
LIFT-GS: Cross-Scene Render-Supervised Distillation for 3D Language Grounding

Anonymous Authors¹

Abstract

We present an approach to train a 3D Vision Language Mask Grounding model without requiring 3D supervision. This is achieved by reconstructing the Gaussian splatting field from the input point cloud, and supervising the mask decoder model using 2D labels and losses. This pipeline allows us to distill the knowledge from powerful 2D foundation models to 3D grounding models, demonstrating impressive performance for both zero-shot and pretraining. We show this approach outperforms baselines/SOTA for 3D vision-language grounding, and also outperforms other 3D pretraining techniques.

1. Introduction

When a user mentions *the keys by the door* or *the blue mug on the table*, they use language to indicate a specific set of objects and 3D locations in space. Such *3D language grounding* provides a particularly natural interface for people to communicate about their surroundings.

For AI systems operating in physical spaces, identifying the set of 3D masks or bounding boxes indexed by a language query represents a core functionality; with applications across autonomous navigation, robotic manipulation, and AR/VR. Yet, despite its importance and recent advances in both 2D vision-language grounding and 3D reconstruction, the gap between human and machine performance in *3D Vision-Language Grounding* remains one of the most significant challenges in embodied AI.

3D vision-language grounding (**3D VLG**) is severely constrained by data scarcity. Generative multimodal models are routinely trained on billions to trillions of tokens from language (Achiam et al., 2023; Touvron et al., 2023), images (Radford et al., 2021; Labs, 2023), videos (Polyak et al., 2024; Brooks et al., 2024). Nearer to the task of 3D VLG, models trained for promptable image and video mask segmentation use millions to billions of labeled masks (Kirillov et al., 2023; Ravi et al., 2024). In contrast, existing 3D VLG models are limited to training on only tens of thousands of labeled 3D masks and scenes, restricting their ability.

In this paper, we propose a scalable pretraining approach for 3D VLG that enables grounding language into a set of target 3D masks; without requiring 3D mask labels. *Language-Indexed Field Transfer with Gaussian Splatting (LIFT-GS)*, is supervised on 2D frames directly. It uses differentiable rendering to render the predicted 3D features and masks into 2D frames (e.g. via Gaussian Splatting (Kerbl et al., 2023)). LIFT-GS is then distilled directly from powerful frame-based foundation models, using pseudolabels from SAM (Kirillov et al., 2023), CLIP (Radford et al., 2021), and Llama 3.2 (Meta AI, 2024).

LIFT-GS conducts both *Promptable 3D Segmentation* and *Reconstruction*. The render-supervised formulation is highly flexible; as it imposes essentially no constraints on the model input or network architecture. Therefore, in order to demonstrate the effectiveness of differentiable rendering for structured prediction tasks, we choose model inputs and meta-architectures that are scalable and offer advantages for 3D vision-language grounding.

As visual input, LIFT-GS exclusively utilizes point clouds (positions and colors) to align with the preferences for many modern embodied AI applications that operate in real time. Point clouds from SLAM, for example, serve as widely adopted input in robotics and AR/VR. By relying on raw point clouds rather than 2D feature-enhanced point clouds like ConceptFusion (Jatavallabhula et al., 2023b), LIFT-GS eliminates the need for preprocessing and feature fusion during inference, reducing inference time from approximately 60 seconds per frame to just 1 second for the forward pass.

LIFT-GS follows the common MaskFormer (Cheng et al., 2021b) meta-architecture. However, the output tokens from the *Encoder* and a *Mask Decoder* are instead interpreted as 3D Gaussians rather than image patches. The entire pipeline is then trained end-to-end by optimizing the 2D loss between the rendered 2D masks and the pseudo-grounding masks.

We validate the effectiveness of LIFT-GS on two popular 3D vision-language grounding (3D VLG) downstream tasks: open-vocabulary 3d instance segmentation and 3D referential grounding. The results show that distillation provides significant improvements over models trained from scratch, and achieves state-of-the-art performance on both tasks.

More importantly, the approach consistently improves from both more data and better 2D foundation models, which indicates the potential for further improvement.

Intriguingly, we find that LIFT-GS “pretraining effectively multiplies the fine-tuning dataset” (Hernandez et al., 2021) by a constant factor (roughly 2x), across varying amounts of fine-tuning data. This somewhat counterintuitive observation indeed matches empirical data scaling laws for pretraining in other modalities (Hernandez et al., 2021), where pretraining shifts the power-law scaling in log-dataset size to the left by that constant factor. The fact that the coefficient does not show diminishing returns, even when we use all available fine-tuning data, underscores that 3D VLG models are currently operating in the data scarce regime.

The seeming universality of scaling laws across modalities, and the effectiveness of LIFT-GS pretraining on in 3D VLG settings, suggests that the underlying tools may also be useful at least in other 3D understanding settings. Specifically, render supervised framework can be used with essentially any 3D/4D task or model, provided the results are renderable. This can be combined with knowledge distillation from frame-based models in order to overcome data scarcity in 3D, and LIFT-GS specifically addresses a scarcity of 3D grounded masks. For summary, our contributions are:

- **Differentiable rendering as a tool for training large-scale 3D promptable segmentation models.** Instead of proposing a new architecture, the contribution is to train the model using differentiable rendering with a structured 2D grounding loss.
- **A pseudo-labeling strategy for distilling a 2D vision-language grounding pipeline into 3D versions.** The pretrained 2D models are needed only for pseudo-labeling and they are not needed during inference.
- **State-of-the-art performance and realistic evaluations.** We demonstrate the effectiveness of the approach using sensor pointclouds, common in embodied settings. Rigorous experiments show state-of-the-art performance and reveal scaling properties.

2. Related Work

3D Vision-Language Grounding 3D Vision-Language Grounding (3D VLG) refers to the task of mapping language descriptions of objects in a scene, to a set of corresponding 3D masks or bounding boxes that indicate the location of the objects in the observed scene. Despite its importance, there is very little annotated data for 3D VLG available — mainly because there are very little 3D data with annotated masks or bounding boxes period, as well as language annotation for each instance. This provides a major challenge for essentially all existing models, since they

[The] [telephone] [that] [is] [besides] [the] [chair]

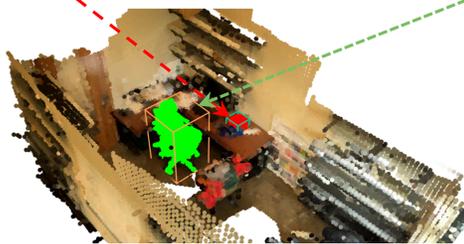


Figure 1: **Illustration of 3D Referential Grounding Task.** Given an object description, the task requires the model to predict 3D masks for all mentioned noun phrases while ensuring that the located objects satisfy the semantic relationships described in the entire sentence.

require access to use ground-truth 3D masks or bounding boxes for trainings (Yuan et al., 2021; Roh et al., 2021; Yang et al., 2021; Zhu et al., 2023c; 2024), and many require them during inference in the form of proposal masks or bounding boxes (Fang et al., 2024; Zhang et al., 2023b). This lack of annotated data is a major challenge for all existing models, as the number of 3D scenes with annotated masks or 3D bounding boxes is in the thousands (Dai et al., 2017; Yeshwanth et al., 2023; Somasundaram et al., 2023), compared to the millions of images and videos used for image and video segmentation (Kirillov et al., 2023).

Unlike prior approaches, LIFT-GS introduces a render-supervised approach for 3D VLG. It uses differentiable rendering to train directly on image masks, without requiring 3D bounding boxes or masks during training or inference. This formulation is architecture agnostic and allows the model to be trained using only image and language losses, without the need for 3D masks or bounding boxes. LIFT-GS demonstrates this by training an end-to-end 3D feedforward model. LIFT-GS takes as input a RGB pointcloud and a language utterance as input, and outputs 3D masks.

3D Instance Segmentation 3D instance segmentation refers to the task of predicting masks for a set of instance in a 3D scene, not necessarily given a language prompt. For example, using a predefined set of category labels (Qian et al., 2021; Schult et al., 2022; Hou et al., 2018) or interactive segmentation using input “prompts” (Osep et al., 2024; Zhou et al., 2024; Ma et al., 2024; Chen et al., 2024a).

Because of the lack of 3D data with masks with which to train a 3D segmentation model, a class of recent work has emerged that trains no predictive 3D segmentation model at all, but instead directly lifts powerful pretrained image and video models to 3D using *per-scene optimization and multiview geometry*. For example, by using depth-unprojection with heuristic view-merging strategies such as voxel-voting (Jatavallabhula et al., 2023a; Zhou et al., 2024; Xu et al., 2023b). Another more recent approach to lift 2D segmentation to 3D is by using differentiable rendering to

optimize the 3D representations to minimize the 2D loss between rendered outputs and ground-truth RGB values, 2D features (Kerr et al., 2023; Kim et al., 2024; Dou et al., 2024; Chen et al., 2024b; Gu et al., 2024), or masks (Cen et al., 2023; Xu et al., 2023a). These approaches leverage the large-scale data available for image and video segmentation, but use a fixed per-scene lifting process that does not improve with more training data and takes a long time to run (e.g. minutes per scene on a H100).

These lifted 3D pseudolabels can be used to partially address the lack of 3D data with ground-truth masks, by pre-training a 3D instance segmentation model on the 3D pseudolabels (Genova et al., 2021; Peng et al., 2023). However, this fixed lifting pipeline introduces accumulated errors from inaccuracies in 3D reconstruction and label-merging strategies, creating a bottleneck for overall performance.

LIFT-GS instead trains a promptable 3D segmentation model using only 2D losses, eliminating the need for a fixed lifting process and label-merging strategies. Not relying on heuristic lifting methods or specialized losses (e.g., cross-mask contrastive loss), LIFT-GS is trained on a large-scale dataset using the same mask losses (*Dice*, *Cross-Entropy Focal*) as the original frame-level segmentation models. As shown in Appendix Section ??, LIFT-GS features trained directly on 2D pseudolabels outperform these lifted 3D pseudolabels. And we show effective data scaling in Section 4.4.

Render-Supervised 3D While photometric losses using differentiable rendering (Mildenhall et al., 2020; Kerbl et al., 2023) were originally used for per-scene optimization, *render-supervision* is emerging as a powerful and generic technique to train predictive 3D models using only 2D losses. For 3D reconstruction, 2D photometric losses can be used to train strong feedforward models (Hong et al., 2024; Tang et al., 2024). PonderV2 (Zhu et al., 2023a) instead uses render-supervision for representation learning, by adding additional contrastive losses on the rendered pixels, using ground-truth category labels to pretrain an *Encoder*.

LIFT-GS introduces render-supervision for 3D VLG, using mask- and grounding-losses typically used for 2D VLG. LIFT-GS jointly trains an *Encoder* and *Mask Decoder*, and for the first time demonstrates using differentiable rendering to train a 3D mask decoder.

3. Method

LIFT-GS is a (pre)training pipeline for 3D vision-language grounding models that uses only 2D supervision. It accomplishes this using differentiable rendering and structured 2D losses, and uses pseudo-labeling for frame annotations.

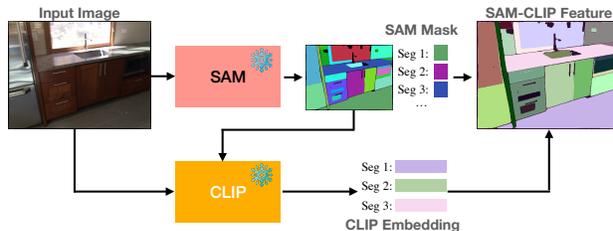


Figure 2: **SAM-CLIP Pseudo-Label Generation.** We leverage powerful 2D foundation models to generate *pseudo language queries*, i.e., CLIP embeddings, along with their corresponding ground-truth 2D masks for training. All pixels within the same mask share the same features.

3.1. Task Formulation

The formulation of 3D Vision-Language Grounding used in LIFT-GS is shown in Figure 1, and based on MDETR (Kamath et al., 2021). The model takes as input a language query, such as “the black chair close to the table near the wall” and a pointcloud input. The objective is to predict 3D gaussian masks, for all mentioned noun phrases (chair, table, wall); ensuring that the predicted masks adhere to the semantic relationships described in the language query, via a mask-to-query correspondence matrix.

As shown in Figure 3, LIFT-GS takes as input a point cloud, \mathbf{P} , and language query embeddings, \mathbf{Q} . It outputs 3D feature Gaussians \mathbf{G} with a correspondence matrix \mathbf{C} .

$$\text{LIFT-GS: } (\mathbf{P}, \mathbf{Q}) \mapsto (\mathbf{G}, \mathbf{C}), \quad (1)$$

The pointcloud $\mathbf{P} \in \mathbb{R}^{|\mathbf{P}| \times 6}$ contains positions and RGB colors. LIFT-GS is trained with pointclouds from unprojected posed multi-view RGBD, but any pointcloud reconstruction will work, including pointmap-based methods that do not require depth or camera pose (Wang et al., 2023; Yang et al., 2025; Wang et al., 2025). This makes the method highly flexible, applicable across diverse scenarios, from sparse-view settings to extremely long video sequences.

The Language queries are encoded into embeddings $\mathbf{Q} \in \mathbb{R}^{|\mathbf{Q}| \times F_Q}$, i.e. via CLIP (Radford et al., 2021) in our design. Each embedding can be either from a single word or a full sentence, depending on the desired granularity of grounding.

The model predicts 3D Gaussians $\mathbf{G} \in \mathbb{R}^{|\mathbf{G}| \times |F_G|}$ with auxiliary features, enabling the differentiable rendering of 3D information into 2D. In LIFT-GS, the first part of channels of F_G are standard Gaussian splatting parameters (i.e. positions μ , covariances Σ , 3rd-degree SH coefficients ψ). The remaining channels are renderable *auxiliary 3D attributes*: predicted 3D masks $\mathbf{M} \in \mathbb{R}^{|\mathbf{G}| \times m}$, and renderable *auxiliary features*: predicted features $\mathbf{F} \in \mathbb{R}^{|\mathbf{G}| \times F}$.

Following the trend in mask grounding (Liu et al., 2023), the predicted 3D masks $\mathbf{M} \in \mathbb{R}^{|\mathbf{G}| \times m}$ contains m predicted

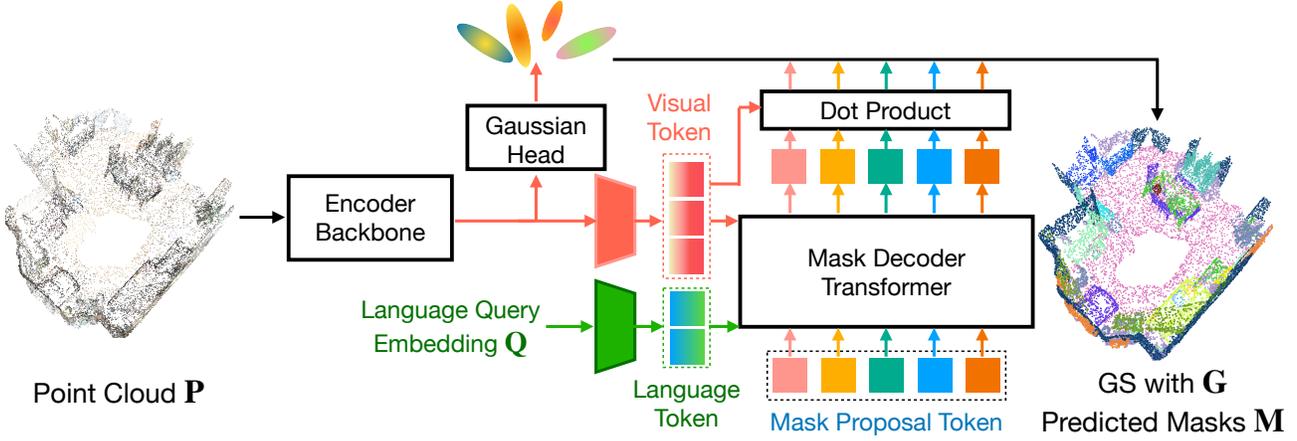


Figure 3: **Architecture Design.** LIFT-GS predicts 3D Gaussian Splatting \mathbf{G} and 3D masks \mathbf{M} given a point cloud \mathbf{P} and language query embeddings \mathbf{Q} as inputs. The 3D masks \mathbf{M} are generated by a Transformer-based Mask Decoder.

masks for each Gaussian, where m is the mask proposal number. Consequently, the rows in the correspondence matrix $\mathbf{C} \in \mathbb{R}^{m \times |Q|}$ represent the log-probability that the i -th predicted mask corresponds to each embedding of \mathbf{Q} .

3.2. Training 3D with 2D Supervision

Predicting a 3D Gaussian Splatting with desired 3D attributes enables training a 3D model using only 2D supervision, eliminating the need for large-scale 3D annotations. In this section, we first discuss the 2D losses used for training and then describe the process of generating pseudo-labels when perfect labels are unavailable. We show ablations for both losses and pseudo-labels in the experiments.

LIFT-GS utilizes three 2D losses to train the model: \mathcal{L}_{RGB} , $\mathcal{L}_{\text{feat}}$ and $\mathcal{L}_{\text{ground}}$. The rendered image, feature map, and 2D mask are denoted in the equations below as $\tilde{I} \in \mathbb{R}^{H \times W \times 3}$, $\tilde{\mathbf{F}}_{2\text{D}} \in \mathbb{R}^{H \times W \times F}$, and $\tilde{\mathbf{M}}_{2\text{D}} \in \mathbb{R}^{H \times W \times m}$, respectively. Their corresponding ground-truth counterparts are I , $\mathbf{F}_{2\text{D}}$, and $\mathbf{M}_{2\text{D}}$, where $\mathbf{M}_{2\text{D}} \in \mathbb{R}^{H \times W \times K}$.

Reconstruction losses: \mathcal{L}_{RGB} is the 2D photometric loss, formulated as a combination of \mathcal{L}_1 loss and SSIM loss (Wang et al., 2004):

$$\mathcal{L}_{\text{RGB}} = \lambda_1 \mathcal{L}_1(I, \tilde{I}) + \lambda_2 \mathcal{L}_{\text{SSIM}}(I, \tilde{I}) \quad (2)$$

Grounding losses: The grounding losses, $\mathcal{L}_{\text{mask}}$, measure the prediction quality between the rendered 2D masks, $\tilde{\mathbf{M}}_{2\text{D}}$, and the ground-truth masks, $\mathbf{M}_{2\text{D}}$. Since the predicted and ground-truth masks differ in number, we employ Hungarian matching to compute the set-to-set loss. $\sigma(i)$ is the index of predicted masks matching to i -th ground truth mask according to the matching distance $\mathbf{d}_{\text{match}}$.

$$\sigma(i) = \arg \min_j \mathbf{d}_{\text{match}}(\tilde{\mathbf{M}}, \mathbf{M}_i, \mathbf{C}) \quad (3)$$

The mask shape is supervised with $\mathcal{L}_{\text{mask}}$, the combination of *Focal* (Lin et al., 2017) and *Dice* (Sudre et al., 2017) loss used in SAM (Kirillov et al., 2023):

$$\mathcal{L}_{\text{ground}} = \frac{1}{K} \sum_i \lambda_3 \mathcal{L}_{\text{mask}}(\tilde{\mathbf{M}}_{2\text{D}}^{\sigma(i)}, \mathbf{M}_{2\text{D}}^i) + \lambda_4 \mathcal{L}_{\text{CE}}(\mathbf{C}_{\sigma(i)}, i) \quad (4)$$

\mathcal{L}_{CE} is the correspondence loss, encouraging correct correspondences between predicted masks and language tokens. Specifically, the ground-truth mask set $\mathbf{M}_{2\text{D}}$ consists of K binary masks, each corresponding to a span $\mathbf{Q}_k \subseteq \mathbf{Q}$ of language query embeddings \mathbf{Q} . $\mathbf{C}_{\sigma(i)}$ represents the correspondence probability of the $\sigma(i)$ -th mask proposal across spans of the $|Q|$ language embeddings. We optimize this using a cross-entropy loss.

$$\mathcal{L}_{\text{CE}}(\mathbf{C}_{\sigma(i)}, i) = -\log \frac{\exp(\mathbf{C}_{\sigma(i), i})}{\sum_j^K \exp(\mathbf{C}_{\sigma(i), j})} \quad (5)$$

Feature loss: $\mathcal{L}_{\text{feat}}$ is the feature rendering loss which measures the difference between rendered feature map $\tilde{\mathbf{F}}_{2\text{D}}$ and ground-truth feature map $\mathbf{F}_{2\text{D}}$. As in (Zhu et al., 2023b), LIFT-GS uses a contrastive loss.

$$\mathcal{L}_{\text{feat}} = \frac{1}{H \times W} \sum_{u,v} -\log \frac{\exp(\tilde{\mathbf{f}}_{(u,v)} \cdot \mathbf{f}_k)}{\sum_j \exp(\tilde{\mathbf{f}}_{(u,v)} \cdot \mathbf{f}_j)} \quad (6)$$

$\tilde{\mathbf{f}}(u, v)$ is the feature vector of $\tilde{\mathbf{F}}_{2\text{D}}$ at location (u, v) . \mathbf{f}_j denotes a batch of unique feature vectors from \mathbf{F} , and \mathbf{f}_k is the ground-truth feature vector corresponding to $\tilde{\mathbf{f}}(u, v)$.

3.3. SAM-CLIP 2D Pseudo-Label

Although LIFT-GS doesn't need 3D annotations, getting high-quality 2D supervision remains challenging. We show how leveraging 2D foundation models for pseudo-label generation enables reasonable zero-shot performance and significantly enhances downstream tasks after fine-tuning.



Figure 4: **Zero-Shot 3D Segmentation.** Trained using only 2D pseudo-labels, LIFT-GS can localize objects in 3D from real text inputs in a zero-shot manner. From left to right, we visualize the *input point clouds*, *segmented 3D masks* (in yellow), *rendered images from predicted 3DGS*, and *rendered segmentation masks*. Language queries include both high-level abstract concepts (e.g., *white*) and detailed descriptions (e.g., *black cabinet near the wall*).

As shown in Figure 2, we leverage 2D foundation models to generate pseudo-labels, including *pseudo language queries* and corresponding 2D masks. For each image, we apply SAM (Kirillov et al., 2023) to obtain segmentation masks. For each segmented region, we extract a CLIP (Radford et al., 2021) image embedding as the *pseudo language query embedding*. Since CLIP’s text and image embeddings share the same feature space, LIFT-GS can take text embeddings as input during inference. During pretraining, we concatenate these CLIP embeddings to form \mathbf{Q} and construct \mathbf{C} , using the corresponding 2D masks as ground truth.

Trained with 2D pseudo-labels, LIFT-GS can perform zero-shot 3D grounding using real text queries without fine-tuning, as shown in Figure 4. However, the zero-shot model suffers from low accuracy and struggles with complex expressions, a common limitation of CLIP-based methods that function as bag-of-words models (Yuksekonul et al., 2023). Future improvements in pseudo-labeling, such as captioning (Meta AI, 2024) and 2D language grounding models (Liu et al., 2023), could alleviate the need for fine-tuning altogether. In this work, we focus on fine-tuning.

Overall, LIFT-GS shows that even simple pseudo-labeling strategies can be effectively distilled into 3D models. Our experiments show that pretraining with 2D pseudo-labels substantially boost performance on downstream tasks.

3.4. Architecture

LIFT-GS is network-agnostic, imposing few architectural constraints and being readily adapted to other architectures. We describe the network used in experiments, as in Figure 3.

Encoder: Predicting 3DGS from Point Cloud

Encoder takes point clouds as input and predicts *3D Gaussian splatting* (without masks) and *per-point features* as latent variables. Generally, the number of predicted Gaus-

sians $|G|$ can differ from the number of input points $|P|$.

However, our focus is on demonstrating the effectiveness of differentiable rendering for structured prediction tasks like *Promptable 3D segmentation*. So in our experiments, we set $|G| = |P|$ with a bijective mapping to ensure consistency with point cloud-based evaluation tasks. For comparison to prior work (Zhu et al., 2023b), LIFT-GS uses a SparseConv UNet (Contributors, 2022) as the *Encoder Backbone* to get *per-point features*, and uses an MLP as *Gaussian Head* to regress Gaussian parameters of \mathbf{G} with features \mathbf{F} .

Decoder: Mask Decoder for 3D VLG

The *Decoder* takes *per-point features* from the *Encoder* and language embeddings \mathbf{Q} as input to predict the 3D mask $\mathbf{M} \in \mathbb{R}^{N \times m}$ and the correspondence matrix $\mathbf{C} \in \mathbb{R}^{m \times |Q|}$. The predicted \mathbf{M} serves as a renderable feature of the 3DGS \mathbf{G} and is subsequently used to generate 2D masks. While this work focuses on 3D vision-language grounding (3D VLG), the proposed framework is a general pipeline that can be extended to other 3D tasks with renderable outputs.

We adopt a 3D variant of the Transformer-based mask decoder in MaskFormer (Cheng et al., 2021a; Jain et al., 2025), inspired by the success of MDETR (Kamath et al., 2021) and BUTD (Jain et al., 2021). As illustrated in Figure 3, the 3D Mask Decoder Transformer takes three inputs: *visual tokens* from *per-point features*, *language tokens* from \mathbf{Q} , and m learnable *mask proposal tokens*. It outputs m predicted *mask embeddings* and a correspondence matrix \mathbf{C} . The 3D masks \mathbf{M} are computed as the dot product between *visual tokens* and output *mask embeddings*. These similarity scores are then differentially rendered into mask images \mathbf{M}_{2D} .

4. Experiments

Table 1: **Open-Vocabulary 3D Instance Segmentation.** We evaluate our model on ScanNet200 by using category names as text queries and compare it against SOTA models.

Model	mAP \uparrow	mAP25 \uparrow	mAP50 \uparrow
OpenScene (Peng et al., 2023)	11.7	17.8	15.2
OpenMask3D (Takmaz et al., 2023)	15.4	23.1	19.9
PQ3D (Zhu et al., 2024)	20.2	32.5	28.0
LIFT-GS-Scratch	22.5	35.1	30.7
LIFT-GS	25.7	40.2	35.0
Δ	+3.2 \uparrow	+5.1 \uparrow	+4.3 \uparrow

In this section, we first provide training details and demonstrate how pretraining significantly enhances downstream task performance through a set of carefully designed ablations. Additionally, we uncover several intriguing findings by scaling pretraining and fine-tuning data, as well as exploring the impact of different 2D foundation models.

4.1. Training Details

We provide details below with more details in Appendix.

Training Datasets We use ScanNet (Dai et al., 2017) as the primary dataset for downstream task fine-tuning and evaluation, as its annotations form the basis for established benchmarks. Our method’s flexibility enables training on diverse 3D datasets without requiring language annotations or instance masks. For pretraining, we primarily use ScanNet (Dai et al., 2017) and ScanNet++(Yeshwanth et al., 2023) and also explore Taskonomy(Zamir et al., 2018) and Aria Synthetic (Somasundaram et al., 2023).

Architecture Our pipeline imposes no constraints on the 3D architecture or inductive biases, allowing flexibility in model selection. For experiments, we use SparseConv UNet as the encoder backbone, a widely adopted architecture for point cloud processing. The Mask Decoder is an 8-layer Transformer decoder with a hidden state size of 512.

Training Details We train the *Encoder* and *Decoder* jointly using a batch size of 32 for 76k steps on 32 A100 GPUs. We use AdamW (Loshchilov & Hutter, 2017) optimizer with a learning rate of $1e-4$ and weight decay of $1e-4$.

4.2. Evaluation on 3D Vision-Language Grounding

We fine-tune and evaluate our pretrained model on two representative 3D VLG tasks: 3D open-vocabulary instance segmentation and 3D referential grounding. Our results, shown in Tables 1 and 2, demonstrate significant improvements over models trained from scratch and achieve state-of-the-art performance with pertaining.

4.2.1. GROUNDING SIMPLE NOUNS IN 3D

We first evaluate simple grounding for simple noun-phrases, using object categories without spatial relationships. Following the protocol in (Zhu et al., 2024), we convert the standard 3D instance segmentation benchmark on ScanNet into an open-vocabulary 3D instance segmentation task. The categories of objects are used as language queries, which are input to the model to predict the corresponding 3D masks.

Evaluation setting: We evaluate using the standard metric mAP, a measure of mask overlap averaged across categories. We fine-tune LIFT-GS for 500 epochs.

Results: Compared against the state-of-the-art baselines PQ3D (Zhu et al., 2024) and OpenMask3D (Takmaz et al., 2023), our pretrained model (LIFT-GS) achieves substantial performance gains (mAP 25.7% vs 20.2%), as shown in Table 1. It significantly outperforms its counterpart trained from scratch (LIFT-GS-Scratch mAP +3.2%).

4.2.2. GROUNDING COMPLEX PHRASES IN 3D

Next, we examine grounding multiple objects using more complex phrases that contain spatial references, referred to as *3D Referential Grounding* (3D RG).

Evaluation Setting. We evaluate LIFT-GS on the most common *3D Referential Grounding* benchmarks: ScanRefer (Chen et al., 2019), SR3D, and NR3D (Achlioptas et al., 2020; Abdelreheem et al., 2022). We use standard top-1 accuracy as the evaluation metric, considering a predicted bounding box correct if its IoU with the ground truth exceeds 0.25 or 0.5. Since LIFT-GS outputs masks instead of axis-aligned bounding boxes, we derive bounding boxes by extracting the extreme corner points from the point cloud within the predicted masks.

LIFT-GS is designed to be practical. To ensure a more realistic evaluation for embodied applications, we introduce two modifications: (1) we predict 3D masks without assuming the availability of ground-truth 3D bounding boxes, and (2) we utilize sensor point clouds (*Sensor PC*) from RGB-D scans instead of mesh-derived point clouds (*Mesh PC*).

Existing methods are typically trained on mesh-derived point clouds (*Mesh PC*), requiring creating scene meshes from RGB-D scans, processing, and sampling point clouds. They generally rely on *segments* derived from clustering centers generated during human annotation. Since such information is unavailable in real-world applications, we instead use raw sensor point clouds (*Sensor PC*) from RGB-D scans. This setting is inherently more challenging, as reflected in the significant performance drop of BUTD-DETR (Jain et al., 2021) when transitioning from *Mesh PC* to *Sensor PC* (Table 2), consistent with findings in (Jain et al., 2024).

Baselines We compare LIFT-GS against the state-of-the-art two-stage methods, 3D-VisTA (Zhu et al., 2023c) and PQ3D (Zhu et al., 2024), as well as the SOTA single-stage method, BUTD-DETR (Jain et al., 2021). All two-stage baselines assume access to ground-truth 3D masks or boxes during inference, so we re-evaluate them using predicted boxes from the SOTA object detector Mask3D (Schult et al., 2022). For fairness, we re-train 3D-VisTA and BUTD-DETR on sensor point clouds. Because PQ3D uses multiple backbones and a multi-stage training pipeline, we were not able to reproduce PQ3D on the sensor point cloud setting.

Results Our model without pretraining (LIFT-GS-Scratch) achieves slightly better performance than the state-of-the-art single-stage method BUTD-DETR (Jain et al., 2021), likely due to architectural similarities with extra modifications.

With pretraining, LIFT-GS achieves significant improvements across all three datasets, with relative gains of 10% – 30%, demonstrating the effectiveness of our pretraining approach. Notably, LIFT-GS outperforms 3D-VisTA in

Table 2: **3D Referential Grounding.** We report top-1 accuracy with various IoU thresholds (0.25, 0.5).

Method	SR3D		NR3D		ScanRefer	
	Acc@25	Acc@50	Acc@25	Acc@50	Acc@25	Acc@50
Mesh PC						
LanguageRefer (Roh et al., 2021)	39.5	-	28.6	-	-	-
SAT-2D (Yang et al., 2021)	35.4	-	31.7	-	44.5	30.1
BUTD-DETR (Jain et al., 2021)	52.1	-	43.3	-	52.2	39.8
3D-VisTA (Zhu et al., 2023c)	56.5	51.5	47.7	42.2	51.0	46.2
PQ3D (Zhu et al., 2024)	62.0	55.9	52.2	45.0	56.7	51.8
Sensor PC + Bounding Box Proposals using Mesh PC						
3D-VisTA (Zhu et al., 2023c)	47.2	43.2	42.1	37.4	46.4	42.5
Sensor PC						
BUTD-DETR (Jain et al., 2021)	43.3	28.9	32.2	19.4	42.2	27.9
LIFT-GS-Scratch	44.0	28.8	37.2	23.1	45.0	29.5
LIFT-GS	50.9	36.5	43.7	29.7	49.7	36.4
Δ	+6.9(16%)	+7.7(27%)	+6.5(17%)	+6.6(29%)	+4.7(10%)	+6.9(23%)

Table 3: **Comparison with other Pretraining Baseline.** LIFT-GS clearly outperforms Ponder-v2 and its variant Ponder-v2 \dagger , which is trained on the same SAM-CLIP features as ours.

Model	Acc@0.25	Acc@0.5	Acc@0.75
Scratch	42.19	27.23	9.66
Ponder-v2 (official)	40.92	25.97	8.84
Ponder-v2 \dagger	45.40	29.36	9.29
LIFT-GS	47.53	33.75	13.49

Acc@25, despite 3D-VisTA being a two-stage method with bounding box proposals from Mask3D using *Mesh PC*.

4.3. Ablation and Analyses

We conduct an in-depth analysis of the proposed method through a series of ablation and scaling experiments. For these evaluations, we use a model pretrained only on ScanNet as the baseline. To simplify the presentation for the ablations, we report results on the combined evaluation set of ScanRefer, SR3D, and NR3D. Additionally, we report the higher accuracy threshold Acc@0.75.

Compare to SOTA pretraining methods We compare to Ponder-v2 (Zhu et al., 2023b), a state-of-the-art method for point cloud pretraining. Ponder-v2 takes point clouds as input and predicts voxel grids, which are then used to render 2D images via NeRF (specifically NeuS, which additionally models surfaces (Wang et al., 2021)). Pretraining is supervised by 2D photometric loss and CLIP features, where features are computed based on per-pixel text labels for the mask (GT category labels and masks for each instance).

We found that the official Ponder-v2 checkpoint fails to improve performance on 3D referential grounding, likely due to the limited size and diversity of its per-pixel text labels. Therefore, we retrain the Ponder-v2 while using the SAM-CLIP pseudolabels described in Section 3.3. Training

Table 4: **Loss Ablation.** We show the impact of different pretraining losses on 3D referential grounding task. $\mathcal{L}_{\text{ground}}$ significantly improves results, particularly at high IoU thresholds.

Model	$\mathcal{L}_{\text{ground}}$	\mathcal{L}_{RGB}	$\mathcal{L}_{\text{feat}}$	Acc@0.25	Acc@0.5	Acc@0.75
Scratch				42.19	27.23	9.66
-	✓			46.34	31.54	<u>12.50</u>
-	✓	✓		46.67	<u>31.81</u>	12.45
-		✓	✓	47.69	31.35	11.36
-	✓	✓	✓	<u>47.53</u>	33.75	13.49

on the pseudolabels (indicated by Ponder-v2 \dagger in Table 3) significantly improves Acc@0.25 +4.5% vs. the official Ponder-v2, and +3.2% vs. training from scratch. Since both approaches use render-supervision, this experiment underscores the advantages of pseudolabel distillation in data-scarce regimes. Moreover our method does not rely on human-annotated per-pixel text labels, making it highly scalable to other datasets, as shown in Table 6.

Loss Ablation Existing pretraining pipelines primarily focus on the encoder (Zhu et al., 2023b; Banani et al., 2021), whereas the render-supervised formulation can pretrain the entire architecture in a unified manner. To assess the impact of the grounding loss, which applies only to the decoder, we compare models pretrained with different losses in Table 4.

A model trained with $\mathcal{L}_{\text{ground}}$ alone (row 2) still substantially improves downstream task performance, performing only slightly worse than the model trained with the full loss (row 5), demonstrating the effectiveness of the grounding loss. Furthermore, comparing models with and without $\mathcal{L}_{\text{ground}}$ (row 5 vs. row 4) clearly shows that $\mathcal{L}_{\text{ground}}$ significantly enhances downstream performance, particularly in more challenging scenarios (IoU thresholds of 0.5 and 0.75).

4.4. Data Scaling

Finetuning Data Scaling We limit the fine-tuning data size to 0.1, 0.2, and 0.5 of the full dataset for 3D referential

Table 5: **Fine-tune Data Scaling.** We show Acc@0.5 results with different ratio of fine-tuning data on referential grounding task.

Finetuning Data Ratio	10%	20%	50%	100%
Scratch	6.93	15.04	23.00	27.23
LIFT-GS	14.70	23.03	28.89	33.75

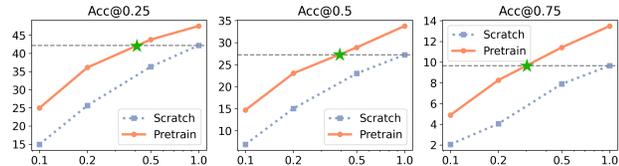


Figure 5: **Fine-tune Data Scaling.** We show how *Grounding Accuracy* changes with increasing *Data Ratio* from 0.1 to 1.0.

grounding to analyze performance variation with different fine-tuning ratios. The results are presented in Figure 5 and Table 5, with additional findings provided in Appendix ??.

Intriguingly, we find that LIFT-GS “pretraining effectively multiplies the fine-tuning dataset” (Hernandez et al., 2021) by a constant factor (roughly 2x), across varying amounts of fine-tuning data. For instance, the pretrained model fine-tuned on 20% of the data achieves performance comparable to training from scratch with 50% of the data. The benefits of pretraining are even more pronounced at higher IoU thresholds. Overall, a pretrained model can achieve the same performance as a model trained from scratch on all the data using only 30–40% of the fine-tuning data.

This observation matches empirical data scaling laws for pretraining in other modalities (Hernandez et al., 2021). And interestingly, the fact that the coefficient does not show diminishing returns, even when we use all available fine-tuning data, which underscores that 3D VLG models are currently operating in the data scarce regime.

Pretraining Data Scaling A key advantage of our framework is its ability to pretrain on large-scale RGB-D scans without requiring 3D or language annotations, ensuring scalability. We scale the pretraining dataset and analyze its impact on fine-tuning performance, as shown in Table 6.

Scaling the pretraining dataset generally improves performance. Adding ScanNet++ (Yeshwanth et al., 2023) yields a notable 0.8% improvement, despite its limited data size. Incorporating Taskonomy (Zamir et al., 2018) and Aerial Synthetic (Somasundaram et al., 2023) also enhances performance, though the gains are less pronounced due to distributional differences from ScanNet, likely due to the mesh reconstruction quality in Taskonomy as well.

Since our method doesn’t need 3D masks or language annotations, expanding the pretraining dataset with real-world scans is fairly easy. We anticipate continuous improvements as larger-scale data collections become available.

Table 6: **Pretraining on OOD data.** Adding more pretraining data from ScanNet++ improves performance. Taskonomy and Aerial helped less than ScanNet++, likely due to distribution difference.

Pretraining Data	Acc@0.25	Acc@0.5	Acc@0.75
ScanNet	47.53	33.75	13.49
+ScanNet++	48.29	34.35	14.06
++ Taskonomy and Aerial	48.49	34.41	14.35

Table 7: **2D Foundation Model Exploration.**

2D Models	Acc@0.25	Acc@0.5	Acc@0.75
SAM-B + CLIP-B	46.31	31.50	12.41
SAM-H + CLIP-L	47.53	33.75	13.49
SAM-H + LLAMA-Caption	47.50	32.78	13.25

4.5. 2D Foundation Models Scaling and Exploration

Our pipeline leverages powerful 2D foundation models to generate pseudo-labels. Here, we investigate their impact by analyzing performance variations with different 2D foundation models, with results presented in Table 7.

Weaker CLIP and SAM The main experiments use SAM-H and CLIP-L for pseudo-labeling. Replacing them with smaller models, MobileSAM (Zhang et al., 2023a) (ViT-tiny) and CLIP-B, leads to a noticeable performance drop, especially at higher accuracy thresholds. This suggests that our model benefits from advancements in 2D foundation models, highlighting the importance of stronger 2D models.

Captions from VLMs We explore an alternative pseudo-labeling strategy by combining SAM with the LLAMA-V model (details in Appendix). After segmenting objects in 2D images using SAM, we prompt LLAMA-V to describe the segmented regions and encode these descriptions into CLIP embeddings.

Pretraining with this pseudo-labeling method achieves performance comparable to our original pipeline with SAM-H and CLIP-L, highlighting the flexibility and extensibility of our framework. We believe text-based captions hold significant potential for future research, and our pipeline is well-positioned to benefit from advancements in this area.

5. Conclusion

We present LIFT-GS, a model-agnostic pipeline for training 3D VLG models without 3D supervision by leveraging Gaussian Splatting and 2D foundation models. Beyond enabling zero-shot 3D grounding, LIFT-GS significantly improves downstream performance when fine-tuned with limited 3D data. It also scales effectively, benefiting from larger datasets and stronger 2D foundation models, suggesting continuous improvements with increased data. Limitations and future directions are discussed in the Appendix.

References

- Abdelreheem, A., Olszewski, K., Lee, H.-Y., Wonka, P., and Achlioptas, P. Scanents3d: Exploiting phrase-to-3d-object correspondences for improved visio-linguistic models in 3d scenes. *2024 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 3512–3522, 2022. URL <https://api.semanticscholar.org/CorpusID:254591182>.
- Achiam, O. J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F. L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., Avila, R., Babuschkin, I., Balaji, S., Balcom, V., Baltescu, P., ing Bao, H., Bavarian, M., Belgum, J., Bello, I., Berdine, J., Bernadett-Shapiro, G., Berner, C., Bogdonoff, L., Boiko, O., laine Boyd, M., Brakman, A.-L., Brockman, G., Brooks, T., Brundage, M., Button, K., Cai, T., Campbell, R., Cann, A., Carey, B., Carlson, C., Carmichael, R., Chan, B., Chang, C., Chantzis, F., Chen, D., Chen, S., Chen, R., Chen, J., Chen, M., Chess, B., Cho, C., Chu, C., Chung, H. W., Cummings, D., Currier, J., Dai, Y., Decareaux, C., Degry, T., Deutsch, N., Deville, D., Dhar, A., Dohan, D., Dowling, S., Dunning, S., Ecoffet, A., Eleti, A., Eloundou, T., Farhi, D., Fedus, L., Felix, N., Fishman, S. P., Forte, J., abella Fulford, I., Gao, L., Georges, E., Gibson, C., Goel, V., Gogineni, T., Goh, G., Gontijo-Lopes, R., Gordon, J., Grafstein, M., Gray, S., Greene, R., Gross, J., Gu, S. S., Guo, Y., Hallacy, C., Han, J., Harris, J., He, Y., Heaton, M., hannes Heidecke, J., Hesse, C., Hickey, A., Hickey, W., Hoeschele, P., Houghton, B., Hsu, K., Hu, S., Hu, X., Huizinga, J., Jain, S., Jain, S., Jang, J., Jiang, A., Jiang, R., Jin, H., Jin, D., Jomoto, S., Jonn, B., Jun, H., Kaf-tan, T., Kaiser, L., Kamali, A., Kanitscheider, I., Keskar, N. S., Khan, T., Kilpatrick, L., Kim, J. W., Kim, C., Kim, Y., Kirchner, H., Kiros, J. R., Knight, M., Kokotajlo, D., Kondruciuk, L., Kondrich, A., Konstantinidis, A., Kopic, K., Krueger, G., Kuo, V., Lampe, M., Lan, I., Lee, T., Leike, J., Leung, J., Levy, D., Li, C. M., Lim, R., Lin, M., Lin, S., teusz Litwin, M., Lopez, T., Lowe, R., Lue, P., Makanju, A., Malfacini, K., Manning, S., Markov, T., Markovski, Y., Martin, B., Mayer, K., Mayne, A., McGrew, B., McKinney, S. M., McLeavey, C., McMillan, P., McNeil, J., Medina, D., Mehta, A., Menick, J., Metz, L., Mishchenko, A., Mishkin, P., Monaco, V., Morikawa, E., Mossing, D. P., Mu, T., Murati, M., Murk, O., M’ely, D., Nair, A., Nakano, R., Nayak, R., Neelakantan, A., Ngo, R., Noh, H., Long, O., O’Keefe, C., Pachocki, J. W., Paino, A., Palermo, J., Pantuliano, A., Parascandolo, G., Parish, J., Parparita, E., Passos, A., Pavlov, M., Peng, A., Perelman, A., de Avila Belbute Peres, F., Petrov, M., de Oliveira Pinto, H. P., Pokorny, M., Pokrass, M., Pong, V. H., Powell, T., Power, A., Power, B., Proehl, E., Puri, R., Radford, A., Rae, J. W., Ramesh, A., Raymond, C., Real, F., Rimbach, K., Ross, C., Rotsted, B., Roussez, H., Ryder, N., Saltarelli, M. D., Sanders, T., Santurkar, S., Sastry, G., Schmidt, H., Schnurr, D., Schulman, J., Sel-sam, D., Sheppard, K., Sherbakov, T., Shieh, J., Shoker, S., Shyam, P., Sidor, S., Sigler, E., Simens, M., Sitkin, J., Slama, K., Sohl, I., Sokolowsky, B., Song, Y., Stau-dacher, N., Such, F. P., Summers, N., Sutskever, I., Tang, J., Tezak, N. A., Thompson, M., Tillet, P., Tootoonchian, A., Tseng, E., Tuggle, P., Turley, N., Tworek, J., Uribe, J. F. C., Vallone, A., Vijayvergiya, A., Voss, C., Wainwright, C. L., Wang, J. J., Wang, A., Wang, B., Ward, J., Wei, J., Weinmann, C., Welihinda, A., Welinder, P., Weng, J., Weng, L., Wiethoff, M., Willner, D., Winter, C., Wolrich, S., Wong, H., Workman, L., Wu, S., Wu, J., Wu, M., Xiao, K., Xu, T., Yoo, S., Yu, K., ing Yuan, Q., Zaremba, W., Zellers, R., Zhang, C., Zhang, M., Zhao, S., Zheng, T., Zhuang, J., Zhuk, W., and Zoph, B. Gpt-4 technical report. 2023. URL <https://api.semanticscholar.org/CorpusID:257532815>.
- Achlioptas, P., Abdelreheem, A., Xia, F., Elhoseiny, M., and Guibas, L. J. Referit3d: Neural listeners for fine-grained 3d object identification in real-world scenes. In *European Conference on Computer Vision*, 2020. URL <https://api.semanticscholar.org/CorpusID:221378802>.
- Banani, M. E., Gao, L., and Johnson, J. Unsupervisedr&r: Unsupervised point cloud registration via differentiable rendering. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7125–7135, 2021. URL <https://api.semanticscholar.org/CorpusID:232014069>.
- Brooks, T., Peebles, B., Holmes, C., DePue, W., Guo, Y., Jing, L., Schnurr, D., Taylor, J., Luhman, T., Luhman, E., Ng, C., Wang, R., and Ramesh, A. Video generation models as world simulators. 2024. URL <https://openai.com/research/video-generation-models-as-world-simulators>.
- Cen, J., Zhou, Z., Fang, J., Shen, W., Xie, L., Jiang, D., Zhang, X., and Tian, Q. Segment anything in 3d with nerfs. *ArXiv*, abs/2304.12308, 2023. URL <https://api.semanticscholar.org/CorpusID:271201271>.
- Chen, D. Z., Chang, A. X., and Nießner, M. Scanrefer: 3d object localization in rgb-d scans using natural language. *ArXiv*, abs/1912.08830, 2019. URL <https://api.semanticscholar.org/CorpusID:209414687>.
- Chen, M., Shapovalov, R., Laina, I., Monnier, T., Wang, J., Novotný, D., and Vedaldi, A. Partgen: Part-level 3d generation and reconstruction with

- multi-view diffusion models. 2024a. URL <https://api.semanticscholar.org/CorpusID:274992225>.
- Chen, Z., Gebru, I. D., Richardt, C., Kumar, A., Laney, W., Owens, A., and Richard, A. Real acoustic fields: An audio-visual room acoustics dataset and benchmark. *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21886–21896, 2024b. URL <https://api.semanticscholar.org/CorpusID:268723955>.
- Cheng, B., Schwing, A. G., and Kirillov, A. Per-pixel classification is not all you need for semantic segmentation. In *Neural Information Processing Systems*, 2021a. URL <https://api.semanticscholar.org/CorpusID:235829267>.
- Cheng, B., Schwing, A. G., and Kirillov, A. Per-pixel classification is not all you need for semantic segmentation, 2021b. URL <https://arxiv.org/abs/2107.06278>.
- Contributors, S. Spconv: Spatially sparse convolution library. <https://github.com/traveller59/spconv>, 2022.
- Dai, A., Chang, A. X., Savva, M., Halber, M., Funkhouser, T., and Nießner, M. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 2017.
- Dou, Y., Yang, F., Liu, Y., Loquercio, A., and Owens, A. Tactile-augmented radiance fields. *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 26519–26529, 2024. URL <https://api.semanticscholar.org/CorpusID:269614492>.
- Fang, J., Tan, X., Lin, S., Vasiljevic, I., Guizilini, V. C., Mei, H., Ambrus, R., Shakhnarovich, G., and Walter, M. R. Transcrib3d: 3d referring expression resolution through large language models. *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 9737–9744, 2024. URL <https://api.semanticscholar.org/CorpusID:269457175>.
- Genova, K., Yin, X., Kundu, A., Pantofaru, C., Cole, F., Sud, A., Brewington, B., Shucker, B., and Funkhouser, T. Learning 3d semantic segmentation with only 2d image supervision. In *2021 International Conference on 3D Vision (3DV)*, pp. 361–372, 2021. doi: 10.1109/3DV53792.2021.00046.
- Gu, Q., Lv, Z., Frost, D., Green, S., Straub, J., and Sweeney, C. Egolifter: Open-world 3d segmentation for egocentric perception. *arXiv preprint arXiv:2403.18118*, 2024.
- Hernandez, D., Brown, T., Greenwald, E., Kaplan, J., Abbeel, P., and McCandlish, S. Scaling laws for transfer. *arXiv preprint arXiv:2102.01293*, 2021.
- Hong, Y., Zhang, K., Gu, J., Bi, S., Zhou, Y., Liu, D., Liu, F., Sunkavalli, K., Bui, T., and Tan, H. Lrm: Large reconstruction model for single image to 3d. In *International Conference on Learning Representations (ICLR)*, 2024.
- Hou, J., Dai, A., and Nießner, M. 3d-sis: 3d semantic instance segmentation of rgb-d scans. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4416–4425, 2018. URL <https://api.semanticscholar.org/CorpusID:56171922>.
- Jain, A., Gkanatsios, N., Mediratta, I., and Fragkiadaki, K. Bottom up top down detection transformers for language grounding in images and point clouds. *ArXiv*, abs/2112.08879, 2021. URL <https://api.semanticscholar.org/CorpusID:250921818>.
- Jain, A., Katara, P., Gkanatsios, N., Harley, A. W., Sarch, G. H., Aggarwal, K., Chaudhary, V., and Fragkiadaki, K. Odin: A single model for 2d and 3d segmentation. *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3564–3574, 2024. URL <https://api.semanticscholar.org/CorpusID:266756014>.
- Jain, A., Swerdlow, A., Wang, Y., Sax, A., Meier, F., and Fragkiadaki, K. RODIN: Injecting 2d foundational features to 3d vision language understanding, 2025. URL <https://openreview.net/forum?id=Pt31fU1NqC>.
- Jatavallabhula, K., Kuwajerwala, A., Gu, Q., Omama, M., Chen, T., Li, S., Iyer, G., Saryazdi, S., Keetha, N., Tewari, A., Tenenbaum, J., de Melo, C., Krishna, M., Paull, L., Shkurti, F., and Torralba, A. Conceptfusion: Open-set multimodal 3d mapping. *Robotics: Science and Systems (RSS)*, 2023a.
- Jatavallabhula, K. M., Kuwajerwala, A., Gu, Q., Omama, M., Chen, T., Li, S., Iyer, G., Saryazdi, S., Keetha, N. V., Tewari, A. K., Tenenbaum, J. B., de Melo, C. M., Krishna, M., Paull, L., Shkurti, F., and Torralba, A. Conceptfusion: Open-set multimodal 3d mapping. *ArXiv*, abs/2302.07241, 2023b. URL <https://api.semanticscholar.org/CorpusID:256846496>.
- Kamath, A., Singh, M., LeCun, Y., Misra, I., Synnaeve, G., and Carion, N. Mdetr - modulated detection for end-to-end multi-modal understanding. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 1760–1770,

2021. URL <https://api.semanticscholar.org/CorpusID:233393962>.
- Kerbl, B., Kopanas, G., Leimkuehler, T., and Drettakis, G. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics (TOG)*, 42:1–14, 2023. URL <https://api.semanticscholar.org/CorpusID:259267917>.
- Kerr, J., Kim, C. M., Goldberg, K., Kanazawa, A., and Tancik, M. Lerf: Language embedded radiance fields. In *International Conference on Computer Vision (ICCV)*, 2023.
- Kim, C. M., Wu, M., Kerr, J., Tancik, M., Goldberg, K., and Kanazawa, A. Garfield: Group anything with radiance fields. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.
- Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A. C., Lo, W.-Y., Dollar, P., and Girshick, R. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4015–4026, October 2023.
- Labs, B. F. Flux. <https://github.com/black-forest-labs/flux>, 2023.
- Lin, T.-Y., Goyal, P., Girshick, R. B., He, K., and Dollár, P. Focal loss for dense object detection. *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 2999–3007, 2017. URL <https://api.semanticscholar.org/CorpusID:47252984>.
- Liu, S., Zeng, Z., Ren, T., Li, F., Zhang, H., Yang, J., Li, C., Yang, J., Su, H., Zhu, J., et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. *arXiv preprint arXiv:2303.05499*, 2023.
- Loshchilov, I. and Hutter, F. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2017. URL <https://api.semanticscholar.org/CorpusID:53592270>.
- Ma, Z., Yue, Y., and Gkioxari, G. Find any part in 3d. *ArXiv*, abs/2411.13550, 2024. URL <https://api.semanticscholar.org/CorpusID:274149857>.
- Meta AI. Llama 3: The llama 3 herd of models. <https://ai.meta.com/llama/>, 2024. [Large language model].
- Mildenhall, B., Srinivasan, P. P., Tancik, M., Barron, J. T., Ramamoorthi, R., and Ng, R. Nerf: Representing scenes as neural radiance fields for view synthesis. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020.
- Osep, A., Meinhardt, T., Ferroni, F., Peri, N., Ramanan, D., and Leal-Taixé, L. Better call sal: Towards learning to segment anything in lidar. In *European Conference on Computer Vision (ECCV)*, 2024.
- Peng, S., Genova, K., Jiang, C. M., Tagliasacchi, A., Pollefeys, M., and Funkhouser, T. Openscene: 3d scene understanding with open vocabularies. In *CVPR*, 2023.
- Polyak, A., Zohar, A., Brown, A., Tjandra, A., Sinha, A., Lee, A., Vyas, A., Shi, B., Ma, C.-Y., Chuang, C.-Y., Yan, D., Choudhary, D., Wang, D., Sethi, G., Pang, G., Ma, H., Misra, I., Hou, J., Wang, J., ran Jagadeesh, K., Li, K., Zhang, L., Singh, M., Williamson, M., Le, M., Yu, M., Singh, M. K., Zhang, P., Vajda, P., Duval, Q., Girdhar, R., Sumbaly, R., Rambhatla, S. S., Tsai, S. S., Azadi, S., Datta, S., Chen, S., Bell, S., Ramaswamy, S., Sheynin, S., Bhattacharya, S., Motwani, S., Xu, T., Li, T., Hou, T., Hsu, W.-N., Yin, X., Dai, X., Taigman, Y., Luo, Y., Liu, Y.-C., Wu, Y.-C., Zhao, Y., Kirstain, Y., He, Z., He, Z., Pumarola, A., Thabet, A. K., Sanakoyeu, A., Mallya, A., Guo, B., Araya, B., Kerr, B., Wood, C., Liu, C., Peng, C., Vengertsev, D., Schonfeld, E., Blanchard, E., Juefei-Xu, F., Nord, F., Liang, J., Hoffman, J., Kohler, J., Fire, K., Sivakumar, K., Chen, L., Yu, L., Gao, L., Georgopoulos, M., Moritz, R., Sampson, S. K., Li, S., Parmeggiani, S., Fine, S., Fowler, T., Petrovic, V., and Du, Y. Movie gen: A cast of media foundation models. *ArXiv*, abs/2410.13720, 2024. URL <https://api.semanticscholar.org/CorpusID:273403698>.
- Qian, S., Kirillov, A., Ravi, N., Chaplot, D. S., Johnson, J., Fouhey, D. F., and Gkioxari, G. Recognizing scenes from novel viewpoints. *ArXiv*, abs/2112.01520, 2021. URL <https://api.semanticscholar.org/CorpusID:244799191>.
- Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., and Sutskever, I. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, 2021. URL <https://api.semanticscholar.org/CorpusID:231591445>.
- Ravi, N., Gabeur, V., Hu, Y.-T., Hu, R., Ryali, C., Ma, T., Khedr, H., Rädle, R., Rolland, C., Gustafson, L., Mintun, E., Pan, J., Alwala, K. V., Carion, N., Wu, C.-Y., Girshick, R., Dollár, P., and Feichtenhofer, C. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024. URL <https://arxiv.org/abs/2408.00714>.
- Roh, J., Desingh, K., Farhadi, A., and Fox, D. Language-refer: Spatial-language model for 3d visual grounding. *ArXiv*, abs/2107.03438, 2021. URL <https://arxiv.org/abs/2107.03438>.

- 605 //api.semanticscholar.org/CorpusID:
606 235765540.
- 607 Schult, J., Engelmann, F., Hermans, A., Litany, O., Tang, S.,
608 and Leibe, B. Mask3d: Mask transformer for 3d semantic
609 instance segmentation. *2023 IEEE International Confer-*
610 *ence on Robotics and Automation (ICRA)*, pp. 8216–8223,
611 2022. URL <https://api.semanticscholar.org/CorpusID:258079403>.
- 614 Somasundaram, K. K., Dong, J., Tang, H., Straub, J., Yan,
615 M., Goesele, M., Engel, J. J., Nardi, R. D., and New-
616 combe, R. A. Project aria: A new tool for egocen-
617 tric multi-modal ai research. *ArXiv*, abs/2308.13561,
618 2023. URL <https://api.semanticscholar.org/CorpusID:261243365>.
- 620 Sudre, C. H., Li, W., Vercauteren, T. K. M., Ourselin, S.,
621 and Cardoso, M. J. Generalised dice overlap as a deep
622 learning loss function for highly unbalanced segmen-
623 tations. *Deep learning in medical image analysis and mul-*
624 *timodal learning for clinical decision support : Third*
625 *International Workshop, DLMIA 2017, and 7th Inter-*
626 *national Workshop, ML-CDS 2017, held in conjunc-*
627 *tion with MICCAI 2017 Quebec City, QC,...*, 2017:240–248,
628 2017. URL <https://api.semanticscholar.org/CorpusID:21957663>.
- 631 Takmaz, A., Fedele, E., Sumner, R. W., Pollefeys, M.,
632 Tombari, F., and Engelmann, F. OpenMask3D: Open-
633 Vocabulary 3D Instance Segmentation. In *Advances in*
634 *Neural Information Processing Systems (NeurIPS)*, 2023.
- 635 Tang, J., Chen, Z., Chen, X., Wang, T., Zeng, G., and
636 Liu, Z. Lgm: Large multi-view gaussian model for
637 high-resolution 3d content creation. In *European*
638 *Conference on Computer Vision*, 2024. URL <https://api.semanticscholar.org/CorpusID:267523413>.
- 642 Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux,
643 M.-A., Lacroix, T., Rozière, B., Goyal, N., Ham-
644 bro, E., Azhar, F., Rodriguez, A., Joulin, A., Grave,
645 E., and Lample, G. Llama: Open and efficient
646 foundation language models. *ArXiv*, abs/2302.13971,
647 2023. URL <https://api.semanticscholar.org/CorpusID:257219404>.
- 649 Wang, P., Liu, L., Liu, Y., Theobalt, C., Komura, T., and
650 Wang, W. Neus: Learning neural implicit surfaces by
651 volume rendering for multi-view reconstruction. *arXiv*
652 *preprint arXiv:2106.10689*, 2021.
- 654 Wang, Q., Zhang, Y., Holynski, A., Efros, A. A.,
655 and Kanazawa, A. Continuous 3d perception
656 model with persistent state. 2025. URL <https://api.semanticscholar.org/CorpusID:275789153>.
- 659 Wang, S., Leroy, V., Cabon, Y., Chidlovskii, B., and
Revaud, J. Dust3r: Geometric 3d vision made
easy. *2024 IEEE/CVF Conference on Computer Vi-*
sion and Pattern Recognition (CVPR), pp. 20697–20709,
2023. URL <https://api.semanticscholar.org/CorpusID:266436038>.
- Wang, Z., Bovik, A. C., Sheikh, H. R., and Simoncelli,
E. P. Image quality assessment: from error visibility
to structural similarity. *IEEE Transactions on Im-*
age Processing, 13:600–612, 2004. URL <https://api.semanticscholar.org/CorpusID:207761262>.
- Xu, C., Wu, B., Hou, J., Tsai, S. S., Li, R., Wang,
J., Zhan, W., He, Z., Vajda, P., Keutzer, K., and
Tomizuka, M. Nerf-det: Learning geometry-aware
volumetric representation for multi-view 3d object
detection. *2023 IEEE/CVF International Confer-*
ence on Computer Vision (ICCV), pp. 23263–23273,
2023a. URL <https://api.semanticscholar.org/CorpusID:260202833>.
- Xu, M., Yin, X., Qiu, L., Liu, Y., Tong, X., and
Han, X. Sampro3d: Locating sam prompts in 3d for
zero-shot scene segmentation. *ArXiv*, abs/2311.17707,
2023b. URL <https://api.semanticscholar.org/CorpusID:265498885>.
- Yang, J., Sax, A., Liang, K. J., Henaff, M., Tang, H., Cao,
A., Chai, J., Meier, F., and Feiszli, M. Fast3r: Towards
3d reconstruction of 1000+ images in one forward pass.
2025. URL <https://api.semanticscholar.org/CorpusID:275820456>.
- Yang, Z., Zhang, S., Wang, L., and Luo, J. Sat: 2d
semantics assisted training for 3d visual ground-
ing. *2021 IEEE/CVF International Conference on*
Computer Vision (ICCV), pp. 1836–1846, 2021.
URL <https://api.semanticscholar.org/CorpusID:235166799>.
- Yeshwanth, C., Liu, Y.-C., Nießner, M., and Dai,
A. Scannet++: A high-fidelity dataset of 3d in-
door scenes. *2023 IEEE/CVF International Con-*
ference on Computer Vision (ICCV), pp. 12–22,
2023. URL <https://api.semanticscholar.org/CorpusID:261064784>.
- Yuan, Z., Yan, X., Liao, Y., Zhang, R., Li, Z., and Cui,
S. Instancerefer: Cooperative holistic understanding for
visual grounding on point clouds through instance multi-
level contextual referring. *2021 IEEE/CVF International*
Conference on Computer Vision (ICCV), pp. 1771–1780,
2021. URL <https://api.semanticscholar.org/CorpusID:232092539>.

- 660 Yuksekogonul, M., Bianchi, F., Kalluri, P., Jurafsky, D.,
661 and Zou, J. When and why vision-language models be-
662 have like bags-of-words, and what to do about it? In
663 *International Conference on Learning Representations*,
664 2023. URL [https://openreview.net/forum?](https://openreview.net/forum?id=KRLUvvh8uaX)
665 [id=KRLUvvh8uaX](https://openreview.net/forum?id=KRLUvvh8uaX).
- 666 Zamir, A., Sax, A., Shen, B. W., Guibas, L. J., Malik,
667 J., and Savarese, S. Taskonomy: Disentangling task
668 transfer learning. *2018 IEEE/CVF Conference on Com-*
669 *puter Vision and Pattern Recognition*, pp. 3712–3722,
670 2018. URL [https://api.semanticscholar.](https://api.semanticscholar.org/CorpusID:5046249)
671 [org/CorpusID:5046249](https://api.semanticscholar.org/CorpusID:5046249).
- 673 Zhang, C., Han, D., Qiao, Y., Kim, J. U., Bae, S.-H., Lee,
674 S., and Hong, C. S. Faster segment anything: Towards
675 lightweight sam for mobile applications. *arXiv preprint*
676 *arXiv:2306.14289*, 2023a.
- 677 Zhang, Y., Gong, Z., and Chang, A. X. Multi3drefer:
678 Grounding text description to multiple 3d objects. *2023*
679 *IEEE/CVF International Conference on Computer*
680 *Vision (ICCV)*, pp. 15179–15179, 2023b. URL [https:](https://api.semanticscholar.org/CorpusID:261681990)
681 [//api.semanticscholar.org/CorpusID:](https://api.semanticscholar.org/CorpusID:261681990)
682 [261681990](https://api.semanticscholar.org/CorpusID:261681990).
- 684 Zhou, Y., Gu, J., Chiang, T. Y., Xiang, F., and
685 Su, H. Point-sam: Promptable 3d segmentation
686 model for point clouds. *ArXiv*, abs/2406.17741,
687 2024. URL [https://api.semanticscholar.](https://api.semanticscholar.org/CorpusID:270711268)
688 [org/CorpusID:270711268](https://api.semanticscholar.org/CorpusID:270711268).
- 689 Zhu, H., Yang, H., Wu, X., Huang, D., Zhang, S., He, X.,
690 He, T., Zhao, H., Shen, C., Qiao, Y., and Ouyang, W.
691 Ponderv2: Pave the way for 3d foundation model with a
692 universal pre-training paradigm. *ArXiv*, abs/2310.08586,
693 2023a. URL [https://api.semanticscholar.](https://api.semanticscholar.org/CorpusID:263908802)
694 [org/CorpusID:263908802](https://api.semanticscholar.org/CorpusID:263908802).
- 696 Zhu, H., Yang, H., Wu, X., Huang, D., Zhang, S., He,
697 X., He, T., Zhao, H., Shen, C., Qiao, Y., and Ouyang,
698 W. Ponderv2: Pave the way for 3d foundation model
699 with a universal pre-training paradigm. *arXiv preprint*
700 *arXiv:2310.08586*, 2023b.
- 702 Zhu, Z., Ma, X., Chen, Y., Deng, Z., Huang, S., and
703 Li, Q. 3d-vista: Pre-trained transformer for 3d vision
704 and text alignment. *2023 IEEE/CVF International Con-*
705 *ference on Computer Vision (ICCV)*, pp. 2899–2909,
706 2023c. URL [https://api.semanticscholar.](https://api.semanticscholar.org/CorpusID:260704493)
707 [org/CorpusID:260704493](https://api.semanticscholar.org/CorpusID:260704493).
- 708 Zhu, Z., Zhang, Z., Ma, X., Niu, X., Chen, Y.,
709 Jia, B., Deng, Z., Huang, S., and Li, Q. Unify-
710 ing 3d vision-language understanding via promptable
711 queries. In *European Conference on Computer Vision*,
712 2024. URL [https://api.semanticscholar.](https://api.semanticscholar.org/CorpusID:269921315)
713 [org/CorpusID:269921315](https://api.semanticscholar.org/CorpusID:269921315).
- 714