COS3D: Collaborative Open-Vocabulary 3D Segmentation

Runsong Zhu[1] **Ka-Hei Hui**[2] **Zhengzhe Liu**[3] **Qianyi Wu**[4] **Weiliang Tang**[1]
Shi Qiu[1] **Pheng-Ann Heng**[1] **Chi-Wing Fu**[1]
[1] The Chinese University of Hong Kong [2] Autodesk AI Lab
[3] Lingnan University [4] Monash University

Abstract

Open-vocabulary 3D segmentation is a fundamental yet challenging task, requiring a mutual understanding of both segmentation and language. However, existing Gaussian-splatting-based methods rely either on a single 3D language field, leading to inferior segmentation, or on pre-computed class-agnostic segmentations, suffering from error accumulation. To address these limitations, we present COS3D, a new collaborative prompt-segmentation framework that contributes to effectively integrating complementary language and segmentation cues throughout its entire pipeline. We first introduce the new concept of collaborative field, comprising an instance field and a language field, as the cornerstone for collaboration. During training, to effectively construct the collaborative field, our key idea is to capture the intrinsic relationship between the instance field and language field, through a novel instance-to-language feature mapping and designing an efficient two-stage training strategy. During inference, to bridge distinct characteristics of the two fields, we further design an adaptive language-to-instance prompt refinement, promoting high-quality prompt-segmentation inference. Extensive experiments not only demonstrate COS3D's leading performance over existing methods on two widely-used benchmarks but also show its high potential to various applications, *i.e.*, novel image-based 3D segmentation, hierarchical segmentation, and robotics.

1 Introduction

Open-vocabulary 3D segmentation (OV3DS) aims to predict 3D segmentation of scenes according to given natural language queries. Beyond traditional 3D segmentation, which is often restricted to fixed object categories [1–8], the OV3DS task supports flexible text queries, allowing for diverse semantic categories, physical properties, affordance, and more. This flexibility is crucial to making OV3DS a practical and valuable tool for applications in fields such as AR, VR, and robotics.

Recent efforts focus on transferring 2D vision-language models (VLMs) to 3D scenes represented by learned radiance fields. These works can be roughly divided into two classes: *language-based* and *segmentation-based*. Specifically, language-based methods [9–13] propose to distill language features (*e.g.*, CLIP [14]) from the 2D image space to a 3D language field by leveraging differentiable rendering to support OV3DS; see Fig. 1 (a). However, directly learning language features through a pixel-wise language distillation demonstrates limited distinctiveness, leading to severe artifacts and errors around boundaries in the segmentation results.

On the other hand, segmentation-based methods, *e.g.*, [15–17], directly decompose the OV3DS task into two sub-tasks: (i) a class-agnostic 3D segmentation, followed by (ii) a post-selection of the best-matched 3D segments, using a 2D vision-language model [14, 18]; see Fig. 1 (b). Though this approach bypasses direct language distillation, it faces two major challenges, leading to limited performance. First, without semantic cues, accurately segmenting all objects in a 3D scene is highly

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

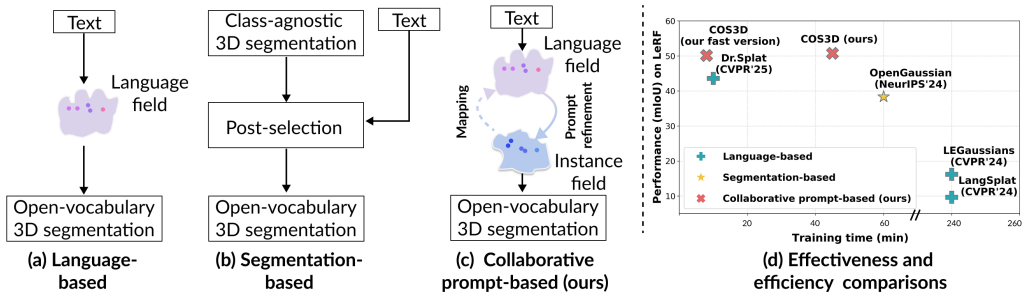

Figure 1: Comparing different paradigms. (a) Language-based methods [9–13] directly learn a 3D language field for open-vocabulary segmentation. (b) Segmentation-based methods [15–17] perform class-agnostic segmentation then post selection. (c) COS3D addresses existing limitations with a new collaborative prompt-segmentation framework that connects knowledge from the language and instance fields in the training and inference. The solid line (dotted line) indicates inference (training). (d) COS3D clearly outperforms existing methods on both segmentation quality and training efficiency. Note that "fast version" refers to a result of our approach under a short optimization time setting (see Sec.4.3), while the time and performance of the baselines are sourced from their publications [12, 16].

challenging, so under- and over-segmentation errors often occur in the class-agnostic segmentation, which further affects the post selection. Second, the hand-crafted matching strategies in the post selection easily introduce additional inaccuracies that further degrade the performance.

Revisiting the existing methods, we attribute their limitations to the lack of integrating language and segmentation information. In particular, these two types of information provide complementary knowledge: segmentation information is typically *discriminative* and *boundary-aware*, whereas language information facilitates high-level *understanding of objects and scenes*. Fundamentally, to achieve OV3DS requires a mutual understanding of *both* language and segmentation.

To this end, we introduce COS3D, a new COllaborative approach for prompt-Segmentation of 3D scenes, in which we collectively incorporate segmentation and language cues in our framework; see Fig. 1 (c). There are three technical components in COS3D. First, we propose the new concept of *collaborative field*, comprising an instance field and a language field, as the foundation in COS3D. To effectively construct the two fields, *our key insight lies in their intrinsic relationship: regions within the same object segment should share similar semantics and exhibit similar language information.* Second, we propose modeling the intrinsic relationship through a feature mapping process from a learned, boundary-aware instance field to the text-aligned language field. Here, we first train the instance field to implicitly encode the segmentation information, then formulate an instance-to-language mapping learning to facilitate the language-field construction. Third, at inference, given a text query, we generate the segmentation from the text-aligned language field. Importantly, considering the limited expressivity of the language feature, we leverage the distinct characteristics of the instance field and introduce an adaptive language-to-instance prompt refinement to exploit the intermediate 3D relevance map from the instance field as a prompt, then design a further refinement on the boundary-aware instance field for prompt segmentation. With these new designs, COS3D is able to arrive at a surprisingly **effective** and **efficient** solution; see Fig. 1 (d).

We evaluate COS3D on two standard benchmarks for OV3DS. Both quantitative and qualitative results show that our method *significantly* outperforms existing approaches. Also, the ablation studies validate the effectiveness of our designs for both training and inference stages. Furthermore, we present three example applications of COS3D, including novel image-based 3D segmentation, explicit hierarchical OV3DS, and robotics, demonstrating its potential and practical values.

Our major contributions are summarized as follows:

- We present COS3D, a new collaborative prompt-segmentation framework that integrates segmentation and language cues, enabling top-quality open-vocabulary 3D segmentation.
- For training, we propose a novel instance-to-language mapping with two optional implementations that effectively and efficiently leverage the instance field to enhance the construction of a semantically meaningful language field.
- For inference, we propose an adaptive language-to-instance prompt refinement that utilizes the 3D relevance map from the language field to guide the refinement in the instance field.

- Our method sets a new state-of-the-art performance on two standard benchmarks and shows strong potential for image-based segmentation, hierarchical segmentation, and robotics.

2 Related work

Radiance field. Radiance fields have emerged as a powerful representation for supporting 3D scene reconstruction with diverse properties such as geometry, color, and semantics, from only 2D inputs such as multi-view RGB images and extracted feature maps. Neural Radiance Fields (NeRF) [19] model the radiance field using neural networks composed of MLPs, enabling photorealistic novel view synthesis. Subsequent works focus on improving the efficiency of NeRF by introducing explicit 3D structures, such as voxel grids [20, 21] and hash grids [22]. More recently, 3D Gaussian Splatting (3D-GS) [23–32] is proposed as an alternative representation by modeling the radiance field as a set of explicit 3D Gaussian points. This approach supports splatting-based rendering [33], which is highly efficient, significantly enhancing its potential for real-time applications. Given these advantages, we adopt 3D-GS as the backbone representation in our 3D segmentation framework.

Open-vocabulary 3D segmentation. Open-vocabulary 3D scene segmentation has made significant progress in recent years, empowered by 2D foundation vision-language models (VLMs) (*e.g.*, CLIP [14], LSeg [34], DINO [35]), together with 3D representations, ranging from point clouds to radiance fields. Early approaches [36–43] project 3D point clouds to 2D views to align them with image-based features, enabling zero-shot open-vocabulary 3D segmentation. However, as discussed in [44], point cloud representation suffers from the discrete structure and typically has a lower resolution compared to images, limiting their effectiveness and applications.

To overcome limitations of the point cloud representation, recent methods [9–13, 15–17, 45, 44, 46] propose distilling dense VLM features into continuous radiance field representations, enabling high-resolution novel-view synthesis for effective feature alignment and downstream tasks. Specifically, LeRF [9] first introduced the concept of language field distillation into NeRF via a 2D CLIP supervision. Besides, LangSplat [10], LEGGaussians [11], Dr.Splat [12], and FastLGS [13] adopt 3D-GS as an explicit scene representation, which integrates language features, for supporting open-vocabulary 3D scene understanding. While these models achieve promising results, the segmentation quality is still limited by the weak expressiveness of the directly learned language features. In contrast, other methods [15–17] tackle the task sequentially by first performing class-agnostic 3D segmentation, followed by selecting the best-matched 3D segment using language queries. For instance, OpenGaussian [16] and InstanceGaussian [17] propose to align 3D segmentations with 2D segmentations produced by SAM to leverage 2D CLIP features to enable subsequent text grounding. Besides, Gaussian Grouping [15] employs a 2D vision-language grounding model (*i.e.*, GroundingDINO [18]) to associate 2D grounding result with 3D segments using handcrafted 2D matching techniques. However, these methods suffer from error accumulation, which restricts overall performance. To address these limitations, we introduce a novel 3D prompt-based segmentation framework that collaboratively engage both segmentation cues and language cues, throughout both the training and inference stages, to optimize for high-quality open-vocabulary segmentation.

Promptable segmentation. Promptable segmentation, which aims to generate segmentations based on input prompts by specifying the target to be segmented within an image, was introduced by the Segment Anything Model (SAM) [47]. Its effectiveness has been widely demonstrated through a number of follow-up works [48–52]. To adopt it to 3D understanding, some recent methods [53–56] propose to learn the discriminative instance field that encode 3D segmentation information. At the inference, these methods leverage user-click prompts to specify the target, facilitating the production of more accurate segmentation results. Though these methods achieve notable improvement, they require additional manual interaction in screen space as prompt, thereby limiting their applicability in autonomous 3D systems.

Going beyond the prior works, we realize a novel prompt-segmentation framework to directly support open-vocabulary 3D queries as prompts for segmentation inference through innovations that actively integrate the segmentation-aware instance field and text-aligned language field.

3 Method

Given a set of multi-view posed images, we utilize the 2D foundation models SAM [57] and CLIP [14] to produce associated 2D instance segmentation masks $\{\mathcal{K}_I\}$ and 2D language feature maps $\{\mathcal{F}_L\}$.

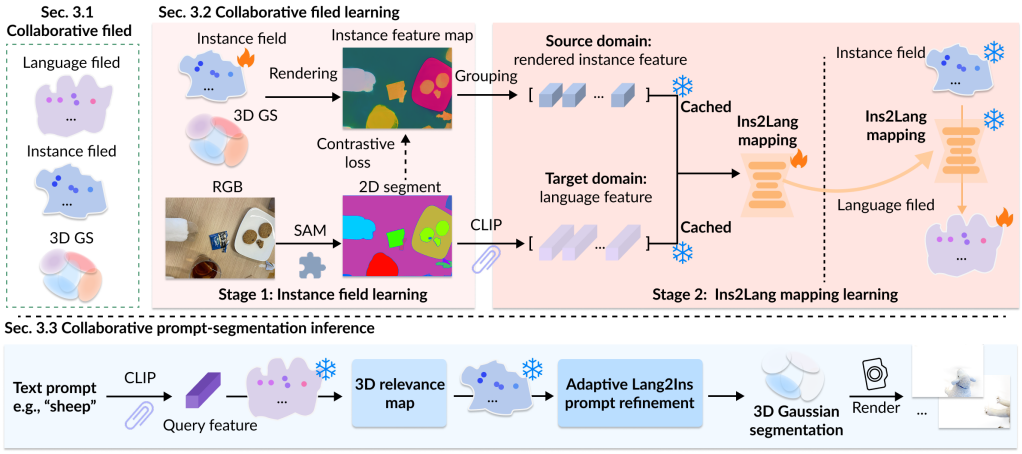

Figure 2: Overview of our proposed COS3D method. We first introduce collaborative field, comprising an instance field and a language field (see Sec.3.1). During training, we first learn the instance field to encode instance information and propose a novel instance-to-language (Ins2Lang) mapping to construct the language field (see Sec.3.2). During inference, leveraging the 3D relevance map from the language field, we design an adaptive language-to-instance (Lang2Ins) prompt refinement to further guide the instance field, enabling accurate segmentation (see Sec.3.3).

Based on reconstructed 3D Gaussian representations, our objective is to enrich these 3D Gaussians with high-quality, text-guided segmentations that align with the open-vocabulary query. Fig. 2 illustrates the overview of our approach. First, to support effective open-vocabulary segmentation, we propose the new concept of collaborative field, which comprise an instance field and a language field, integrating both segmentation and language cues (see Sec. 3.1). Next, to enable collaboration between the instance and language fields throughout the whole pipeline, we further design a novel instance-to-language mapping within a two-stage learning strategy in Sec. 3.2 and an adaptive language-to-instance prompt refinement for collaborative prompt-based inference in Sec. 3.3.

3.1 Collaborative field

3D-GS backbone. Specifically, our collaborative field utilize the explicit 3D Gaussian Splatting (3D-GS) [23], as the underlying 3D scene representation. Specifically, the 3D-GS model represents a 3D scene using a collection of explicit 3D Gaussians and leverages differentiable rasterization for efficient rendering. Mathematically, 3D-GS represents a 3D scene as a set of N Gaussians, $G = \{g_i\}_{i=1}^{N}$, where each $g_i = (\mathbf{p}_i, \mathbf{s}_i, \mathbf{q}_i, o_i, \mathbf{c}_i)$ denotes the center, scale, orientation (quaternion), opacity, and color coefficients (in spherical harmonics) of the i-th Gaussian. Each 3D Gaussian is projected to the image plane as a 2D Gaussian G_i' via tile-based rasterization [23]. The color at a query pixel u is computed using α-blending:

$$\mathbf{C}_u = \sum_{i \in \mathcal{N}} \mathbf{c}_i \alpha_i \prod_{t=1}^{i-1} (1 - \alpha_t), \quad \alpha_i = o_i G_i'(u), \tag{1}$$

where $\mathcal{N}$ is the set of sorted 2D Gaussians contributing to pixel u. The parameters $\{g_i\}$ are optimized via photometric reconstruction loss.

Collaborative field. To enable effective open-vocabulary 3D segmentation of the scene represented by 3D-GS, we propose the collaborative field to fully exploit the complementary strengths of segmentation and language information. Specifically, our collaborative field comprise two components: an instance field $\Theta_{\mathbf{I}}$ and a language field $\Theta_{\mathbf{L}}$. Mathematically, the instance field is obtained by augmenting each Gaussian g with a feature vector $\mathbf{I} \in \mathbb{R}^{d_I}$, where d_I is the feature dimension (set to 16 by default). Besides, the language field is encoded as a high-dimensional language feature vector $\mathbf{L} \in \mathbb{R}^{d_L}$ for each Gaussian g, where d_L is the dimension of language feature, *i.e.*, 512 for CLIP [14]. We refer to these representations as collaborative field because they interact during both training and inference. During training, the instance field serves as a distinctive representation that

simplifies learning the language field, allowing us to construct it more effectively and efficiently. At inference time, the language field provides an initial 3D relevance map in response to a text query, which guides the instance field to refine and produce accurate open-vocabulary segmentation results.

3.2 Collaborative field learning

In our collaborative field learning, the key idea is to introduce the instance-to-language mapping to model the intrinsic relationship of two fields. To this end, we propose a two-stage training strategy. First, we learn a segmentation-aware instance field supervised by the 2D SAM [47] segmentation. Next, based on the learned instance field, we construct the language field by learning a mapping function from paired instance and CLIP features extracted from multi-view images. Once trained, this mapping function is applied to the instance field to generate a language field, completing the construction of the collaborative field.

Stage 1: instance field learning. For the instance field Θ_I, we adopt contrastive learning to optimize the rendered features. Similar to color rendering, we apply differentiable rasterization to efficiently render the instance feature $\mathbf{I}_u$ at each pixel u as: $\mathbf{I}_u = \sum_{i\in\mathcal{N}} \mathbf{I}_i \alpha_i \prod_{t=1}^{i-1}(1-\alpha_t), \alpha_i = o_i G'_i(u)$. Then, we apply widely-used InfoNCE loss in existing works [54, 55, 58, 59] for supervision:

$$\mathcal{L}_{\text{ins}} = -\frac{1}{|\Omega|}\sum_{\Omega_j\in\Omega}\sum_{u\in\Omega_j}\log\frac{\exp\left(\text{sim}\left(\mathbf{I}_u, \bar{\mathbf{I}}_j\right)\right)}{\sum_{\Omega_I\in\Omega}\exp\left(\text{sim}\left(\mathbf{I}_u, \bar{\mathbf{I}}_l\right)\right)}, \tag{2}$$

where similarity kernel function sim uses the dot product operation here and Ω is the set of pixel samples. In specific, Ω_j, Ω_l denotes the pixel samples with the same instance ID j, l according to the 2D instance segmentation $\mathcal{K}_I$, $\bar{\mathbf{I}}_j$ and $\bar{\mathbf{I}}_l$ represent the mean instance features (centroids) for Ω_j and Ω_l, respectively. Notably, the instance segmentation mask $\mathcal{K}_I$ is automatically generated using SAM by creating a grid of point prompts across the image, following common practices [10, 16, 12, 11]. By minimizing this loss across training views, the instance field learns to produce discriminative and view-consistent features that capture 3D instance-level information.

Stage 2: instance-to-language (Ins2Lang) mapping learning. Based on the learned discriminative instance field, we introduce an Ins2Lang mapping to transfer the instance feature source domain to the language feature target domain. Mathematically, the mapping function Φ is defined as

$$\Phi : \mathbf{L} = \Phi(\mathbf{I}), \quad \mathbf{I} \in \Theta_{\mathbf{I}}, \tag{3}$$

where $\mathbf{I}$ denotes the instance feature and $\mathbf{L}$ denotes the corresponding language feature. To learn the mapping function Φ, we first construct training pairs between the instance features and their corresponding language features. Specifically, for each individual view, we render the multi-view consistent instance feature map, enabling us to directly use the 2D language feature map $\mathcal{F}_L$ from CLIP as the corresponding pair. Moreover, since the CLIP features are inherently patch-wise, and to reduce redundancy, we utilize the SAM mask to group rendered instance feature maps and CLIP feature maps by averaging the features with the same 2D instance ID. This process results in segment-wise training feature pairs, denoted as $(\mathbf{I}^m, \mathbf{L}^m)_{m=1}^M$, where M is the total number of pairs. Based on the training pairs, we further provide two implementation strategies, *i.e.*, shallow MLPs and kernel regression for the mapping function.

i) **Shallow MLPs**. We can choose shallow MLPs, denoted as Φ_{network}, to represent the instance-to-language mapping. Then, we utilize the prepared mapping pair $(\mathbf{I}^m, \mathbf{L}^m)_{m=1}^M$ to supervise the learning of Φ_{network}. Specifically, we use the following mapping loss: $\mathcal{L}_{\text{mapping}} = |\mathbf{L}^m - \Phi_{\text{network}}(\mathbf{I}^m)|$. Notably, the mapping learning is highly efficient, requiring less than three minutes on a single GPU.

ii) **Kernel regression**. We can also utilize traditional kernel regression, denoted as Φ_{kernel}, to represent the mapping function. In particular, we adopt the widely used Nadaraya-Watson estimator [60] because of its simplicity and learning-free nature. Mathematically, the kernel regression function is defined as: $\Phi_{\text{kernel}}(\mathbf{I}) = \sum_{m=1}^M (\exp\left(-\frac{\|\mathbf{I}-\mathbf{I}^m\|^2}{2\sigma^2}\right)\mathbf{L}^m)/\sum_{m=1}^M \exp\left(-\frac{\|\mathbf{I}-\mathbf{I}^m\|^2}{2\sigma^2}\right)$, where σ is a hyperparameter that controls the bandwidth of the kernel function, and is set to 0.1 by default.

Based on the learned Ins2Lang mapping Φ, we can obtain our language feature field Θ_L by calculating the corresponding language feature $\Phi(\mathbf{L})$ for each Gaussian.

Discussion. We introduce the Ins2Lang mapping within a two-stage training strategy for language field construction, offering the advantages of effectiveness and efficiency. (i) **Effectiveness**: In contrast to existing approaches that directly learn language features from scratch, our method constructs the language field via a mapping function from the learned instance field. This strategy leverages the discriminative, segmentation-aware features already captured by the instance field, enabling more semantically meaningful and spatially coherent language representations. As a result, it significantly improves the quality of open-vocabulary segmentation. (ii) **Efficiency**: Unlike approaches that directly optimize language features for each Gaussian point, requiring extensive supervision and per-point updates, our method employs a shared mapping function that generalizes across all Gaussians. This significantly reduces the number of parameters and the training overhead. Moreover, instead of using dense pixel-level supervision, we construct training pairs at the segment (patch) level using SAM masks, which further lowers redundancy and improves learning efficiency. Together, these design choices result in a highly efficient training process. In addition, the experimental comparisons with other alternatives are provided in our ablation (see Sec. 4.3).

3.3 Collaborative prompt-segmentation inference

During inference, we further design an adaptive language-to-instance prompt refinement, enabling a collaborative prompt-segmentation inference. As illustrated at the bottom of Fig. 2, given a query text, we utilize the text-aligned language field to generate a 3D relevance map as a prompt and introduce the adaptive prompt refinement in the boundary-aware instance field for producing accurate 3D Gaussian segmentation.

3D relevance map in language field. Based on the text-aligned language field, for the text query q_{text}, we first generate the 3D relevance map, which indicates the correspondence between the input text and 3D regions. Specifically, we first utilize the CLIP [14] text encoder to obtain the corresponding language feature $\mathbf{L}_{\text{text}}$. Then, we compute the 3D point-level relevance R as: $R = \min_i \frac{\exp(\mathbf{L}\cdot\mathbf{L}_{\text{text}})}{\exp(\mathbf{L}\cdot\mathbf{L}_{\text{text}})+\exp(\mathbf{L}\cdot\mathbf{L}^i_{\text{canon}})}$, where $\mathbf{L}_{\text{canon}}$ is the CLIP embedding of a predefined canonical phrase [9]. Intuitively, we can obtain segmentation by identifying Gaussian points $\mathcal{S}$ with high relevance, using a predefined threshold τ (set to 0.5 by default), following common practices [9–12, 16].

Adaptive language-to-instance (Lang2Ins) prompt refinement. Directly extracting the segmentation by the above process easily produces inferior segmentation (see Fig. 3 (b)). To address this, we treat the relevance map as a prompt to guide the instance field in refining segmentation via an adaptive Lang2Ins prompt refinement process. Particularly, starting from the initial high-relevance Gaussian point set $\mathcal{S}$, we aim to obtain a refined Gaussian point set $\mathcal{S}_t$ that represents 3D Gaussian segmentation results. Concretely, for each center Gaussian point $g' \in \mathcal{S}$, we leverage the learned instance field to define a local neighborhood set $\mathcal{S}_{g'}$, consisting of points whose instance features have cosine similarity above a threshold $\mathcal{T}$ with that of g'. Here, threshold $\mathcal{T}$ is based on the statistical value from instance field. Considering the presence of errors in $\mathcal{S}$, and the risk that expansion from noisy points may include undesired objects, we further perform an adaptive filtering operation for $\mathcal{S}_{g'}$ based on a region-level relevance. We define this region-level relevance as the opacity-weighted average of relevance scores in $\mathcal{S}_{g'}$: $(\sum_{w\in\mathcal{S}_{g'}} o_w * R_w)/(\sum_{w\in\mathcal{S}_{g'}} o_w)$, where o_w and R_w are the opacity and relevance of point $w \in \mathcal{S}_{g'}$. We process the regions in descending order based on the relevance score of their center points. A region is included in the final segmentation only if its region score exceeds the threshold τ (as defined earlier). By applying this process to the initial set $\mathcal{S}$, we gradually aggregate retained expanded point sets, producing the final segmentation $\mathcal{S}_t$. More details on the algorithm and automatic threshold generation are provided in Supp.

Figure 3: Visual comparisons on LeRF [9].

Discussion. Unlike the existing methods [10–12] that solely rely on the relevance map, bounded by the limited expressivity of language features, our adaptive Lang2Ins prompt refinement further leverages the discriminative instance field to aggregate neighboring points with spatial and semantic coherence, thus enabling boundary-aware segmentation. Further, it helps adaptively filter noisy points using robust region-wise relevance. As shown in Fig. 3 (c), this algorithm significantly enhances the Gaussian segmentation quality, enabling clean object rendering. Moreover, the proposed adaptive Lang2Ins prompt refinement is efficient, adding small overhead to the query time (see Sec. 4.3).

3.4 Implementation details

We adopt the official implementation of 3D-GS [23] with a default of 30K training iterations as our base architecture. For the instance field and instance-to-language mapping (*e.g.*, MLPs version), we also set the training iterations to 30K by default, following common practice as in [10]. For kernel regression version in mapping, the function is directly formulated without requiring training. All experiments are conducted on a single RTX-4090 GPU. More details are provided in Supp.

4 Experiments

4.1 Results on LeRF dataset

Settings. **(i) Task**: For open-vocabulary text queries, we first select the corresponding Gaussians and render them into multi-view 2D images. **(ii) Dataset and metrics**: Following OpenGaussian [16] and LangSplat [10], we evaluate our method on the LeRF dataset [9]. After rendering the selected 3D objects into 2D views, we compute mean Intersection over Union (mIoU) and mean Accuracy (mAcc). **(iii) Baselines**: As this task requires explicit 3D point-level segmentation, we compare our method with other explicit Gaussian-based methods, including language-based methods LangSplat [61], LEGaussians [11], and segmentation-based method OpenGaussian [16]. Moreover, we provide the quantitative comparison with the most recent works, *i.e.*, InstanceGaussian [17] and Dr. Splat [12].

Results. Quantitative results are presented in Tab. 1, demonstrating that our proposed method achieves significantly improved results compared to all existing language-based and segmentation-based baselines. The qualitative results in Fig. 4 further show that the rendered objects using our method contain more complete boundaries and significantly fewer artifacts. Note that Instance-Gaussian [17] and Dr. Splat [12] are not open-sourced, which prevents further visual comparisons.

Table 1: Performance of Gaussian segmentation in 3D space from text query on LeRF [9]. Following [16], we report mIoU and mAcc. The performance of all prior works is sourced from [16, 17, 12].

Method	Venue	Type	**mean**		*figurines*		*teatime*		*ramen*		*waldo_kitchen*	
			mIoU ↑	**mAcc ↑**	mIoU ↑	mAcc ↑	mIoU ↑	mAcc ↑	mIoU ↑	mAcc ↑	mIoU ↑	mAcc ↑
LangSplat [10]	CVPR'24	Language	9.66	12.41	10.16	8.93	11.38	20.34	7.92	11.27	9.18	9.09
LEGaussians [11]	CVPR'24	Language	16.21	23.82	17.99	23.21	19.27	27.12	15.79	26.76	11.78	18.18
Dr.Splat [12]	CVPR'25	Language	43.58	63.87	53.36	80.36	57.20	76.27	24.70	35.21	39.07	63.64
OpenGaussian [16]	NeurIPS'24	Segmentation	38.36	51.43	39.29	55.36	60.44	76.27	31.01	42.25	22.70	31.82
IntanceGaussian [17]	CVPR'25	Segmentation	45.30	58.44	-	-	-	-	-	-	-	-
Ours (shallow MLPs)	-	Collaborative prompt	49.75	70.60	53.90	76.79	**66.91**	88.14	**36.61**	**49.30**	41.56	68.18
Ours (kernel regression)	-	Collaborative prompt	**50.76**	**72.08**	**60.03**	**82.14**	65.07	**91.53**	35.86	46.48	**42.10**	**68.18**

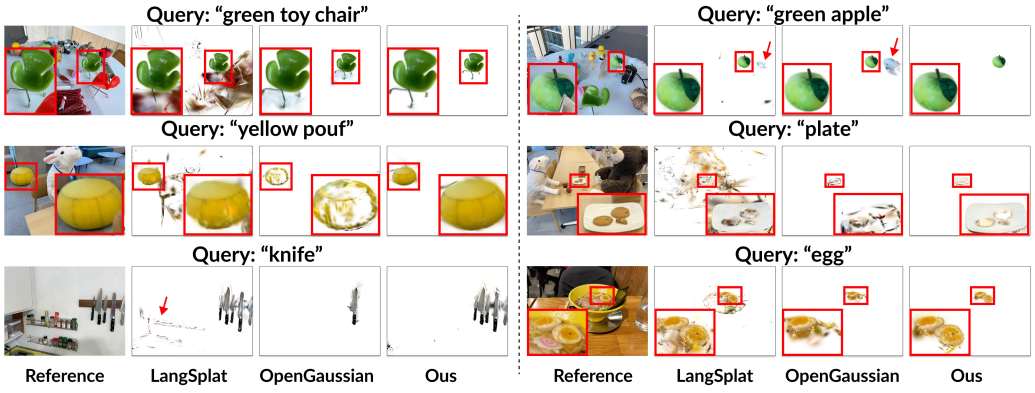

Figure 4: Open-vocabulary 3D Gaussian segmentation on the LeRF dataset. Our method outperforms previous open-sourced SOTA methods (*i.e.*, LangSplat, OpenGaussian) in accurately identifying the 3D objects corresponding to text queries with fewer artifacts. Here, we present our results using kernel regression for visual comparison, and more results are provided in Supp.

Table 2: Performance of Gaussian segmentation on the ScanNetv2 [62] dataset compared to baselines [10, 11, 16] based on text query. The performance of all prior work has been sourced from [16].

Methods	Type	19 classes		15 classes		10 classes	
		mIoU ↑	mAcc. ↑	mIoU ↑	mAcc. ↑	mIoU ↑	mAcc. ↑
LangSplat [10]	Language	3.78	9.11	5.35	13.20	8.40	22.06
LEGaussians [11]	Language	3.84	10.87	9.01	22.22	12.82	28.62
OpenGaussian [16]	Segmentation	24.73	41.54	30.13	48.25	38.29	55.19
Ours (shallow MLPs)	Collaborative prompt	26.72	39.89	31.02	46.30	37.28	55.41
Ours (kernel regression)	Collaborative prompt	**32.47**	**49.05**	**35.95**	**54.35**	**44.32**	**63.66**

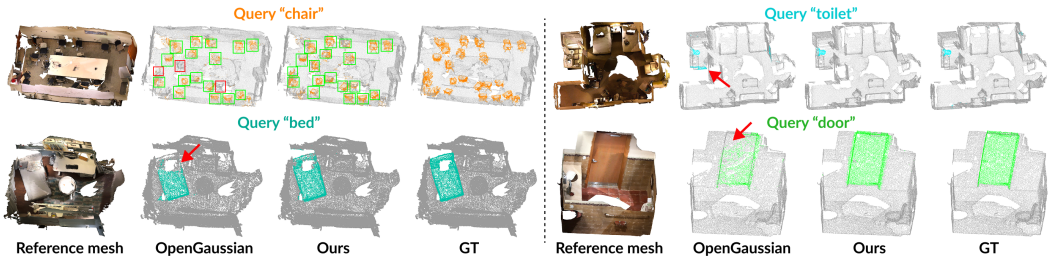

Figure 5: Open-vocabulary Gaussian segmentation on ScanNetv2 [62] dataset. Our method outperforms the previous open-sourced SOTA approach (*i.e.*, OpenGaussian) in accurately identifying 3D objects for various text queries. In addition, we use green boxes to indicate regions of accurate predictions and red boxes to indicate regions of missing predictions. Here, we present our results using kernel regression for visual comparison, and more results are provided in Supp.

4.2 Results on ScanNetv2 dataset

Settings. **(i) Task**: In this task, we focus on direct 3D evaluation without rendering processing. Specifically, the model takes text queries as input and selects the corresponding Gaussian points. **(ii) Dataset and metrics**: Following the protocol established by OpenGaussian [16], we adopt 19, 15, and 10 categories from ScanNetv2 [62] as text queries. Moreover, we evaluate performance using mIoU and mAcc for the 10 scenes selected by OpenGaussian [16]. **(iii) Baselines:** We compare our method against recent Gaussian-based approaches, including LangSplat [10], LEGaussians [11], and OpenGaussian [16], following prior work [16]. Note that the latest baselines (*i.e.*, InstanceGaussian and Dr. Splat) use different evaluation protocols or segmentation inputs, which hinder direct evaluation using the results from their papers. We provide additional comparisons in Supp.

Results. Quantitative results, as shown in Tab. 2, consistently demonstrate that our method significantly outperforms Gaussian-based methods. The 3D visualization results are presented in Fig. 5, illustrating that our method achieves accurate and complete 3D Gaussian point-level segmentation for various queries, especially in challenging scenarios (*e.g.*, the dense "chair" query shown in Fig. 5).

4.3 Ablation study

The influence of learning designs for collaborative field. For collaborative field learning, we propose an instance-to-language mapping design within a two-stage learning strategy. Alternatively, there are two other technically feasible training solutions for our proposed collaborative field. One straightforward approach is the one-stage joint learning of instance and language branches, where the instance field is also influenced by the mapping loss, leading to costly optimization and an additional risk that the supervision for the mapping may make the instance feature space less discriminative. Another alternative is parallel learning, where the two branches are trained independently, failing to fuse the instance field and the language field. As shown in Tab. 3, compared to alternatives, both of our two implementations for the proposed two-stage instance-to-language mapping design not only improve final performance but also significantly reduce the required training time. Moreover, we find that our kernel regression implementation achieves the best performance. We attribute this to the choice of discriminative instance features as the source domain, which makes the mapping process

inherently an easy regression task, and the traditional kernel regression method is well-suited for such a case. Thus, we use the kernel regression implementation for subsequent analysis. Moreover, we provide a more detailed analysis of kernel regression and the MLP counterpart in Supp.

The influence of collaborative prompt-segmentation inference. To analyze the influence of collaborative prompt-based segmentation inference, we compare our results with two other alternatives. Specifically, the first inference alternative conducts the class-agnostic 3D segmentation by clustering [7] and uses a similar strategy as OpenGaussian [16] to select the 3D segment results, and the second inference alternative solely utilizes the language branch in our collaborative field to generate the segmentation results [10, 11]. The comparisons are presented in Tab. 4, indicating that collaborative prompt-based inference significantly improves segmentation by integrating language and instance knowledge, with only a slight increase in inference time per query.

Table 3: Ablation of learning on LeRF [9].

Learning solution / (**Mean**)	mIoU ↑	mAcc ↑	Training time
Joint learning	49.15	69.19	165 min
Parallel learning	43.84	59.81	95 min
Our (shallow MLPs)	49.75	70.60	53 min
Our (kernel regression)	**50.76**	**72.08**	50 min

Table 4: Ablation of inferences on LeRF [9].

Inference solution / (**Mean**)	mIoU ↑	mAcc ↑	Query time
Instance branch	44.07	59.83	0.12 s
Language branch	48.99	71.31	0.13 s
Collaborative prompt (ours)	**50.76**	**72.08**	0.22 s

Training efficiency. To analyze the training efficiency of our method, we compare performance under different training times. As training the instance field with the default 30K optimization steps requires the majority of the total training time (45 out of 50 minutes), we conducted experiments with shorter training times by reducing the number of optimization steps (*i.e.*, 3K and 6K). The results, presented in Tab. 5, demonstrate that our method converges quickly and achieves significantly better performance than baselines even with less training time, highlighting our superior training efficiency.

The influence of different 2D foundation vision-language models (VLMs). We utilize CLIP [14] and SAM [47] as the default 2D language and segmentation models in our main experiments, following common practice in recent baselines [10, 11, 16, 12, 17], to ensure that the improvements are attributed to the proposed collaborative fields design. Furthermore, we conducted an ablation study comparing different 2D foundation VLMs (e.g., CLIP [14] vs. SigLIP [63], SAM [47] vs. SAM2 [64], Semantic SAM [48]). The results, shown in the Tab. 6, demonstrate that our framework is compatible with different 2D foundation models. Additionally, we empirically observed that using more advanced models (*e.g.*, SAM2 [64] for the segmentation model and SigLIP [63] for the language model) can lead to performance improvements.

Table 5: Training efficiency analysis on LeRF [9]. We report mean mIoU.

Method	Training time	mIoU ↑
Langsplat [10]	240 min	9.66
LEGaussian [11]	240 min	16.21
Dr.Splat [12]	10 min	43.58
OpenGaussian [16]	60 min	38.36
Instance Gaussian [17]	-	45.30
Ours (3k for instance)	8 min	50.16
Ours (6k for instance)	15 min	50.24
Ours (default)	50 min	50.76

Table 6: Comparisons of various 2D foundation VLMs on LeRF [9]. Model A is the default implementation for our proposed COS3D.

Model	Segmentation	Language	mIoU ↑	mAcc ↑
A	SAM [47]	CLIP [14]	50.76	72.08
B	SAM [47]	SigLIP [63]	51.08	73.79
C	SAM2 [64]	CLIP [14]	51.94	75.05
D	Semantic SAM [48]	CLIP [14]	49.93	70.94

4.4 Applications

Novel image-based 3D segmentation. Beyond text queries, our method inherently supports 3D segmentation using a novel image as queries. Specifically, given a novel query image, we utilize the CLIP vision backbone to extract visual features and apply our inference algorithm to obtain the 3D segmentation. As shown in Fig. 6 (a), our method enables accurate image-based segmentation when the query image contains a similar, but not identical, object to those in the original 3D scene.

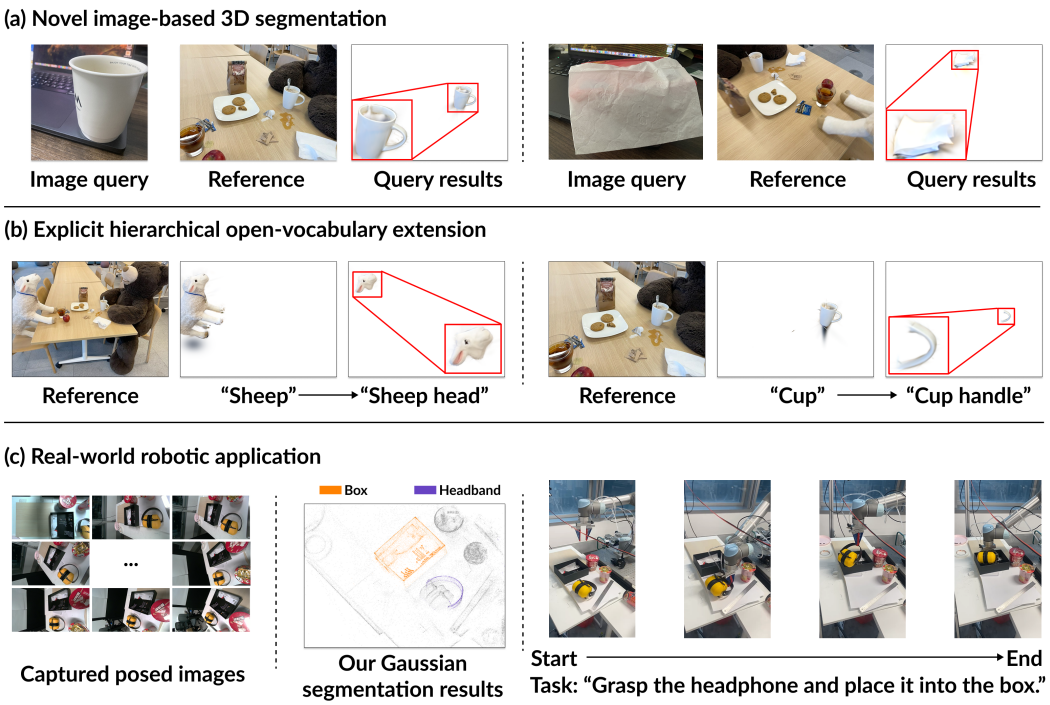

Figure 6: (1) We illustrate the novel image-based 3D segmentation results. (b) We perform explicit hierarchical open-vocabulary 3D segmentation on LERF [9]. (c) Following prior works [65–67], we leverage our method to provide accurate 3D segmentation for real-world robotic grasping, enabling the successful execution of grasp manipulation. We provide the video demo in Supp.

Explicit hierarchical open-vocabulary extension. Furthermore, our method can be naturally extended to support explicit hierarchical queries. Inspired by Click-Gaussian [54], we apply our mapping function to construct a hierarchical language field based on a two-level feature representation to capture hierarchical information. As shown in Fig. 6 (b), our approach produces accurate 3D segmentation results across coarse and fine-grained queries, enabling explicit hierarchical understanding.

Real-world robotic application. We further demonstrate the applicability of COS3D in real-world robotic scenarios. Specifically, we leverage our method to provide effective 3D perception for robotic grasping tasks, building upon prior work [65–67]. As illustrated in Fig. 6 (c), the accurate open-vocabulary segmentation produced by our method assists the robotic arm in completing grasping operations. More details and the video demo are provided in Supp.

5 Conclusion

We presented COS3D, a new collaborative 3D prompt-segmentation approach for open-vocabulary queries. We introduce the new concept of collaborative field comprising the instance and language fields. To achieve collaboration, we model the instance-to-language mapping during training and design an adaptive language-to-instance prompt refinement during inference. Extensive experimental results manifest the effectiveness of our method over the state of the art.

Limitations. Although COS3D provides effective open-vocabulary 3D segmentation, it has the following limitations. First, COS3D lacks reasoning capabilities for 3D segmentation, as the text-aligned language field struggles with relational or multi-object queries. Moreover, following recent approaches [10, 16, 12, 11], COS3D adopts the offline setting, and extending COS3D to the online setting would be beneficial. We provide more discussion on these potential extensions in Supp.

Acknowledgements

This study was funded by the InnoHK initiative of the Innovation and Technology Commission of the Hong Kong Special Administrative Region Government via the Hong Kong Centre for Logistics Robotics; and Hong Kong Innovation and Technology Fund under Project MHP/092/22.

References

[1] C. R. Qi, H. Su, K. Mo, and L. 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*, pp. 652–660, 2017.

[2] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "PointNet++: Deep hierarchical feature learning on point sets in a metric space," *Advances in neural information processing systems*, vol. 30, 2017.

[3] G. Qian, Y. Li, H. Peng, J. Mai, H. Hammoud, M. Elhoseiny, and B. Ghanem, "PointNext: Revisiting PointNet++ with improved training and scaling strategies," *Advances in neural information processing systems*, vol. 35, pp. 23192–23204, 2022.

[4] D. Robert, H. Raguet, and L. Landrieu, "Efficient 3D semantic segmentation with superpoint transformer," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 17195–17204, 2023.

[5] G. Riegler, A. Osman Ulusoy, and A. Geiger, "OctNet: Learning deep 3D representations at high resolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3577–3586, 2017.

[6] Y. Siddiqui, L. Porzi, S. R. Bulò, N. Müller, M. Nießner, A. Dai, and P. Kontschieder, "Panoptic lifting for 3D scene understanding with neural fields," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9043–9052, 2023.

[7] Y. Bhalgat, I. Laina, J. F. Henriques, A. Zisserman, and A. Vedaldi, "Contrastive Lift: 3D object instance segmentation by slow-fast contrastive fusion," *arXiv preprint arXiv:2306.04633*, 2023.

[8] R. Zhu, S. Qiu, Q. Wu, K.-H. Hui, P.-A. Heng, and C.-W. Fu, "PCF-Lift: Panoptic lifting by probabilistic contrastive fusion," in *European Conference on Computer Vision*, pp. 92–108, Springer, 2024.

[9] J. Kerr, C. M. Kim, K. Goldberg, A. Kanazawa, and M. Tancik, "LERF: Language embedded radiance fields," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 19729–19739, 2023.

[10] M. Qin, W. Li, J. Zhou, H. Wang, and H. Pfister, "LangSplat: 3D language Gaussian splatting," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 20051–20060, 2024.

[11] J.-C. Shi, M. Wang, H.-B. Duan, and S.-H. Guan, "Language embedded 3D Gaussians for open-vocabulary scene understanding," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5333–5343, June 2024.

[12] K. Jun-Seong, G. Kim, K. Yu-Ji, Y.-C. F. Wang, J. Choe, and T.-H. Oh, "Dr. Splat: Directly referring 3D Gaussian splatting via direct language embedding registration," *arXiv preprint arXiv:2502.16652*, 2025.

[13] Y. Ji, H. Zhu, J. Tang, W. Liu, Z. Zhang, X. Tan, and Y. Xie, "FastLGS: Speeding up language embedded Gaussians with feature grid mapping," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 39, pp. 3922–3930, 2025.

[14] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, *et al.*, "Learning transferable visual models from natural language supervision," in *International conference on machine learning*, pp. 8748–8763, PMLR, 2021.

[15] M. Ye, M. Danelljan, F. Yu, and L. Ke, "Gaussian grouping: Segment and edit anything in 3D scenes," *arXiv preprint arXiv:2312.00732*, 2023.

[16] Y. Wu, J. Meng, H. Li, C. Wu, Y. Shi, X. Cheng, C. Zhao, H. Feng, E. Ding, J. Wang, *et al.*, "OpenGaussian: Towards point-level 3D Gaussian-based open vocabulary understanding," *arXiv preprint arXiv:2406.02058*, 2024.

[17] H. Li, Y. Wu, J. Meng, Q. Gao, Z. Zhang, R. Wang, and J. Zhang, "InstanceGaussian: Appearance-semantic joint Gaussian representation for 3D instance-level perception," *arXiv preprint arXiv:2411.19235*, 2024.

[18] S. Liu, Z. Zeng, T. Ren, F. Li, H. Zhang, J. Yang, Q. Jiang, C. Li, J. Yang, H. Su, *et al.*, "Grounding DINO: Marrying DINO with grounded pre-training for open-set object detection," in *European Conference on Computer Vision*, pp. 38–55, Springer, 2024.

[19] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "NeRF: Representing scenes as neural radiance fields for view synthesis," *Communications of the ACM*, vol. 65, no. 1, pp. 99–106, 2021.

[20] A. Chen, Z. Xu, A. Geiger, J. Yu, and H. Su, "TensoRF: Tensorial radiance fields," in *European Conference on Computer Vision*, pp. 333–350, Springer, 2022.

[21] L. Liu, J. Gu, K. Zaw Lin, T.-S. Chua, and C. Theobalt, "Neural sparse voxel fields," *Advances in Neural Information Processing Systems*, vol. 33, pp. 15651–15663, 2020.

[22] T. Müller, A. Evans, C. Schied, and A. Keller, "Instant neural graphics primitives with a multiresolution hash encoding," *ACM transactions on graphics (TOG)*, vol. 41, no. 4, pp. 1–15, 2022.

[23] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis, "3D Gaussian splatting for real-time radiance field rendering," *ACM Transactions on Graphics*, vol. 42, no. 4, 2023.

[24] T.-X. Xu, W. Hu, Y.-K. Lai, Y. Shan, and S.-H. Zhang, "Texture-GS: Disentangling the geometry and texture for 3D Gaussian splatting editing," *arXiv preprint arXiv:2403.10050*, 2024.

[25] Z. Liang, Q. Zhang, W. Hu, Y. Feng, L. Zhu, and K. Jia, "Analytic-Splatting: Anti-aliased 3D Gaussian splatting via analytic integration," *arXiv preprint arXiv:2403.11056*, 2024.

[26] Z. Zhang, W. Hu, Y. Lao, T. He, and H. Zhao, "Pixel-GS: Density control with pixel-aware gradient for 3D Gaussian splatting," *arXiv preprint arXiv:2403.15530*, 2024.

[27] K. Cheng, X. Long, K. Yang, Y. Yao, W. Yin, Y. Ma, W. Wang, and X. Chen, "GaussianPro: 3D Gaussian splatting with progressive propagation," in *Forty-first International Conference on Machine Learning*, 2024.

[28] B. Huang, Z. Yu, A. Chen, A. Geiger, and S. Gao, "2D Gaussian splatting for geometrically accurate radiance fields," in *ACM SIGGRAPH 2024 Conference Papers*, pp. 1–11, 2024.

[29] Z. Yu, A. Chen, B. Huang, T. Sattler, and A. Geiger, "Mip-Splatting: Alias-free 3D Gaussian splatting," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19447–19456, 2024.

[30] O. Seiskari, J. Ylilammi, V. Kaatrasalo, P. Rantalankila, M. Turkulainen, J. Kannala, E. Rahtu, and A. Solin, "Gaussian splatting on the move: Blur and rolling shutter compensation for natural camera motion," in *European Conference on Computer Vision*, pp. 160–177, Springer, 2024.

[31] D. Zhang, C. Wang, W. Wang, P. Li, M. Qin, and H. Wang, "Gaussian in the wild: 3D Gaussian splatting for unconstrained image collections," in *European Conference on Computer Vision*, pp. 341–359, Springer, 2024.

[32] J. Kulhanek, S. Peng, Z. Kukelova, M. Pollefeys, and T. Sattler, "WildGaussians: 3D Gaussian splatting in the wild," *arXiv preprint arXiv:2407.08447*, 2024.

[33] G. Kopanas, J. Philip, T. Leimkühler, and G. Drettakis, "Point-based neural rendering with per-view optimization," in *Computer Graphics Forum*, vol. 40, pp. 29–43, Wiley Online Library, 2021.

[34] B. Li, K. Q. Weinberger, S. Belongie, V. Koltun, and R. Ranftl, "Language-driven semantic segmentation," *arXiv preprint arXiv:2201.03546*, 2022.

[35] M. Caron, H. Touvron, I. Misra, H. Jégou, J. Mairal, P. Bojanowski, and A. Joulin, "Emerging properties in self-supervised vision transformers," in *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 9650–9660, 2021.

[36] R. Ding, J. Yang, C. Xue, W. Zhang, S. Bai, and X. Qi, "PLA: Language-driven open-vocabulary 3D scene understanding," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 7010–7019, 2023.

[37] K. M. Jatavallabhula, A. Kuwajerwala, Q. Gu, M. Omama, T. Chen, A. Maalouf, S. Li, G. Iyer, S. Saryazdi, N. Keetha, *et al.*, "ConceptFusion: Open-set multimodal 3D mapping," *arXiv preprint arXiv:2302.07241*, 2023.

[38] L. Jiang, S. Shi, and B. Schiele, "Open-vocabulary 3D semantic segmentation with foundation models," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 21284–21294, 2024.

[39] M. Liu, Y. Zhu, H. Cai, S. Han, Z. Ling, F. Porikli, and H. Su, "PartSLIP: Low-shot part segmentation for 3D point clouds via pretrained image-language models," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 21736–21746, 2023.

[40] S. Peng, K. Genova, C. Jiang, A. Tagliasacchi, M. Pollefeys, T. Funkhouser, *et al.*, "OpenScene: 3D scene understanding with open vocabularies," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 815–824, 2023.

[41] J. Yang, R. Ding, W. Deng, Z. Wang, and X. 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*, pp. 19823–19832, 2024.

[42] J. Zhang, R. Dong, and K. Ma, "CLIP-FO3D: Learning free open-world 3D scene representations from 2D dense CLIP," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2048–2059, 2023.

[43] M. E. A. Boudjoghra, A. Dai, J. Lahoud, H. Cholakkal, R. M. Anwer, S. Khan, and F. S. Khan, "Open-YOLO 3D: Towards fast and accurate open-vocabulary 3D instance segmentation," *arXiv preprint arXiv:2406.02548*, 2024.

[44] F. Engelmann, F. Manhardt, M. Niemeyer, K. Tateno, M. Pollefeys, and F. Tombari, "OpenNeRF: Open set 3D neural scene segmentation with pixel-wise features and rendered novel views," *arXiv preprint arXiv:2404.03650*, 2024.

[45] Y. Bhalgat, I. Laina, J. F. Henriques, A. Zisserman, and A. Vedaldi, "N2F2: Hierarchical scene understanding with nested neural feature fields," in *European Conference on Computer Vision*, pp. 197–214, Springer, 2024.

[46] J. Guo, X. Ma, Y. Fan, H. Liu, and Q. Li, "Semantic Gaussians: Open-vocabulary scene understanding with 3D Gaussian splatting," *arXiv preprint arXiv:2403.15624*, 2024.

[47] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, P. Dollar, and R. Girshick, "Segment anything," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4015–4026, October 2023.

[48] F. Li, H. Zhang, P. Sun, X. Zou, S. Liu, J. Yang, C. Li, L. Zhang, and J. Gao, "Semantic-SAM: Segment and recognize anything at any granularity," *arXiv preprint arXiv:2307.04767*, 2023.

[49] Y. Xiong, B. Varadarajan, L. Wu, X. Xiang, F. Xiao, C. Zhu, X. Dai, D. Wang, F. Sun, F. Iandola, *et al.*, "EfficientSAM: Leveraged masked image pretraining for efficient segment anything," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16111–16121, 2024.

[50] L. Ke, M. Ye, M. Danelljan, Y.-W. Tai, C.-K. Tang, F. Yu, *et al.*, "Segment anything in high quality," *Advances in Neural Information Processing Systems*, vol. 36, pp. 29914–29934, 2023.

[51] C. Zhang, D. Han, Y. Qiao, J. U. Kim, S.-H. Bae, S. Lee, and C. S. Hong, "Faster segment anything: Towards lightweight SAM for mobile applications," *arXiv preprint arXiv:2306.14289*, 2023.

[52] C. Zhang, D. Han, S. Zheng, J. Choi, T.-H. Kim, and C. S. Hong, "MobileSAMv2: Faster segment anything to everything," *arXiv preprint arXiv:2312.09579*, 2023.

[53] J. Cen, Z. Zhou, J. Fang, W. Shen, L. Xie, D. Jiang, X. Zhang, Q. Tian, *et al.*, "Segment anything in 3D with NeRFs," *Advances in Neural Information Processing Systems*, vol. 36, pp. 25971–25990, 2023.

[54] S. Choi, H. Song, J. Kim, T. Kim, and H. Do, "Click-Gaussian: Interactive segmentation to any 3D Gaussians," *arXiv preprint arXiv:2407.11793*, 2024.

[55] H. Ying, Y. Yin, J. Zhang, F. Wang, T. Yu, R. Huang, and L. Fang, "OmniSeg3D: Omniversal 3D segmentation via hierarchical contrastive learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 20612–20622, 2024.

[56] C. M. Kim, M. Wu, J. Kerr, K. Goldberg, M. Tancik, and A. Kanazawa, "GARField: Group anything with radiance fields," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 21530–21539, 2024.

[57] A. Kirillov, K. He, R. Girshick, C. Rother, and P. Dollár, "Panoptic segmentation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9404–9413, 2019.

[58] M. C. Silva, M. Dahaghin, M. Toso, and A. Del Bue, "Contrastive Gaussian clustering: Weakly supervised 3D scene segmentation," *arXiv preprint arXiv:2404.12784*, 2024.

[59] R. Zhu, S. Qiu, Z. Liu, K.-H. Hui, Q. Wu, P.-A. Heng, and C.-W. Fu, "Rethinking end-to-end 2D to 3D scene segmentation in Gaussian splatting," *arXiv preprint arXiv:2503.14029*, 2025.

[60] E. A. Nadaraya, "On estimating regression," *Theory of Probability & Its Applications*, vol. 9, no. 1, pp. 141–142, 1964.

[61] H. Li, R. Qin, Z. Zou, D. He, B. Li, B. Dai, D. Zhang, and J. Han, "LangSurf: Language-embedded surface Gaussians for 3D scene understanding," *arXiv preprint arXiv:2412.17635*, 2024.

[62] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner, "Scannet: Richly-annotated 3D reconstructions of indoor scenes," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5828–5839, 2017.

[63] X. Zhai, B. Mustafa, A. Kolesnikov, and L. Beyer, "Sigmoid loss for language image pre-training," in *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 11975–11986, 2023.

[64] N. Ravi, V. Gabeur, Y.-T. Hu, R. Hu, C. Ryali, T. Ma, H. Khedr, R. Rädle, C. Rolland, L. Gustafson, *et al.*, "SAM 2: Segment anything in images and videos," *arXiv preprint arXiv:2408.00714*, 2024.

[65] Y. Zheng, X. Chen, Y. Zheng, S. Gu, R. Yang, B. Jin, P. Li, C. Zhong, Z. Wang, L. Liu, *et al.*, "GaussianGrasper: 3D language Gaussian splatting for open-vocabulary robotic grasping," *IEEE Robotics and Automation Letters*, 2024.

[66] W. Tang, J.-H. Pan, Y.-H. Liu, M. Tomizuka, L. E. Li, C.-W. Fu, and M. Ding, "GeoManip: Geometric constraints as general interfaces for robot manipulation," *arXiv preprint arXiv:2501.09783*, 2025.

[67] A. Rashid, S. Sharma, C. M. Kim, J. Kerr, L. Y. Chen, A. Kanazawa, and K. Goldberg, "Language embedded radiance fields for zero-shot task-oriented grasping," in *7th Annual Conference on Robot Learning*, 2023.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: We clarify our contributions in Sec. 1, and experimentally verify them in Sec. 4.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: We discuss the current limitations regarding 3D reasoning segmentation for handling complicated queries in Sec. 5.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

Answer: [NA]

Justification: This paper does not involve theoretical results.

Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: We disclose the experimental information in Sec. 3.4, Sec. 4.1, Sec. 4.2, and the supp.

   Guidelines:

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

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

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

Answer: [Yes]

Justification: Both data and code of our work will be publicly available on github.

Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: The experimental settings and details are presented in Sec. 4.1, Sec. 4.2, and the supp.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [No]

   Justification: We follow the common practice of existing works to report the experimental results, particularly OpenGaussian (published in NeurIPS 2024)

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

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

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: We provide machine information in Sec. 3.4. We provide the training time and inference time in the ablation study Sec. 4.3.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: We have already confirmed with the NeurIPS Code of Ethics.

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [Yes]

    Justification: We discuss possible societal impacts of it in Sec. 4.4 about its robitics applications.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

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

    Answer: [NA]

    Justification: The paper does not involve the release of data or models that pose a high risk for misuse.

    Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: We apply LeRF and ScanNetv2 dataset and 3D-GS, OpenGaussian codebase for our method implementation and testing, all of them are publicly available and have already used in published papers.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

    Answer: [NA]

    Justification: The paper does not release new assets.

    Guidelines:

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

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

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

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

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

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

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

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

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

    Answer: [NA]

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

    Guidelines:

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