Unifying 3D Vision-Language Understanding via Promptable Queries

Ziyu Zhu[1,2†⋆], Zhuofan Zhang[1,2], Xiaojian Ma[2], Xuesong Niu[2], Yixin Chen[2], Baoxiong Jia[2], Zhidong Deng[1✉†], Siyuan Huang[2✉], Qing Li[2✉]

[1]Tsinghua University [2]State Key Laboratory of General Artificial Intelligence, BIGAI
pq3d.github.io

Abstract. A unified model for 3D vision-language (3D-VL) understanding is expected to take various scene representations and perform a wide range of tasks in a 3D scene. However, a considerable gap exists between existing methods and such a unified model, due to the independent application of representation and insufficient exploration of 3D multi-task training. In this paper, we introduce **PQ3D**, a unified model capable of using **P**romptable **Q**ueries to tackle a wide range of **3D**-VL tasks, from low-level instance segmentation to high-level reasoning and planning. This is achieved through three key innovations: (1) unifying various 3D scene representations (*i.e.*, voxels, point clouds, multi-view images) into a shared 3D coordinate space by segment-level grouping, (2) an attention-based query decoder for task-specific information retrieval guided by prompts, and (3) universal output heads for different tasks to support multi-task training. Tested across ten diverse 3D-VL datasets, PQ3D demonstrates impressive performance on these tasks, setting new records on most benchmarks. Particularly, PQ3D improves the state-of-the-art on ScanNet200 by 4.9% (AP25), ScanRefer by 5.4% (acc@0.5), Multi3DRefer by 11.7% (F1@0.5), and Scan2Cap by 13.4% (CIDEr@0.5). Moreover, PQ3D supports flexible inference with individual or combined forms of available 3D representations, *e.g.*, solely voxel input.
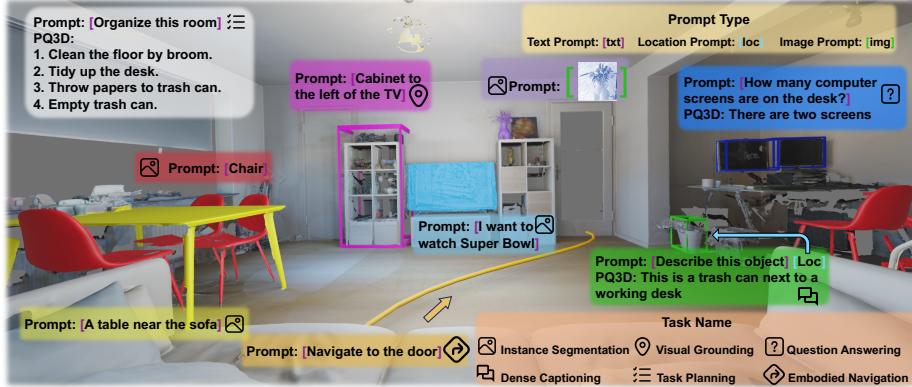
Keywords: 3D Vision-Language · 3D Scene Understanding · Visual Reasoning

1 Introduction

Recent advancements in embodied artificial intelligence have emphasized the importance of connecting 3D scene understanding with natural language [21,32, 34,52,79]. This step is crucial for embodied agents to understand and execute human instructions in real-world scenarios [5,59]. In recent years, numerous tasks and datasets for benchmarking 3D scene understanding with languages have been proposed, including 3D semantic segmentation [60], 3D vision-language

---

⋆ Work done as an intern at BIGAI. ✉ Corresponding author. † BNRist, THUAI, Department of Computer Science, Tsinghua University, Beijing 100084, China.

Fig. 1: PQ3D is a unified model for 3D vision-language understanding, capable of taking various prompts (object categories, referring sentences, images, locations) to perform a wide range of tasks in a 3D scene, including instance segmentation, visual grounding, question answering and dense captioning. Remarkably, PQ3D can take a novel prompt type unseen during training, *e.g.*, an image sketch of a vase, to locate the related object in the scene. If further instruction-tuned with a large language model and plugged into an embodied agent, PQ3D can also plan a complex task and navigate the agent to desired objects.

(3D-VL) reasoning (referring [1, 2, 8, 37, 74], question answering [3, 49, 76], and captioning [12]), and open-vocabulary 3D understanding [25, 27, 39, 64].

The state-of-the-art (SOTA) approaches typically address these tasks utilizing specific scene representations [3, 10, 26, 28, 48, 62]. These representations can all be derived from RGB-D streams, each with unique advantages and drawbacks. For instance, voxels offer a uniform, grid-like structure ideal for instance segmentation [60, 62] but struggle to capture fine geometric details. Point clouds provide detailed spatial information crucial for visual grounding [10, 79], but they miss texture details. Multi-view images include a rich visual context beneficial for open-vocabulary scene understanding [53, 64] but lack accurate 3D location. The full potential of combining all these representations for holistic scene understanding has not been well studied. Additionally, several works in 3D-VL reasoning explored multi-task training [7, 13] and pretraining [73, 79] to seek mutual benefits from related tasks; however, they still require an off-the-shelf 3D mask proposal module. Thus, despite substantial progress, there is still a gap toward a **unified model** achieving comprehensive 3D-VL understanding, from low-level instance segmentation to high-level reasoning and planning.

We expect such a unified model for 3D scene understanding to integrate various scene representations and select appropriate features based on specific task instructions. However, two significant challenges exist: Firstly, these different scene representations have varying granularities, making it difficult to unify them in a single feature space. Secondly, there have been limited research efforts

regarding developing a unified training paradigm that can accommodate 3D-VL tasks at different levels with diverse input instructions and output formats.

To address these challenges, we introduce **PQ3D**, a unified model using **P**romptable **Q**ueries to concurrently manage various **3D** scene *representations*, *prompts*, and *outputs* in numerous 3D-VL tasks, as depicted in Fig. 1. An overview of the proposed PQ3D model is shown in Fig. 2(c), featuring three key innovations: (1) We unify dense point cloud features with multi-scale voxel features and multi-view image features into a shared 3D coordinate space. This process involves the unsupervised grouping of 3D points into larger segments and pooling features to the segment level, significantly reducing the number of points and facilitating training. (2) A novel attention-based query decoder is introduced that progressively retrieves task-specific information from aligned scene features under the guidance of task prompts. (3) Each query is processed through three universal output heads to predict an instance mask, a task-relevance score, and a sentence; these are then combined to produce the required task outputs.

We conduct extensive experiments on ten 3D-VL datasets, including Scan-Net200 [60] / Replica [63] for instance segmentation, ScanRefer [8] / ReferIt3D [2] / Multi3DRef [74] for visual grounding, ScanQA [3] / SQA3D for question-answering, Scan2Cap [12] for dense captioning, and ObjNav from CortexBench [50] for embodied navigation. The proposed PQ3D achieves impressive results across these tasks, setting new records on most tasks as shown in Fig. 2(a). For example, our model boosts the state-of-the-art on ScanNet200 by 4.9% (AP25), ScanRefer by 5.4% (acc@0.5), Multi3DRef by 11.7% (F1@0.5), and Scan2Cap by 13.4% (CIDEr@0.5). More importantly, PQ3D is the first **unified model** capable of handling all these tasks simultaneously. The proposed model also shows zero-shot capability with novel prompt types; for instance, we can prompt it with an image sketch to locate all related objects in a scene as in Fig. 1.
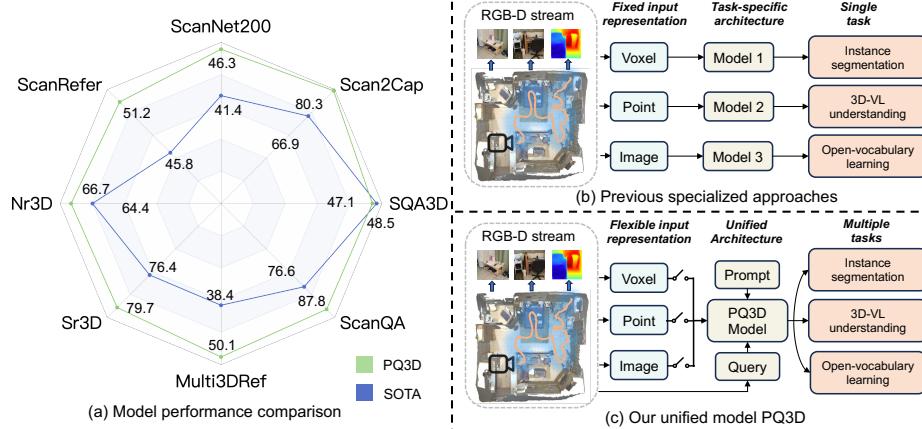
Our main contributions can be summarized as follows:

– We introduce PQ3D, a unified model adept at solving a broad spectrum of 3D-VL tasks with promptable queries. The tasks range from low-level instance segmentation to visual grounding, and high-level reasoning and planning.
– Our model uniquely aligns voxels, point clouds, and multi-view images into a shared 3D space and employs an attention-based query decoder to adaptively extract task-relevant features guided by prompts, offering a flexible approach to model all 3D-VL tasks.
– In our extensive experimentation across various 3D-VL tasks, PQ3D not only achieves competitive results but also sets new records in most of the tasks.

2   Related Work

2.1   3D Vision-language Learning

In recent times, there has been a surging interest in the field of 3D vision-language (3D-VL) learning. 3D-VL tasks establish a vital connection between the physical world and natural language, contributing to the development of

Fig. 2: Comparison between PQ3D and other models. (a) When comparing PQ3D to other state-of-the-art (SOTA) methods, PQ3D demonstrates superior performance on most tasks. (b) Previous models have been designed for specific tasks and representations, often limiting the potential for developing a unified model. (c) Our PQ3D can flexibly accommodate various input representations, effectively addressing a wide range of tasks.

embodied intelligence [37, 79]. In this emerging domain, objects refer [1, 2, 8, 74], question answering [3, 71], and dense captioning [12] tasks are introduced to evaluate natural language grounding concerning 3D object properties and relationships. Additional tasks include embodied question answering [49] and navigation [61], which explore models' capabilities within an embodied environment.

Numerous models have been proposed to tackle these benchmarks in task-specific ways. MVT [33], ViL3DRel [10], and ViewRefer [26] perform 3D visual grounding by explicitly incorporating spatial relation information into their designs. While these endeavors [10, 26, 33, 36, 48, 66, 70, 75] have demonstrated impressive results, they often require specialized model architectures tailored to specific tasks. In contrast, our model is unified, performing multiple tasks in one model. In a recent line of research, several works [11, 30, 32] explore 3D instruction tuning to leverage the power of large language model (LLM) to enhance 3D reasoning. PQ3D can further benefit these models by providing a robust 3D scene encoding module guided by language, which can be seamlessly integrated into these instruction tuning pipelines.

2.2   Query-based Model

Query-based models have gained prominence in 2D perception [20, 22, 29, 31, 78]. DETR [78] has allowed queries to attend to visual features from the backbone network dynamically, enabling object detection without the need for post-processing techniques like NMS [22, 78]. Building upon this concept, subsequent works such as [14, 22, 29] have extended the query-based approach to universal

image segmentations, where queries are employed to match pixel-level features and generate high-quality masks. Additionally, the use of queries initialized from language has led to advancements in referring segmentation [17,65].

More recent literature has further demonstrated the potential of query-based models in achieving unified understanding [35,43,69,80]. For instance, Perceiver-IO [35] leverages queries to extract information from multi-modal features, while XDecoder [80] formulates multiple vision-language tasks within a generalized decoding framework. However, in the context of 3D vision-language understanding, a query-based paradigm for handling multiple tasks is still lacking.

2.3   Promptable Segmentation

The concept of promptable segmentation, as presented in the SAM framework [40], centers around the utilization of prompts to direct the process of segmentation. These prompts can manifest in diverse forms, such as foreground/background points, approximate bounding boxes or masks, and free-form text, which furnish guidance on the desired elements to be segmented in an image [40, 46, 47].
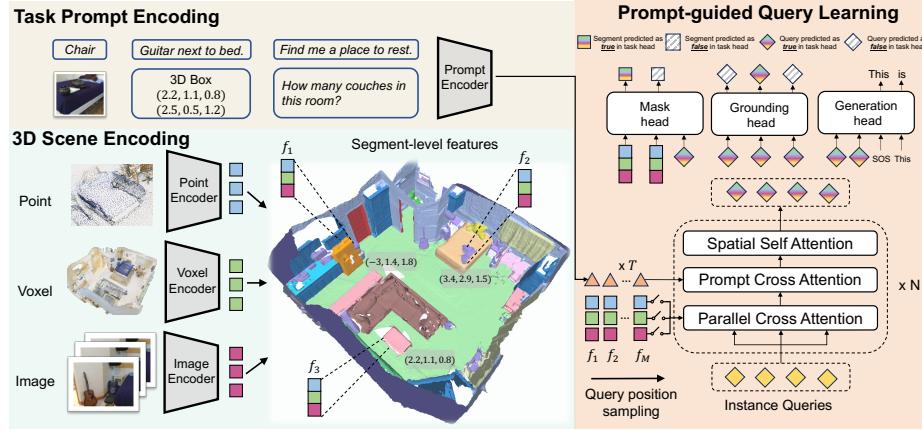
Promptable segmentation is intimately linked with open-vocabulary learning. Previous studies such as OWL-ViT [51] and OVR-CNN [72] have leveraged contrastive learning with extensive image-text pairs to demonstrate object detection generalization capabilities [51, 72, 77]. OpenSeg [24] and OVSeg [45] extend open-vocabulary detection from the object level to pixel-level segmentation [24,42,45,56,67,68]. In the field of 3D vision, methods like OpenMask3D [64] and OpenScene [53] have used 2D features to accomplish zero-shot segmentation. The PLA model [18] harnessed captions to align vision and language for novel-class instance segmentation. Our work takes cues from these existing methodologies, leveraging the benefits of prompting images, text, and point-based data to achieve open-vocabulary promptable segmentation.

3   Method

In this section, we present PQ3D, which consists of three main modules: Task Prompt Encoding, 3D Scene Encoding, and Prompt-guided Query Learning, as depicted in Fig. 3. Next, we will explain the details of each module.

3.1   Task Prompt Encoding

In various 3D-VL tasks, a task prompt can be of diverse formats, including object categories, referring sentences, questions, 3D bounding boxes, 3D locations, images, *etc*. Diverse prompt formats are one of the key obstacles to building a unified model for 3D-VL understanding. However, we figured out that all these diverse prompts can be divided into three categories: *textual*, *visual*, and *numerical*. We encode the textual and visual prompts by the pre-trained CLIP model [57], which allows us to train using a text prompt and perform inference using an image prompt in a zero-shot manner. Numerical prompts, *e.g.*,

Fig. 3: The model architecture of PQ3D, which consists of **Task Prompt Encoding**, **3D Scene Encoding**, and **Prompt-guided Query Learning** modules. In prompt encoding, task prompts in diverse formats are projected to a shared feature space. In scene encoding, point clouds, voxel grids, and multi-view images of a scene are first encoded by corresponding encoders and then aligned into a shared 3D coordinate space. The prompt-guided query learning module takes in zero-initialized instance queries and progressively retrieves task-relevant information from aligned scene features under the guidance of task prompts. Finally, each updated instance query is fed into three output heads to predict an instance mask, a task-relevance score, and a sentence.

3D bounding boxes and locations, are projected by fully-connected layers into the same feature space as CLIP. With such unification, we do not distinguish different prompt formats anymore and this design enables the model to transfer knowledge between different prompts. The encoded task prompt is denoted as $\mathbf{t} \in \mathcal{R}^{T \times D}$, T is the number of prompt tokens, and D is the hidden dimension.

3.2   3D Scene Encoding

There are three widely used representations for 3D scenes: point clouds, voxel grids, and multi-view images, which have their unique advantages in different tasks. For example, voxels offer a uniform, grid-like structure ideal for instance segmentation [60, 62], point clouds provide spatial information crucial for visual grounding [10, 79], and multi-view images include a rich visual context beneficial for open-vocabulary learning [53, 64].

To simultaneously handle various tasks in a 3D scene, we aim to achieve seamless unification of these representations. To accomplish this, we first group the dense 3D points into larger segments through unsupervised graph-based segmentation and masks [23]. The grouped segments are much fewer than the original points, making the training feasible by reducing the number of tokens in cross-attention. Then, we encode these scene representations by the corresponding encoders and pool the features to the segments in total of M.

Point Cloud To process the point cloud of a 3D scene, we first partition the full point cloud into the pre-generated segments. For each segment, we sample 1,024 points, normalize their coordinates into a unit sphere and then feed them into a pre-trained PointNet++ backbone [54, 55] to obtain the point features, denoted by $\mathbf{P} = \{\mathbf{p}_0, \mathbf{p}_1, ..., \mathbf{p}_M\} \in \mathcal{R}^{M \times D}$, where M is the number of segments.
Voxel To extract voxel features, we follow Mask3D [62] to first discretize the 3D scene into voxels and then channel these voxels into a sparse convolutional U-net backbone. The sparse U-Net includes downsampling and upsampling stages to extract hierarchical information from the given scene. The extracted features of each voxel are remapped to the pre-generated segments and average-pooled to obtain segment-level features. We also apply a linear layer to project the voxel features to the hidden dimension D, denoted by $\mathbf{V} = \{\mathbf{v}_0, \mathbf{v}_1, ..., \mathbf{v}_M\} \in \mathcal{R}^{M \times D}$.
Multi-View Image Given multi-view images of a 3D scene, we follow Open-Scene [53] to get the multi-view image features for each 3D point. We first compute per-pixel embeddings for each image using the pre-trained OpenSeg [24] segmentation model. We then back-project each 2D pixel into 3D point and aggregate the features from the associated multi-view pixels. Finally, we obtain the segment-level image features by simply average pooling, denoted as $\mathbf{I} = \{\mathbf{i}_0, \mathbf{i}_1, ..., \mathbf{i}_M\} \in \mathcal{R}^{M \times D}$.

The final 3D scene representation is composed of these three segment-level features $\{\mathbf{V}, \mathbf{I}, \mathbf{P}\}$. Besides, we add positional encoding to segment-level features. In our implementation, We compute the average coordinate across all points within each segment as its 3D location, which is further encoded by an MLP into the same hidden dimension, denoted as $\mathbf{L} = \{\mathbf{l}_0, \mathbf{l}_1, ..., \mathbf{l}_M\} \in \mathcal{R}^{M \times D}$. Unifying all features to the pre-generated segments is crucial for instance queries to effectively interact with these features in query learning.

3.3 Prompt-guided Query Learning

We propose a novel Transformer-like decoder to instruct the instance queries to assimilate scene and prompt information. This process begins with a set of instance queries $\mathbf{Q}_0$, whose values are initialized to zeros and positions are sampled via Farthest Point Sampling from 3D points [64]. Within the decoder layer l, the instance queries $\mathbf{Q}_l$ retrieve task-relevant information by first attending to the scene features $\{\mathbf{V}, \mathbf{I}, \mathbf{P}\}$ in parallel and then the task prompt $\mathbf{t}$, followed by a spatial self-attention as [79]. All attention layers are followed by the forward layer (FFN). The spatial self-attention utilizes location information from the coordinates of the farthest sampled points. Formally, we have:

$$\mathbf{Q}_l^{'} = \text{FFN}(\text{Norm}(\mathbf{Q}_l + \sum_{\mathbf{F} \in \{\mathbf{V}, \mathbf{I}, \mathbf{P}\}} \text{MaskedCrossAttn}(\mathbf{Q}_l, \mathbf{F}))), \qquad (1)$$

$$\mathbf{Q}_l^{''} = \text{FFN}(\text{Norm}(\mathbf{Q}_l^{'} + \text{CrossAttn}(\mathbf{Q}_l^{'}, \mathbf{t}))), \qquad (2)$$

$$\mathbf{Q}_{l+1} = \text{FFN}(\text{Norm}(\text{SpatialSelfAttn}(\mathbf{Q}_l^{''}))). \qquad (3)$$

Following [14, 62], we adopt masked attention when cross-attending to the scene features, which restricts the attention to localized features centered around the

query for faster convergence and improved performance. To support flexible inference when only some representations are available, we randomly drop out some scene features with rate 0.6 in masked-attention computation during training. After N decoder layers, we expect the final instance queries $\mathbf{Q}$ to have collected enough information for solving the given task.

3.4   Output Heads and Losses

We adopt the following three output heads to support a variety of 3D-VL tasks:
Mask head For each instance query, we apply a mask head to predict a binary mask over the pre-generated segments. Formally, we have:

$$p_{\mathrm{mask}} = \sigma(f_s(\mathbf{V} + \mathbf{I} + \mathbf{P}) \cdot f_q(\mathbf{Q})^T) \tag{4}$$

where f_s, f_q are projection layers. The dimension of $f_s(\mathbf{V} + \mathbf{I} + \mathbf{P})$ is $M \times D$ and $f_q(\mathbf{Q})$ is $Q \times D$, where M, Q, and D represent the number of segments, queries, and hidden dimensions, respectively. The multiplication of f_s and f_q followed by sigmoid function σ results in a binary mask in $M \times Q$ dimension. During training, we follow [62, 78] to apply Hungarian Matching between queries and ground-truth objects, then calculate the mask loss:

$$\mathcal{L}_{\mathrm{mask}} = \lambda_{\mathrm{bce}}\mathcal{L}_{\mathrm{bce}} + \lambda_{\mathrm{dice}}\mathcal{L}_{\mathrm{dice}} \tag{5}$$

where $\mathcal{L}_{\mathrm{bce}}$ is the binary cross-entropy loss and $\mathcal{L}_{\mathrm{dice}}$ is the Dice loss [14, 29, 64].
Grounding head We apply a grounding head f_g to predict if an instance query is related to the task. f_g is implemented as linear projection layers. Formally, we have:

$$p_{\mathrm{grd}} = \sigma(f_g(\mathbf{Q})) \tag{6}$$

During training, if grounding labels are provided as supervision, we calculate a binary cross-entropy loss as the grounding loss $\mathcal{L}_{\mathrm{grd}}$.
Generation head We choose the decoder of a pre-trained T5-small [15, 58] as the generation head to generate a text response, using all instance queries as the encoded inputs. During training, if text responses are provided as supervision for dense caption and QA task, we calculate the cross-entropy loss as the generation loss $\mathcal{L}_{\mathrm{gen}}$.

During training, the total loss is the weighted sum of losses from the above three heads:

$$\mathcal{L}_{\mathrm{total}} = \lambda_{\mathrm{mask}}\mathcal{L}_{\mathrm{mask}} + \lambda_{\mathrm{grd}}\mathcal{L}_{\mathrm{grd}} + \lambda_{\mathrm{gen}}\mathcal{L}_{\mathrm{gen}} \tag{7}$$

4   Experiments

4.1   Experimental setting

Training Datasets Tab. 1 shows a summary of the datasets used for the multitask training of PQ3D. Notably, we combine eight datasets for training, including about 662K training samples for various tasks.

Training Details The training procedure consists of two stages. In the first stage, we train the model with instance segmentation alone on ScanNet200 for 800 epochs. At this stage, instance segmentation is trained with a classification head on instance queries, instead of prompting object categories. At the second stage, we continue train the model on the full training set using the training objectives in Eq. (7) for 50 epochs. We set the hidden dimension D to 768, and query decoder layer N to 4. We utilize the AdamW optimizer with a learning rate of 1e-4, batch size of 16, $\beta_1 = 0.9$, and $\beta_2 = 0.98$. The loss balance weights $\lambda_{\mathrm{mask}}, \lambda_{\mathrm{gen}}$ are set to 1, and λ_{grd} is set to 10. To further demonstrate the capability of PQ3D, we also transfer it to an embodied agent for object navigation using the ObjNav task from CortexBench [50] and instruction-tune it with a large language model (LLM) Vicuna-7B using the instruction-following dataset from [32]. The whole training process is conducted on four NVIDIA A100 GPUs. More details can be found in the appendix.

Table 1: Datasets for unified training. The size of ScanNet200 is #scenes (1202) × #categories (200).

Dataset	Task	Prompt	Heads	Size
ScanNet200 [60]	instance segmentation	category	mask,grounding	240K
ScanRefer [8]	visual grounding	sentence	grounding	37K
Nr3D [2]	visual grounding	sentence	grounding	119K
Sr3D [2]	visual grounding	sentence	grounding	66K
Multi3DRefer [74]	visual grounding	sentence	grounding	44K
ScanQA [3]	question answering	question	grounding,generation	30K
SQA3D [49]	question answering	question	generation	89K
Scan2Cap [12]	dense captioning	3D box	generation	37K
Total	-	-	-	662K

4.2 Quantative Results

Instance segmentation on ScanNet200 As shown in Tab. 2, PQ3D demonstrates SOTA performance for instance segmentation tasks on ScanNet200. Our approach achieves 20.2% for AP, 28.0% for AP_{50}, and 32.5% for AP_{25} in promptable manner, which uses the output logits from the ground head after prompting all classes in ScanNet200. Notably, our methods surpass other open-vocabulary approaches, offering a more versatile language interface. However, our model's performance with tail classes is relatively less robust due to biases in the CLIP text encoder, which is analyzed in the appendix. To mitigate this issue, we incorporate a closed-vocabulary task head, which enables our model to surpass the closed-vocabulary SOTA method, Mask3D [62], especially in the tail classes.

Zero-shot transfer to Replica In addition to evaluating our method's performance on ScanNet200, we investigate its generalization capabilities to an out-

of-distribution dataset Replica [63]. Our observations reveal that PQ3D displays improved generalization abilities on unseen data, surpassing other open-vocabulary approaches in terms of AP, AP_{50}, and AP_{25} metrics by 0.2%, 4.1%, and 7.7%. These findings imply a notable capability of PQ3D for effective transfer to different datasets.

Table 2: Instance Segmentation results on the ScanNet200 validation set and zero-shot performance on Replica. The Average Precision (AP) is averaged over an overlapping range, and the AP_{50}, AP_{25} is evaluated at 50% and 25% overlaps. Additionally, we provide AP scores for the head, common, and tail classes. The notation "PQ3D (*w/cls*)" represents results using a closed-vocabulary classification head, whereas "PQ3D (*prompt*)" denotes segmentation results in a promptable way, which corresponds to an open-vocabulary setting.

Model	ScanNet200						Replica		
	AP	AP_{50}	AP_{25}	head	common	tail	AP	AP_{50}	AP_{25}
Closed-vocabulary									
Mask3D [62]	26.9	36.2	41.4	**39.8**	21.7	17.9	-	-	-
PQ3D (*w/cls*)	**27.0**	**38.9**	**46.3**	35.8	**24.2**	**20.0**	-	-	-
Open-vocabulary									
OpenScene [53]	11.7	15.2	17.8	13.4	11.6	9.9	10.9	15.6	17.3
OpenMask3D [64]	15.4	19.9	23.1	17.1	14.1	**14.9**	13.1	18.4	24.2
PQ3D (*prompt*)	**20.2**	**28.0**	**32.5**	**30.9**	**17.0**	11.3	**13.3**	**22.5**	**31.9**

Visual Grounding Tab. 3 provides an assessment of grounding accuracy for various methods on four benchmarks: ScanRefer, Nr3D, Sr3D, and Multi3DRefer. Our model consistently outperforms the other methods in most categories. On the ScanRefer, Nr3D, and Sr3D benchmarks, our model outperforms SOTA by 5.4%, 2.3%, and 3.3%, respectively. Furthermore, on the Multi3DRefer benchmark, our model outperforms others in the ST (single target) and MT (multiple targets) categories and achieves the highest average score of 50.1%. In the ZT (zero target) metric, our model lags behind the state-of-the-art. This performance gap could potentially be attributed to the fact that ZT is only present in the Multi3DRefer dataset. However, our model trained only on the Multi3DRefer dataset "PQ3D (*sg.*)" exhibits better performance in the ZT and MT metric, but falls short of the unified trained model in other categories.

Question Answering On the ScanQA test set, PQ3D outperforms all other methods in terms of the BLEU-1, METEOR, and CIDEr metrics. Specifically, our model surpasses SOTA by 8.6% / 5.4% for BLEU-1, 2.6% / 1.0% for METEOR, 11.2% / 2.6% for CIDEr on "w/ object" and "w/o object" test set. However, in terms of the EM@1 metric, the 3D-VisTA method outperforms our model with scores of 27.0% "w/ object" and 23.0% "w/o object", compared to our model's 26.1% and 20.0%, respectively. Different from 3D-VisTA, our model does not use a classification head for QA, which causes a performance drop in EM metric.

Table 3: Grounding accuracy (%) on 3D visual grounding benchmarks. The results of ScanRefer and Multi3DRefer are reported under IoU@0.5. The results of Nr3D and Sr3D are reported using ground-truth masks during masked cross-attention. The ZT and ST results from Multi3DRefer are with distractors of the same class. "PQ3D (*sg.*)" signifies a model trained on a single dataset.

| Method | ScanRefer | | | Nr3D | | | Sr3D | | | Multi3DRefer | | | |
	Unique	Multiple	Avg.	Easy	Hard	Avg.	Easy	Hard	Avg.	ZT	ST	MT	Avg.
ViL3DRel [10]	68.6	30.7	37.7	70.2	57.4	64.4	74.9	67.9	72.8	-	-	-	-
3DJCG [7]	64.3	30.8	37.3	-	-	-	-	-	-	**66.9**	16.7	26.2	26.6
UniT3D [13]	73.1	31.1	39.1	-	-	-	-	-	-	-	-	-	-
M3DRef-CLIP [74]	77.2	36.8	44.7	55.6	43.4	49.4	-	-	-	39.4	30.6	37.9	38.4
3D-VisTA [79]	75.1	39.1	45.8	72.1	56.7	64.2	78.8	71.3	76.4	-	-	-	-
PQ3D (*sg.*)	76.6	42.0	47.4	73.3	56.7	64.9	78.8	68.2	75.6	61.1	40.5	**41.7**	48.6
PQ3D	**78.2**	**46.2**	**51.2**	**75.0**	**58.7**	**66.7**	**82.7**	**72.8**	**79.7**	57.7	**43.6**	40.9	**50.1**

Table 4: Answer accuracy on ScanQA. Each entry denotes "test w/ object" and "test w/o object". EM@1 refers to the top 1 exact match accuracy, while BLEU-1, METEOR, and CIDEr denote text similarity scores between the predicted answer and the ground-truth answer. The notation "PQ3D (*sg.*)" indicates a model trained on a single dataset rather than through unified joint training.

Method	EM@1	BLEU-1	METEOR	CIDEr
ScanQA [3]	23.5 / 20.9	31.6 / 30.7	13.6 / 12.6	67.3 / 60.2
3D-VisTA [79]	**27.0** / **23.0**	34.4 / 30.2	15.2 / 12.9	76.6 / 62.6
PQ3D (*sg.*)	18.9 / 16.1	34.7 / 30.5	14.5 / 12.1	69.3 / 56.0
PQ3D	26.1 / 20.0	**43.0** / **36.1**	**17.8** / **13.9**	**87.8** / **65.2**

For the SQA3D task, it is worth noting that our proposed model falls slightly behind the SOTA in SQA3D, with a difference of 1.4%. As our model utilizes the CLIP text encoder, it may face limitations in understanding long sentences.

Dense Captioning For the dense captioning task, PQ3D outperforms all other models in the CIDEr, METEOR, and ROUGE metrics. With a CIDEr score of 80.3%, it significantly surpasses the next best, 3D-VisTA, which has a CIDEr score of 66.9%. Importantly, the performance of PQ3D trained on multiple tasks and datasets exceeds that of PQ3D trained on a single task and dataset, showcasing the effectiveness of multi-task joint training.

Object Navigation To further verify PQ3D's effectiveness, we finetune it on the ObjNav task from CortexBench [50] for navigating an embodied agent to a desired object. The proposed PQ3D provides global 3D features to the navigation agent that can improve the baseline VC-1 by a significant margin, achieving a 22.9% increase in success rate. Importantly, our method showcases consistent and stable performance, even in the absence of a compass or directional guidance.

Table 5: Answer accuracy on SQA3D under question types.

Method	Test set						Avg.
	What	Is	How	Can	Which	Other	
ClipBERT [49]	30.2	60.1	38.7	63.3	42.5	42.7	43.3
SQA3D [49]	31.6	**63.8**	**46.0**	69.5	43.9	45.3	46.6
3D-VisTA [79]	34.8	63.3	45.4	**69.8**	**47.2**	**48.1**	**48.5**
PQ3D (*sg.*)	35.6	62.7	45.2	66.3	43.3	43.3	46.8
PQ3D	**37.1**	61.3	44.5	60.9	47.0	45.1	47.1

Table 6: Captioning results on Scan2Cap under IoU@0.5 with various text similarity scores. "PQ3D (*sg.*)" indicates a model trained on a single dataset rather than through unified training.
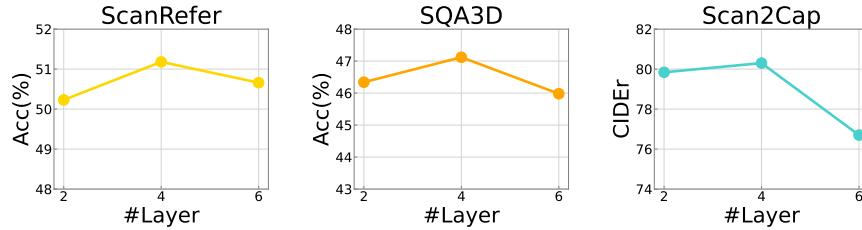
Table 7: Results on ObjNav from CortexBench [50]. Note we reproduce the result of "VC-1 (ViT-B)" ourselves due to the slight mismatch we have found. Only variants with PQ3D use 3D input. The result with * is also without a compass sensor.

Method	CIDEr	BLEU-4	METEOR	ROUGE
Scan2Cap [12]	35.2	22.4	21.4	43.5
3DJCG [7]	47.7	31.5	24.3	51.8
3D-VisTA [79]	66.9	34.0	27.1	54.3
PQ3D (*sg.*)	75.6	34.4	28.6	57.1
PQ3D	**80.3**	**36.0**	**29.1**	**57.9**

Model	Success ↑	SPL ↑	Soft-SPL ↑
VC-1 (ViT-B) [50]	57.1	0.31	0.41
PQ3D	**80.0**	**0.50**	**0.60**
PQ3D w/o GPS*	75.0	0.45	0.50

4.3   Ablation study

Query Decoder Depth In this study, we examine the influence of decoder depth on downstream tasks. The results suggest that a 4-layer decoder outperforms both 2-layer and 6-layer ones on all tasks. Consequently, we choose a 4-layer query decoder for PQ3D.

Fig. 4: Ablation study of query decoder depth.

Scene Features Our result in Tab. 8 reveals the benefits of incorporating voxel, point, and image features on refering (ScanRefer Acc), question answering (SQA3D Acc), and captioning (Scan2Cap CIDEr) tasks. Specifically, the addition of point features results in performance changes of $+3.1\%$, $+1.7\%$, $+6.8\%$ and while the inclusion of image features, based on voxel and point, leads to

improvements of $+2.0\%$, $+1.7\%$, and $+5.7\%$ on these respective tasks. These results exemplify the effectiveness of our feature alignment approach, as it enables enhanced performance across grounding, QA, and captioning tasks.

Flexible Inference To assess how our model performs under the constraints of limited scene representations, we conduct inference while omitting certain scene features, using a single set of model weights for testing. From Tab. 8, we can observe that PQ3D achieves comparable performance with the model trained with specific scene features when the image feature is excluded. When both image and point features are absent, the PQ3D outperforms the specific-tuned model, demonstrating the improved generalization ability through training with multiple representations.

Influence between tasks. In Tab. 9, we examine the interplay among Refer, QA, and captioning tasks. Our findings indicate that incorporating data from the Refer task yields improvements of $+1.8\%$ for QA and $+2.2\%$ for captioning. While adding QA data shows no significant benefit for Refer, it does contribute to a $+1.2\%$ enhancement in captioning performance. Conversely, data from the captioning task positively impacts both Refer and QA tasks, with gains of $+0.6\%$ and $+0.7\%$, respectively.

Table 8: Ablation study of scene features. Each entry denotes PQ3D "trained with specific scene features" and "trained with all features but some removed during inference". These results indicate that PQ3D can support flexible inference.
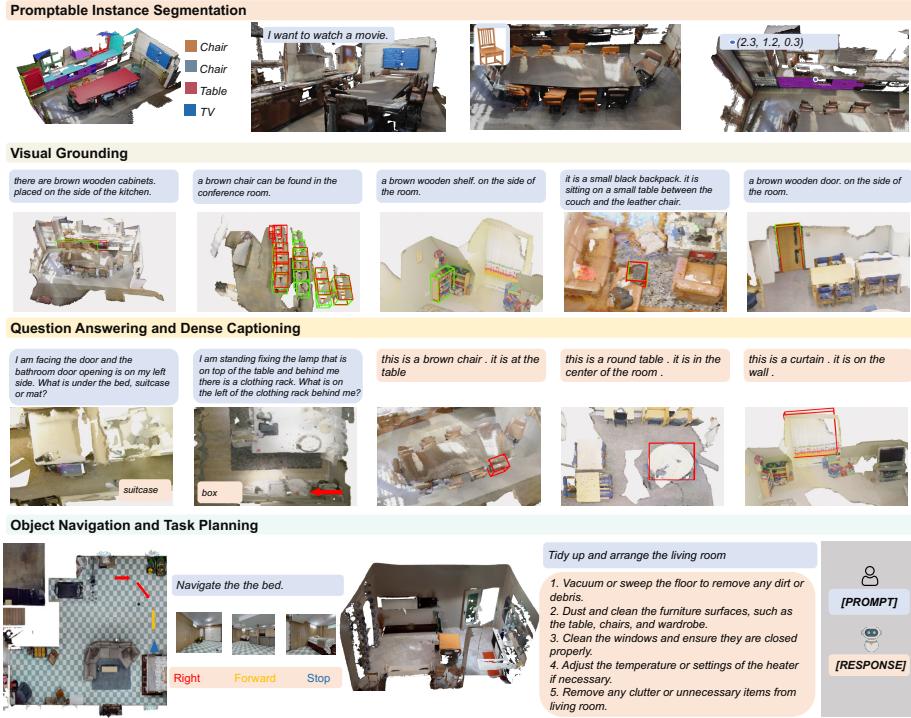
Voxel	Point	Image	Refer	QA	Caption
✓			46.1 / 47.1	43.7 / 44.2	67.8 / 68.1
✓	✓		49.2 / 49.4	45.4 / 45.8	74.6 / 74.7
✓	✓	✓	51.2	47.1	80.3

Table 9: Ablation study of influence between tasks. Each entry denotes performance gain by introducing extra task data for joint training.

Task Data	Refer	QA	Caption
+Refer	-	1.8 ↑	2.2 ↑
+QA	0.0 ↑	-	1.2 ↑
+Caption	0.6 ↑	0.7 ↑	-

4.4 Qualitative results

Fig. 5 presents the qualitative results of our model, PQ3D. The first row displays the outcome of promptable segmentation, which includes prompt forms like class names, a sentence, an image, and a 3D location. The second and third rows demonstrate the results of 3D-VL tasks, encompassing visual grounding, question answering, and dense captioning. These results exhibit the model's capabilities in solving various 3D-VL tasks. The last row illustrates the model's proficiency in object navigation and task planning. PQ3D successfully navigates towards the intended destination and formulates a plan to organize a living room. These qualitative results emphasize that PQ3D is a unified model with the potential to be applied to more embodied agent tasks as a fundamental 3D encoding module.

Fig. 5: Qualitative examples from PQ3D. Red bounding box denotes the result from PQ3D, and green denotes ground truth.

5   Conclusions and Future Works

In conclusion, our proposed PQ3D addresses the challenges in 3D vision-language learning (3D-VL) by offering a unified approach that integrates multiple representations and supports a wide range of tasks. By leveraging the cross-attention mechanism between instance queries and representations, our model generates task-aware instance queries, guided by prompts specific to each task. Through extensive experiments on various 3D-VL benchmarks, we demonstrate the effectiveness of unifying different representations in achieving state-of-the-art results. Notably, our model demonstrates promotable 3D instance segmentation, which contributes to advancing open-world 3D scene understanding. With these accomplishments, PQ3D may potentially impact embodied intelligence more, representing a step towards aligning natural language with the 3D physical world.

At present, the scale and generalization capability of PQ3D still exhibits a gap compared to 2D vision language foundational models. We aim to enhance PQ3D by scaling it with more scenes and language data. Additionally, we plan to utilize the 2D foundational model to guide 3D training.

References

1. Abdelreheem, A., Olszewski, K., Lee, H.Y., Wonka, P., Achlioptas, P.: Scanents3d: Exploiting phrase-to-3d-object correspondences for improved visio-linguistic models in 3d scenes. In: Proceedings of Winter Conference on Applications of Computer Vision (WACV). pp. 3524–3534 (2024) 2, 4

2. Achlioptas, P., Abdelreheem, A., Xia, F., Elhoseiny, M., Guibas, L.: Referit3d: Neural listeners for fine-grained 3d object identification in real-world scenes. In: European Conference on Computer Vision (ECCV) (2020) 2, 3, 4, 9

3. Azuma, D., Miyanishi, T., Kurita, S., Kawanabe, M.: Scanqa: 3d question answering for spatial scene understanding. In: The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2022) 2, 3, 4, 9, 11, A12

4. Bakr, E.M., Alsaedy, Y., Elhoseiny, M.: Look around and refer: 2d synthetic semantics knowledge distillation for 3d visual grounding. Advances in Neural Information Processing Systems (NeurIPS) (2022) A11

5. Brohan, A., Chebotar, Y., Finn, C., Hausman, K., Herzog, A., Ho, D., Ibarz, J., Irpan, A., Jang, E., Julian, R., et al.: Do as i can, not as i say: Grounding language in robotic affordances. In: Conference on Robot Learning (CoRL) (2022) 1

6. Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J.D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al.: Language models are few-shot learners. Advances in Neural Information Processing Systems (NeurIPS) **33**, 1877–1901 (2020) A12

7. Cai, D., Zhao, L., Zhang, J., Sheng, L., Xu, D.: 3djcg: A unified framework for joint dense captioning and visual grounding on 3d point clouds. In: The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2022) 2, 11, 12, A10, A11, A13

8. Chen, D.Z., Chang, A.X., Nießner, M.: Scanrefer: 3d object localization in rgb-d scans using natural language. In: European Conference on Computer Vision (ECCV) (2020) 2, 3, 4, 9

9. Chen, D.Z., Wu, Q., Nießner, M., Chang, A.X.: D3net: a speaker-listener architecture for semi-supervised dense captioning and visual grounding in rgb-d scans. arXiv preprint arXiv:2112.01551 (2021) A11

10. Chen, S., Guhur, P.L., Tapaswi, M., Schmid, C., Laptev, I.: Language conditioned spatial relation reasoning for 3d object grounding. Advances in Neural Information Processing Systems (NeurIPS) (2022) 2, 4, 6, 11, A10, A11

11. Chen, S., Chen, X., Zhang, C., Li, M., Yu, G., Fei, H., Zhu, H., Fan, J., Chen, T.: Ll3da: Visual interactive instruction tuning for omni-3d understanding reasoning and planning. In: The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2024) 4

12. Chen, Z., Gholami, A., Nießner, M., Chang, A.X.: Scan2cap: Context-aware dense captioning in rgb-d scans. In: The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2021) 2, 3, 4, 9, 12, A13

13. Chen, Z., Hu, R., Chen, X., Nießner, M., Chang, A.X.: Unit3d: A unified transformer for 3d dense captioning and visual grounding. In: International Conference on Computer Vision (ICCV). pp. 18109–18119 (2023) 2, 11

14. Cheng, B., Misra, I., Schwing, A.G., Kirillov, A., Girdhar, R.: Masked-attention mask transformer for universal image segmentation. In: The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2022) 4, 7, 8

15. Cho, J., Lei, J., Tan, H., Bansal, M.: Unifying vision-and-language tasks via text generation. In: International Conference on Machine Learning (ICML) (2021) 8

16. Choy, C., Gwak, J., Savarese, S.: 4d spatio-temporal convnets: Minkowski convolutional neural networks. In: The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2019) A1

17. Ding, H., Liu, C., Wang, S., Jiang, X.: Vision-language transformer and query generation for referring segmentation. In: International Conference on Computer Vision (ICCV). pp. 16321–16330 (2021) 5

18. Ding, R., Yang, J., Xue, C., Zhang, W., Bai, S., Qi, X.: Pla: Language-driven openvocabulary 3d scene understanding. In: The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 7010–7019 (2023) 5

19. Ding, Z., Han, X., Niethammer, M.: Votenet: A deep learning label fusion method for multi-atlas segmentation. In: Medical Image Computing and Computer Assisted Intervention (MICCAI). Springer (2019) A10

20. Dong, B., Zeng, F., Wang, T., Zhang, X., Wei, Y.: Solq: Segmenting objects by learning queries. Advances in Neural Information Processing Systems (NeurIPS) **34**, 21898–21909 (2021) 4

21. Duan, J., Yu, S., Tan, H.L., Zhu, H., Tan, C.: A survey of embodied ai: From simulators to research tasks. IEEE Transactions on Emerging Topics in Computational Intelligence **6**(2), 230–244 (2022) 1

22. Fang, Y., Yang, S., Wang, X., Li, Y., Fang, C., Shan, Y., Feng, B., Liu, W.: Instances as queries. In: International Conference on Computer Vision (ICCV). pp. 6910–6919 (2021) 4

23. Felzenszwalb, P.F., Huttenlocher, D.P.: Efficient graph-based image segmentation. International Journal of Computer Vision (IJCV) **59**, 167–181 (2004) 6

24. Ghiasi, G., Gu, X., Cui, Y., Lin, T.Y.: Scaling open-vocabulary image segmentation with image-level labels. In: European Conference on Computer Vision (ECCV). pp. 540–557. Springer (2022) 5, 7, A1

25. Guo, J., Ma, X., Fan, Y., Liu, H., Li, Q.: Semantic gaussians: Open-vocabulary scene understanding with 3d gaussian splatting. arXiv preprint arXiv:2403.15624 (2024) 2

26. Guo, Z., Tang, Y., Zhang, R., Wang, D., Wang, Z., Zhao, B., Li, X.: Viewrefer: Grasp the multi-view knowledge for 3d visual grounding. In: International Conference on Computer Vision (ICCV). pp. 15372–15383 (2023) 2, 4

27. Ha, H., Song, S.: Semantic abstraction: Open-world 3d scene understanding from 2d vision-language models. In: Conference on Robot Learning (CoRL) (2022) 2

28. He, D., Zhao, Y., Luo, J., Hui, T., Huang, S., Zhang, A., Liu, S.: Transrefer3d: Entity-and-relation aware transformer for fine-grained 3d visual grounding. In: Proceedings of the 29th ACM International Conference on Multimedia (2021) 2, A11

29. He, J., Li, P., Geng, Y., Xie, X.: Fastinst: A simple query-based model for real-time instance segmentation. In: The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 23663–23672 (2023) 4, 8, A2

30. Hong, Y., Zhen, H., Chen, P., Zheng, S., Du, Y., Chen, Z., Gan, C.: 3d-llm: Injecting the 3d world into large language models. Advances in Neural Information Processing Systems (NeurIPS) **36**, 20482–20494 (2023) 4

31. Hu, J., Cao, L., Lu, Y., Zhang, S., Wang, Y., Li, K., Huang, F., Shao, L., Ji, R.: Istr: End-to-end instance segmentation with transformers. arXiv preprint arXiv:2105.00637 (2021) 4

32. Huang, J., Yong, S., Ma, X., Linghu, X., Li, P., Wang, Y., Li, Q., Zhu, S.C., Jia, B., Huang, S.: An embodied generalist agent in 3d world. In: International Conference on Machine Learning (ICML) (2024) 1, 4, 9, A4

33. Huang, S., Chen, Y., Jia, J., Wang, L.: Multi-view transformer for 3d visual grounding. In: The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2022) 4, A10, A11

34. Huang, W., Abbeel, P., Pathak, D., Mordatch, I.: Language models as zero-shot planners: Extracting actionable knowledge for embodied agents. In: International Conference on Machine Learning (ICML). pp. 9118–9147. PMLR (2022) 1

35. Jaegle, A., Borgeaud, S., Alayrac, J.B., Doersch, C., Ionescu, C., Ding, D., Koppula, S., Zoran, D., Brock, A., Shelhamer, E., et al.: Perceiver io: A general architecture for structured inputs & outputs. arXiv preprint arXiv:2107.14795 (2021) 5

36. Jain, A., Gkanatsios, N., Mediratta, I., Fragkiadaki, K.: Bottom up top down detection transformers for language grounding in images and point clouds. In: European Conference on Computer Vision (ECCV) (2022) 4

37. Jia, B., Chen, Y., Yu, H., Wang, Y., Niu, X., Liu, T., Li, Q., Huang, S.: Sceneverse: Scaling 3d vision-language learning for grounded scene understanding. European Conference on Computer Vision (ECCV) (2024) 2, 4

38. Jiang, L., Zhao, H., Shi, S., Liu, S., Fu, C.W., Jia, J.: Pointgroup: Dual-set point grouping for 3d instance segmentation. In: The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2020) A10

39. Kerr, J., Kim, C.M., Goldberg, K., Kanazawa, A., Tancik, M.: Lerf: Language embedded radiance fields. In: International Conference on Computer Vision (ICCV). pp. 19729–19739 (2023) 2

40. Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A.C., Lo, W.Y., et al.: Segment anything. In: International Conference on Computer Vision (ICCV). pp. 4015–4026 (2023) 5

41. Lei, J., Li, L., Zhou, L., Gan, Z., Berg, T.L., Bansal, M., Liu, J.: Less is more: Clipbert for video-and-language learning via sparse sampling. In: The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2021) A12

42. Li, B., Weinberger, K.Q., Belongie, S., Koltun, V., Ranftl, R.: Language-driven semantic segmentation (2022) 5

43. Li, J., Li, D., Savarese, S., Hoi, S.: Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. arXiv preprint arXiv:2301.12597 (2023) 5

44. Li, Y., Si, S., Li, G., Hsieh, C.J., Bengio, S.: Learnable fourier features for multi-dimensional spatial positional encoding. Advances in Neural Information Processing Systems (NeurIPS) **34**, 15816–15829 (2021) A1

45. Liang, F., Wu, B., Dai, X., Li, K., Zhao, Y., Zhang, H., Zhang, P., Vajda, P., Marculescu, D.: Open-vocabulary semantic segmentation with mask-adapted clip. In: The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 7061–7070 (2023) 5

46. Lin, Z., Zhang, Z., Chen, L.Z., Cheng, M.M., Lu, S.P.: Interactive image segmentation with first click attention. In: International Conference on Computer Vision (ICCV). pp. 13339–13348 (2020) 5

47. Liu, Q., Xu, Z., Bertasius, G., Niethammer, M.: Simpleclick: Interactive image segmentation with simple vision transformers. In: International Conference on Computer Vision (ICCV). pp. 22290–22300 (2023) 5

48. Luo, J., Fu, J., Kong, X., Gao, C., Ren, H., Shen, H., Xia, H., Liu, S.: 3d-sps: Single-stage 3d visual grounding via referred point progressive selection. In: The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2022) 2, 4, A10, A11

49. Ma, X., Yong, S., Zheng, Z., Li, Q., Liang, Y., Zhu, S.C., Huang, S.: Sqa3d: Situated question answering in 3d scenes. International Conference on Learning Representations (ICLR) (2023) 2, 4, 9, 12, A12

50. Majumdar, A., Yadav, K., Arnaud, S., Ma, Y.J., Chen, C., Silwal, S., Jain, A., Berges, V.P., Abbeel, P., Malik, J., et al.: Where are we in the search for an artificial visual cortex for embodied intelligence? arXiv preprint arXiv:2303.18240 (2023) 3, 9, 11, 12

51. Minderer, M., Gritsenko, A., Stone, A., Neumann, M., Weissenborn, D., Dosovitskiy, A., Mahendran, A., Arnab, A., Dehghani, M., Shen, Z., et al.: Simple open-vocabulary object detection. In: European Conference on Computer Vision (ECCV). pp. 728–755. Springer (2022) 5

52. Padmakumar, A., Thomason, J., Shrivastava, A., Lange, P., Narayan-Chen, A., Gella, S., Piramuthu, R., Tur, G., Hakkani-Tur, D.: Teach: Task-driven embodied agents that chat. In: AAAI Conference on Artificial Intelligence (AAAI) (2022) 1

53. Peng, S., Genova, K., Jiang, C., Tagliasacchi, A., Pollefeys, M., Funkhouser, T., et al.: Openscene: 3d scene understanding with open vocabularies. In: The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 815–824 (2023) 2, 5, 6, 7, 10

54. Qi, C.R., Su, H., Mo, K., Guibas, L.J.: Pointnet: Deep learning on point sets for 3d classification and segmentation. In: The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 652–660 (2017) 7

55. Qi, C.R., Yi, L., Su, H., Guibas, L.J.: Pointnet++: Deep hierarchical feature learning on point sets in a metric space. Advances in Neural Information Processing Systems (NeurIPS) (2017) 7, A1

56. Qin, J., Wu, J., Yan, P., Li, M., Yuxi, R., Xiao, X., Wang, Y., Wang, R., Wen, S., Pan, X., et al.: Freeseg: Unified, universal and open-vocabulary image segmentation. In: The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 19446–19455 (2023) 5

57. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: International Conference on Machine Learning (ICML). pp. 8748–8763. PMLR (2021) 5

58. Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., Liu, P.J.: Exploring the limits of transfer learning with a unified text-to-text transformer. The Journal of Machine Learning Research **21**(1), 5485–5551 (2020) 8, A2

59. Rana, K., Haviland, J., Garg, S., Abou-Chakra, J., Reid, I., Suenderhauf, N.: Sayplan: Grounding large language models using 3d scene graphs for scalable task planning. In: 7th Annual Conference on Robot Learning (2023) 1

60. Rozenberszki, D., Litany, O., Dai, A.: Language-grounded indoor 3d semantic segmentation in the wild. In: European Conference on Computer Vision (ECCV). pp. 125–141. Springer (2022) 1, 2, 3, 6, 9

61. Savva, M., Kadian, A., Maksymets, O., Zhao, Y., Wijmans, E., Jain, B., Straub, J., Liu, J., Koltun, V., Malik, J., et al.: Habitat: A platform for embodied ai research. In: The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 9339–9347 (2019) 4

62. Schult, J., Engelmann, F., Hermans, A., Litany, O., Tang, S., Leibe, B.: Mask3d: Mask transformer for 3d semantic instance segmentation. In: International Conference on Robotics and Automation (ICRA). pp. 8216–8223. IEEE (2023) 2, 6, 7, 8, 9, 10, A1, A2, A10

63. Straub, J., Whelan, T., Ma, L., Chen, Y., Wijmans, E., Green, S., Engel, J.J., Mur-Artal, R., Ren, C., Verma, S., et al.: The replica dataset: A digital replica of indoor spaces. arXiv preprint arXiv:1906.05797 (2019) 3, 10

64. Takmaz, A., Fedele, E., Sumner, R.W., Pollefeys, M., Tombari, F., Engelmann, F.: OpenMask3D: Open-Vocabulary 3D Instance Segmentation. In: Advances in Neural Information Processing Systems (NeurIPS) (2023) 2, 5, 6, 7, 8, 10

65. Wu, J., Jiang, Y., Sun, P., Yuan, Z., Luo, P.: Language as queries for referring video object segmentation. In: The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 4974–4984 (2022) 5

66. Wu, Y., Cheng, X., Zhang, R., Cheng, Z., Zhang, J.: Eda: Explicit text-decoupling and dense alignment for 3d visual grounding. In: The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 19231–19242 (2023) 4

67. Xu, J., Hou, J., Zhang, Y., Feng, R., Wang, Y., Qiao, Y., Xie, W.: Learning open-vocabulary semantic segmentation models from natural language supervision. In: The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2935–2944 (2023) 5

68. Xu, X., Xiong, T., Ding, Z., Tu, Z.: Masqclip for open-vocabulary universal image segmentation. In: International Conference on Computer Vision (ICCV). pp. 887–898 (2023) 5

69. Yang, Z., Jiang, L., Sun, Y., Schiele, B., Jia, J.: A unified query-based paradigm for point cloud understanding. In: The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 8541–8551 (2022) 5

70. Yang, Z., Zhang, S., Wang, L., Luo, J.: Sat: 2d semantics assisted training for 3d visual grounding. In: International Conference on Computer Vision (ICCV) (2021) 4, A10, A11

71. Ye, S., Chen, D., Han, S., Liao, J.: 3d question answering. IEEE Transactions on Visualization and Computer Graphics (2022) 4

72. Zareian, A., Rosa, K.D., Hu, D.H., Chang, S.F.: Open-vocabulary object detection using captions. In: The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 14393–14402 (2021) 5

73. Zhang, T., He, S., Tao, D., Chen, B., Wang, Z., Xia, S.T.: Vision-language pre-training with object contrastive learning for 3d scene understanding. arXiv preprint arXiv:2305.10714 (2023) 2

74. Zhang, Y., Gong, Z., Chang, A.X.: Multi3drefer: Grounding text description to multiple 3d objects. In: International Conference on Computer Vision (ICCV). pp. 15225–15236 (2023) 2, 3, 4, 9, 11, A11

75. Zhao, L., Cai, D., Sheng, L., Xu, D.: 3dvg-transformer: Relation modeling for visual grounding on point clouds. In: International Conference on Computer Vision (ICCV) (2021) 4, A10, A11

76. Zhao, L., Cai, D., Zhang, J., Sheng, L., Xu, D., Zheng, R., Zhao, Y., Wang, L., Fan, X.: Towards explainable 3d grounded visual question answering: A new benchmark and strong baseline. IEEE Transactions on Circuits and Systems for Video Technology (2022) 2

77. Zhong, Y., Yang, J., Zhang, P., Li, C., Codella, N., Li, L.H., Zhou, L., Dai, X., Yuan, L., Li, Y., et al.: Regionclip: Region-based language-image pretraining. In: The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 16793–16803 (2022) 5

78. Zhu, X., Su, W., Lu, L., Li, B., Wang, X., Dai, J.: Deformable detr: Deformable transformers for end-to-end object detection. In: International Conference on Learning Representations (ICLR) (2020) 4, 8

79. Zhu, Z., Ma, X., Chen, Y., Deng, Z., Huang, S., Li, Q.: 3d-vista: Pre-trained transformer for 3d vision and text alignment. In: International Conference on Computer Vision (ICCV). pp. 2911–2921 (2023) 1, 2, 4, 6, 7, 11, 12, A2, A10, A11, A12, A13

80. Zou, X., Dou, Z.Y., Yang, J., Gan, Z., Li, L., Li, C., Dai, X., Behl, H., Wang, J., Yuan, L., et al.: Generalized decoding for pixel, image, and language. In: The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 15116–15127 (2023) 5

Appendix

A   Implementation Details

This section introduces more details about model architecture, training details, and downstream task settings.

A.1   Model architecture

Prompt Encoding To encode textual prompts, we utilize the CLIP (clip-vit-large-patch14) text encoder with a projection layer. This allows us to encode text and images into a joint embedding space. For numerical prompts, we employ a linear layer to project locations and bounding box sizes into $\mathbb{R}^D$ space. This projection aligns the feature space with CLIP. When it comes to image prompts, our goal is to utilize image semantics to replace text prompts for zero-shot transfer. To achieve this, we encode the text in the form of "`[SOS] [object] [EOS]`", where `[object]` is substituted with image semantic features encoded by the CLIP image encoder. This approach allows us to leverage the semantic information contained within images and use it as a replacement for traditional text prompts, facilitating zero-shot transfer learning.

Scene Representation For point cloud encoding, we utilize a three-layer Point-Net++ [55] architecture. This architecture operates on the input points with different radii: 0.2, 0.4, and sampling all points within this ball. The output of this PointNet++ is an aggregated 768-dimensional feature representation. For voxel encoding, we use MinkowskiRes16UNet34C [16] with a voxel size of 0.02m. It combines the ResNet-16 backbone with the UNet architecture in the Minkowski Engine framework. It applies 3D convolutions and pooling operations to extract hierarchical features from the input voxelized data. The network progressively downsamples the spatial resolution in the encoder path and upsamples it in the decoder path, while capturing and fusing features at multiple scales through skip connections. We use four levels of voxel features and project them all to dimension 768. For image encoding, we use a pre-trained OpenSeg feature backbone [24].

Prompt-guided Query Learning In query initialization, we first use furthest point sampling [62] to sample points from original point clouds. After we acquire the location of sampled points, we apply Fourier spatial position encoding [44] to encode points $\mathbf{p}_i$ as follows:

$$\mathbf{l_i} = \left[\cos(2\pi \mathbf{p}_i W_f^T) \| \sin(2\pi \mathbf{p}_i W_f^T)\right] \tag{A1}$$

During masked cross-attention, the location information $\mathbf{l}$ is incorporated into the computation of attention by adding it to the queries $\tilde{\mathbf{Q}}_l$ and keys $\tilde{\mathbf{F}}$. Here, $\tilde{\mathbf{Q}}_l$ represents the instance queries, and $\tilde{\mathbf{F}}$ denotes the features augmented by locations. Following the approach described in [62], we enforce attention restriction using a mask generated from the previous layer, denoted as $\mathbf{M}^{l-1}$. This

mask ensures that instance queries can only attend to voxel, image, and point features within their corresponding intermediate instance mask predicted by the previous layer. This masked-attention mechanism is constrained to focus on relevant features within the instance mask, which helps in capturing context and information specific to each instance, preventing cross-contamination between different instances during the attention computation.

$$\mathbf{Q}'_l = \text{softmax}\left(W_q^T \tilde{\mathbf{Q}}_l \tilde{\mathbf{F}}^T W_k^T / \sqrt{D} + \mathbf{M}_{l-1}\right) W_v \mathbf{F} \tag{A2}$$

Output Head The mask head employs two separate MLPs for queries and features, using a hidden dimension of 768 to map them to a shared space [62]. The grounding head consists of a two-layer MLP with a hidden dimension of 384 and produces a probability score as output [79]. The generation head uses a pre-trained T5-small [58].

A.2   Training settings

In this section, we provide a detailed explanation of our training procedure. Image features are preserved before training, and only voxel and point features are trained. For the unsupervised generation of segment masks, we maintain the same hyperparameters as those used in ScanNet, which are applied to both ScanNet and Replica datasets. For data augmentation, we utilize techniques such as horizontal flipping, random rotations around the z-axis, elastic distortion, and random scaling. We also implement color augmentations including jittering, as well as brightness and contrast adjustments. We set the number of queries to 120 to strike a balance between speed and the number of objects present in the scene. During the training phase of instance segmentation, we adopt the use of ground truth mask guidance [29] for the initial 200 epochs to expedite the model's convergence. We employ a cosine decay model for our learning rate scheduler. During instruction tuning, we replace the T5 decoder with Vicuna and conduct 10 epochs of fine-tuning. We use a learning rate of 3e-5 and train the model using 4 A100 GPUs. For object navigation, we connect our query decoder to a recurrent neural network and use the AdamW optimizer with a learning rate of 1e-3. The object navigation agent is trained in the HM3D environment for 100 million steps with 16 parallel environments.

A.3   Details for Each Task

Instance Segmentation In our instance segmentation experiments, we utilize the ScanNet200 dataset, which is an extension of the ScanNet dataset. ScanNet is an annotated dataset that comprises 3D reconstructed indoor scenes, offering a rich representation of various room types such as hotels, libraries, and offices. The dataset is divided into three distinct subsets: 1,202 scenes for training, 312 scenes for validation, and 100 scenes for the hidden test set. In our evaluation, we utilize the mean average precision (mAP) metric at various Intersections over Union (IoU) thresholds. This metric provides a comprehensive assessment of instance

segmentation performance. In closed vocabulary experiments, we employ 120 queries and select the top 200 output masks based on their classification scores. In open vocabulary experiments, we prompt all 200 classes and select the top 200 masks using binary cross-entropy (BCE) logits. This setup allows for the exploration of a broader range of object classes, extending beyond the limitations of a closed vocabulary.

Visual Grounding In our experiments, we evaluate our model on three different datasets for visual grounding tasks: ScanRefer, Nr3D/Sr3D, and Multi3DRefer.

ScanRefer dataset consists of 51,583 sentences written by humans to describe 800 scenes from the ScanNet dataset. The dataset is categorized into unique and multiple subsets based on whether the target object described in the sentence is a unique class within the scene. The evaluation metric for this task is accuracy, measured under intersection over union (IoU) thresholds of 0.25 and 0.5. In this task, we aim to predict the location of the target object described in the sentence.

Nr3D/Sr3D datasets comprise synthetic and human utterances, respectively. Sr3D contains 83,572 synthetic utterances, while Nr3D contains 45,503 human utterances. Both datasets are split into "Easy"/"Hard" and "ViewDep"/"ViewIndep" subsets. The "Hard" samples involve scenes with more than two distractors. The evaluation metric for these datasets is based on the F1 score, measured at IoU thresholds of 0.25 and 0.5. Ground truth masks and locations are used for evaluation, following the original settings of these benchmarks to ensure fair comparison.

Multi3DRefer dataset consists of 61,926 language samples, categorized into zero-target (ZT), single-target (ST), and multi-target (MT) referring samples. In this dataset, we predict masks for the referred objects. The evaluation metric used is the F1 score at IoU thresholds of 0.25 and 0.5. Again, we follow the original settings of this benchmark to ensure fair and consistent evaluation.

Question Answering ScanQA is a 3D question answering dataset with 41,363 questions and 58,191 answers, focusing on spatial relations. Evaluation metrics include exact matches (EM@1, EM@10) and text similarity metrics (BLEU-1, ROUGE, METEOR, CIDEr). We also evaluate PQ3D on SQA3D, a benchmark for scene understanding with 6.8k situations and 33.4k diverse questions. The evaluation metric in SQA3D is answer accuracy across different question types. Both benchmarks use a generation approach for answering questions.

Dense Captioning Scan2Cap is a dense captioning benchmark that utilizes texts from the ScanRefer dataset. In this benchmark, we evaluate the performance of models in generating dense captions for 3D scenes. To assess the quality of the generated captions, we report text similarity scores, including CIDEr, BLEU-4, METEOR, and ROUGE, under different IoU (Intersection over Union) scores. These metrics provide a measure of the similarity between the generated captions and the reference captions, taking into account the level of overlap between the predicted and ground truth regions of interest.

Object Navigation We utilize the object navigation benchmark from CortexBench, which is based on the HM3D-SEM dataset. The dataset consists of 80 training scenes, 20 validation scenes, and 20 test scenes from the HABITAT
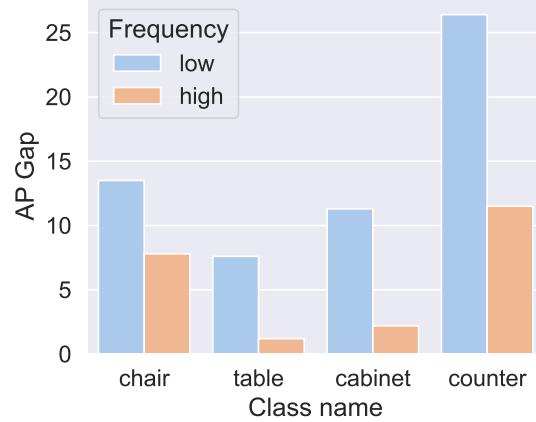
platform. For our evaluation, we use the validation split for comparison purposes. In this benchmark, the agent is modeled after the LocoBot, and the sensors are positioned at the top of the agent's head. The RGB camera used has a resolution of 640×480 and a horizontal field of view of 79 degrees. The objective for the agent is to navigate within the environment and locate objects belonging to one of six categories: "chair", "bed", "plant", "toilet", "tv/monitor", and "sofa". The agent has a maximum of 500 steps to complete the task successfully. The determination of successful episodes is based on the agent stopping within 0.1m of a viewpoint that is within 1m of any instance of the target object. This proximity criterion is used to determine if the agent has successfully located the target object.

Task Planning To demonstrate the task planning ability of our model, we employ the LEO [32] instruction following data for fine-tuning. This dataset comprises scenes from 3RScan and ScanNet, enriched with 3D dialogues, scene-aware task planning information, and paired 3D question answering data. By utilizing this data, we aim to enhance our model's capability to understand instructions, follow them in a 3D environment, and effectively plan and execute tasks based on the given instructions. The inclusion of scene-aware task planning information allows our model to incorporate contextual knowledge about the environment and optimize its decision-making process accordingly. Furthermore, the paired 3D question answering data enables our model to not only follow instructions but also answer questions related to the scene and the performed tasks.

B   Extra Ablation Studies
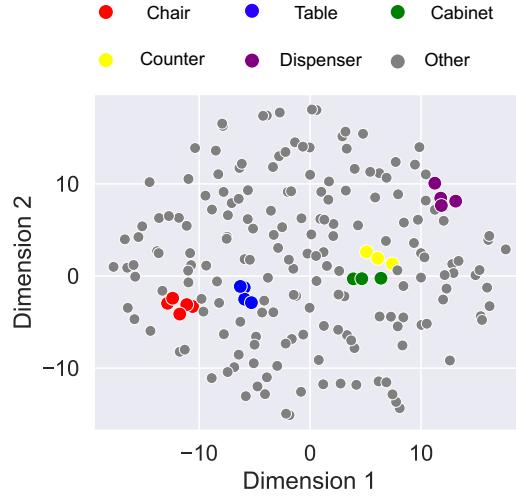
Bias in CLIP text encoder From Fig. A1, we find that when comparing classes with similar semantics, such as "cabinet" and "kitchen cabinet", or "chair" and "armchair", we observe differences in the instance segmentation quality between a closed vocabulary head and an open vocabulary approach. In the closed vocabulary setting, where specific class labels are predefined, the instance segmentation quality tends to be higher. This is because the model is trained to recognize and segment instances based on the pre-defined class labels, leading to more accurate and precise results. On the other hand, in the open vocabulary setting, where the model has the flexibility to generate class labels on the fly, the instance segmentation quality may vary. The model may struggle to distinguish between closely related classes, resulting in less accurate segmentation boundaries and potentially merging instances that should be separated.

These differences arise from the challenge of disambiguating between classes that have similar meanings but subtle text differences in CLIP embedding space as shown in Fig. A2. The closed vocabulary approach benefits from pre-defined class boundaries, while the open vocabulary approach relies on the model's ability to generalize and make fine-grained distinctions between similar classes. Therefore, when dealing with classes that have similar semantics but differ in instance segmentation quality, the choice between closed vocabulary and open

Fig. A1: Differences in classes with similar semantics but have different frequencies in the ScanNet200 dataset. AP Gap denotes the difference between closed vocabulary and open vocabulary.

vocabulary approaches should be carefully considered, taking into account the specific requirements and trade-offs of the task at hand.

Fig. A2: t-SNE visualization of ScanNet200 class names encoded by CLIP text encoder.

Ablation of structures Apart from our proposed decoder structure in Fig. 3, we explore two alternative structures for the cross-attention process: The parallel

structure and the sequential structure. In the parallel structure, instance queries cross-attend to all visual features and the encoded task prompt in parallel:

$$\mathbf{Q}_l^{'} = \sum_{\mathbf{F} \in \{\mathbf{V}, \mathbf{I}, \mathbf{P}, \mathbf{t}\}} \text{MaskedCrossAttn}(\mathbf{Q}_l, \mathbf{F}) \tag{A3}$$

$$\mathbf{Q}_{l+1} = \text{SpatialSelfAttn}(\mathbf{Q}_l^{'}) \tag{A4}$$

Conversely, in the sequential structure, instance queries cross-attend to all visual features and the encoded task prompt in sequence:

$$\mathbf{Q}_l^{'} = \text{MaskedCrossAttn}(\mathbf{Q}_l, \mathbf{V}) \tag{A5}$$

$$\mathbf{Q}_l^{''} = \text{MaskedCrossAttn}(\mathbf{Q}_l^{'}, \mathbf{I}) \tag{A6}$$

$$\mathbf{Q}_l^{'''} = \text{MaskedCrossAttn}(\mathbf{Q}_l^{''}, \mathbf{P}) \tag{A7}$$

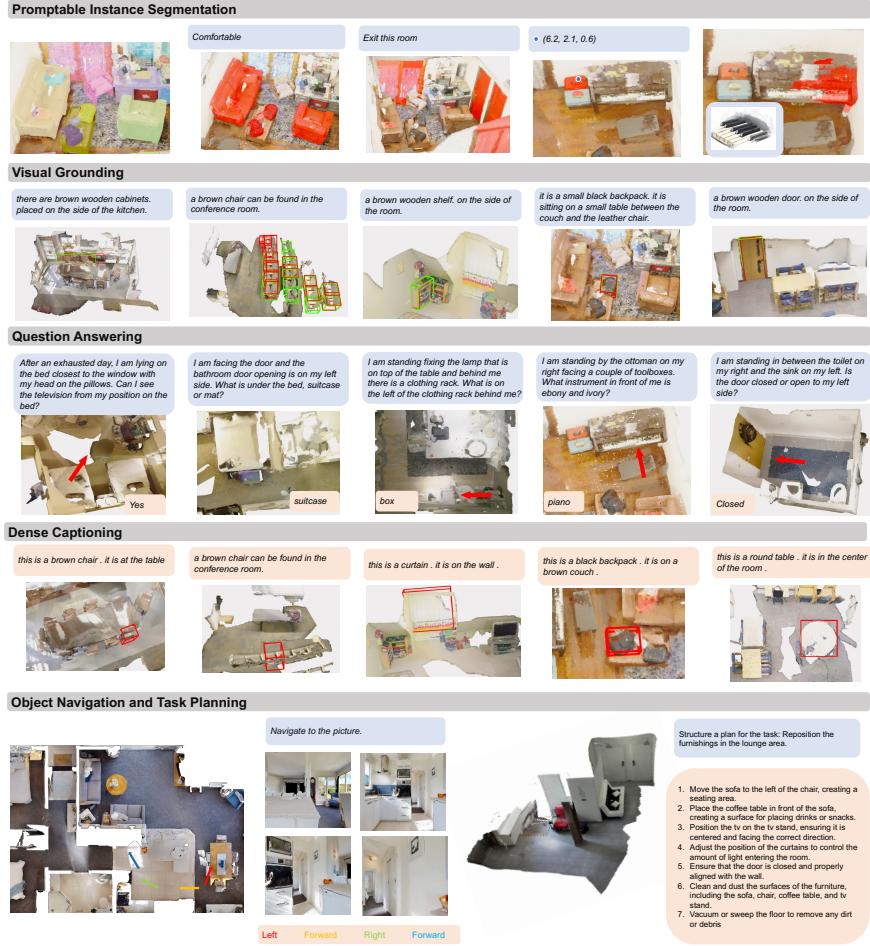$$\mathbf{Q}_l^{''''} = \text{CrossAttn}(\mathbf{Q}_l^{'''}, \mathbf{t}) \tag{A8}$$

$$\mathbf{Q}_{l+1} = \text{SpatialSelfAttn}(\mathbf{Q}_l^{''''}) \tag{A9}$$

From Tabs. A1 to A6, it is evident that the structure outlined in our main paper delivers superior performance compared to both parallel and sequential approaches.

C   More qualitative results

More visualization In Fig. A3, we present results from various tasks conducted on a range of scenes. In the case of promptable instance segmentation, PQ3D is capable of producing high-fidelity masks and reasoning about affordances such as comfort and exit. Interestingly, when an image prompt featuring a part of a piano keyboard is provided, PQ3D segments not the entire piano but a small section of the keyboard. This suggests that the query can extract semantic information from the image prompt and segment a specific part of an instance. In the visual grounding task, it is observed that PQ3D is capable of grounding both single and multiple objects, and it demonstrates an understanding of the relations between objects. In the question answering task, PQ3D exhibits an understanding of the surrounding environment and can infer object classes and states based on the given situation. In the dense captioning task, our model can describe an object and its relationships with surrounding objects. In object navigation and planning, the examples presented demonstrate our model's capacity to support embodied agent tasks.

Failure cases We present failure cases of PQ3D in Fig. A4. It can be observed that PQ3D may segment unrelated objects when text prompts lack sufficient clarity, such as in the case of prompts like "classical", "exam", or "video games". Additionally, when image prompts contain background noise, it may lead to confusion for the model, causing it to segment random objects across different regions in a scene. In visual grounding and question answering tasks, the model may struggle to recognize complex spatial relations and comprehend long

Fig. A3: More visualization on promptable segmentation, visual grounding, question answering, dense captioning, object navigation, and task planning. Red bounding box denotes predicted result, green denotes groundtruth result.

sentences. This limitation can potentially be mitigated by employing a more powerful text encoder. In dense captioning tasks, the model may face challenges in accurately understanding small objects in terms of their semantics.

Effect of different features As illustrated in Fig. A5, it is clear that integrating image and point features leads to a more refined understanding of the scene. In tasks such as visual grounding and question answering, having more features enables the model to better comprehend the semantics, such as objects like backpacks and cluttered desks, as well as spatial relationships like "other end", "corner", "beneath", and attributes like color and object state. In the case of dense captioning, incorporating image features is particularly beneficial for the

Fig. A4: Failure cases in promptable segmentation, visual grounding, question answering, and dense captioning.

model to describe more detailed information about the instance, including class, shape, and color.

D Full quantitative results

We provide full quantitative results in Tab. A1, Tab. A2, Tab. A3, Tab. A4, Tab. A5, Tab. A6, including results of baselines and all variants of PQ3D for abalation study. Specifically, PQ3D (sg.) denotes the model trained on a single dataset rather than through unified joint training, PQ3D (V) denotes the model with only voxel features, PQ3D (V,P) denotes the model with only point and voxel features, PQ3D (rm I,P), and PQ3D (rm I) denotes training with all features but removing image, voxel or image only during inference. PQ3D (2-layer) denotes the model with 2-layer decoder, PQ3D (6-layer) denotes the model with 6-layer decoder, PQ3D (par.) denotes the model with parallel structure, and PQ3D (seq.) denotes the model with sequential structure.

Fig. A5: Comparison between different features. The blue color represents the results obtained using only the voxel feature. The magenta color signifies the results from using both voxel and point features. The results derived from using all features are denoted by the color red. The ground truth results are represented by the color green.

Table A1: Grounding accuracy (%) on ScanRefer with detected object masks. "Det." represents the 3D object detection module used in the model. "VN" stands for VoteNet [19], "PG" for PointGroup [38], and M3D for Mask3D [62], while "Opt." denotes jointly optimizing the object detector on ScanRefer.

Method	Det.	Unique		Multiple		Overall	
		acc@0.25	acc@0.5	acc@0.25	acc@0.5	acc@0.25	acc@0.5
3DVG-Trans [75]	Opt.	81.9	60.6	39.3	28.4	47.6	34.7
3D-SPS [48]	Opt.	84.1	66.7	40.3	29.8	48.8	37.0
3DJCG [7]	Opt.	83.5	64.3	41.4	30.8	49.6	37.3
SAT [70]	VN	73.2	50.8	37.6	25.2	44.5	30.1
MVT [33]	PG	77.7	66.5	31.9	25.3	40.8	33.3
ViL3DRel [10]	PG	81.6	68.6	40.3	30.7	47.9	37.7
3D-VisTA [79]	PG	77.0	67.9	37.9	30.4	45.2	37.3
3D-VisTA [79]	M3D	81.6	75.1	43.7	39.1	50.6	45.8
PQ3D (*sg.*)	Opt.	85.2	76.6	46.8	42.0	52.8	47.4
PQ3D	Opt.	86.7	78.3	51.5	46.2	57.0	51.2
PQ3D (2-layer)	Opt.	86.6	78.2	50.6	45.2	56.1	50.2
PQ3D (6-layer)	Opt.	85.0	77.1	51.2	45.9	56.4	50.7
PQ3D (V)	Opt.	82.9	75.1	45.9	40.9	51.6	46.1
PQ3D (rm I,P)	Opt.	84.0	76.4	46.8	41.7	52.6	47.1
PQ3D (V,P)	Opt.	84.4	76.1	49.7	44.4	55.0	49.2
PQ3D (rm I)	Opt.	86.2	77.9	49.1	44.3	54.8	49.4
PQ3D (par.)	Opt.	85.8	77.5	50.9	45.6	56.3	50.5
PQ3D (seq.)	Opt.	83.7	76.1	46.8	41.9	52.5	47.1

Table A2: Grounding accuracy (%) on Nr3D and Sr3D with ground-truth object masks.

Method	Nr3D					Sr3D				
	Overall	Easy	Hard	View Dep	View Indep	Overall	Easy	Hard	View Dep	View Indep
3DVG-Trans [75]	40.8	48.5	34.8	34.8	43.7	51.4	54.2	44.9	44.6	51.7
TransRefer3D [28]	48.0	56.7	39.6	42.5	50.7	57.4	60.5	50.2	49.9	57.7
LAR [4]	48.9	58.4	42.3	47.4	52.1	59.4	63.0	51.2	50.0	59.1
SAT [70]	56.5	64.9	48.4	54.4	57.6	57.9	61.2	50.0	49.2	58.3
3D-SPS [48]	51.5	58.1	45.1	48.0	53.2	62.6	56.2	65.4	49.2	63.2
MVT [33]	59.5	67.4	52.7	59.1	60.3	64.5	66.9	58.8	58.4	64.7
ViL3DRel [10]	64.4	70.2	57.4	62.0	64.5	72.8	74.9	67.9	63.8	73.2
3D-VisTA [79]	64.2	72.1	56.7	61.5	65.1	76.4	78.8	71.3	58.9	77.3
PQ3D (*sg.*)	64.9	73.3	56.7	60.7	67.0	75.6	78.8	68.2	51.5	76.7
PQ3D	66.7	75.0	58.7	62.8	68.6	79.7	82.7	72.8	62.9	80.5
PQ3D (2-layer)	64.2	72.7	56.3	57.2	67.8	78.2	81.3	70.9	62.1	78.9
PQ3D (6-layer)	66.5	74.1	59.4	60.9	69.3	80.0	82.6	73.9	65.8	80.7
PQ3D (V)	58.8	66.7	51.4	55.2	60.7	71.4	73.5	66.5	56.9	72.1
PQ3D (rm I,P)	60.3	69.0	52.0	56.2	62.4	71.9	75.0	64.9	55.5	72.7
PQ3D (V,P)	64.4	73.3	56.0	61.4	65.9	77.5	80.7	70.0	61.0	78.3
PQ3D (rm I)	63.7	71.8	56.1	59.1	66.0	77.3	80.4	70.1	61.7	78.0
PQ3D (par.)	65.9	74.2	58.1	62.5	67.6	78.7	81.8	71.5	65.7	79.3
PQ3D (seq.)	56.1	63.8	48.9	47.6	60.4	70.8	73.6	64.3	56.1	71.5

Table A3: Grounding accuracy (%) on Multi3DRefer. Results of 3DVG-Trans+, D3Net and 3DJCG are provided by [74].

Method	F1@0.5 (Pred boxes)					
	ZT w/o D	ZT w/D	ST w/o D	ST w/D	MT	All
3DVG-Trans+ [75]	87.1	45.8	27.5	16.7	26.5	25.5
D3Net (Grounding) [9]	81.6	32.5	38.6	23.3	35.0	32.2
3DJCG (Grounding) [7]	94.1	66.9	26.0	16.7	26.2	26.6
M3DRef-CLIP [74]	81.8	39.4	47.8	30.6	37.9	38.4
PQ3D (*sg.*)	87.1	61.1	66.2	40.5	41.7	48.6
PQ3D	85.4	57.7	68.5	43.6	40.9	50.1
PQ3D (2-layer)	85.2	56.9	66.3	41.4	39.0	48.1
PQ3D (6-layer)	86.5	58.5	67.4	44.0	41.6	50.3
PQ3D (V)	84.1	55.1	63.0	36.8	37.5	44.8
PQ3D (rm I,P)	84.1	57.7	62.9	38.2	38.1	45.7
PQ3D (V,P)	82.4	54.6	66.0	40.9	39.4	47.7
PQ3D (rm I)	85.6	55.4	66.9	40.8	39.8	48.1
PQ3D (par.)	87.8	58.5	66.8	42.4	41.1	49.4
PQ3D (seq.)	84.5	58.8	62.3	34.5	35.5	43.2

Table A4: Answer accuracy on ScanQA. Each entry denotes "test w/ object" / "test w/o object".

Method	EM@1	BLEU-1	ROUGE	METEOR	CIDEr
Image+MCAN [3]	22.3 / 20.8	26.7 / 26.3	31.3 / 29.2	12.1 / 11.5	60.4 / 55.6
ScanRefer+MCAN [3]	20.6 / 19.0	27.9 / 27.0	30.7 / 28.6	12.0 / 11.4	57.4 / 53.4
ScanQA [3]	23.5 / 20.9	31.6 / 30.7	34.3 / 31.1	13.6 / 12.6	67.3 / 60.2
3D-VisTA [79]	27.0 / 23.0	34.4 / 30.2	38.6 / 32.8	15.2 / 12.9	76.6 / 62.6
PQ3D (*sg.*)	18.9 / 16.1	34.7 / 30.5	35.6 / 30.4	14.5 / 12.1	69.3 / 56.0
PQ3D	26.1 / 20.0	43.0 / 36.1	42.9 / 34.0	17.8 / 13.9	87.8 / 65.2

Table A5: Answer accuracy on SQA3D under question types.

Method	Test set						Avg.
	What	Is	How	Can	Which	Other	
GPT-3 [6]	39.7	46.0	40.5	45.6	36.1	38.4	41.0
ClipBERT [41]	30.2	60.1	38.7	63.3	42.5	42.7	43.3
SQA3D(w/o s) [49]	28.6	65.0	47.3	66.3	43.9	42.9	45.3
SQA3D [49]	31.6	63.8	46.0	69.5	43.9	45.3	46.6
3D-VisTA [79]	34.8	63.3	45.4	69.8	47.2	48.1	48.5
PQ3D (*sg.*)	35.6	62.7	45.2	66.3	43.3	43.3	46.8
PQ3D	37.1	61.4	44.5	61.0	47.0	45.1	47.1
PQ3D (2-layer)	37.7	62.1	41.4	62.9	36.2	46.2	46.3
PQ3D (6-layer)	36.2	62.2	42.0	64.6	40.4	42.5	46.0
PQ3D (V)	32.3	59.2	40.1	61.5	43.2	41.3	43.7
PQ3D (rm I,P)	31.5	60.3	41.1	61.8	42.7	44.5	44.2
PQ3D (V,P)	33.5	59.7	40.3	66.7	41.8	46.5	45.4
PQ3D (rm I)	35.4	61.0	42.4	62.3	43.5	43.9	45.8
PQ3D (par.)	37.7	60.9	42.0	64.1	42.1	41.4	46.2
PQ3D (seq.)	33.6	58.9	43.3	65.5	42.6	43.7	45.2

Table A6: Captioning results on Scan2Cap dataset. "C" stands for "CIDEr", "B-4" for "BLEU-4", "M" for "METEOR", and "R" for "ROUGE", respectively. "@0.25" and "@0.5" represent the overlap ratios between the predicted boxes and ground truth boxes.

Method	@0.25				@0.5			
	C	B-4	M	R	C	B-4	M	R
Scan2Cap [12]	53.7	34.3	26.1	55.0	35.2	22.4	21.4	43.5
3DJCG [7]	60.9	39.7	27.5	59.0	47.7	31.5	24.3	51.8
3D-VisTA [79]	71.0	36.5	28.4	57.6	66.9	34.0	27.1	54.3
PQ3D (*sg.*)	81.5	37.5	30.4	60.6	75.6	34.5	28.6	57.1
PQ3D	87.1	39.2	30.9	61.5	80.3	36.0	29.1	57.9
PQ3D (2-layer)	86.4	38.8	30.7	61.0	79.8	35.5	28.8	57.3
PQ3D (6-layer)	83.2	36.2	30.0	59.5	76.7	33.1	28.1	56.0
PQ3D (V)	73.3	33.8	28.8	58.5	67.8	31.0	27.1	54.9
PQ3D (rm I,P)	73.2	34.2	29.1	59.3	68.1	31.5	27.4	55.9
PQ3D (V,P)	80.2	37.7	30.2	60.9	74.6	34.8	28.4	57.5
PQ3D (rm I)	80.4	37.8	30.1	60.7	74.7	34.7	28.4	57.2
PQ3D (par.)	84.9	39.1	30.5	61.2	78.3	35.7	28.6	57.5
PQ3D (seq.)	66.1	29.1	27.9	56.1	61.4	27.0	26.3	53.0