

SceneSplat: Gaussian Splatting-based Scene Understanding with Vision-Language Pretraining

*Yue Li¹, *Qi Ma^{2,3}, Runyi Yang³, Huapeng Li², Mengjiao Ma^{3,4}, †Bin Ren^{3,5,6}, Nikola Popovic³, Nicu Sebe⁶, Ender Konukoglu², Theo Gevers¹, Luc Van Gool^{2,3}, Martin R. Oswald¹, Danda Pani Paudel³

¹University of Amsterdam ²Computer Vision Lab, ETH Zurich ³INSAIT, Sofia University "St. Kliment Ohridski"

⁴Nanjing University of Aeronautics and Astronautics ⁵University of Pisa ⁶University of Trento

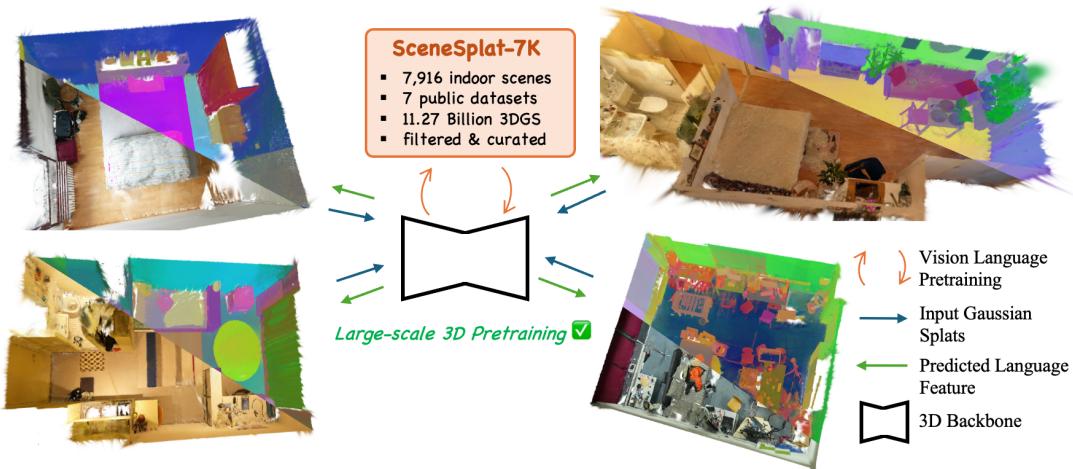


Figure 1. We present the 3DGS indoor dataset **SceneSplat-7K** which includes 7K scenes generated from ARKitScenes [1], Replica [45], ScanNet [5], ScanNet++ [57], Hypersim[42], 3RScan [48], and Matterport3D [2]. Leveraging this high-quality dataset, we propose **SceneSplat**, the first model to predict open-vocabulary language features for millions of 3D Gaussians in a single forward pass.

Abstract

Recognizing arbitrary or previously unseen categories is essential for comprehensive real-world 3D scene understanding. Currently, all existing methods rely on 2D or textual modalities during training or together at inference. This highlights the clear absence of a model capable of processing 3D data alone for learning semantics end-to-end, along with the necessary data to train such a model. Meanwhile, 3D Gaussian Splatting (3DGS) has emerged as the de facto standard for 3D scene representation across various vision tasks. However, effectively integrating semantic reasoning into 3DGS in a generalizable manner remains an open challenge. To address these limitations, we introduce **SceneSplat** in Fig. 1, to our knowledge the first large-scale 3D indoor scene understanding approach that operates na-

tively on 3DGS. Furthermore, we propose a self-supervised learning scheme that unlocks rich 3D feature learning from unlabeled scenes. To power the proposed methods, we introduce **SceneSplat-7K**, the first large-scale 3DGS dataset for indoor scenes, comprising 7916 scenes derived from seven established datasets, such as ScanNet and Matterport3D. Generating **SceneSplat-7K** required computational resources equivalent to 150 GPU days on an L4 GPU, enabling standardized benchmarking for 3DGS-based reasoning for indoor scenes. Our exhaustive experiments on **SceneSplat-7K** demonstrate the significant benefit of the proposed method over the established baselines. Our code, model, and datasets will be released at [SceneSplat](#).

1. Introduction

The ability to interpret arbitrary queries rather than being limited to a closed set of categories is crucial for 3D understanding models to generalize across diverse real-world scenarios. Traditional 3D vision systems are typically trained

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

on fixed closed-set category labels drawn from datasets like ScanNet [5]. Such label spaces fail to capture the diversity of concepts in real-world environments. This gap has spurred significant interest in open-vocabulary 3D scene understanding, which aims to recognize arbitrary or unseen categories beyond a predefined taxonomy [8, 15, 54]. Achieving this capability would empower the model to reason about novel objects in a scene using natural language descriptors.

Achieving open-vocabulary recognition in 3D is challenging due to the scarcity of large-scale 3D-text paired data. Breakthroughs in 2D vision, driven by internet-scale image-text pre-training, cannot be directly leveraged in 3D due to the absence of analogous 3D datasets with rich textual annotations. To address this gap, current methods resort to multi-modality fusion, distilling knowledge from 2D vision-language models into 3D data. Some approaches project 3D points into source images and use a pretrained 2D backbone (*e.g.*, CLIP) to supervise 3D feature learning [33, 64, 70]; others generate synthetic captions for 3D scenes to explicitly associate point clouds with semantic-rich text [7, 43]. However, all current methods rely on 2D or textual modalities during training, or together at inference, to compensate for limited 3D semantics. This highlights a key limitation: *the absence of a robust model for processing 3D data end-to-end for semantic learning, along with the lack of sufficient data for training such a model.*

The field of 3D representation is rapidly evolving. While classical 3D networks rely on point clouds or voxel grids [50, 51, 55], recent approaches inspired by radiance fields have greatly improved scene representation. Notably, 3D Gaussian Splatting (3DGS) [16] has emerged as an efficient representation, achieving state-of-the-art performance in view synthesis and geometry modeling [58, 61]. Unlike discrete point clouds, 3DGS provides a compact formulation that can be optimized per scene for 2D synthesis, while the underlying Gaussians also softly encode rich 3D structures (positions, shapes, and opacities of the regions). This makes it a unique candidate for 3D scene cues, as it naturally fuses geometry and appearance information. However, integrating semantic reasoning into 3DGS is nontrivial. The naive way of optimizing additional semantic features in 3DGS [17] is inefficient and limited to a single scene. Consequently, generalizable open-vocabulary understanding in 3DGS remains unexplored, as no existing model directly processes 3D Gaussian parameters. Moreover, no large-scale scene-level 3DGS dataset is available to train such models, further limiting the progress in this direction.

We address these limitations with SceneSplat, to our knowledge the first large-scale 3D indoor scene understanding approach that operates natively on 3D Gaussian splats. The proposed model is powered by SceneSplat-7K, a carefully curated 3DGS indoor scene dataset spanning around

7K scenes. SceneSplat introduces a 3DGS encoder that takes as input the parameters of a Gaussian-splat scene (center, scale, color, opacity) and outputs semantic features in a per-primitive manner, in a single forward pass. We leverage supervision from vision-language models to train SceneSplat’s encoder to produce CLIP-aligned embeddings, effectively bridging language and 3D without explicit 2D fusion at runtime. Furthermore, we propose GaussSSL, a self-supervised learning scheme that unlocks rich 3D feature learning from unlabeled scenes. GaussSSL operates through three synergistic strategies: Masked Gaussian Modeling (MGM), self-distillation, and optional Language-Gaussian Feature Alignment, allowing the model to separate high-level semantic signals from raw parameters alone. This self-supervised pretraining is fueled by large amounts of 3DGS scenes in the SceneSplat-7K dataset. Our contributions can be summarized as follows:

- We present SceneSplat-7K, a high-quality large-scale Gaussian splats dataset spanning 7K indoor scenes, which boosts 3DGS scene understanding research.
- We propose SceneSplat, a model that unlocks open-vocabulary recognition for 3D Gaussian splats and achieves state-of-the-art zero-shot semantic segmentation performance on three fine-grained benchmarks.
- We incorporate annotation-free self-supervised training mechanisms on this large-scale indoor dataset and demonstrate their effectiveness in the downstream indoor segmentation task.

2. Related Work

3D Indoor Datasets. The advancement of 3D deep learning has driven the development of various indoor datasets for scene understanding, reconstruction, and representation learning [1, 2, 5, 42, 45, 48, 57]. ScanNet [5] serves as a fundamental benchmark with 1,513 real-world indoor scenes and dense annotations, while ScanNet++ [57] extends it with additional real and synthetic scenes. HyperSim [42] offers photorealistic synthetic environments, and ARKitScenes [1] captures 1,661 real-world indoor scenes using ARKit. Replica [45] provides high-fidelity reconstructions, and 3RScan [48] supports multi-view analysis with 1,482 scans. Habitat-Matterport3D [2] integrates Matterport3D into an embodied AI simulation platform. These datasets are essential for 3D perception but lack large-scale support for emerging 3D representations like 3DGS. Prior works [13, 29] have explored NeRF-based representations, and ShapeSplat [28] introduced Gaussian-splatted objects, but no dataset currently supports indoor scene understanding with Gaussian Splatting. To address this limitation, we introduce a 3D Gaussian Splatting scene dataset derived from 7 widely used indoor datasets [1, 2, 5, 42, 45, 48, 57]. Our dataset offers a rich collection of indoor scenes from both real-world and synthetic sources, featur-

Metric	ScanNet[5]	ScanNet++[57]	ScanNet++v2[57]	Replica[45]	Hypersim[42]	3RScan[48]	ARKitScenes[1]	Matterport3D[2]	Scenesplat-7K
Raw Scenes	1613	380	1006	8	461	1482(scans)	1970	2194(regions)	9114
GS Scenes	1613	330	956	8	448	632(scans)	1947	1982(regions)	7916
RGB Frames	2.5M	228K	1.1M	16K	77K	156K	450K	194K	4.72M
Storage	600GB	152GB	447GB	7GB	251GB	235GB	577GB	492GB	2.76TB
PSNR	29.07	29.49	29.11	41.25	25.93	27.46	29.18	32.34	29.64
Depth Loss	0.031	0.019	0.015	0.002	0.228	0.018	0.0131	0.033	0.035
SSIM	0.869	0.924	0.933	0.980	0.894	0.881	0.885	0.916	0.897
LPIPS	0.236	0.133	0.116	0.0396	0.157	0.335	0.294	0.145	0.212
GS per scene	1.50M	1.56M	1.89M	1.50M	2.84M	1.50M	1.19M	1.0M	1.42M
Total GS	2419.5M	513.4M	1810.3M	12.0M	1,237.5M	948.0M	2,316.9M	1,982M	11.27B
GPU Time (L4)	593h	177h	594h	4h	176h	576h	811h	661h	3592 h

Table 1. Dataset Statistics. The proposed SceneSplat-7K dataset includes various 3D Gaussian Splatting datasets generated from ScanNet [5], ScanNet++ [57], ScanNet++ v2, Replica[45], Hypersim[42], 3RScan[48], ARKitScenes[1], and Matterport3D[2]. The dataset contains **7,916 scenes** and **11.27 Billion 3DGS**. Constructing this dataset required computational resources equivalent to **150 GPU-days** on one NVIDIA L4 GPU. SceneSplat-7K achieves high-fidelity reconstruction quality with an average PSNR **29.64 dB**.

ing high-quality Gaussian splats, facilitating the transition from object-level to scene-level Gaussian splatting and advancing large-scale 3D scene understanding.

Open Vocabulary Scene Understanding. Recent advancements in open-vocabulary models have significantly expanded the capabilities of vision-language understanding. Foundation models such as DINO [31, 62] have enabled self-supervised visual feature extraction at scale, effectively supporting tasks like detection and segmentation. SAM [20, 39] introduced the capability of prompt-driven segmentation, generalizing robustly across diverse datasets and tasks. CLIP [37] pioneered aligning visual and textual embedding spaces, allowing for zero-shot transfer across numerous downstream tasks. Subsequently, SigLIP [46, 60] improved alignment by introducing non-linear activation techniques, enhancing open-vocabulary performance marginally. Recent works [4, 10, 33, 36, 67, 69] have also introduced 2D open-vocabulary foundation models into 3D by utilizing NeRF [30] or 3DGS [16]. LERF [17, 38] integrated language queries into NeRF-based models, enabling rich semantic querying within 3D scenes. LangSplat [36] combined 3DGS with open-vocabulary embeddings from SAM [39] and CLIP [37], facilitating open-vocabulary semantic scene understanding. OccamLGS [4] further optimized this process by employing feature lifting, directly projecting all 2D features into 3DGS, and reducing computation time from hours to seconds. However, these methods require time-consuming preprocessing of images using 2D foundation models. In contrast, we propose a pipeline and dataset for training a 3D foundation model, enabling feed-forward open-vocabulary understanding of 3DGS.

3D Representation Learning. Representation learning in 3D, akin to its success in the 2D image domain, is crucial for extracting meaningful features from 3D data. Previous approaches have focused on either architectural advancements or learning paradigms. Many methods have sought to map 3D data into 2D patches, leveraging conventional 2D CNNs or Vision Transformers [9, 40, 47]. Although

promising, this approach often overlooks the inherent properties of 3D data due to the embedding strategy. To address this, research has shifted toward specialized architectures (*i.e.*, PointCNN [22], PointNet [34], Point Transformer [50, 51, 65], and Pointmamba [24]). However, labeling 3D data requires extensive effort and domain expertise, making it quite challenging. This has highlighted the importance of self-supervised learning (SSL) for 3D data, with methods typically falling into contrastive or generative categories. Contrastive methods aim to differentiate similar and dissimilar examples, but often suffer from issues such as overfitting and mode collapse, exacerbated by the sparse nature of 3D data [35, 41]. On the other hand, generative methods, inspired by BERT’s *mask-and-reconstruct* strategy [6], have proven more effective. Masked Autoencoders [11], originally designed for 2D images, have been adapted for 3D data types, including point clouds [32, 63], meshes [25], voxels [12], and Gaussians [28], allowing the reconstruction of masked regions and facilitating the learning of robust spatial and semantic features. In this work, we propose a novel approach for vision-language 3DGS pre-training, as well as a self-supervised learning scheme that operates via Gaussian masking and self-distillation.

3. SceneSplat Dataset

We introduce SceneSplat-7K – a carefully curated dataset of 3D Gaussian Splats representing indoor scenes. The main goal was to obtain a dataset to facilitate generalizable 3DGS indoor scene understanding. The dataset contains about seven thousand scenes, including both real-world and synthetic environments. The comprehensive statistics of the introduced dataset are presented in Tab. 1. The SceneSplat-7K dataset will be publicly released, with additional details (licenses) available in the supplement.

3.1. Data Processing

Multiple measures are applied before, during, and after the 3DGS optimization to guarantee high-quality 3DGS scenes.

Starting with the training views, we select scenes with at least 400 frames to ensure sufficient multi-view coverage. We remove blurry frames by using the variance of the Laplacian as a sharpness metric. We use gsplat [56] for 3DGS optimization. For scenes with available depth input, we apply depth loss to achieve better geometry modeling. To efficiently compress the 3DGS scene, we employ a Markov Chain Monte Carlo strategy [18] and add opacity and scale regularization. Once optimized, we filtered 3DGS scenes based on the PSNR metric before using them as inputs for our pretraining. We refer to the supplement for per-dataset processing details.

3.2. Data Statistic

SceneSplat-7K dataset includes various 3D Gaussian Splatting datasets generated from ScanNet [5], ScanNet++ [57], ScanNet++ v2, Replica [45], Hypersim [42], 3RScan [48], ARKitScenes [1], and Matterport3D [2], comprising approximately 9,000 raw scenes. SceneSplat-7K contains **7,916** processed Gaussian splatting scenes, with an average of 1.42 Million 3D Gaussians per scene and **11.27 Billion Gaussians** in total, from which 4,114 high-quality Gaussian splatting scenes are selected for pretraining. Spanning a total of 4.72 Million RGB frames, SceneSplat-7K achieves high-fidelity appearance and competitive reconstruction quality, having an average PSNR of 29.64 dB, average depth loss of 0.035 m, average SSIM of 0.897, and average LPIPS of 0.212. Constructing this dataset required an equivalent of **150 days** of computation on an NVIDIA L4 GPU.

4. Methodology

Building upon the SceneSplat-7K dataset, we carry out both vision-language 3DGS pretraining, which enables open-vocabulary scene understanding, and self-supervised pretraining, which regularizes the latent space during 3DGS parameter encoding, as shown in Fig. 2. For vision-language pretraining, we first need to collect primitive-level language labels for 3DGS scenes (Sec. 4.1). The SceneSplat model then learns to robustly predict vision-language features from the Gaussian parameters and their surroundings (Sec. 4.2). Furthermore, for self-supervised pretraining, we employ a multi-objective self-supervised training framework that integrates reconstruction and self-distillation alignment (Sec. 4.3).

4.1. 3DGS Language Label Collection

Our language label collection aims to establish 3D-language paired data by associating each 3D Gaussian primitive G_i with a rich semantic feature $F_i \in \mathbb{R}^d$.

Unlike methods that align 3D primitives with text embeddings [8, 21] or use visual captioning [26, 59], we directly align Gaussians with the image embedding space

Algorithm 1 3DGS Language Label Collection

```

1: Input: Training views  $\{I_j\}_{j=1}^M$ , 3D Gaussian scene
    $\{G_i\}_{i=1}^N$ , SAMv2, SigLip2
2: Output: 3D Gaussian-feature pairs  $\{(G_i, F_i)\}_{i=1}^N$ 
3: Step 1: 2D Feature Map Generation
4: for each training view  $I_j$  do
5:    $M_{\text{seg}} \leftarrow \text{SAMv2}(I_j)$   $\triangleright$  Get object-level seg. masks
6:    $f_g \leftarrow \text{SigLip2}(I_j)$   $\triangleright$  Feature from full frame
7:   Initialize feature map  $F_j$  for view  $I_j$ 
8:   for each segment  $s$  in  $M_{\text{seg}}$  do
9:      $f_l, f_m \leftarrow \text{SigLip2}(\text{crop}(I_j, s))$ 
10:     $\triangleright$  Local features for crops w/ and w/o background
11:    Dynamic Weighting:
12:     $w_g, w_l, w_m \leftarrow \text{compute\_weights}(f_g, f_l, f_m)$ 
13:     $\triangleright$  Weights based on context
14:     $f_s \leftarrow w_g \cdot f_g + w_l \cdot f_l + w_m \cdot f_m$ 
15:     $\triangleright$  Fuse features
16:    Update feature map  $F_j$  with  $f_s$  at segment  $s$ 
17:  end for
18: end for
19: Step 2: Lifting 2D Features to 3D Gaussian Feature Field
20:  $\{F_i\}_{i=1}^N \leftarrow \text{Occam's\_LGS}(\{F_j\}_{j=1}^M, \{G_i\}_{i=1}^N)$   $\triangleright$  Lift
   2D features to 3D
21:  $\{F_i\}_{i=1}^N \leftarrow \text{normalize}(\{F_i\}_{i=1}^N)$   $\triangleright$  Normalization
22: Return: 3D Gaussian-feature pairs  $\{(G_i, F_i)\}_{i=1}^N$ 

```

of vision-language models (VLM), preserving richer latent semantic information. Our approach also avoids scene-specific compression [36, 44] which limits scalability and generalization.

As outlined in Alg. 1, we employ SAMv2 [39] for object-level segmentation and SigLIP2 [46] for feature extraction. We then use Occam’s LGS [4] to efficiently lift these 2D feature maps to a 3D Gaussian feature field in an optimization-free manner. This results in a comprehensive collection of 3DGS-feature pairs $\{(G_i, F_i)\}_{i=1}^N$ across multiple datasets, providing a solid foundation for our vision-language pretraining.

4.2. Vision-Language 3DGS Pretraining

We first adapt the transformer encoder-decoder backbone from [51] to efficiently predict high-dimensional per-primitive features corresponding to collected 3DGS language labels. More specifically, our model $g(\cdot)$, parameterized by θ , maps the input Gaussians to their language features:

$$\hat{F} = g_\theta(\{G_i\}_{i=1}^N) , \quad (1)$$

where $\hat{F} \in \mathbb{R}^{N \times d}$ is the predicted per-gaussian feature.

We apply three training objectives for supervision. The cosine similarity loss minimizes the angular difference be-

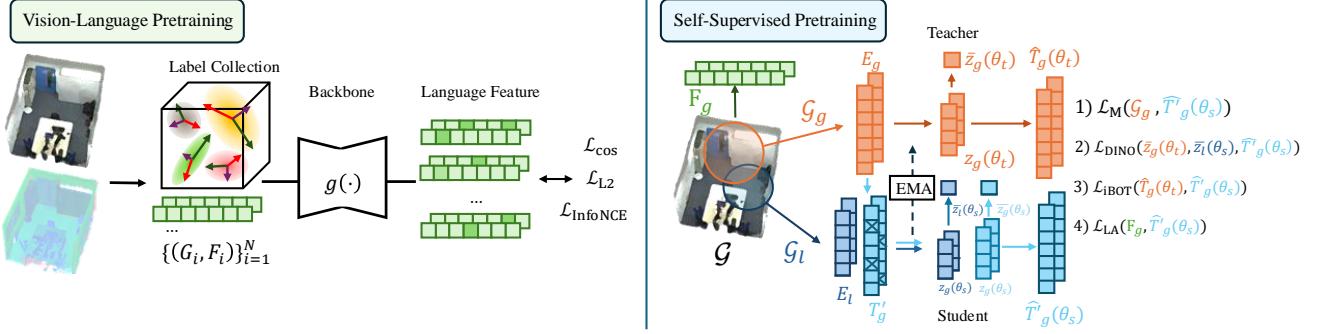


Figure 2. SceneSplat Overview. The SceneSplat-7K dataset enables **Vision-Language Pretraining** and **Self-Supervised Pretraining**. For vision-language pretraining, we associate each 3D Gaussian primitive with semantic features based on our label collection process and train a generalizable open-vocabulary learner that predict per-gaussian embeddings. For self-supervised pretraining, we employ Masked Gaussian Modeling to reconstruct masked primitives, Self-Distillation Learning for augmentation-invariant features, and Language-Gaussian Alignment for scenes with collected labels. The former achieves state-of-the-art zero-shot segmentation results on ScanNet200 [5], ScanNet++ [57], and Matterport3D [2] benchmarks and the latter unlocks training on large-scale 3DGS data.

tween the predicted and ground truth language labels:

$$\mathcal{L}_{\cos} = \frac{1}{|\mathcal{V}|} \sum_{i \in \mathcal{V}} \left(1 - \frac{\hat{F}_i \cdot F_i}{\|\hat{F}_i\| \cdot \|F_i\|} \right), \quad (2)$$

where \mathcal{V} is the set of Gaussians with language feature labels. To enforce feature similarity in Euclidean space, we use L2 loss:

$$\mathcal{L}_2 = \frac{1}{|\mathcal{V}|} \sum_{i \in \mathcal{V}} \|\hat{F}_i - F_i\|^2. \quad (3)$$

Lastly, we use an aggregated contrastive loss to encourage the separation of class-level features. Instead of contrasting every Gaussian feature individually (which would be computationally prohibitive in large scenes), we apply class-wise *mean pooling*. For each semantic class c with sufficiently many Gaussians, we randomly split its Gaussians into two disjoint sets \mathcal{G}_c^A and \mathcal{G}_c^B , and compute the pooled features:

$$\bar{F}_c^A = \text{Pool}(\hat{F}, \mathcal{G}_c^A), \quad \bar{F}_c^B = \text{Pool}(\hat{F}, \mathcal{G}_c^B). \quad (4)$$

We then apply a bidirectional contrastive loss. First, we normalize \bar{F}_c^A and \bar{F}_c^B to unit length. Let $Z^A = \bar{F}_c^A (\bar{F}_c^B)^\top / \tau$ and $Z^B = \bar{F}_c^B (\bar{F}_c^A)^\top / \tau$, where each row in \bar{F}^A or \bar{F}^B corresponds to a different semantic class. The diagonal elements of Z^A and Z^B are positive matches. Hence, we compute a cross-entropy loss in both directions:

$$\mathcal{L}_{\text{contrast}} = \frac{1}{2|C|} \sum_{X \in \{A, B\}} \sum_{i \in C} -\log \frac{\exp(Z_{i,i}^X)}{\sum_{j \in C} \exp(Z_{i,j}^X)}, \quad (5)$$

where C is the set of semantic classes with sufficient Gaussians, τ is a learnable temperature controlling the softness of the distribution.

The total loss is the weighted sum $\mathcal{L}_{\text{total}} = \lambda_{\cos} \mathcal{L}_{\cos} + \lambda_{L2} \mathcal{L}_{L2} + \lambda_{\text{con}} \mathcal{L}_{\text{contrast}}$, where $\lambda_{(.)}$ denotes each weight. In practice, we found that applying the contrastive loss later during the training (*i.e.*, “warm starting” with \mathcal{L}_{\cos} and \mathcal{L}_2) helps promote early feature learning while effectively refining class distinctions in later stages.

Through this training, our model learns to predict semantically rich language features for each Gaussian primitive, enabling downstream open-vocabulary scene understanding task without requiring additional finetuning or 2D input.

4.3. Self Supervised Pretraining

The proposed GaussianSSL method is presented in Fig. 2 (right). It incorporates multiple losses with different objectives into SceneSplat’s large-scale pretraining.

Masked Gaussian Modeling. This part supervises the model to predict masked Gaussian primitives. Given a 3D Gaussian splatting scene represented by $\mathcal{G} = \{G_i\}_{i=1}^N$, where $G_i \in \mathbb{R}^{59}$, the training proceeds as follows: (1) A subset $\{G_j\}_{j=1}^{N'}$ is sampled from \mathcal{G} using dense grid sampling of size S . (2) Samples are projected into a latent space with the embedding function P to obtain tokens $E = P(\{G_j\}_{j=1}^{N'}) \in \mathbb{R}^{N' \times d_e}$, where d_e is the embedding dimension. (3) The tokens E are masked with ratio $r \in [0, 1]$ by replacing $N' \cdot r$ randomly chosen tokens with a learnable mask token $t \in \mathbb{R}^{d_e}$ to obtain $T_m \in \mathbb{R}^{N' \times d_e}$. (4) The masked tokens T_m are processed using the 3D backbone $g_\theta(\cdot)$ to obtain $\hat{T}_m = h_\phi(f_\varphi(T_m))$, where $f_\varphi(\cdot)$ is the encoder and $h_\phi(\cdot)$ is the decoder. (5) The output tokens \hat{T}_m are mapped to the input Gaussian space with the reconstruction projector $\hat{G}_m = \Phi(\hat{T}_m) \in \mathbb{R}^{N' \times F}$. (6) Finally, the \mathcal{L}_2 reconstruction loss between the predicted masked Gaussians and the original Gaussians is used: $\mathcal{L}_{\text{MGM}} = \mathbb{E}_{G_j \sim \mathcal{G}} [\|G_m - \hat{G}_m\|_2^2]$.

Self-Distillation Representation Learning. Self-distillation [31] learns augmentation-invariant representations by aligning the predictions of a student network θ_s with an EMA-updated teacher θ_t [3]. For a batch of Gaussian scenes $\{\mathcal{G}_n\}_{n=1}^B$ (global/local views G_g^b, G_l^b), we extract tokenized bottleneck features $z \in \mathbb{R}^{M \times d_e}$, compute global representations \bar{z} via mean pooling, and align student-teacher outputs via cosine similarity loss \mathcal{L}_{sim} [52]. Feature diversity is regularized with a coding rate term \mathcal{L}_{cr} [23, 52] resulting in $\mathcal{L}_{\text{DINO}} = \omega_{\text{sim}}\mathcal{L}_{\text{sim}} + \omega_{\text{cr}}\mathcal{L}_{\text{cr}}$ with corresponding weights. Inspired by [52, 68], the student network also predicts masked features aligned with the teacher via $\mathcal{L}_{\text{iBOT}}$, computed using cosine similarity. We propose to mitigate the decoder collapse issues by multi-task reconstruction \mathcal{L}_{MGM} , as coding rate regularization stabilizes only the hierarchical encoder. Following [11, 53], the reconstruction is limited to weakly augmented views to avoid degradation.

Language-Gaussian Alignment. As shown in Sec. 4.2, the precomputed language feature enables effective knowledge distillation. For scenes with existing language labels, we seek to leverage them to further regulate self-supervised learning. However, the high dimensionality of these language features (dimension $N \times d_L$) can substantially increase the computational cost of supervision. To address this, we replace the original features with a compressed representation learned via an autoencoder [19], drastically reducing the memory overhead while preserving semantic information. Similar to \mathcal{L}_{MGM} , we use \mathcal{L}_{LA} following Eqs. (2) and (3) to train the network to predict low-dimensional language features from unmasked neighbors.

5. Experiments

In this section, we evaluate the performance of vision-language pretraining on open-vocabulary task and examine the impact of large-scale Gaussian self-supervised pretraining on downstream indoor semantic segmentation. We further justify our design choices through ablation studies. The implementation details are provided in the supplement.

5.1. Vision-Language Pretraining

Tab. 2 reports the zero-shot 3D semantic segmentation results on the fine-grained ScanNet++ (100 classes) [57], Matterport3D (160 classes) [2] and ScanNet200 (200 classes) [5] benchmarks, where methods are trained on specified data sources. When trained on ScanNet, SceneSplat achieves state-of-the-art results, leading to 5.9% and 2.2% f-mIoU increases on the ScanNet200 and Matterport3D benchmarks. By extending the training sources, we obtain 5.7%, 0.7%, and 10.4% f-mIoU increases on the ScanNet200, Matterport3D, and ScanNet++ benchmarks, respectively, compared to concurrent work [21]. Notably, [21] uses $8.32 \times$ training scenes to achieve its best results.

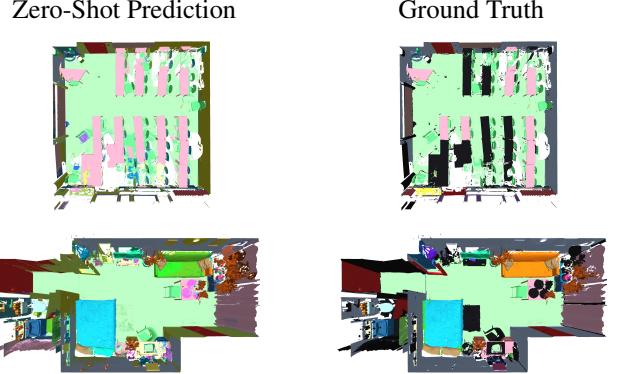


Figure 3. **Qualitative Results of Zero-Shot 3D Semantic Segmentation on ScanNet++.** SceneSplat demonstrates competitive zero-shot performance, note how our model correctly annotate the regions lacking ground truth labels, e.g., `desks` on the top row. Best viewed zoomed in and in color.

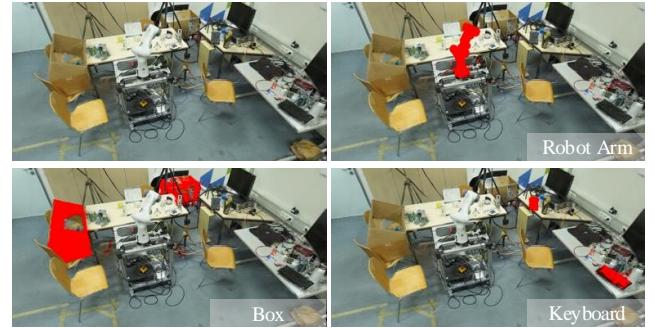


Figure 4. **Text-Based 3DGS Scene Query.** Given text queries and SceneSplat inference results for a 3DGS scene, we can effectively localize the corresponding splats (highlighted in red for queries "Robot Arm", "Box", and "Keyboard").

Fig. 3 shows the zero-shot segmentation results on evaluation scenes. SceneSplat not only achieves competitive segmentation performance but also correctly annotates the missing semantic labels (e.g., desks shown above). We demonstrate text-based queries on the inference results of the predicted language features in Fig. 4. Our vision-language pretraining enables the effective localization of complex objects within the scene.

5.2. Label-free 3DGS Pretraining

We conduct extensive pretraining experiments with the proposed GaussianSSL method. To assess the efficacy of the pretrained model, we report the segmentation results in Tab. 3. Our method achieves a +0.1% improvement over supervised-only baselines on ScanNet20 and +0.5% on ScanNet200, while observing a performance drop on ScanNet++ primarily due to pretraining dataset quality variations (Tab. 1). Furthermore, compared with our reproduced im-

Method	Training Source	#Training Scenes	ScanNet200 (200)	Matterport3D (160)	ScanNet++ (100)	
			f-mIoU	f-mAcc	f-mIoU	f-mAcc
OpenScene [†] [33]	SN	×1	6.4	12.2	5.7	10.7
PLA [8]	SN	—	1.8	3.1	—	—
RegionPLC [54]	SN	—	9.2	16.4	6.2	13.3
OV3D [15]	SN	—	8.7	—	—	—
Mosaic3D [21]	SN	—	13.0	24.5	8.6	17.8
SceneSplat (Ours)	SN	—	18.9	31.7	10.8	18.7
Mosaic3D [21]	SN, SN++, ARKitS, MP3D, S3D	×24.3	15.7	28.3	13.1	27.7
SceneSplat (Ours)	SN++	×0.75	11.8	19.2	10.6	18.6
SceneSplat (Ours)	SN, SN++, MP3D	×2.92	21.4	38.7	13.8	31.8

Table 2. **Zero-Shot 3D Semantic Segmentation on the Fine-Grained ScanNet++ (100 classes) [57], Matterport3D (160 classes) [2] and ScanNet200 (200 classes) [5] Benchmarks.** We report the foreground mean IoU (f-mIoU) and foreground mean accuracy (f-mAcc) excluding background classes (wall, floor, ceiling), following [8, 14, 33, 54]. [†] denotes the official checkpoint and the results of the baselines are taken from [21]. Dataset abbreviations SN, SN++, ARKitS, MP3D, and S3D respectively denote ScanNet [5], ScanNet++ [57], ARKitScenes [1], Matterport3D [2] and Structured3D [66]. SceneSplat achieves noticeably better segmentation performance, *i.e.*, a 5.9% f-mIoU increase on the ScanNet200 benchmark when trained on a single source, and an 11.1% f-mIoU increase on ScanNet++ when trained on two sources, while using significantly less training data compared to the concurrent work [21].

Method	ScanNet20 (20)		ScanNet200 (200)		ScanNet++ (100)	
	mIoU	mAcc	mIoU	mAcc	mIoU	mAcc
No-Pre	77.1	84.1	35.4	44.0	42.4	53.3
MGM	76.7	83.5	35.5	44.5	41.7	51.9
+DINO	77.0	84.6	35.9	46.1	42.0	52.4
+iBOT	77.2	84.2	35.2	44.3	41.1	52.0
+LA	77.2	84.2	34.7	44.4	41.4	52.5

Table 3. **GaussianSSL Ablation Experiments.** We adopt the pre-training on the Scenensplat-7K dataset and report fine-tuning mIOU and mAcc on indoor semantic segmentation tasks. For details of specific losses please refer to Sec. 4.3 and Fig. 2.

Method	ScanNet20		ScanNet200		ScanNet++	
	mIoU	mAcc	mIoU	mAcc	mIoU	mAcc
PTv1 [65]	70.6	—	27.8	—	—	—
PTv2 [50]	75.4	—	30.2	—	—	—
PTv3 [51]	76.4	83.5	35.0	44.2	42.6	53.0
SceneSplat (Ours)	77.2	84.6	35.9	46.1	42.4	53.5

Table 4. **Supervised Semantic Segmentation Experiments.** We report our best results from Tab. 3 comparing against the state-of-the-art Point Transformer method.

plementation of PTv3 [51], we outperform by +0.8% on ScanNet20 and +0.9% on ScanNet200 (Tab. 4). More qualitative results are provided in the supplement.

5.3. Further Statistical Evaluation

SceneSplat Inference Results vs. Collected Language Labels. One may assume that the collected language labels cap the upper bound zero-shot performance since they provide the supervision signal during vision-language pretraining. Interestingly, this is not always the case. Tab. 5 compares the performance of SceneSplat inference features with

Method	ScanNet200 (200)		ScanNet++ (100)	
	f-mIoU	f-mAcc	f-mIoU	f-mAcc
Language Labels	22.8	35.9	22.6	46.5
SceneSplat	18.9	31.7	26.8	45.3

Table 5. **SceneSplat Inference Results vs. Collected Language Labels on Zero-Shot 3D Semantic Segmentation.** The result on ScanNet++ shows the inference performance can be even better than using the collected labels. SceneSplat here is trained on the single dataset respectively.

the collected language labels. On ScanNet++, our method outperforms the labels with a 4.2% increase in f-mIoU. Although the collected labels are not perfect, large-scale pre-training can filter noise and learn meaningful patterns.

Impact of the Input 3DGS Scene PSNR on Open-Vocabulary Performance. Reported on the Matterport3D test split with 370 scenes, Fig. 5 indicates a clear positive trend of the input 3DGS scene PSNR of the training views and the resulting zero-shot mIoU performance with SceneSplat model. Low PSNRs usually come out of blurry input images, poor Gaussian centers optimization, and insufficient scene coverage, where the 3DGS parameters cannot resolve the scene well. This trend highlights the importance of data curation for the collected 3DGS scenes.

Nearest Neighbors Voting During Zero-Shot Experiments. The centers of the input Gaussians differ from the point locations where the semantic predictions are evaluated; thus, we have to aggregate predictions from neighboring Gaussians. We perform majority voting using the nearest neighboring Gaussians for each evaluation location. Fig. 6 ablates the number of nearest neighbors on the IoU results using the ScanNet++ validation split. We observe the overall trend of the mIoU increase *w.r.t.* to the number

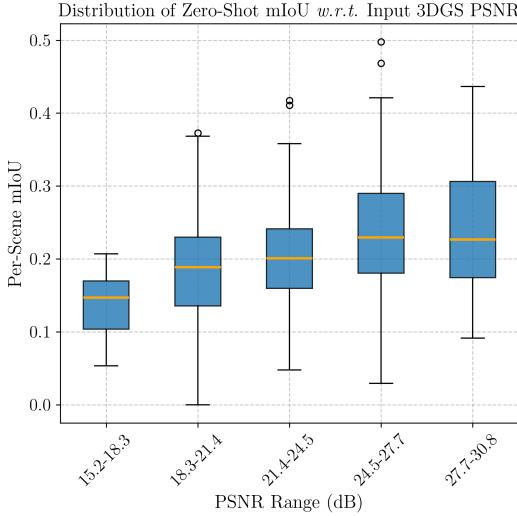


Figure 5. Distribution of SceneSplat Zero-Shot Semantic Segmentation mIoU w.r.t. Input 3DGS Scene PSNR. Reported on the Matterport3D test split labeled in 21 semantic classes, the box plot shows a clear positive trend between the input 3DGS scene training PSNR and the resulted mIoU once applied SceneSplat language pretraining for zero-shot semantic segmentation. This encourages the careful curation of the collected 3DGS scene dataset.

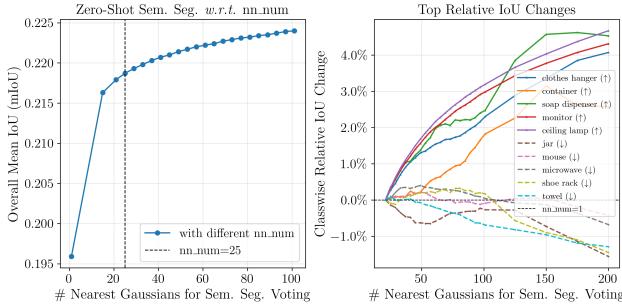


Figure 6. Overall and Class-Wise IoU Changes w.r.t. to the Nearest Neighbor Number During Majority Voting. We evaluate SceneSplat using different nearest 3DGS neighbors for zero-shot task at the point locations on ScanNet++ validation split. Overall mIoU increases with different class-wise relative IoU changes.

SceneSplat Training Input	ScanNet200 (200)		ScanNet++ (100)	
	mIoU	mAcc	mIoU	mAcc
Point Parameters	17.1	27.9	23.8	40.2
3DGS Parameters	18.9	31.7	26.8	45.3

Table 6. Zero-Shot Performance of Using Point Clouds vs. 3DGS for SceneSplat Vision-Language Pretraining. SceneSplat trained on 3DGS parameters consistently outperforms the variant trained on point cloud properties. The models here are trained on the single dataset respectively.

Contrastive Loss	ScanNet200 (200)		ScanNet++ (100)	
	f-mIoU	f-mAcc	f-mIoU	f-mAcc
w/o	13.7	22.5	19.6	34.4
always apply	13.2	23.4	23.2	39.3
last 75% epochs	15.7	24.0	23.8	40.2

Table 7. Ablation on Contrastive Loss During Vision-Language Pretraining Using Subsets. Having a warm-up period and applying the contrastive loss later leads to better performance.

Method	Steps Required	Runtime / Scene
Occam’s LGS	2D fusion + lifting	107 min
SceneSplat	single inference	0.24 min

Table 8. Runtime. SceneSplat is significantly faster compared to the current fastest language-embedded 3DGS method.

of nearest neighbors and list the classes with the top relative changes. To balance the performance and inference speed, 25 nearest neighbors are selected for voting.

Effectiveness of Using 3DGS in Vision-Language Pre-training Compared to Point Clouds. To further justify the effectiveness of using 3DGS parameters for scene understanding, we apply the same vision-language pretraining on point properties (color and normal). Tab. 6 indicates that the model takes point properties as input consistently get outperformed by SceneSplat using 3DGS parameters.

Ablation on Contrastive Loss in the Vision-Language Pre-training. Tab. 7 ablates the contrastive loss applied during vision-language pretraining, where applying contrastive loss at the late stage of training outperforms other variants.

Runtime vs. Per-Scene Language Gaussian Splatting Method. Thanks to the feed-forward ability after vision-language pretraining, SceneSplat is shown in Tab. 8 to be 445.8× faster than the state-of-the-art language-embedded Gaussian Splatting method, as there is no need for 2D feature extraction and fusion.

6. Conclusion

In this work, we introduce SceneSplat, the first large-scale 3D scene understanding model for indoor environments operating directly on 3D Gaussian splats. Powered by SceneSplat-7K, a dataset comprising 7,916 scenes, we propose a novel 3D Gaussian splat encoder that generates semantic features in a single pass, enabling open-vocabulary scene recognition without relying on 2D fusion. Through self-supervised techniques, we unlock label-free 3DGS pre-training at the scene level. Our approach achieves state-of-the-art performance in zero-shot semantic segmentation, establishing new benchmarks and laying the foundation for future advancements in open-vocabulary 3D understanding.

7. Acknowledgment

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), the MUR PNRR project FAIR (PE00000013), the EU Horizon project ELIAS (No. 101120237), and the computational resources provided by the Google Cloud Platform (GCP).

SceneSplat: Gaussian Splatting-based Scene Understanding with Vision-Language Pretraining

Supplementary Material

Contents

A Implementation Details	1
A.1. Language Label Collection	1
A.2 Vision-Language Pretraining	1
A.3 Masked Gaussian Modeling	2
A.4 Self-Distillation Representation Learning	2
A.5 Autoencoder for Feature Compression	2
A.6 Gaussian Language Alignment.	3
A.7 Model Architecture	3
B Further Experimental Results	3
B.1. SceneSplat Zero-shot Segmentation	3
B.2. 3DGS Self-Supervised Pretraining	4
C Datasets Curation and Statistics	8
C.1. Analysis	8
C.2. Visualization	9
D Dataset License	9
E Limitations	9
F. Impact Statement	9

A. Implementation Details

A.1. Language Label Collection

Many existing methods align 3D primitives with text embeddings from vision-language models (VLM) [8, 21]. While effective for basic understanding, this approach inherently limits the information captured, as textual descriptions typically only convey categorical and spatial properties, lacking fine-grained visual details and relationships. Methods employing visual captioning models [26, 59] face similar challenges as they struggle to describe all aspects of the target scene content. Even detailed captions inevitably result in information loss during the text embedding alignment process. In contrast, we directly align Gaussians with the image embedding space of VLM, thereby preserving richer latent semantic information.

Dynamic Weighting Mechanism. Unlike previous approaches that use a single tight crop around each segment, we adapt the three-crop strategy [49] with dynamic weighting to capture the context. For each segment identified by SAMv2, we extract three distinct features: (1) global feature f_g from the entire RGB frame, capturing the full scene context; (2) local feature f_l from the image crop with

background; (3) masked feature f_m from the image crop without background, focusing solely on the object. During the process, SAMv2/sam2-hiera-large and SigLIP2/siglip2-base-patch16-512 models are used.

Our dynamic weighting mechanism combines the three extracted features through the following equations. First, we calculate the cosine similarity between features with and without background and use it to create a fused local feature:

$$r_{lm} = \text{sim}(f_l, f_m) \quad (6)$$

$$F_l = r_{lm} \cdot f_m + (1 - r_{lm}) \cdot f_l, \quad (7)$$

$$F_l = \text{normalize}(F_l) \quad (8)$$

We then compute the similarity between this fused local feature and the global feature to determine the final weights:

$$\phi_{lG} = \text{sim}(F_l, f_g), \quad w_i = \text{softmax}(\phi_{lG}) \quad (9)$$

$$w_g = w_i, \quad w_m = (1 - w_i) \cdot r_{lm}, \quad (10)$$

$$w_l = (1 - w_i) \cdot (1 - r_{lm}) \quad (11)$$

The final fused feature is computed as a weighted combination and normalized as:

$$f_s = w_g \cdot f_g + w_l \cdot f_l + w_m \cdot f_m, \quad f_s = \text{normalize}(f_s) \quad (12)$$

This mechanism adaptively balances the global context influence via w_g , local context with background via w_l , and object-specific features via w_m . These three features are dynamically combined to create a representation that adapts to the segment’s relationship with its context. For objects that are highly integrated with their surroundings (*e.g.*, a keyboard in front of a monitor), background-inclusive features receive a higher weight. For isolated objects (*e.g.*, a coffee mug), background-excluded features dominate.

A.2. Vision-Language Pretraining

Our vision-language 3DGS pretraining model is built on a transformer encoder-decoder architecture adapted from [51]. As detailed in Tab. A, the encoder consists of 4 stages with depths [2,2,2,6], channels [32,64,128,256], and heads [2,4,8,16], while the decoder employs 3 stages with depths [2,2,2], channels [768,512,256], and 16 attention heads each. We process 3D Gaussian primitives with all parameters (center, color, opacity, quaternion, and scale). The model is trained with the AdamW optimizer (initial LR

$= 0.006$, weight decay = 0.05) using a OneCycle scheduler with cosine annealing. Our loss weights are set to $\lambda_{\text{cos}} = 1.0$, $\lambda_{\text{L2}} = 1.0$, and $\lambda_{\text{con}} = 0.02$, with contrastive loss (temperature $\tau = 0.2$) activated only in the later 75% of training. During training, we employ extensive data augmentation, including random rotations, scaling, flipping, jittering, and elastic distortions (see Tab. C). We train for 800 data epochs using 4 NVIDIA H100 GPUs.

A.3. Masked Gaussian Modeling

For all self-supervised pretraining experiments on the full dataset, we employ 4 NVIDIA H100 GPUs (94GB each) and the AdamW optimizer, with learning rates of 1×10^{-3} for the embedding layer and 1×10^{-4} for the attention blocks, coupled with a weight decay of 1×10^{-3} . We use mixed-precision training (FP16) to accelerate convergence and reduce memory overhead. In total, we train for 300k steps. We use an \mathcal{L}_2 loss to enforce consistency between predicted and ground truth Gaussian parameters, focusing only on masked regions. For parameter reconstruction, we design a three-layer MLP projector for each Gaussian attribute, incorporating output activations that adhere to physical constraints: color uses a tanh activation (bounded in $[-1, 1]$), opacity and scale use a sigmoid activation (bounded in $[0, 1]$) because in most indoor scenes, most Gaussians have small scales, rotation, and normals apply tanh followed by ℓ_2 -normalization to ensure valid quaternions and normals.

A.4. Self-Distillation Representation Learning

In self-distilled representation learning, we adopt the framework of [52], combining DINO loss and coding rate regularization on pooled encoder features. We configure a batch size of 24 and set the grid sampling resolution to 0.02 for partitioning Gaussian scenes. After grid sampling, the scenes are randomly partitioned into base views \mathcal{G}_b , which are then used to generate global and local crops. For global crops, we randomly select a Gaussian splat from \mathcal{G}_b and sample its K neighbors, where $K \sim [0.4, 1.0] \times 256,000$. For local crops, we sample $K \sim [0.1, 0.4] \times 102,400$. Each batch includes $N_g = 2$ global views and $N_l = 3$ local views to balance the scene coverage and granularity. For the DINO loss $\mathcal{L}_{\text{DINO}}$, we extract tokenized features from the student encoder $f_\theta(\cdot)$ and teacher encoder $f_\phi(\cdot)$, where ϕ is updated via exponential moving average (EMA). A global representation \bar{z} is obtained by mean pooling over spatial tokens. This representation is projected into a latent space using a 3-layer MLP P_{DINO} (dimensions: 2048 \rightarrow 2048 \rightarrow 256) followed by ℓ_2 -normalization. Then, we calculate the

similarity loss:

$$\mathcal{L}_{\text{sim}} = \frac{1}{N_g \times N_l} \sum_{i=1}^{N_l} \sum_{j=1}^{N_g} \frac{P_{\text{DINO}}(\bar{z}_l^{(i)}) \cdot P_{\text{DINO}}(\bar{z}_g^{(j)})}{\|P_{\text{DINO}}(\bar{z}_l^{(i)})\| \|P_{\text{DINO}}(\bar{z}_g^{(j)})\|} \quad (13)$$

For regularization, we use the negative of the coding rate:

$$R_\epsilon(\Gamma) := \frac{1}{2} \log \det \left(\mathbf{I} + \frac{d}{\epsilon^2} \Gamma \right), \quad (14)$$

R_ϵ approximates the rate-distortion function of a Gaussian random variable with covariance Γ , becoming exact as $\epsilon \rightarrow 0$. More specifically, it quantifies the spread of covariance, even when the underlying variables are not strictly Gaussian.

For the iBOT loss, we mask 50% of the global views with masking ratios sampled uniformly from $r \sim [0.2, 0.7]$. The masked tokens are replaced by a learnable token (T_{mask}). Note here the masked global view embedding feature goes through the student network and outputs $\hat{T}'(\theta_s) \in \mathbb{R}^{N' \times d_r}$, where d_r denotes the representation dimension, while the unmasked global view embedding is forwarded to the teacher network and outputs $\hat{T}'(\theta_t) \in \mathbb{R}^{N' \times d_r}$. We detach the teacher's output and calculate the simple iBOT loss using 3-layer MLP $P_{\text{iBOT}}(\cdot)$ (dimensions: 256 \rightarrow 256 \rightarrow 32) as the projector:

$$\mathcal{L}_{\text{iBOT}} = \frac{1}{|M|} \sum_{i \in M} \underbrace{\frac{P_{\text{iBOT}}(\hat{T}'(\theta_s)^{(i)}) \cdot P_{\text{iBOT}}(\hat{T}'(\theta_t)^{(i)})}{\|P_{\text{iBOT}}(\hat{T}'(\theta_s)^{(i)})\| \|P_{\text{iBOT}}(\hat{T}'(\theta_t)^{(i)})\|}}}_{\text{Pairwise Cosine Similarity}} \quad (15)$$

Only the masked regions M contribute to the iBOT loss calculation. Following [52], we simplify the original iBOT implementation by removing the centering operation and online tokenizer and replacing them with a pairwise cosine-similarity loss for feature alignment. When MGM loss is enabled, for one global view, we will employ less aggressive augmentation, and this view will contribute to both iBOT loss and MGM loss.

A.5. Autoencoder for Feature Compression

Building on the idea from [36], we train a scene-specific language autoencoder on precomputed vision-language embeddings. This model compresses high-dimensional SigLIP features into a low-dimensional latent space, enabling the efficient storage of language features within Gaussian representations. The encoder (by default 5-layer architecture: [384, 192, 96, 48, 16]) generates compact latent codes, while the symmetric decoder (by default 5-layer architecture: [48, 96, 192, 384, 768]) reconstructs the original high-dimensional embeddings. This design reduces memory overhead while preserving semantic fidelity. Unlike [36], our model is trained on 3D Gaussian features rather

Config	Value
embedding depth	2
embedding channels	32
encoder depth	[2, 2, 2, 6]
encoder channels	[32, 64, 128, 256]
encoder num heads	[2, 4, 8, 16]
encoder patch size	[1024, 1024, 1024, 1024]
decoder depth	[2, 2, 2]
decoder channels	[768, 512, 256]
decoder num heads	[16, 16, 16]
decoder patch size	[1024, 1024, 1024]
down stride	[$\times 2$, $\times 2$, $\times 2$, $\times 2$]
mlp ratio	4
qkv bias	True
drop path	0.3

Table A. Model Configs for Vision-Language Pretraining.

Config	Value
embedding depth	2
embedding channels	32
encoder depth	[2, 2, 2, 6, 2]
encoder channels	[32, 64, 128, 256, 512]
encoder num heads	[2, 4, 8, 16, 32]
encoder patch size	[1024, 1024, 1024, 1024]
decoder depth	[2, 2, 2, 2]
decoder channels	[64, 64, 128, 256]
decoder num heads	[4, 4, 8, 16]
decoder patch size	[1024, 1024, 1024]
down stride	[$\times 2$, $\times 2$, $\times 2$, $\times 2$]
mlp ratio	4
qkv bias	True
drop path	0.3

Table B. Model Configs for GaussianSSL Pretraining and Downstream Semantic Segmentation.

than 2D image-level data. For detailed ablations on the autoencoder architecture and training paradigm, see Tab. E.

A.6. Gaussian Language Alignment.

We follow the same loss functions in Eqs. (2) and (3) for feature alignment. When combining MGM with language alignment, we adhere to the MGM process by masking the input tokens and applying compressed language alignment loss only to the masked regions. Given the comparable magnitudes of all loss terms, we assign equal weights ($\omega_{\text{MGM}} = \omega_{\text{DINO}} = \omega_{\text{iBOT}} = \omega_{\text{LA}} = 1.0$) to maintain the balance of different objectives during training.

A.7. Model Architecture

The network design choices are described in detail in Tabs. A and B. For the 3D backbone, we adopt the state-of-the-art Point Transformer [51], enhanced with Flash At-

tention, to substantially improve computational efficiency. To optimize feature embedding for scene-level Gaussians, we use sparse convolutions, which preserve geometric details while minimizing the memory overhead.

B. Further Experimental Results

B.1. SceneSplat Zero-shot Segmentation

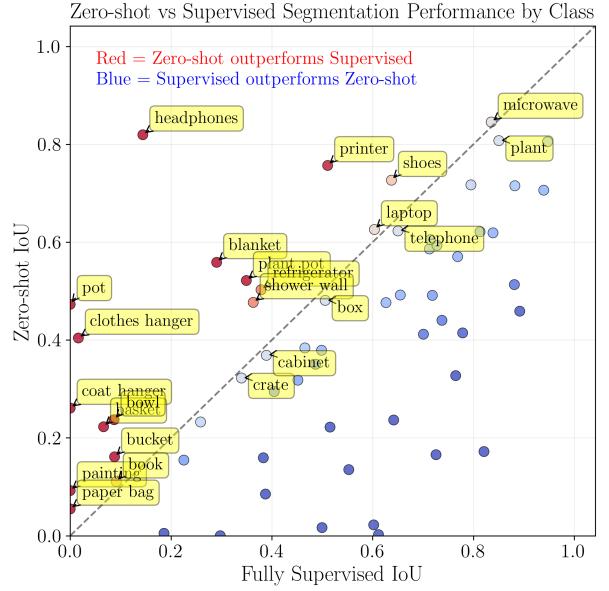


Figure A. Comparison of SceneSplat Zero-Shot Versus Fully Supervised Segmentation Across Object Classes. Notably, our zero-shot segmentation after vision language pretraining demonstrates better performance for 18 object classes, predominantly small objects such as headphone, printer, pot, and clothes hanger.

Zero-shot vs. Fully Supervised. Fig. A compares the class-wise zero-shot 3D segmentation results with those of the current state-of-the-art supervised method [51]. Points above the diagonal line (colored red) represent classes where zero-shot segmentation outperforms fully supervised approaches, whereas points below (colored blue) indicate the opposite. Our zero-shot segmentation achieves superior performance for 18 object classes in the ScanNet++ benchmark, predominantly small objects such as headphones, printers, pots, and clothes hangers. These results demonstrate the robust prior knowledge acquired by our model through vision-language pretraining.

More Qualitative Zero-shot Segmentation Results. Fig. C presents more qualitative zero-shot semantic segmentation results on ScanNet++ validation scenes. SceneSplat effectively segments the scenes and helps annotate regions with missing labels in the ground truth.

Consistency Issue in Label Collection. Tab. D presents our zero-shot segmentation results on the ScanNet20 benchmark. We observe that the performance on this particular

Augmentations	Parameters	Global View	Local View
	Base Transform	✓	✓
random rotate	axis: z, angle: [-1, 1], p: 0.5 axis: x, angle: [-1 / 64, 1 / 64], p: 0.5 axis: y, angle: [-1 / 64, 1 / 64], p: 0.5		
random scale	scale: [0.9, 1.1]		
random flip	p: 0.5		
random jitter	sigma: 0.005, clip: 0.02		
elastic distort	params: [[0.9, 0.1]]		
grid sampling	grid size 0.02		
	Global Base Transform	✓	-
random flip	p: 0.5		
random crop	ratio: (0.4, 1.0) max: 256000		
	Global Transform 0	✓	-
random color jitter	b:0.4, c:0.4, s:0.2 hue:0.1 p:0.8		
random grayscale	p: 0.2		
random Gaussian blur	p: 1.0		
	Global Transform 1	✓	-
random dropout	dropout ratio: 0.2, p: 0.2		
random color jitter	b:0.4, c:0.4, s:0.2 hue:0.1 p:0.8		
random grayscale	p: 0.2		
random Gaussian blur	p: 0.2		
	Local Base Transform	-	✓
elastic distort	params: [[0.2, 0.4], [0.8, 1.6]]		
random flip	p: 0.5		
random crop	ratio: (0.1, 0.4) max: 102400		
	Local Transform	-	✓
random dropout	dropout ratio: 0.2, p: 0.2		
random color jitter	b:0.4, c:0.4, s:0.2 hue:0.1 p:0.8		
random grayscale	p: 0.2		
random Gaussian blur	p: 0.5		

Table C. **Data Augmentations.** Following [31], we design the data augmentations for 3D scenes for global and local views.

benchmark is significantly lower compared to the state-of-the-art results we achieve on the other three benchmarks. Through a detailed analysis, we identify that the issue stems from inconsistencies in the 3DGS language label collection process. Specifically, the SAMv2+SigLIP2 pipeline that we employ does not guarantee temporal consistency, especially for large background objects such as walls and floors, as illustrated in Fig. B. This inconsistency results in corrupted feature fields for Gaussians associated with these regions, subsequently leading to reduced zero-shot segmentation performance. Addressing this temporal consistency issue remains an important direction for future research.

B.2. 3DGS Self-Supervised Pretraining

In Tab. E, we present an ablation study that compares various autoencoder architectures trained on the ScanNet++ training set and evaluated on the validation set. In the “ScanNet++ 2D” columns, the autoencoder is trained using SigLIP2 features preprocessed from images, while in the “ScanNet++ 3D” columns, it is trained directly on Gaussian SigLIP2 features. At inference, we measure both the L2 distance and cosine similarity on 3D Gaussian SigLIP2 features. The results indicate that training the autoencoder directly on 3D data yields notably better performance, whereas increasing the network depth from layer5 to layer6 leads to only marginal improvements. Moreover, the

Method	Training Source	ScanNet20 (20)	
		f-mIoU	f-mAcc
OpenScene	ScanNet	57.5	72.4
Mosaic3D	ScanNet	65.0	82.5
PLA	ScanNet	19.1	41.5
RegionPLC	ScanNet	55.6	76.3
OV3D	ScanNet	64.0	76.3
SceneSplat	ScanNet	35.4	57.9

Table D. **Zero-Shot 3D Semantic Segmentation on ScanNet20 Benchmark.**

The results for the baselines are taken from [21]. Ours observes many faulty predictions and have poor performance, we identify the issue in the inconsistency during 2D feature map collection when labeling Gaussians, which leads to corrupted 3DGS-feature pairs.

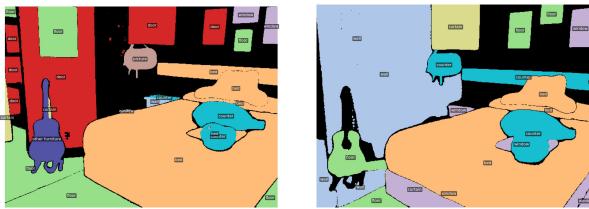


Figure B. **Consistency Issue During 2D Vision-Language Feature Map Collection on ScanNet.** Erroneous 2D feature maps are collected, as shown by the mislabeled regions in the neighboring figures. The root cause is that the SAMv2+SigLip2 process we use does not guarantee temporal consistency.

Method	ScanNet++ 2D		ScanNet++ 3D	
	L2	cosine	L2	cosine
VAE16 layer5	0.008	0.182	0.005	0.028
VAE16 layer6	0.008	0.172	0.005	0.023
VAE64 layer5	0.007	0.081	0.004	0.014
VAE64 layer6	0.007	0.079	0.004	0.013

Table E. **AutoEncoder Ablation Experiments.** We report the feature compression performance with \mathcal{L}_2 loss and cosine similarity on unseen 3D language label. We ablate on different autoencoder architectures and training sources.

64-dimensional latent space consistently outperforms the 16-dimensional counterpart. Consequently, as detailed in Sec. A.5, we adopt the 3D-trained autoencoder with layer5 and a 16-dimensional latent space as our default configuration, striking a favorable balance between efficiency and accuracy.

We evaluate the impact of language feature alignment loss dimensionality (64 vs. 16) when pretraining exclusively on ScanNet++ and testing on the ScanNet20 benchmark. As shown in Tab. F, we implement two 3-layer MLP architectures: one with uniform dimensions ($64 \rightarrow 64 \rightarrow 64$) for 64 dimension language feature and another with progressively reduced dimensions ($64 \rightarrow 32 \rightarrow 16$). We observe that reducing the latent dimension from 64D to 16D im-

Method	ScanNet20 (20)	
	mIoU	mAcc
LA16	76.3	83.9
LA64	75.8	82.2
LA16 (MGM)	76.2	83.7

Table F. **Language Alignment Loss Ablation.** We conduct an ablation study on low-dimensional language feature compression and Masked Gaussian Modeling (MGM)-based pretraining to evaluate their impact on semantic segmentation performance.

Mask Ratio	ScanNet20		Mask Size	ScanNet20	
	mIoU	mAcc		mIoU	mAcc
0.4	76.1	83.6	0.02	76.9	84.5
0.5	76.4	84.1	0.05	77.0	84.5
0.6	77.0	84.4	0.10	76.6	83.8
0.7	76.7	83.8	0.15	76.5	84.1

Table G. **Masked Gaussian Modeling Ablation Experiment.** We analyze the impact of masking ratios and grid sizes in Masked Gaussian Modeling (MGM) on semantic segmentation performance using the ScanNet20 benchmark.

Method	ModelNet10 (10)		Omniobject3D (83)	
	Linear	MLP-3	Linear	MLP-3
MGM	<u>77.3</u>	<u>83.1</u>	50.3	57.5
+DINO	74.5	80.2	40.5	42.4
+iBOT	65.2	73.8	38.4	40.8
+LA	68.1	74.8	26.5	31.2
MGM+LA	83.1	84.6	<u>47.3</u>	<u>53.2</u>

Table H. **Cross Domain Linear Probe Experiments.** We report the mAcc results in the ablation of GaussianSSL on object-level classification tasks using the linear probe.

proves mIoU by +0.5% in mIoU and +0.7% in mAcc. This suggests that lower-dimensional language features with hierarchical dimensionality reduction preserve critical indoor scene semantic information.

We visualize the pretraining results using Masked Gaussian Modeling (MGM) and language feature alignment loss in Fig. D. The model effectively reconstructs masked Gaussian parameters (e.g., position, scale, and rotation) and predicts semantically meaningful language-aligned features for indoor scene understanding.

Tab. G evaluates how masking ratio and grid size in Masked Gaussian Modeling (MGM) affect downstream semantic segmentation performance. Key observations include: (1) small masking ratios (0.4) degrade mIoU by 0.9% due to insufficient context learning compared to the ratio 0.6; (2) using a larger grid size 0.15m for masking reduces fine-grained detail recovery, leading to a 0.5% drop in mIoU compared to a 0.05m grid size.

To evaluate global scene understanding, we design a cross-domain classification task (Tab. H) using object-level Gaussian splats from [28, 29]. We freeze the encoder and

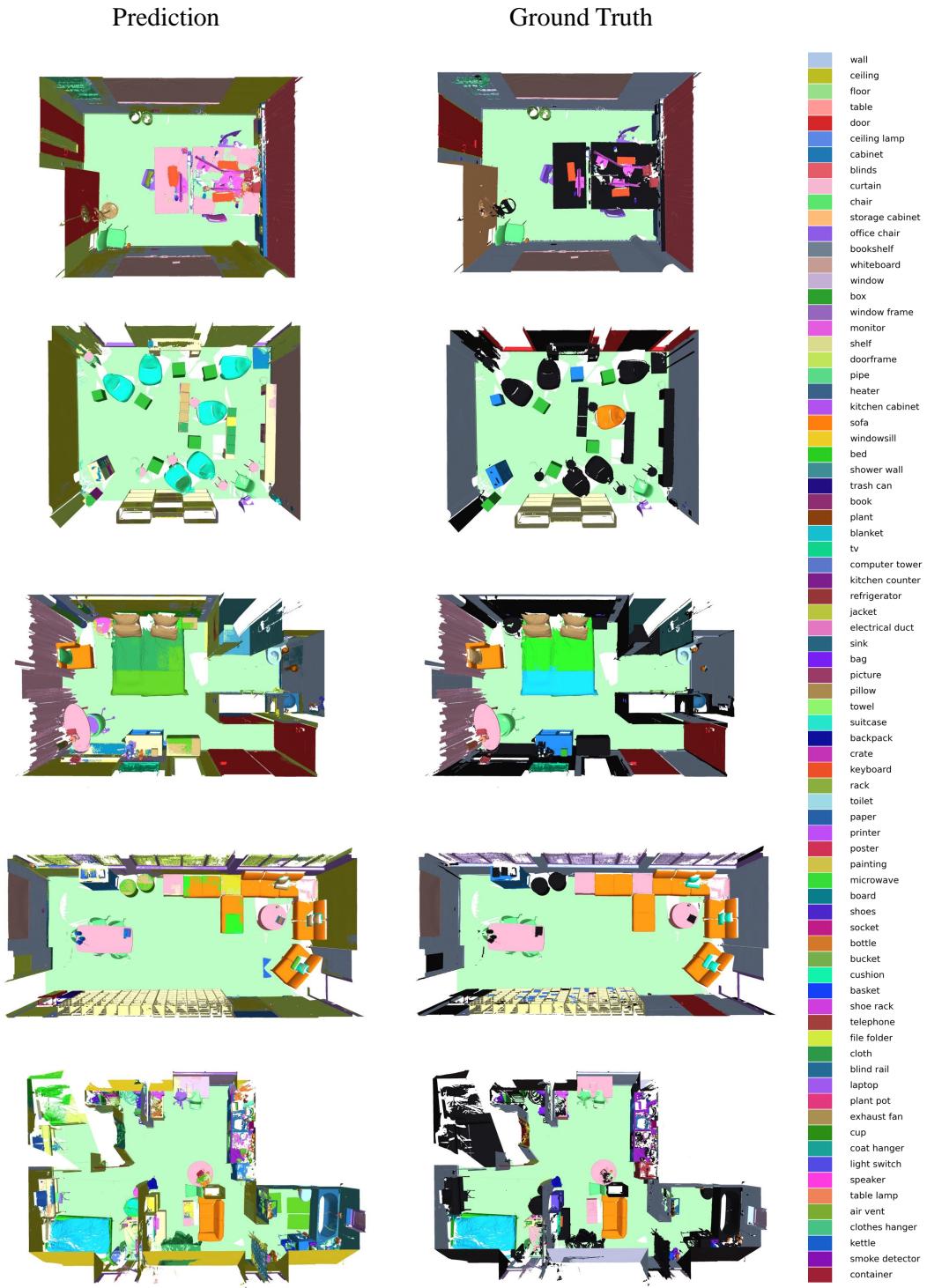


Figure C. Qualitative Zero-shot Semantic Segmentation Results on ScanNet++ Validation Scenes. SceneSplat effectively segments the scenes and helps annotate regions with missing labels in the ground truth.

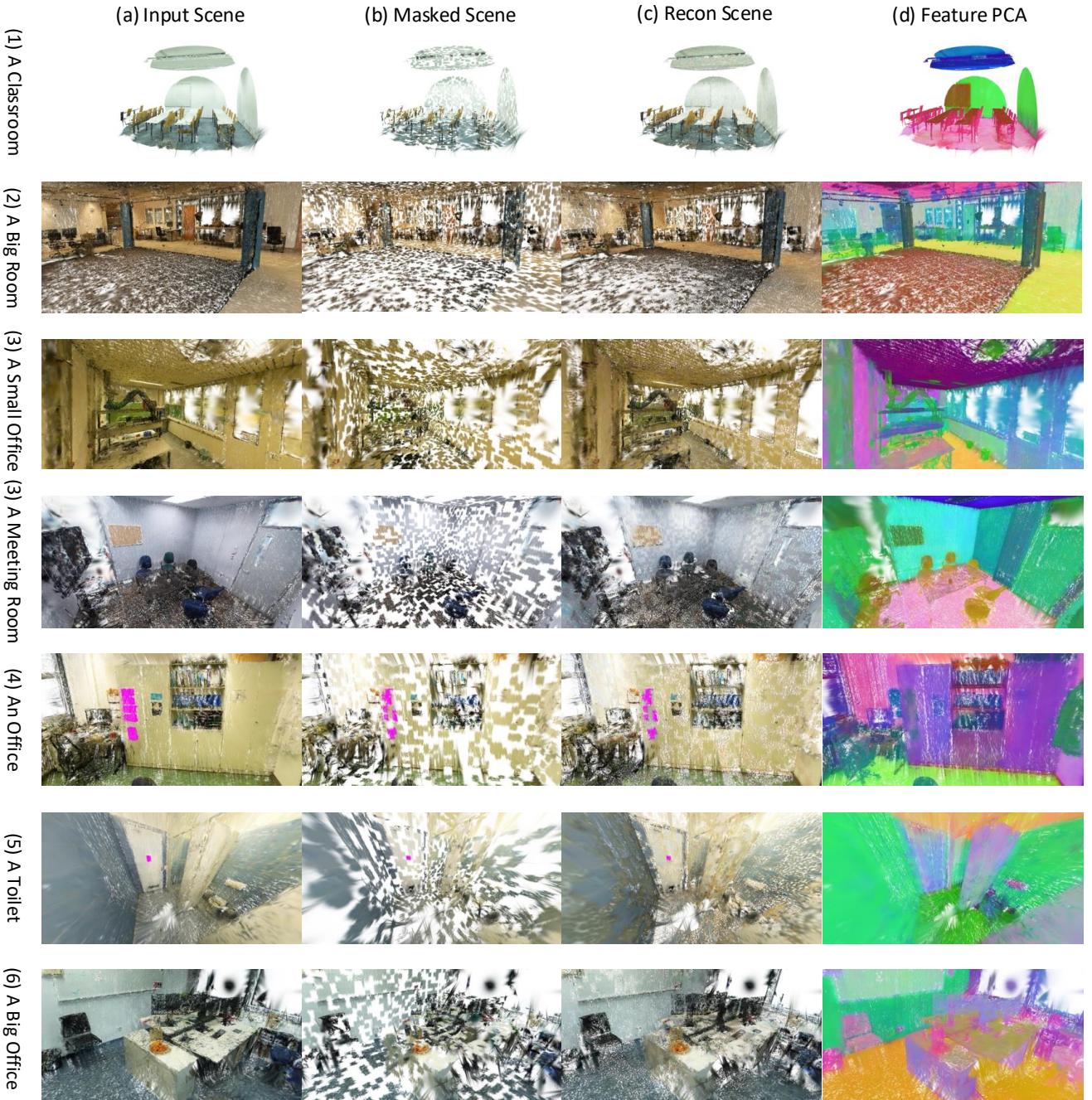


Figure D. **Self-supervised Reconstructions across Multiple Scenes.** Each row shows (left to right) the unmasked input, masked scene, reconstruction, and a PCA projection of features

embedding layers and train only (1) a linear probe (final layer) to map the features to logits. (2) a 3-layer MLP ($512 \rightarrow 256 \rightarrow$ classes) for nonlinear evaluation. We find that progressively adding DINO, iBOT, and language alignment (LA) losses degrades the classification accuracy. This misaligned trend with scene-level benchmarks stems from domain gaps, where the model must map distinct objects

(e.g., fridge, oven) to similar scene-level semantics (e.g., “kitchen”). Using masked pretraining and language alignment loss achieves the best performance on ModelNet10 (furniture), whereas MGM alone excels on OmniObject3D (common objects). For ModelNet10, MGM achieves the best 80% accuracy, whereas for the challenging object dataset (e.g., OmniObject3D), it peaks at 50%.

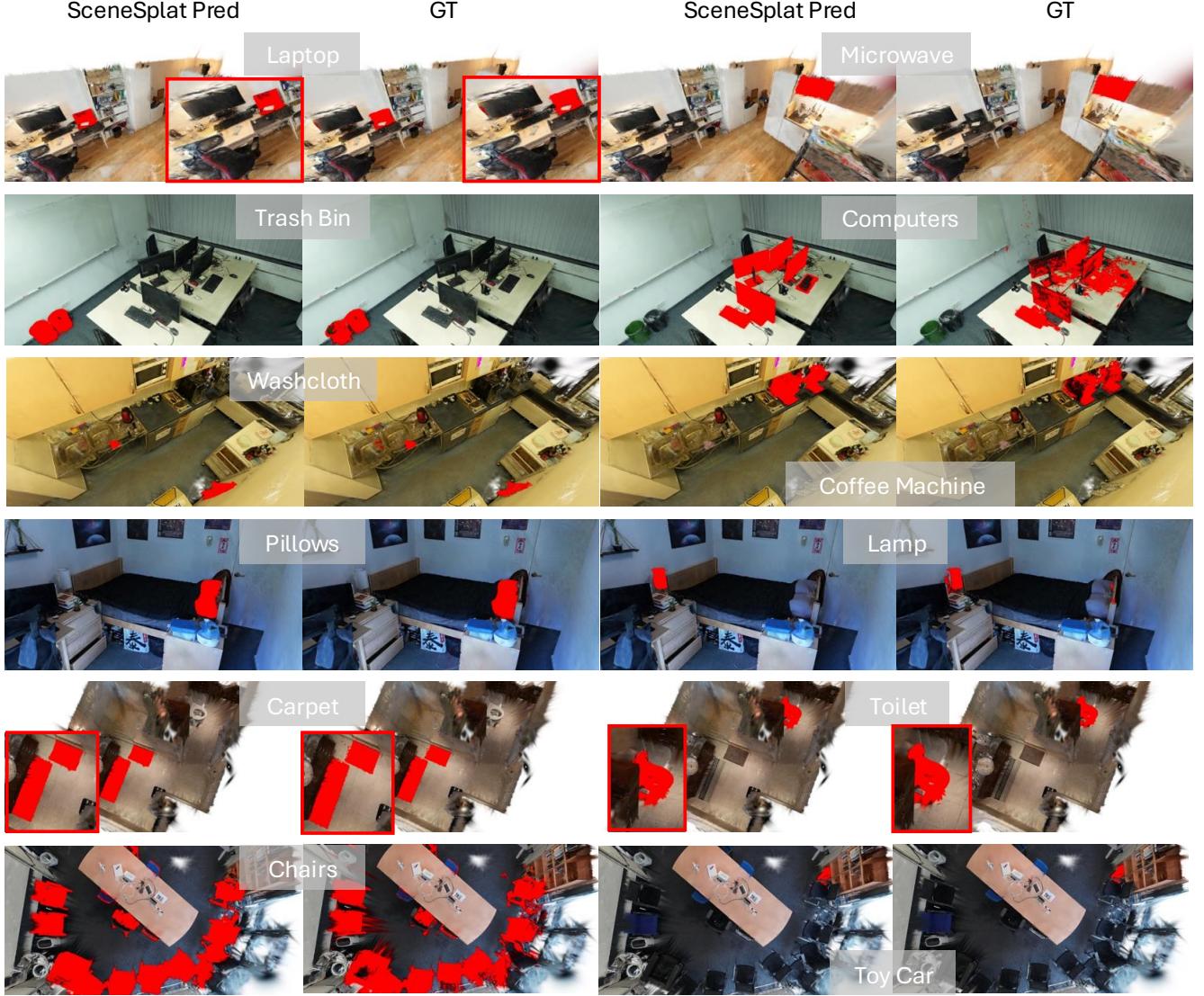


Figure E. Comparison of Scene Query Results Using Our Predictions and GT Language Labels on ScanNet++.

C. Datasets Curation and Statistics

C.1. Analysis

Gaussian Splatting (GS) has demonstrated strong reconstruction capabilities, but its performance is heavily dependent on raw dataset quality and input conditions. We analyzed both quantitative metrics and visualization results of reconstructed scenes and identified several common failure cases. The primary issues include holes, floating artifacts, and non-smooth surfaces. Through our analysis, we categorize the root causes into four main factors.

Lack of Frames and Filming Angles. Scenes with fewer than 400 RGB frames often fail to capture a complete representation of the environment. Limited filming angles result in missing perspectives, leading to incomplete indoor structures.



Figure F. Example Incomplete Scenes in ARKitScenes.

tures. Fig. F from the ARKitScenes dataset show partial room reconstructions where corners or ceiling details are lost.

Motion Blur and Camera Instability. As seen in some



Figure G. **Blurry Scenes from 3RScan.**

blurry input examples Fig. G in the 3RScan dataset, the rapid, unsteady filming leads to edge deformation, ghosting, and smeared textures, and in unrealistic reconstructions, the sharp edges are lost, significantly reducing the fine details in the rendered GS.

Indoor-outdoor Lighting Changes. Scenes with dynamic indoor-outdoor lighting changes, such as the case from the Hypersim dataset, exhibit severe blurring and loss of surface textures. Large glass ceilings, skylights, and reflective floors further disrupt GS training, as the algorithm struggles with light inconsistency and high contrast between illuminated and shadowed areas.

Challenges with Transparent and Glass Objects. Transparent and glass objects pose a unique challenge, as GS often fails to accurately capture windows or furniture glass, resulting in missing elements or floating artifacts. This issue arises from the inherent difficulty in rendering transparency, where reflection and refraction introduce complexity beyond the current GS capabilities.

C.2. Visualization

The SceneSplat-7K dataset achieves an impressive average PSNR of 28.17 dB through all the different sources. We provide visualizations that showcase the photorealistic appearance rendering. See Sec. C.2.

D. Dataset License

SceneSplat-7K is constructed from several established indoor datasets, each attached with a specific license: ARKitScenes [1] (Apple Open Source License), Replica [45] (Replica Research License), ScanNet [5] (ScanNet Terms of Use), ScanNet++ [57] (ScanNet++ Terms of Use), Hypersim [42] (CC BY-SA 3.0), 3RScan [48] (3RScan Terms of Use), and Matterport3D [2] (Matterport Academic License Agreement). We have carefully structured our distribution approach to respect all original licenses while making our dataset accessible to the research community. For sources that permit redistribution, we plan to release our data on

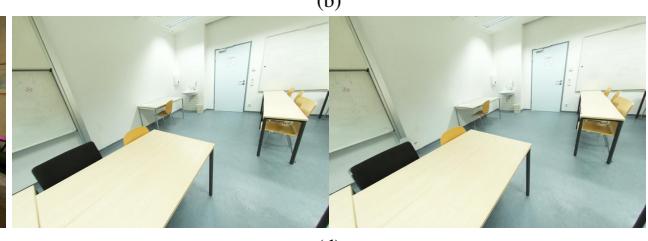
the Hugging Face under their respective terms of use. For datasets that require special permission, we co-host the data only after receiving approval from the original teams or let the original dataset team host the 3DGS data.

E. Limitations

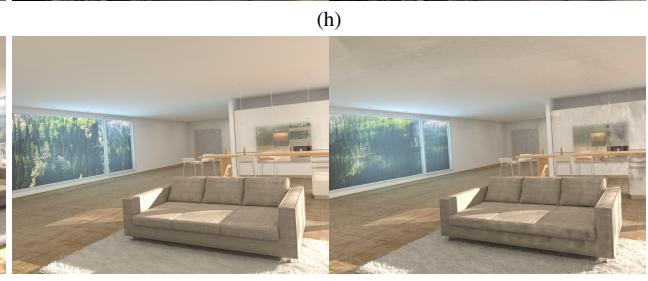
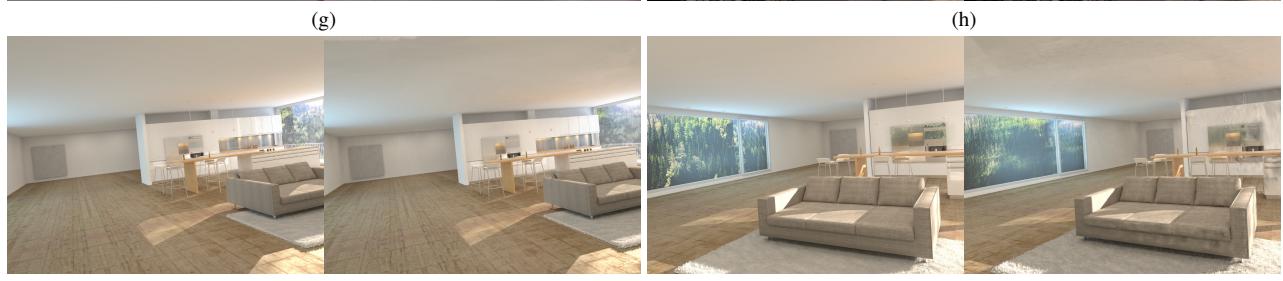
The quality of our dataset is largely influenced by the source indoor datasets. Low-resolution and blurry images can cause floating artifacts. Downstream tasks are also limited by the annotations of the original dataset. Temporal inconsistency in the current language label obtaining process needs to be addressed, as it can pollute vision-language pre-training. In addition, we plan to add bounding boxes [13] and language descriptions [27] in the next step.

F. Impact Statement

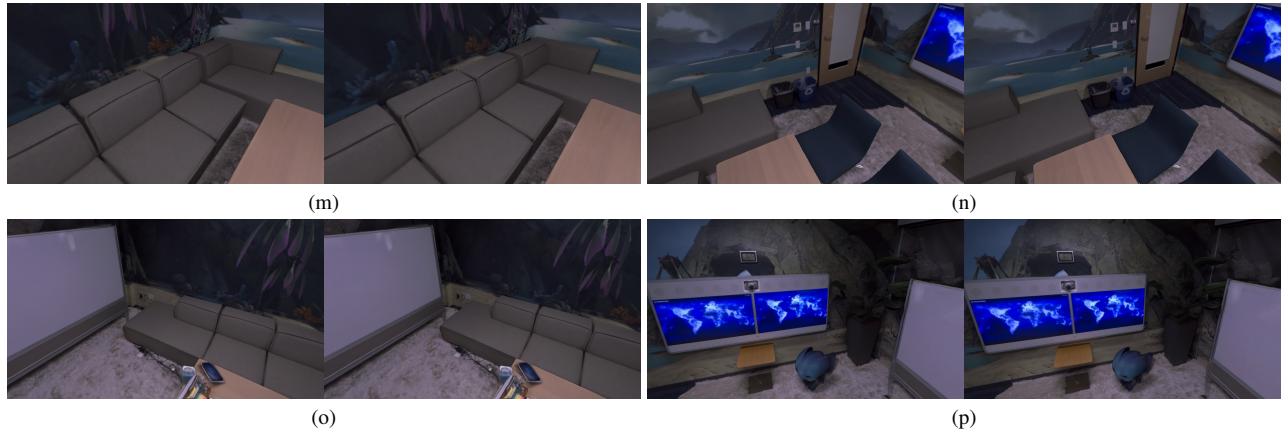
This work introduces SceneSplat-7K, the first large-scale indoor dataset of 3D Gaussian Splatting (3DGS). By providing over 7K annotated scenes, it enables standardized benchmarking for 3DGS-based reasoning in vision tasks. The proposed framework for vision-language pretraining tailored to 3DGS enhances semantic alignment and establishes a clear connection between latent representations and 3DGS scenes. By leveraging knowledge distillation from 2D foundation models, combining contrastive learning and masked gaussian modeling (MGM), our approach significantly outperforms existing methods in 3D semantic segmentation. This study lays the foundation for advancing scalable 3D scene understanding, with broad implications for autonomous systems and augmented reality applications. By publicly releasing the dataset, model, and code, we foster further innovation and facilitate the development of generalizable solutions for 3D scene understanding.



ScanNet++ GS. Ground truth (left) and 3DGS rendering results (right).



Hypersim GS. Ground truth (left) and 3DGS rendering results (right).



Replica Office0 GS. Ground truth (left) and 3DGS rendering results (right). PSNR: 45.55 dB.



Replica Room2 GS. Ground truth (left) and 3DGS rendering results (right). PSNR: 41.57 dB.



Figure G. 3RScan GS. Ground truth (left) and 3DGS rendering results (right). PSNR: 34.43 dB.

References

- [1] 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. 1, 2, 3, 4, 7, 9
- [2] 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. 1, 2, 3, 4, 5, 6, 7, 9
- [3] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning, 2020. 6
- [4] 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. 3, 4
- [5] 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. 1, 2, 3, 4, 5, 6, 7, 9
- [6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 3
- [7] 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 (CVPR)*, pages 7010–7019, 2023. 2
- [8] 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, 4, 7, 1
- [9] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations (ICLR)*, 2020. 3
- [10] 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
- [11] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16000–16009, 2022. 3, 6
- [12] Georg Hess, Johan Jaxing, Elias Svensson, David Hagerman, Christoffer Petersson, and Lennart Svensson. Masked autoencoder for self-supervised pre-training on lidar point clouds. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 350–359, 2023. 3
- [13] Muhammad Zubair Irshad, Sergey Zakharov, Vitor Guizilini, Adrien Gaidon, Zsolt Kira, and Rares Ambrus. Nerf-mae: Masked autoencoders for self-supervised 3d representation learning for neural radiance fields. In *European Conference on Computer Vision (ECCV)*, 2024. 2, 9
- [14] Baoxiong Jia, Yixin Chen, Huangyue Yu, Yan Wang, Xuesong Niu, Tengyu Liu, Qing Li, and Siyuan Huang. Sceneverse: Scaling 3d vision-language learning for grounded scene understanding. In *European Conference on Computer Vision*, pages 289–310. Springer, 2024. 7
- [15] 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, 7
- [16] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):139–1, 2023. 2, 3
- [17] 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. 2, 3
- [18] 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. 4
- [19] Diederik P Kingma and Max Welling. Auto-encoding variational bayes, 2022. 6
- [20] 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
- [21] 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, 6, 7, 1, 5
- [22] Yangyan Li, Rui Bu, Mingchao Sun, Wei Wu, Xinhuan Di, and Baoquan Chen. Pointcnn: Convolution on x-transformed points. *Advances in neural information processing systems*, 31, 2018. 3
- [23] Zengyi Li, Yubei Chen, Yann LeCun, and Friedrich T. Sommer. Neural manifold clustering and embedding, 2022. 6
- [24] Dingkang Liang, Xin Zhou, Wei Xu, Xingkui Zhu, Zhikang Zou, Xiaoqing Ye, Xiao Tan, and Xiang Bai. Pointmamba: A simple state space model for point cloud analysis. *Advances in neural information processing systems*, 37:32653–32677, 2025. 3
- [25] Yaqian Liang, Shanshan Zhao, Baosheng Yu, Jing Zhang, and Fazhi He. Meshmae: Masked autoencoders for 3d mesh data analysis. In *European Conference on Computer Vision*, pages 37–54. Springer, 2022. 3

- [26] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 26296–26306, 2024. 4, 1
- [27] Ruiyuan Lyu, Tai Wang, Jingli Lin, Shuai Yang, Xiaohan Mao, Yilun Chen, Runsen Xu, Haifeng Huang, Chenming Zhu, Dahua Lin, and Jiangmiao Pang. Mmscan: A multi-modal 3d scene dataset with hierarchical grounded language annotations. In *arXiv*, 2024. 9
- [28] 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 *3D Vision*, 2024. 2, 3, 5
- [29] 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. 2, 5
- [30] 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
- [31] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023. 3, 6, 4
- [32] 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. 3
- [33] 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, 7
- [34] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017. 3
- [35] Zekun Qi, Runpei Dong, Guofan Fan, Zheng Ge, Xiangyu Zhang, Kaisheng Ma, and Li Yi. Contrast with reconstruct: Contrastive 3d representation learning guided by generative pretraining. In *International Conference on Machine Learning*, pages 28223–28243. PMLR, 2023. 3
- [36] Minghan Qin, Wanhua Li, Jiawei Zhou, Haoqian Wang, and Hanspeter Pfister. Langsplat: 3d language gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20051–20060, 2024. 3, 4, 2
- [37] 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
- [38] 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
- [39] 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
- [40] Bin Ren, Yahui Liu, Yue Song, Wei Bi, Rita Cucchiara, Nicu Sebe, and Wei Wang. Masked jigsaw puzzle: A versatile position embedding for vision transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20382–20391, 2023. 3
- [41] 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. 3
- [42] 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. 1, 2, 3, 4, 9
- [43] 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
- [44] Jin-Chuan Shi, Miao Wang, Hao-Bin Duan, and Shao-Hua Guan. Language embedded 3d gaussians for open-vocabulary scene understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5333–5343, 2024. 4
- [45] 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. 1, 2, 3, 4, 9
- [46] 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
- [47] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2023. 3
- [48] 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. 1, 2, 3, 4, 9
- [49] Abdelrhman Werby, Chenguang Huang, Martin Büchner, Abhinav Valada, and Wolfram Burgard. Hierarchical open-

- vocabulary 3d scene graphs for language-grounded robot navigation. In *First Workshop on Vision-Language Models for Navigation and Manipulation at ICRA 2024*, 2024. 1
- [50] 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. 2, 3, 7
- [51] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4840–4851, 2024. 2, 3, 4, 7, 1
- [52] Ziyang Wu, Jingyuan Zhang, Druv Pai, XuDong Wang, Chandan Singh, Jianwei Yang, Jianfeng Gao, and Yi Ma. Simplifying dino via coding rate regularization. *arXiv preprint arXiv:2502.10385*, 2025. 6, 2
- [53] Zhenda Xie, Zheng Zhang, Yue Cao, Yutong Lin, Jianmin Bao, Zhiliang Yao, Qi Dai, and Han Hu. Simmim: A simple framework for masked image modeling, 2022. 6
- [54] 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, 7
- [55] Yu-Qi Yang, Yu-Xiao Guo, Jian-Yu Xiong, Yang Liu, Hao Pan, Peng-Shuai Wang, Xin Tong, and Baining Guo. Swin3d: A pretrained transformer backbone for 3d indoor scene understanding. *arXiv preprint arXiv:2304.06906*, 2023. 2
- [56] Vickie Ye, Rui long Li, Justin Kerr, Matias Turkulainen, Brent Yi, Zhuoyang Pan, Otto Seiskari, Jianbo Ye, Jeffrey Hu, Matthew Tancik, et al. gsplat: An open-source library for gaussian splatting. *Journal of Machine Learning Research*, 26(34):1–17, 2025. 4
- [57] 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. 1, 2, 3, 4, 5, 6, 7, 9
- [58] Zehao Yu, Anpei Chen, Binbin Huang, Torsten Sattler, and Andreas Geiger. Mip-splatting: Alias-free 3d gaussian splatting. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 19447–19456, 2024. 2
- [59] Yuqian Yuan, Wentong Li, Jian Liu, Dongqi Tang, Xinjie Luo, Chi Qin, Lei Zhang, and Jianke Zhu. Osprey: Pixel understanding with visual instruction tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 28202–28211, 2024. 4, 1
- [60] 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
- [61] Baowen Zhang, Chuan Fang, Rakesh Shrestha, Yixun Liang, Xiaoxiao Long, and Ping Tan. Rade-gs: Rasterizing depth in gaussian splatting. *arXiv preprint arXiv:2406.01467*, 2024. 2
- [62] 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
- [63] Renrui Zhang, Ziyu Guo, Peng Gao, Rongyao Fang, Bin Zhao, Dong Wang, Yu Qiao, and Hongsheng Li. Point-m2ae: multi-scale masked autoencoders for hierarchical point cloud pre-training. *Advances in neural information processing systems*, 35:27061–27074, 2022. 3
- [64] Renrui Zhang, Ziyu Guo, Wei Zhang, Kunchang Li, Xupeng Miao, Bin Cui, Yu Qiao, Peng Gao, and Hongsheng Li. Pointclip: Point cloud understanding by clip. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8552–8562, 2022. 2
- [65] 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. 3, 7
- [66] Jia Zheng, Junfei Zhang, Jing Li, Rui Tang, Shenghua Gao, and Zihan Zhou. Structured3d: A large photo-realistic dataset for structured 3d modeling. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IX 16*, pages 519–535. Springer, 2020. 7
- [67] 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
- [68] Jinghao Zhou, Chen Wei, Huiyu Wang, Wei Shen, Cihang Xie, Alan Yuille, and Tao Kong. ibot: Image bert pre-training with online tokenizer, 2022. 6
- [69] 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. 3
- [70] 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