. 2

[13] Yasutaka Furukawa and Jean Ponce. Accurate, dense, and robust multiview stereopsis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(8):1362–1376, 2010. 2

[14] Eric L. W. Grimson. *From Images to Surfaces: A Computational Study of the Human Early Visual System*. MIT Press, Cambridge, MA, USA, 1981. 2

[15] Jun Guo, Xiaojian Ma, Yue Fan, Huaping Liu, and Qing Li. Semantic gaussians: Open-vocabulary scene understanding with 3d gaussian splatting. *arXiv preprint arXiv:2403.15624*, 2024. 3

[16] Anna-Maria Halacheva, Jan-Nico Zaech, Xi Wang, Danda Pani Paudel, and Luc Van Gool. Gaussianvlm: Scene-centric 3d vision-language models using language-aligned gaussian splats for embodied reasoning and beyond. *arXiv preprint arXiv:2507.00886*, 2025. 2

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. 2

[18] Heiko Hirschmuller. Stereo processing by semiglobal matching and mutual information. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(2):328–341, 2008. 2

[19] Li Jiang, Shaoshuai Shi, and Bernt Schiele. Open-vocabulary 3d semantic segmentation with foundation models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21284–21294, 2024. 2

[20] Joji Joseph, Bharadwaj Amrutur, and Shalabh Bhatnagar. Gradient-weighted feature back-projection: A fast alternative to feature distillation in 3d gaussian splatting, 2024. 2, 3, 7, 8, 9, 6

[21] Michael Kazhdan, Matthew Bolitho, and Hugues Hoppe. Poisson surface reconstruction. In *Proceedings of the Fourth Eurographics Symposium on Geometry Processing*, SGP '06, page 61–70, 2006. 2

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

[23] 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. 3

[24] Justin Kerr, Chung Min Kim, Ken Goldberg, Angjoo Kanazawa, and Matthew Tancik. Lerf: Language embedded radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19729–19739, 2023. 3, 6, 10

[25] Shakiba Kheradmand, Daniel Rebain, Gopal Sharma, Weiwei Sun, Yang-Che Tseng, Hossam Isack, Abhishek Kar, Andrea Tagliasacchi, and Kwang Moo Yi. 3d gaussian splatting as markov chain monte carlo. *Advances in Neural Information Processing Systems*, 37:80965–80986, 2025. 5

[26] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4015–4026, 2023. 3

[27] K.N. Kutulakos and S.M. Seitz. A theory of shape by space carving. In *Proceedings of the Seventh IEEE International Conference on Computer Vision*, volume 1, pages 307–314 vol.1, 1999. 2

[28] Junha Lee, Chunghyun Park, Jaesung Choe, Yu-Chiang Frank Wang, Jan Kautz, Minsu Cho, and Chris Choy. Mosaic3d: Foundation dataset and model for open-vocabulary 3d segmentation. *arXiv preprint arXiv:2502.02548*, 2025. 4

[29] Boyi Li, Kilian Q Weinberger, Serge Belongie, Vladlen Koltun, and René Ranftl. Language-driven semantic segmentation. *arXiv preprint arXiv:2201.03546*, 2022. 6

[30] Boyi Li, Kilian Q. Weinberger, Serge Belongie, Vladlen Koltun, and René Ranftl. Language-driven semantic segmentation, 2022. 8

[31] Yue Li, Qi Ma, Runyi Yang, Huapeng Li, Mengjiao Ma, Bin Ren, Nikola Popovic, Nicu Sebe, Ender Konukoglu, Theo Gevers, Luc Van Gool, Martin R. Oswald, and Danda Pani Paudel. Scenesplat: Gaussian splatting-based scene understanding with vision-language pretraining, 2025. 2, 3, 4, 6, 7, 8, 9, 1

[32] Yiyi Liao, Jun Xie, and Andreas Geiger. Kitti-360: A novel dataset and benchmarks for urban scene understanding in 2d and 3d. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(3):3292–3310, 2023. 7, 8

[33] Lu Ling, Yichen Sheng, Zhi Tu, Wentian Zhao, Cheng Xin, Kun Wan, Lantao Yu, Qianyu Guo, Zixun Yu, Yawen Lu, et al. Dl3dv-10k: A large-scale scene dataset for deep learning-based 3d vision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22160–22169, 2024. 4, 1

[34] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *NeurIPS*, 2023. 2

[35] 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, 2024. 3

[36] Xingchen Liu, Piyush Tayal, Jianyuan Wang, Jesus Zarzar, Tom Monnier, Konstantinos Tertikas, Jiali Duan, Antoine Toisoul, Jason Y. Zhang, Natalia Neverova, Andrea Vedaldi, Roman Shapovalov, and David Novotny. Uncommon objects in 3d. In *arXiv*, 2024. 3

[37] Yuanhuiyi Lyu, Xu Zheng, Lutao Jiang, Yibo Yan, Xin Zou, Huiyu Zhou, Linfeng Zhang, and Xuming Hu. Realrag: Retrieval-augmented realistic image generation via self-reflective contrastive learning. *arXiv preprint arXiv:2502.00848*, 2025. 4

[38] Yuanhuiyi Lyu, Xu Zheng, Jiazhou Zhou, and Lin Wang. Unibind: Llm-augmented unified and balanced representation space to bind them all. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 26752–26762, 2024. 4

[39] Qi Ma, Yue Li, Bin Ren, Nicu Sebe, Ender Konukoglu, Theo Gevers, Luc Van Gool, and Danda Pani Paudel. Shapesplat: A large-scale dataset of gaussian splats and their self-supervised pretraining. In *3DV*, 2024. 3

[40] Qi Ma, Danda Pani Paudel, Ender Konukoglu, and Luc Van Gool. Implicit-zoo: A large-scale dataset of neural implicit functions for 2d images and 3d scenes, 2024. 4

[41] Qi Ma, Runyi Yang, Bin Ren, Nicu Sebe, Ender Konukoglu, Luc Van Gool, and Danda Pani Paudel. Cityloc: 6dof pose distributional localization for text descriptions in large-scale scenes with gaussian representation. *arXiv preprint arXiv:2501.08982*, 2025. 2

[42] Juliette Marrie, Romain Menegaux, Michael Arbel, Diane Larlus, and Julien Mairal. Ludvig: Learning-free uplifting of 2d visual features to gaussian splatting scenes. *arXiv preprint arXiv:2410.14462*, 2024. 2, 3, 7, 8, 9, 6

[43] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020. 2

[44] 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. 3

[45] Ashkan Mirzaei, Tristan Aumentado-Armstrong, Konstantinos G. Derpanis, Jonathan Kelly, Marcus A. Brubaker, Igor Gilitschenski, and Alex Levinshtein. SPIn-NeRF: Multiview segmentation and perceptual inpainting with neural radiance fields. In *CVPR*, 2023. 3

[46] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Trans. Graph.*, 41(4):102:1–102:15, July 2022. 6

[47] Richard A. Newcombe, Shahram Izadi, Otmar Hilliges, David Molyneaux, David Kim, Andrew J. Davison, Pushmeet Kohi, Jamie Shotton, Steve Hodges, and Andrew Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In *2011 10th IEEE International Symposium on Mixed and Augmented Reality*, pages 127–136, 2011. 2

[48] Maxime Oquab, Timothée Darcet, Theo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Russell Howes, Po-Yao Huang, Hu Xu, Vasu Sharma, Shang-Wen Li, Wojciech Galuba, Mike Rabbat,

Mido Assran, Nicolas Ballas, Gabriel Synnaeve, Ishan Misra, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision, 2023. 2, 3

[49] Yatian Pang, Wenxiao Wang, Francis EH Tay, Wei Liu, Yonghong Tian, and Li Yuan. Masked autoencoders for point cloud self-supervised learning. In *European conference on computer vision*, pages 604–621. Springer, 2022. 4

[50] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 165–174, 2019. 2

[51] Songyou Peng, Kyle Genova, Chiyu Jiang, Andrea Tagliasacchi, Marc Pollefeys, Thomas Funkhouser, et al. Openscene: 3d scene understanding with open vocabularies. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 815–824, 2023. 2, 3

[52] 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 (CVPR)*, pages 20051–20060, June 2024. 2, 3, 7, 8, 9, 10, 4

[53] Yansong Qu, Shaohui Dai, Xinyang Li, Jianghang Lin, Liujuan Cao, Shengchuan Zhang, and Rongrong Ji. Goi: Find 3d gaussians of interest with an optimizable open-vocabulary semantic-space hyperplane. *arXiv preprint arXiv:2405.17596*, 2024. 2, 3, 7, 8, 9, 4

[54] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR, 2021. 3

[55] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763, 2021. 2, 4

[56] Adam Rashid, Satvik Sharma, Chung Min Kim, Justin Kerr, Lawrence Yunliang Chen, Angjoo Kanazawa, and Ken Goldberg. Language embedded radiance fields for zero-shot task-oriented grasping. In *7th Annual Conference on Robot Learning*, 2023. 3

[57] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, et al. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024. 3, 4

[58] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, Eric Mintun, Junting Pan, Kalyan Vasudev Alwala, Nicolas Carion, Chao-Yuan Wu, Ross Girshick, Piotr Dollár, and Christoph Feichtenhofer. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024. 2, 4

[59] Bin Ren, Guofeng Mei, Danda Pani Paudel, Weijie Wang, Yawei Li, Mengyuan Liu, Rita Cucchiara, Luc Van Gool, and Nicu Sebe. Bringing masked autoencoders explicit contrastive properties for point cloud self-supervised learning. In *Proceedings of the Asian Conference on Computer Vision*, pages 2034–2052, 2024. 2

[60] Zhongzheng Ren, Aseem Agarwala, Bryan Russell, Alexander G Schwing, and Oliver Wang. Neural volumetric object selection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6133–6142, 2022. 3

[61] Mike Roberts, Jason Ramapuram, Anurag Ranjan, Atulit Kumar, Miguel Angel Bautista, Nathan Paczan, Russ Webb, and Joshua M Susskind. Hypersim: A photorealistic synthetic dataset for holistic indoor scene understanding. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10912–10922, 2021. 5

[62] Barbara Roessle, Norman Müller, Lorenzo Porzi, Samuel Rota Bulò, Peter Kontschieder, Angela Dai, and Matthias Nießner. L3dg: Latent 3d gaussian diffusion. In *SIGGRAPH Asia 2024 Conference Papers*, December 2024. 2, 3

[63] David Rozenberszki, Or Litany, and Angela Dai. Language-grounded indoor 3d semantic segmentation in the wild. In *European Conference on Computer Vision*, pages 125–141. Springer, 2022. 2

[64] Sayan Deb Sarkar, Ondrej Miksik, Marc Pollefeys, Daniel Barath, and Iro Armeni. Crossover: 3d scene cross-modal alignment. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025. 2

[65] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 2

[66] Yunhang Shen, Chaoyou Fu, Peixian Chen, Mengdan Zhang, Ke Li, Xing Sun, Yunsheng Wu, Shaohui Lin, and Rongrong Ji. Aligning and prompting everything all at once for universal visual perception, 2023. 6

[67] 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 (CVPR)*, pages 5333–5343, June 2024. 2

[68] Yongliang Shi, Runyi Yang, Zirui Wu, Pengfei Li, Caiyun Liu, Hao Zhao, and Guyue Zhou. City-scale mapping system with three-layer sampling and panoptic representation. *Available at SSRN 4568335*, 2023. 2

[69] Julian Straub, Thomas Whelan, Lingni Ma, Yufan Chen, Erik Wijmans, Simon Green, Jakob J Engel, Raul Mur-Artal, Carl Ren, Shobhit Verma, et al. The replica dataset: A digital replica of indoor spaces. *arXiv preprint arXiv:1906.05797*, 2019. 3, 5

[70] Matthew Tancik, Ethan Weber, Evonne Ng, Ruilong Li, Brent Yi, Justin Kerr, Terrance Wang, Alexander Kristoffersen, Jake Austin, Kamyar Salahi, Abhik Ahuja, David McAllister, and Angjoo Kanazawa. Nerfstudio: A modular framework for neural radiance field development. In *ACM SIGGRAPH 2023 Conference Proceedings*, SIGGRAPH '23, 2023. 8

[71] Michael Tschannen, Alexey Gritsenko, Xiao Wang, Muhammad Ferjad Naeem, Ibrahim Alabdulmohsin, Nikhil Parthasarathy, Talfan Evans, Lucas Beyer, Ye Xia, Basil Mustafa, et al. Siglip 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features. *arXiv preprint arXiv:2502.14786*, 2025. 3, 4

[72] Johanna Wald, Armen Avetisyan, Nassir Navab, Federico Tombari, and Matthias Nießner. Rio: 3d object instance re-localization in changing indoor environments. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7658–7667, 2019. 5

[73] Zeming wei, Junyi Lin, Yang Liu, Weixing Chen, Jingzhou Luo, Guanbin Li, and Liang Lin. 3daffordsplat: Efficient affordance reasoning with 3d gaussians, 2025. 3

[74] Andrew P. Witkin. Recovering surface shape and orientation from texture. *Artificial Intelligence*, 17(1):17–45, 1981. 2

[75] Xiaoyang Wu, Daniel DeTone, Duncan Frost, Tianwei Shen, Chris Xie, Nan Yang, Jakob Engel, Richard Newcombe, Hengshuang Zhao, and Julian Straub. Sonata: Self-supervised learning of reliable point representations. In *CVPR*, 2025. 4

[76] Xiaoyang Wu, Li Jiang, Peng-Shuai Wang, Zhijian Liu, Xihui Liu, Yu Qiao, Wanli Ouyang, Tong He, and Hengshuang Zhao. Point transformer v3: Simpler, faster, stronger. In *CVPR*, 2024. 2, 4

[77] Xiaoyang Wu, Yixing Lao, Li Jiang, Xihui Liu, and Hengshuang Zhao. Point transformer v2: Grouped vector attention and partition-based pooling. *Advances in Neural Information Processing Systems*, 35:33330–33342, 2022. 4

[78] 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. In *Advances in Neural Information Processing Systems*, volume 37, 2024. 2, 3, 7, 8, 9, 4

[79] Zirui Wu, Tianyu Liu, Liyi Luo, Zhide Zhong, Jianteng Chen, Hongmin Xiao, Chao Hou, Haozhe Lou, Yuantao Chen, Runyi Yang, Yuxin Huang, Xiaoyu Ye, Zike Yan, Yongliang Shi, Yiyi Liao, and Hao Zhao. Mars: An instance-aware, modular and realistic simulator for autonomous driving. *CICAI*, 2023. 2

[80] Jianfeng Xiang, Zelong Lv, Sicheng Xu, Yu Deng, Ruicheng Wang, Bowen Zhang, Dong Chen, Xin Tong, and Jiaolong Yang. Structured 3d latents for scalable and versatile 3d generation. *arXiv preprint arXiv:2412.01506*, 2024. 2

[81] Runsen Xu, Xiaolong Wang, Tai Wang, Yilun Chen, Jiangmiao Pang, and Dahua Lin. Pointllm: Empowering large language models to understand point clouds. In *European Conference on Computer Vision*, pages 131–147. Springer, 2024. 9

[82] Jihan Yang, Runyu Ding, Weipeng Deng, Zhe Wang, and Xiaojuan Qi. Regionplc: Regional point-language contrastive learning for open-world 3d scene understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19823–19832, 2024. 2

[83] Runyi Yang, Zhenxin Zhu, Zhou Jiang, Baijun Ye, Xiaoxue Chen, Yifei Zhang, Yuantao Chen, Jian Zhao, and Hao Zhao. Spectrally pruned gaussian fields with neural compensation. *arXiv preprint arXiv:2405.00676*, 2024. 8

[84] Mingqiao Ye, Martin Danelljan, Fisher Yu, and Lei Ke. Gaussian grouping: Segment and edit anything in 3d scenes. In *ECCV*, 2024. 2

[85] Vickie Ye, Ruilong Li, Justin Kerr, Matias Turkulainen, Brent Yi, Zhuoyang Pan, Otto Seiskari, Jianbo Ye, Jeffrey Hu, Matthew Tancik, and Angjoo Kanazawa. gsplat: An open-source library for gaussian splatting. *Journal of Machine Learning Research*, 26(34):1–17, 2025. 2, 5

[86] Chandan Yeshwanth, Yueh-Cheng Liu, Matthias Nießner, and Angela Dai. Scannet++: A high-fidelity dataset of 3d indoor scenes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12–22, 2023. 3, 5, 8, 9

[87] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 11975–11986, 2023. 3

[88] Hao Zhang, Feng Li, Shilong Liu, Lei Zhang, Hang Su, Jun Zhu, Lionel M Ni, and Heung-Yeung Shum. Dino: Detr with improved denoising anchor boxes for end-to-end object detection. *arXiv preprint arXiv:2203.03605*, 2022. 3

[89] Kai Zhang, Sai Bi, Hao Tan, Yuanbo Xiangli, Nanxuan Zhao, Kalyan Sunkavalli, and Zexiang Xu. Gs-lrm: Large reconstruction model for 3d gaussian splatting. *European Conference on Computer Vision*, 2024. 2

[90] Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip HS Torr, and Vladlen Koltun. Point transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 16259–16268, 2021. 4

[91] Yuhang Zheng, Xiangyu Chen, Yupeng Zheng, Songen Gu, Runyi Yang, Bu Jin, Pengfei Li, Chengliang Zhong, Zengmao Wang, Lina Liu, et al. Gaussiangrasper: 3d language gaussian splatting for open-vocabulary robotic grasping. *IEEE Robotics and Automation Letters*, 2024. 3

[92] Ding Zhong, Xu Zheng, Chenfei Liao, Yuanhuiyi Lyu, Jialei Chen, Shengyang Wu, Linfeng Zhang, and Xuming Hu. Omnisam: Omnidirectional segment anything model for uda in panoramic semantic segmentation. *arXiv preprint arXiv:2503.07098*, 2025. 4

[93] Shijie Zhou, Haoran Chang, Sicheng Jiang, Zhiwen Fan, Zehao Zhu, Dejia Xu, Pradyumna Chari, Suya You, Zhangyang Wang, and Achuta Kadambi. Feature 3dgs: Supercharging 3d gaussian splatting to enable distilled feature fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21676–21685, 2024. 2, 3, 7, 8, 9, 4

[94] Yichao Zhou, Jingwei Huang, Xili Dai, Shichen Liu, Linjie Luo, Zhili Chen, and Yi Ma. Holicity: A city-scale data platform for learning holistic 3d structures, 2021. 3, 4, 8, 9

[95] Xiangyang Zhu, Renrui Zhang, Bowei He, Ziyu Guo, Ziyao Zeng, Zipeng Qin, Shanghang Zhang, and Peng Gao. Pointclip v2: Prompting clip and gpt for powerful 3d open-world learning. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 2639–2650, 2023. 2

[96] Chen Ziwen, Hao Tan, Kai Zhang, Sai Bi, Fujun Luan, Yicong Hong, Li Fuxin, and Zexiang Xu. Long-lrm: Long-sequence large reconstruction model for wide-coverage gaussian splats. *arXiv preprint 2410.12781*, 2024. 2, 3

[97] Xingxing Zuo, Pouya Samangouei, Yunwen Zhou, Yan Di, and Mingyang Li. Fmgs: Foundation model embedded 3d gaussian splatting for holistic 3d scene understanding. *arXiv preprint arXiv:2401.01970*, 2024. 2, 3, 7, 8, 9, 4, 6

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: We will publicly release our proposed dataset, SceneSplat-49K, mentioned in Section 3 along with all benchmark codes. A complete summary of the benchmark experiments and key findings was mentioned in Section 4.1 and 4.2.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: Limitations are discussed in Section 5.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

Answer: [NA]

Justification: The paper does not include theoretical results.

Guidelines:

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

4. **Experimental result reproducibility**

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

Answer: [Yes]

Justification: We provide detailed implementations of both the proposed and benchmark methods in the main paper and supplementary materials. All code, benchmarks, and datasets will be released publicly.

Guidelines:

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

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

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

Answer: [Yes]

Justification: Our code, benchmark and datasets will be made public.

Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: The experimental settings are mentioned in Section 4.1, as well as the supplementary materials.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [No]

   Justification: We follow the standard practices of Language Gaussian Splatting for reporting zero-shot semantic segmentation metrics. In large-scale validation scenarios, the reported mean metric serves as a reliable performance indicator. Furthermore, performing all experiments was very computationally expensive.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

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

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: We mention the computer resources used for all the experiments in Section 4.1 and for the dataset in Tab. 2, as well as the supplementary materials.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: All authors have thoroughly reviewed the full NeurIPS Code of Ethics and confirm that our work adheres to every guideline therein, no personally identifiable information is processed.

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [Yes]

    Justification: Broader impacts are discussed in the supplementary materials.

    Guidelines:

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

11. **Safeguards**

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

    Answer: [Yes]

    Justification: Models and code are hosted behind a license that prohibits harmful uses. A brief "Acceptable Use" document accompanies all downloads. We reserve the right to revoke repository access if misuse is detected. A dedicated email address is provided for the community to report vulnerabilities or harmful behavior.

    Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: The original authors and owners of the data used to construct our dataset are mentioned in Section 3. The licence details for the mentioned data is also mentioned in the supplementary materials.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.

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

13. **New assets**

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

    Answer: [Yes]

    Justification: The dataset creation workflow is detailed in Section 3 and the supplementary materials, and the benchmark methodology is fully described in 4 and the supplementary materials. The benchmark code will also be released publicly.

    Guidelines:

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

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

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

    Answer: [NA]

    Justification: The paper does not involve crowd-sourcing or research with human participants.

    Guidelines:

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

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

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

    Answer: [NA]

Justification: The paper does not involve crowd-sourcing or research with human participants.

Guidelines:

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

16. **Declaration of LLM usage**

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

Answer: [NA]

Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (https://neurips.cc/Conferences/2025/LLM) for what should or should not be described.

SceneSplat++: A Large Dataset and Comprehensive Benchmark for Language Gaussian Splatting

Supplementary Material

This supplementary material provides additional information about the dataset, licensing, the benchmark and future research directions.

Contents

1 Introduction **2**

2 Related Works **3**

3 Dataset **4**

3.1 Data Collection and Processing 5

3.2 Vision-Language Embedding Collection 6

4 Benchmark **6**

4.1 Benchmark settings 6

4.2 Key Insights 7

5 Conclusion and Limitations **10**

A Dataset Collection and Processing **1**

A.1 Details of Each Data Source 1

A.2 Dataset License 2

B Benchmark Settings **4**

B.1 Details of Running Benchmark Methods 4

B.2 Additional Benchmark Results 7

C Further Research Directions **8**

C.1 Gaussian LLM 8

C.2 Gaussian Reasoning 9

D Broader Impact **10**

A Dataset Collection and Processing

A.1 Details of Each Data Source

SceneSplat-7K. We refer to [31] for processing details of the included ScanNet, ScanNet++, ScanNet++ v2, Replica, Hypersim, 3RScan, ARKitScenes, and Matterport3D data sources. We follow the same language label collection process as detailed in its supplementary.

DL3DV-10K. The DL3DV-10K [33] is a large-scale 3D vision dataset comprising 51.2 million frames from 10,510 videos captured at 65 real-world locations, featuring diverse scenes with varying

lighting, transparency, and reflections. It provides high-resolution 4K images (2160×3840 at 60 fps), which we downsample to 540×960 for training Gaussian Splatting models. Unlike other datasets, DL3DV-10K provides only RGB images and depth supervision is disabled. To address large-scale variations, we employ vanilla Gaussian Splatting without a fixed number of Gaussians, and we filter out low-quality scenes with PSNR values below 21 dB.

HoliCity. The dataset covers 6,253 real-world panoramas of resolution 13312×6656 that are aligned with the CAD model of downtown London with an area of more than $20\,\mathrm{km}^2$. We use the converted perspective images, which are equipped with pseudo depth maps estimated from the CAD map and the semantic segmentation labels. The original semantic labels are: `Sky or nothing, Buildings, Roads, Terrains, Trees, Others`. We map `Sky or nothing` and `Others` to the ignore label of -1 and add the vehicle and pedestrian classes in our query. Since these two classes are not present in the ground truth labels and are flattened, we map them to the terrains and roads classes respectively before calculating the metrics. The original test split does not provide the depth maps and semantic labels, we use the original validation split instead for our benchmark.

Aria Synthetic Environments. We follow the official documentation to undistort the RGB frames and depth maps. Devignetting is performed using the provided gain map. Because the camera frustum has a conical shape, we ignore the black border regions during 3DGS optimization. We fuse the depth maps into point clouds and apply voxel downsampling with grid size $3\,\mathrm{cm}$. The threshold for the total number of Gaussians is set to be 1.5 times the number of initial points.

Dataset Splits for Benchmark. For our benchmark results on full evaluation scenes reported in the main paper, we use the official splits provided by each dataset, *i.e.*, ScanNet `val` split (312 scenes), ScanNet++ `nvs_sem_val` split (50 scenes), Matterport3D `test` split (370 scenes), and HoliCity `val` split (328 scenes). For the benchmarks on 10 selected scenes, the scenes were randomly chosen from the full evaluation scenes as detailed in Tab. A.

Visualization. We provide visualizations that showcase the photorealistic appearance and accurate geometry rendering reconstructions through all the different sources. See Fig. H and Fig. G.

ScanNet	ScanNet++	Matterport3D	HoliCity
scene0011_00	09c1414f1b	2t7WUuJeko7_02	ytwUEEljP6RgoV0MviqvsQ_LD
scene0011_01	0d2ee665be	5ZKStnWn8Zo_15	yuVpITQv74rpaWqX4kEW4Q_HD
scene0153_00	38d58a7a31	ARNzJeq3xxb_07	yxYrxgMjOWCeJ3a_KyBXKA_LD
scene0153_01	3db0a1c8f3	fzynW3qQPVF_00	yy03mA02i0OxVdMb9klRgg_LD
scene0329_01	5ee7c22ba0	jtcxE69GiFV_32	z0IhWSN8uhWDltDqmwdY8g_HD
scene0329_02	5f99900f09	pa4otMbVnkk_27	z37U-BzqOo0_mSeJh7gXbg_LD
scene0435_00	a8bf42d646	q9vSo1VnCiC_05	z5af-sE80KZQc_9Se7Aq_g_LD
scene0435_01	a980334473	rqfALeAoiTq_11	z5l-XERj7c-N4Rdk4JSLeg_LD
scene0578_01	c5439f4607	UwV83HsGsw3_22	z8hAivfNyPaRSobbXCR8jQ_LD
scene0578_02	cc5237fd77	wc2JMjhGNzB_11	z9UseFv2rAwNfe21MuYxZQ_LD

Table A: **Selected 10 Scenes for Benchmarks.**

A.2 Dataset License

Our proposed dataset construction process includes multiple existing 3D datasets, each with specific license requirements. Tab. B details the license of each component dataset.

To respect all component licenses while making our proposed dataset accessible to the research community, we employ the following approaches based on each dataset's licensing terms:

1. **Direct distribution with attribution**: For datasets whose licenses permit redistribution under non-commercial research purposes (ARKitScenes, Hypersim, DL3DV-10K), we include the original licenses as requirements before providing our data and provide proper attribution to the original authors.
2. **Distribution after approval**: For datasets with less clear redistribution terms, we have contacted the original authors and asked for permission to redistribute our processed data. Our proposed use case does not conflict with the existing licenses.

Dataset	License / Terms	Allowed Purposes	Redistribution	Approach
ScanNet	ScanNet Terms of Use	Only for non-commercial research and educational purposes.	No, but the authors agreed to host our GS on their website.	3
ScanNet++	ScanNet++ Terms of Use	Only for non-commercial research and educational purposes.	No, but the authors agreed to host our GS on their website.	3
ARKitScenes	Apple Software License	Non-commercial, non-exclusive license; granted use, installation, modification and redistribution	Yes	1
Hyperim	CC BY-SA 3.0	Free to share and adapt under the Attribution and ShareAlike terms	Yes	1
3RScan	3RScan Terms of Use	Only for non-commercial research and educational purposes.	Yes	1
Matterport3D	Matterport End User License Agreement for Academic Use of Model Data	Non-commercial academic use only.	Yes (Required to include the Agreement, or a hyperlink to this Agreement)	3
Replica	Replica Dataset Research Terms	Research or educational purpose that is non-commercial or not-for-profit.	Yes	1
DL3DV-10K	DL3DV-10K Terms of Use & CC BY-NC 4.0	Only for non-commercial research and educational purposes.	Yes	1
HoliCity	HoliCity Terms of Use	Non-commercial purposes.	No, currently in active contact to reach arrangement.	4
Aria Synthetic Environments	Aria Synthetic Environments Dataset License Agreement	Non-commercial research.	No, currently in active contact to reach arrangement.	4
Internet Collected Data	Polycam Terms of Service, Sketchfab License Agreement	Cannot be used for any commercial or promotional use.	No, but we offer the selected high-quality scenes with URL and SceneId.	4

Table B: **Data Source License Summary.**

3. **Hosting through original dataset platforms**: For datasets with custom terms that restricts free redistribution (ScanNet, ScanNet++), we reach agreement with the original dataset authors and host our processed data on their official channel.

4. **Reproduction by downloading from the original sources**: For datasets that prohibit redistribution, including any derivative works, we will not share the processed data directly. We provide the processing scripts and language label collection, so the users can process the data themselves after downloading the original datasets.

Config	LangSplat[52]	OpenGaussian [78]	Feature-3DGS [93]	FMGS [97]	GOI [53]
2D Feature	SAM+CLIP	SAM+CLIP	LSeg	DINOv2	SAM+CLIP
Text Encoder	CLIP	CLIP	CLIP	CLIP	CLIP
Rasterizer	Custom	3DGS	Custom	Custom	Custom
Iteration	30k	70k	7k	32.5k	1.5k
Feat. Opti. Only	✓	✗	✗	✗	✓
Feature Lr	2.5e-3	2.5e-3	2.5e-3	1e-2	2.5e-3
Feat. Compression	Autoencoder	N.A.	CNN	Hash Code	Feature Codebook
Feature Dim.	3	6	256	512	10
Inference GPU	A6000*1 (48G)	A6000*1 (48G)	A6000*1 (48G)	H200*1 (141G)	A6000*1 (48G)

Table C: **Configurations of Per-Scene Optimization Methods.**

For the that datasets fall under the scripts-only approach, we are actively contacting the respective dataset teams to request potential hosting options.

In conclusion, our SceneSplat-49K dataset components are licensed as follows: **(1)** Language features are made available under CC BY-SA 4.0. **(2)** 3D Gaussian Splatting scenes must be used in accordance with the original license terms as detailed in Tab. B. Users must comply with all component dataset licenses when using our data. This follows similar procedures used in recent datasets such as [40].

B Benchmark Settings

B.1 Details of Running Benchmark Methods

The implementation details of Per-Scene optimization methods are shown in Tab. C, we provide detailed information about the vision-language models used, which fall into two main categories for feature extraction visualized in Fig. B and the zero shot segmentation performance using these features are reported in Tab. G. The tile-based method produces instance-level features by leveraging SAM-generated masks, while the LSeg-based method extracts pixel-level features using the LSeg image encoder. Additional details reported include the rasterizer employed, the number of training iterations, and whether the method is used to optimize only the features or both the features and the Gaussian primitives. We also include information on the feature compression strategy and the type of GPUs used for training. In Tab. D we report details on the selected features, the Gaussian rasterizer configuration, the dimensionality of the lifted features, and whether pruning was applied to reduce the number of Gaussians. The training details of SceneSplats across different benchmarks are provided in Tab. E.

LangSplat. The method first proposes tile-based splatting to learn hierarchical semantics from SAM and CLIP models as shown in Fig. A. Specifically, the approach utilizes the automatic mask generator of SAM, where mesh grid points are used as prompts. By controlling the spacing between grid points, the method generates masks corresponding to different semantic levels, including "whole," "part," and "subpart." Each masked region is then cropped, resized to 224×224 pixels with a black background, and subsequently passed through the CLIP image encoder. This design utilizes the CLIP encoder's capacity to generate global semantic representations of input images, effectively reducing background interference while enhancing focus on the target object. To accelerate rasterization, LangSplat employs an autoencoder to project 512-dimensional CLIP features into a lower-dimensional (3-dimension) space. The autoencoder architecture consists of a 5-layer encoder with dimensions [256, 128, 64, 32, 3] and a decoder with dimensions [16, 32, 64, 128, 256, 256, 512], trained for 1,000 epochs with a learning rate of 7e-4. As shown in Tab. G, this achieves effective compression while maintaining feature quality, measured by average cosine similarity loss. For rasterization, LangSplat introduces a custom rasterizer that treats the compressed 3D features as RGB-like channels, enabling efficient processing without modifying the underlying CUDA kernel functions. In the main paper, we report the best results among the three semantic levels.

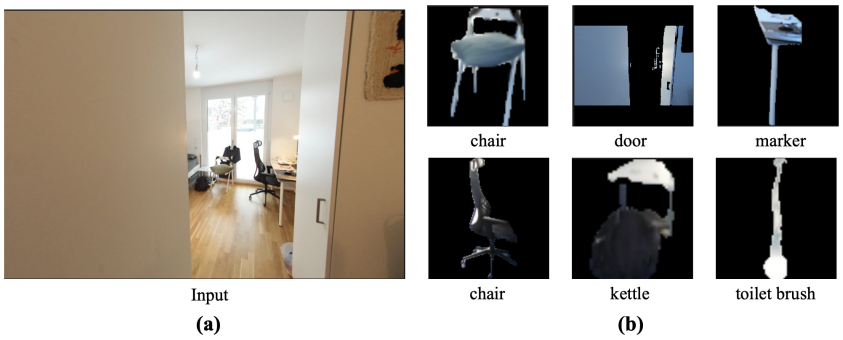


Figure A: **Illustration of Tile-Based Feature Extraction.** (a) The input frame. (b) The SAM-generated masks are cropped and resized to 224×224. The accompanying text shows the zero-shot prediction results for each cropped image. However, some predictions are incorrect due to imprecise object instance composition. For example, in the top-middle prediction on the right, a wall and a wardrobe together form a crop that resembles a door. Similarly, a combination of a bedside cabinet and a backpack is mistakenly predicted as a kettle. Context also plays an important role: as shown on the right, a partial view of a desk is misclassified, and a lamp is incorrectly identified as a toilet brush.

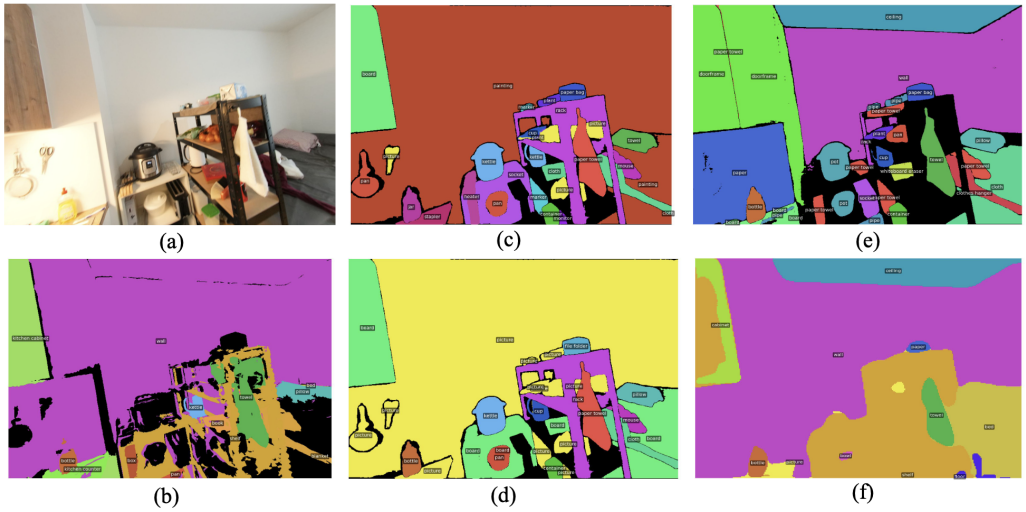


Figure B: **2D Segmentation Using Different 2D Extracted Features.** (a) Input frame, (b) Ground truth segmentation, (c) CLIP zero-shot segmentation, (d) Autoencoder-restored zero-shot segmentation, (e) SigLIP2 zero-shot segmentation, (f) LSeg zero-shot segmentation. The features above demonstrate the strengths and weaknesses of different feature extraction methods. All examples in (c), (d), and (e) use tile-based features extracted from SAM, which clearly preserve object boundaries and capture fine details in small objects, such as the kettle. In contrast, (f), which uses LSeg, treats the entire shelf as a single instance, failing to distinguish finer structural elements. Additionally, the ground truth is noisy and misses some boundary details. We also observe that the SigLIP2 model outperforms CLIP by successfully recognizing elements such as the ceiling and walls. Notably, features decoded by the autoencoder demonstrate improved instance-level understanding in some cases, potentially due to the suppression of high-frequency information.

OpenGaussian. The method follows LangSplat for 2D feature extraction and utilizes whole-object masks for training, employing a two-stage codebook discretization process. In the first stage, a 6D feature space is projected onto instance masks. A smoothing loss is applied to encourage feature consistency within each mask, while a contrastive loss separates features across different masks. This stage is trained for 30,000 iterations. In the second stage, continuous instance features are transformed into quantized features and indices through a coarse-to-fine codebook learning approach, which also requires 30,000 iterations. To handle occlusion, the coarse-level clustering integrates 3D spatial coordinates and applies an L1 loss to minimize discrepancies between rendered instance features and their corresponding pseudo-instance features generated from cluster centers. At the fine level, an L2 loss is applied exclusively within the object silhouette region, focusing on feature-level optimization

Config	Gradient-Weighted 3DGS[20]	LUDVIG[42]	OccamLGS[9]	SceneSplat (PL)[31]
2D Feature	LSeg	DINOv2 + CLIP	SAM + CLIP	SAM + SigLIP2
Text Encoder	CLIP	CLIP	CLIP	SigLIP2
Rasterizer	gsplat	3DGS	Custom	gsplat
Filter	N.A	N.A.	✓	N.A.
Feature Dim.	512	512	512	768
Prune	✓	N.A.	✓	N.A.
Inference GPU	A6000*1 (48G)	A6000*1 (48G)	A6000*1 (48G)	A6000*1 (48G)

Table D: **Configurations of Other Methods.**

BenchMark	ScanNet (val)	ScanNet++ (val)	Matterport3D (test)	HoliCity (val)
Training set	ScanNet++ (v2) + ScanNet + M3D	ScanNet++ (v2) + ScanNet + M3D	ScanNet++ (v2) + ScanNet + M3D	HoliCity
Batch Size	8	8	8	8
Optimizer	AdamW	AdamW	AdamW	AdamW
Lr	0.006	0.006	0.006	0.006
Weight Decay	0.05	0.05	0.05	0.05
Epoch	400	400	400	400
Inference GPU	A6000*1 (48G)	A6000*1 (48G)	A6000*1 (48G)	A6000*1 (48G)

Table E: **Configurations of SceneSplat Method.**

for precise boundary refinement. Similarly, the rasterizer is run twice as RGB-like channels to obtain a 6-dimensional instance feature, while the official rasterizer is still used in subsequent processing.

Feature-3DGS. The method develops a parallel rasterizer capable of processing arbitrary-dimensional features, demonstrating its effectiveness through feature distillation using SAM and LSeg [29]. While the approach optimizes rendering for high-dimensional features similar to GSplat, processing 512-dimensional features imposes significant memory overhead. To address this, it employs a speed-optimized mode that renders only 256-dimensional features and subsequently lifts them to higher dimensions via 2D convolutional networks. All reported results are obtained by projecting these 2D features back to 3D and merging them into the 3D language field and segmentation labels. Following the original implementation, we optimize both Gaussian parameters and features for 7K iterations.

FMGS. The method adopts the official implementation of Foundation-Model-embedded 3D Gaussian Splatting (FMGS) [97], which augments 3DGS with a multi-resolution hash table and distills CLIP and DINO feature maps into the splats for view-consistent semantics. FMGS retrains language gaussian splatting per scene for 32.5k iterations, keeping the RGB-reconstruction and pixel-alignment distillation losses. It follows Instant-NGP [46] and queries a trainable hash grid at the Gaussian centroid, passing the returned code through a light MLP to obtain semantic embedding. This design cuts the memory footprint by two orders of magnitude while retaining view-consistent features.This method in default setting could reach around 50$\sim$80 GB GPU memory, thus all the experiments are conducted on NVIDIA H200.

GOI. Our approach diverges from the original paper's modified APE [66] for pixel-level feature extraction, which requires excessive storage for large datasets. Instead, we implement the SAM and CLIP pipeline with the crop fusion strategy from [31]. We train the language feature field following GOI's official implementation and increase the feature codebook entries to 500 in response to the increase in scene size. Rather than using the paper's computationally intensive fine-tuning approach for relevancy scoring across our numerous scenes and classes, we employ the 3D CLIP relevancy score without fine-tuning for evaluation.

Gradient-Weighted 3DGS. It first performs pruning by removing Gaussians whose gradients remain above a specified threshold. It then computes the gradients of the rendered RGB attributes with zero color, which are equivalent to the alpha weights. Subsequently, 3D features are obtained as

a weighted sum of 2D features extracted by the LSeg image encoder, using the minimal LSeg implementation. This design effectively lifts 2D features into 3D space without compression, leading to strong performance. The method leverages the gsplat rasterizer for rendering.

LUDVIG. The method extracts and uplifts features from SAM, DINOv2, and CLIP, and evaluates them on downstream tasks of segmentation and open-vocabulary object localization. However, for the segmentation task, the method assumes that a foreground mask of the object to be segmented is provided on one reference frame, which does not correspond well with our open-vocabulary 3D evaluation setting. We thus use the 3D CLIP relevancy score returned from the method to perform zero-shot 3D semantic segmentation.

OccamLGS. We use the open-source implementation for our benchmark and keep the original pipeline of weighted aggregation of multi-view features. The only adaptation is that we change the 2D feature map collection process when running the SAM and CLIP models, and adopt the crop fusion strategy as in [31], which we find beneficial for encoding complex scenes.

SceneSplat. We extend the training dataset of SceneSplat [31] to ScanNet, ScanNet++, and Matterport3D, in a total of 3,503 scenes, and report the results from this joint training on all the indoor benchmarks. The results on the HoliCity benchmark are reported using the model trained only on this dataset. The hyperparameters used are the same as in the original method ,except we decrease the data epochs from 800 to 400 for the joint training.

Text Prompts for Benchmark. For all the indoor benchmarks on zero-shot 3D semantic-segmentation, we generate text prompts with the fixed template "`this is a/an <class name>`". For the HoliCity benchmark, we use the following texts:

```
this is a building or other construction
this is a road or street
this is terrain or ground near a road
this is part of a tree or plant
this is a car or vehicle
this is a pedestrian
```

B.2 Additional Benchmark Results

Comparison of CLIP and SigLIP2 Feature. Tab. F provides the baseline benchmark results when using the same SigLIP2 features. LUDVIG shows a relative improvement of 34.6%, 50.5%, and 13.1% in f-mIoU across ScanNet++, ScanNet200, and Matterport3D benchmarks, and OccamLGS observes more improvements, which gains 54.7%, 89.4%, and 59.3% in f-mIoU respectively. However, the generalizable method SceneSplat still outperforms both baselines when all use SigLIP2 features. The comparison of zero-shot performance on 2D tasks using CLIP and SigLIP2 features is presented in Tab. G, with qualitative results shown in Fig. B.

Method	**Features**	**ScanNet++**		**ScanNet200**		**Matterport3D**	
		f–mIoU	f–mAcc	f–mIoU	f–mAcc	f–mIoU	f–mAcc
LUDVIG	CLIP	0.1099	0.2547	0.0602	0.1701	0.1960	0.3738
	SigLIP2	0.1479	0.2958	0.0906	0.2008	0.2217	0.3909
OccamLGS	CLIP	0.1502	0.3312	0.1204	0.2503	0.1725	0.3151
	SigLIP2	0.2324	0.4155	**0.2280**	0.3590	0.2748	0.4384
SceneSplat	SigLIP2	**0.2836**	**0.4992**	0.2140	**0.3874**	**0.3384**	**0.5745**

Table F: **Baseline Methods Performance with CLIP vs. SigLIP2 Features.** When equipped with the same SigLIP2 features, the baseline methods in general do not reach the performance of the generalizable method.

Zero-Shot 3D Outdoor Segmentation on KITTI-360 [32] Dataset. To evaluate the methods' performance in complex outdoor environments, we extend the experiments on KITTI-360 [32] Dataset, which has 37 semantic classes compared to 4 classes in HoliCity. All methods are evaluated on a 16-scene mini-validation set, as shown in Tab. H. We observe that most methods are unable to effectively process large-scale scenes containing over 10 million Gaussians. Therefore, we divide

Features	SAM+CLIP		SAM+AE-CLIP			SAM+CLIP-Fuse		SAM+SigLIP2		LSeg	
	f–mIoU	f–mAcc	f–mIoU	f–mAcc	Cosine	f–mIoU	f–mAcc	f–mIoU	f–mAcc	f–mIoU	f–mAcc
ScanNet++	0.1051	0.2542	0.1328	0.2086	0.2022	0.0873	0.2264	**0.1341**	**0.3504**	0.1265	0.2554
HoliCity	0.1233	0.2353	0.1377	0.2605	0.1644	0.0988	0.1177	**0.2104**	**0.3658**	0.1040	0.2751

Table G: **2D Zero-Shot Segmentation Results Using Different 2D Extracted Features on ScanNet++ (10 scenes) and HoliCity (10 scenes).** Reported f–mIoU / f–mAcc for indoor datasets and mIoU / mAcc for HoliCity. The above features **SAM+CLIP** combines whole level SAM masks with the CLIP encoder as in LangSplat; **SAM+AE+CLIP** compresses the prior features into a 3-dimensional latent space using an autoencoder and reconstructs them. The reconstruction error is evaluated using cosine similarity; SAM+CLIP+Fuse introduces dynamic feature weighting following the method proposed in [31]; SAM+SigLIP2 leverages 2D features from SceneSplat (pseudo labels), incorporating both dynamic weighting and SigLIP2 features; and LSeg extracts pixel-level features using the LSeg image encoder.

each scene into 50m × 50m chunks. Under this setting, Feature-3DGS and Gradient-Weighted 3DGS achieve significantly better results compared to other methods. This performance gap likely stems from differences in 2D feature encoding: both well-performing methods employ LSeg [30], which is specifically pretrained on large-scale outdoor data and excels in outdoor scenes. In contrast, the under performing methods rely on SAM-tiles-based features, which appear less effective in this setting. This trend is also observed in the Holicity experiments. Additionally, Feature-3DGS performs worse than Gradient-Weighted 3DGS, possibly due to its use of low-dimensional feature compression, which may lose discriminative detail compared to gradient-weighted aggregation.

Even though Feature-3DGS and Gradient-Weighted 3DGS performance stands out in complex outdoor settings, they do not stand out in indoor scenes because LSeg's patch-wise ViT encoding struggles to preserve fine object boundaries and small objects. While the data-driven method SceneSplat was trained exclusively on indoor data, our evaluation demonstrates its ability to generalize to outdoor scenes, although extending its training data is vital to improve the outdoor performance.

Method	mIoU	mAcc
Per-Scene Optimization Methods		
LangSplat [52]	0.0340	0.0873
OpenGaussian [78]	0.0360	0.0655
Feature-3DGS [93]	0.0691	0.0427
FMGS [97]	0.0020	0.0020
GOI [53]	0.0265	0.0538
Per-Scene Optimization-Free Methods		
Gradient-Weighted 3DGS [20]	**0.2042**	**0.2751**
LUDVIG [42]	0.0410	0.0912
OccamLGS [9]	0.0962	0.1505
Generalizable Method		
SceneSplat [31]	0.0330	0.0494

Table H: **Zero-Shot 3D Outdoor Segmentation on KITTI-360 [32] Dataset. All methods are evaluated on a randomly selected 16-scene mini-validation set. The metrics exclude the following classes: sky, unknown construction, unknown vehicle, unknown object.**

Qualitative Visualizations. See more qualitative examples of zero-shot 3D semantic segmentation for the benchmark methods in Fig. C, Fig. D, and text query results in Fig. E and Fig. F. SceneSplats accurately predicts the cloth behind the chair, which is not labeled in the ground truth. In query experiments, the tile-based feature splatting method successfully detects the hanger in the corner, while the LSeg-based method fails.

C Further Research Directions

C.1 Gaussian LLM

A promising direction is a framework that integrates 3D Gaussian Splatting with large language models to enable more complex natural language interactions. Building upon the success of generalizable

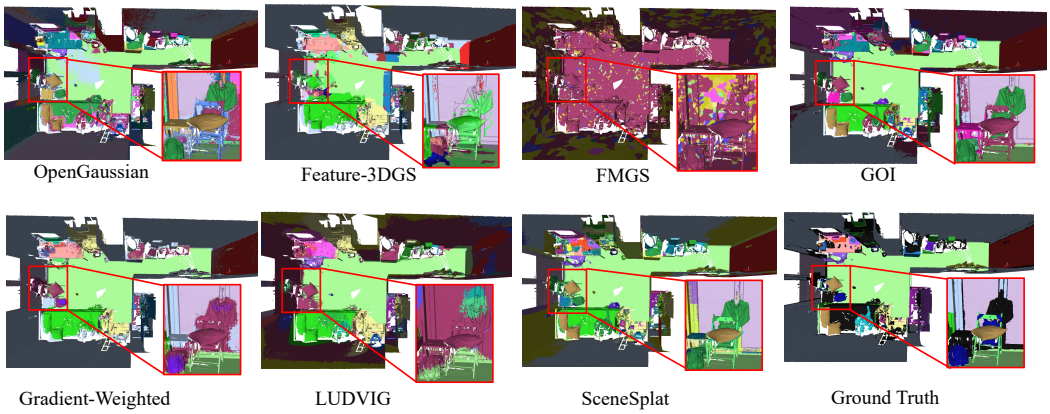


Figure C: **Qualitative Results of Zero-Shot 3D Semantic Segmentation on ScanNet++.** The semantic classes "chair" and "pants" are highlighted, which are not fully labeled in Ground Truth. (Zoom in to check details.)

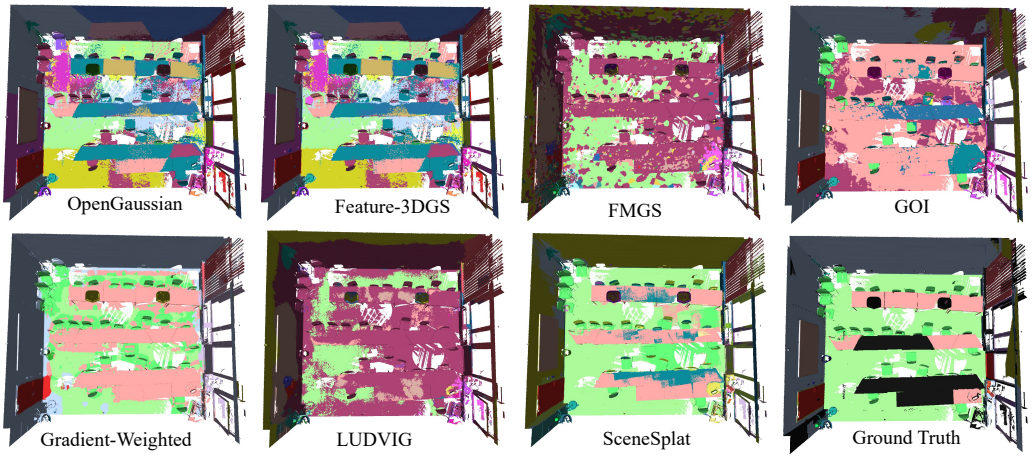


Figure D: **Qualitative Results of Zero-Shot 3D Semantic Segmentation on ScanNet++.** (Zoom in to check details.)

3DGS models demonstrated in our benchmark, Gaussian-LLM could encode 3D scenes as collections of Gaussian primitives with associated features, which are then projected into a language model's embedding space through a specialized adapter. This approach would surpass existing work on a single object level [81] and leverage the inherent advantages of 3DGS—efficient representation, high visual fidelity, and explicit geometry—while harnessing the reasoning capabilities of LLMs for tasks such as spatial relationship understanding, scene description, and multimodal question answering. Moreover, Gaussian-LLM could benefit from training on our SceneSplat-49K dataset of diverse indoor and outdoor scenes.

C.2 Gaussian Reasoning

Beyond language alignment, another promising frontier is Gaussian-based visual reasoning, which extends traditional point cloud reasoning paradigms into a richer, multimodal paradigm. Unlike point clouds that primarily encode geometric structure, 3D Gaussian Splatting provides both fine-grained texture and appearance in addition to geometry—enabling higher-level understanding tasks such as reading text on surfaces, recognizing affordances, or identifying visual semantics that pure geometry cannot capture. This opens a pathway to scene-level reasoning grounded in full visual context, a capability that point-based representations fundamentally lack. Moreover, reinforcement learning

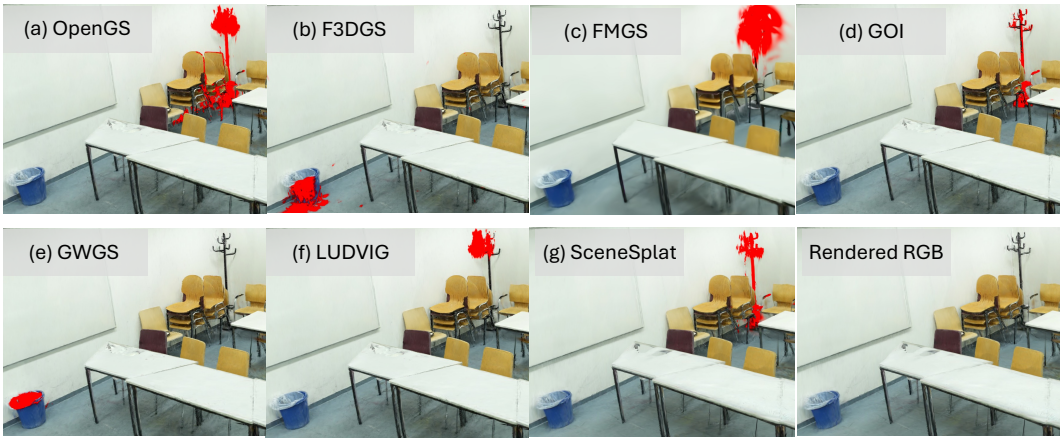


Figure E: **Text-Based Scene Query.** Given the prompt "This is a hanger." to different LGS methods, the queried parts are highlighted in red.

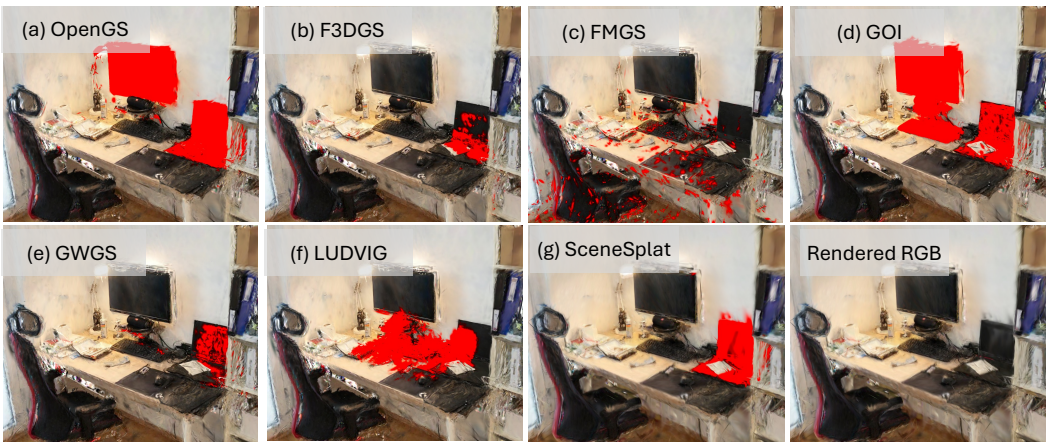


Figure F: **Text-Based Scene Query.** Given the prompt "This is a laptop." to different LGS methods, the queried parts are highlighted in red.

(RL)-based approaches could be integrated to exploit the active and interactive nature of 3DGS environments—learning to reason not just over static inputs, but over dynamic tasks that evolve through interaction (*e.g.*, navigation, embodied QA, or interactive captioning).

D Broader Impact

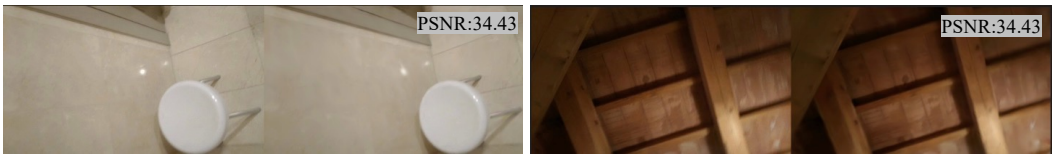
While our work on introducing a large-scale Gaussian dataset for 3D tasks represents a significant advancement in 3D representation learning, it is important to consider potential societal impacts. First, since our dataset is built upon existing public datasets, it shares similar privacy risks, including the potential for unintentional inclusion of identifiable individuals or sensitive scenes. Second, the dataset is currently focused on open-vocabulary tasks while more potential annotation can be added. Third, training large-scale Gaussian Splatting models requires considerable computational resources, leading to high energy consumption and a larger carbon footprint—an increasingly important concern as AI models continue to scale.

Figure G: **Renderings From Outdoor Gaussian Splatting Scenes (HoliCity).** (Left) Gaussian Splats Render Result and (right) Ground Truth.

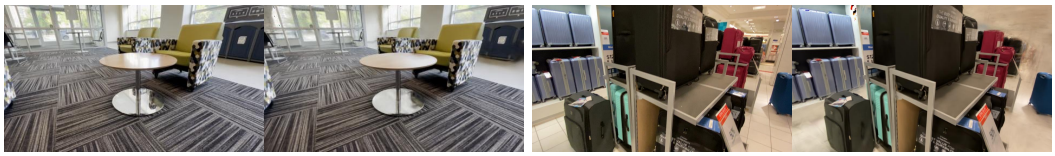
(a) Scannetpp GS. (left) Ground Truth and (right) Gaussian Splats Render Result.

(b) Hypersim GS. (left) Ground Truth and (right) Gaussian Splats Render Result.

(c) Replica GS. (left) Ground Truth and (right) Gaussian Splats Render Result.



(d) 3RScan GS. (left) Ground Truth and (right) Gaussian Splats Render Result.



(e) DL3DV-10K GS. (left) Ground Truth and (right) Gaussian Splats Render Result.

Figure H: **Renderings From Indoor Gaussian Splatting Scenes.** The results showcase the photorealistic appearance reconstructions.