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

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 Text Encoder	SAM+CLIP CLIP	SAM+CLIP CLIP	LSeg CLIP	DINOv2 CLIP	SAM+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.

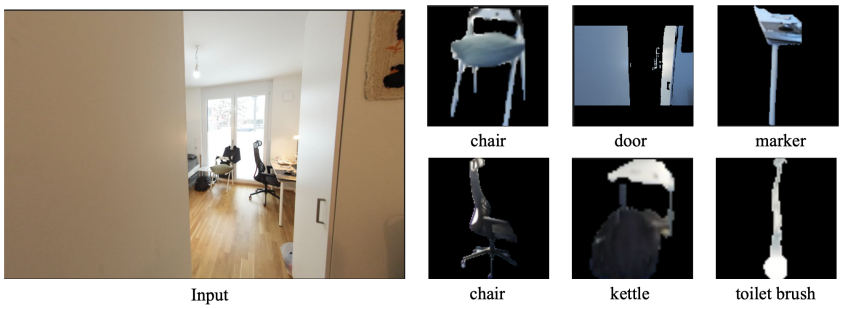


Figure A: **Illustration of Tile-Based Feature Extraction.** (a) The input frame. (b) The SAM-generated masks are cropped and resized to 224×224. The accompanying text shows the zero-shot prediction results for each cropped image. However, some predictions are incorrect due to imprecise object instance 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.

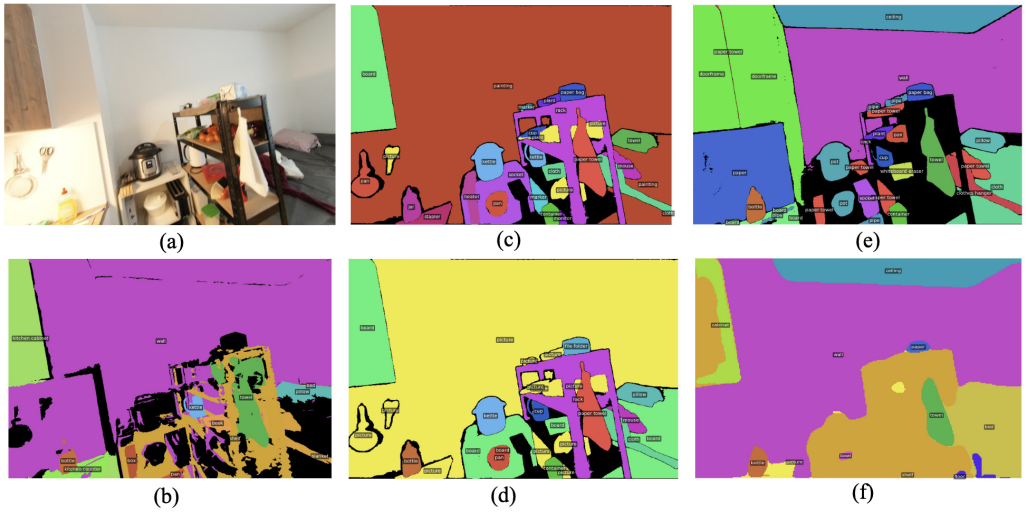


Figure B: **2D Segmentation Using Different 2D Extracted Features.** (a) Input frame, (b) Ground truth segmentation, (c) CLIP zero-shot segmentation, (d) Autoencoder-restored zero-shot segmentation, (e) SigLIP2 zero-shot segmentation, (f) LSeg zero-shot segmentation. The features above demonstrate the 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∼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

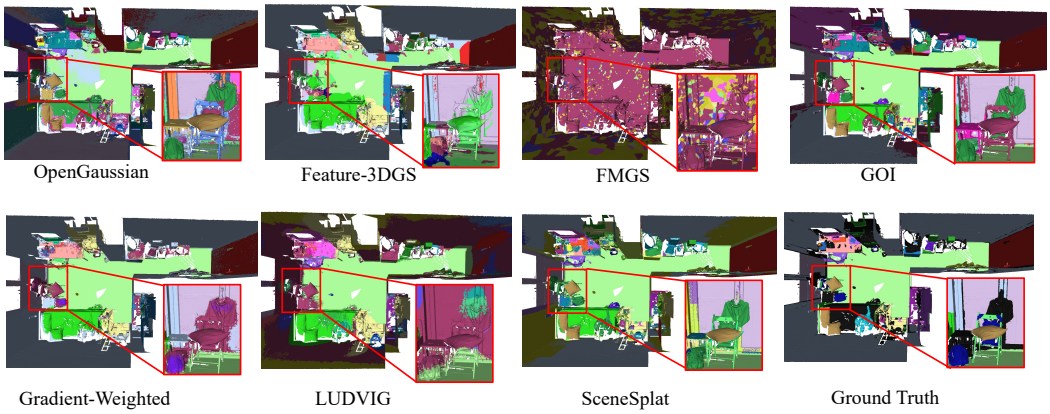


Figure C: **Qualitative Results of Zero-Shot 3D Semantic Segmentation on ScanNet++.** The semantic classes "chair" and "pants" are highlighted, which are not fully labeled in Ground Truth. (Zoom in to check details.)

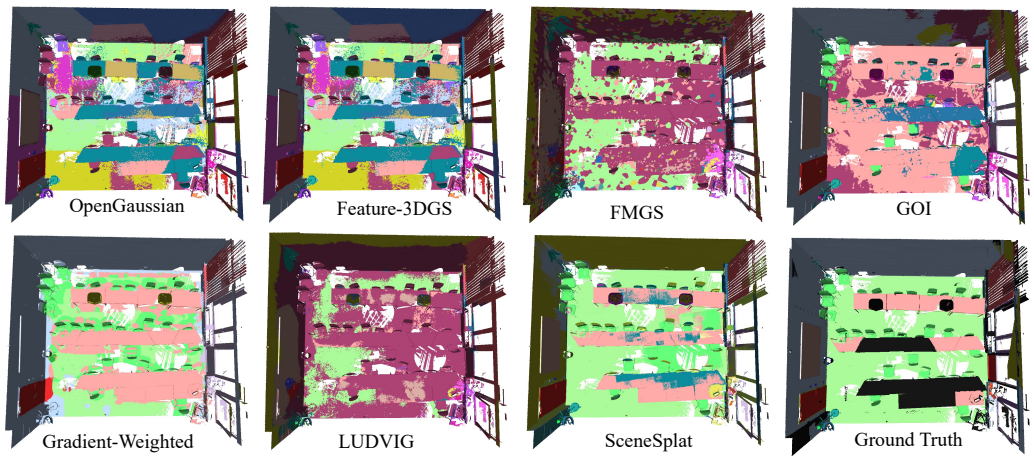


Figure D: **Qualitative Results of Zero-Shot 3D Semantic Segmentation on ScanNet++.** (Zoom in to check details.)

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

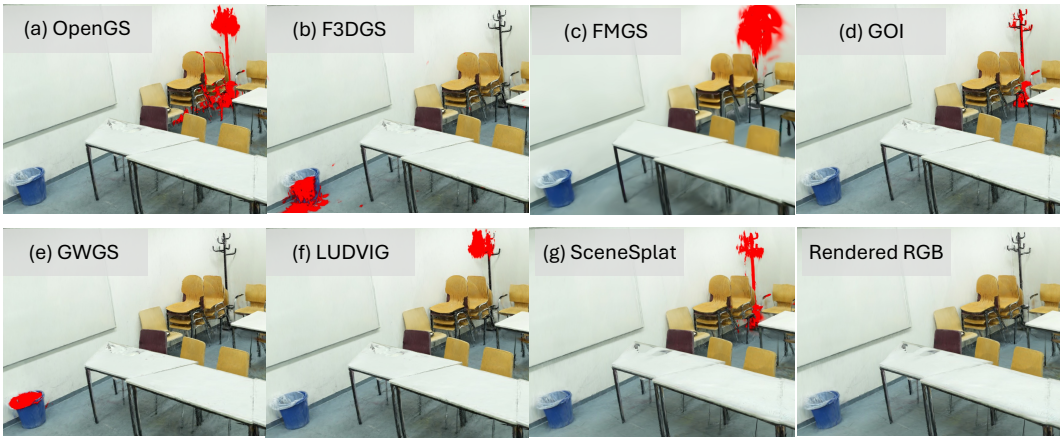


Figure E: **Text-Based Scene Query.** Given the prompt "This is a hanger." to different LGS methods, the queried parts are highlighted in red.

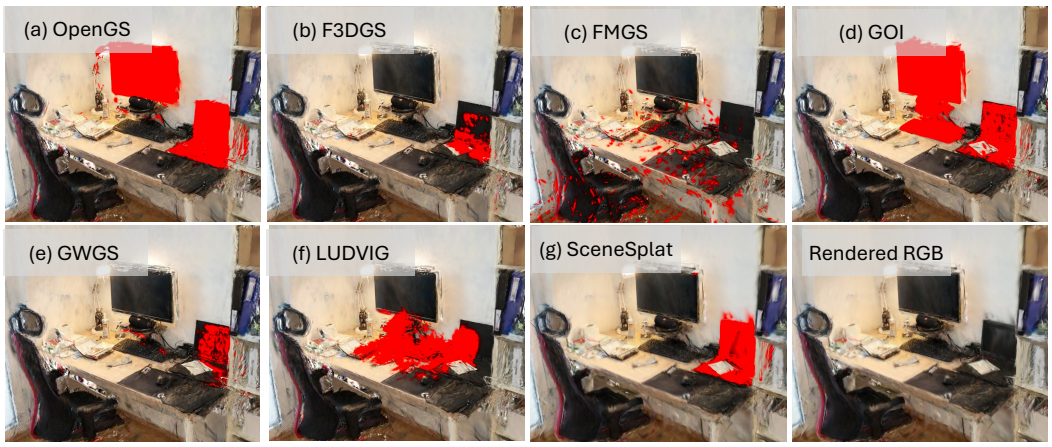


Figure F: **Text-Based Scene Query.** Given the prompt "This is a laptop." to different LGS methods, the queried parts are highlighted in red.

(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

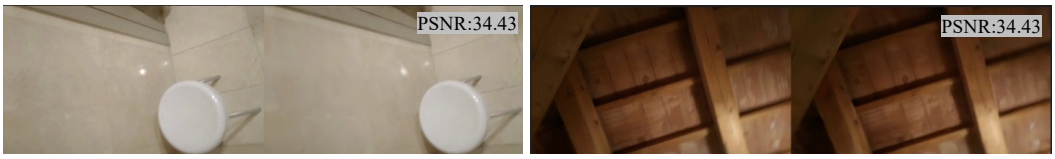
While our work on introducing a large-scale Gaussian dataset for 3D tasks represents a significant advancement in 3D representation learning, it is important to consider potential societal impacts. First, since our dataset is built upon existing public datasets, it shares similar privacy risks, including 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.

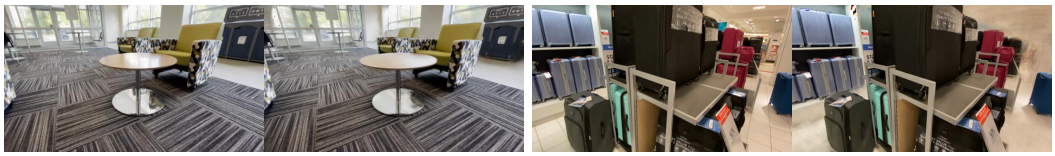
(a) Scannetpp GS. (left) Ground Truth and (right) Gaussian Splats Render Result.

(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.