PerLA : Perceptive 3D language assistant

Guofeng Mei[1] Wei Lin[2] Luigi Riz[1] Yujiao Wu[3] Fabio Poiesi[1] Yiming Wang[1]
[1]Fondazione Bruno Kessler, Italy [2]JKU Linz, Austria [3]CSIRO, Australia
gmei@fbk.eu

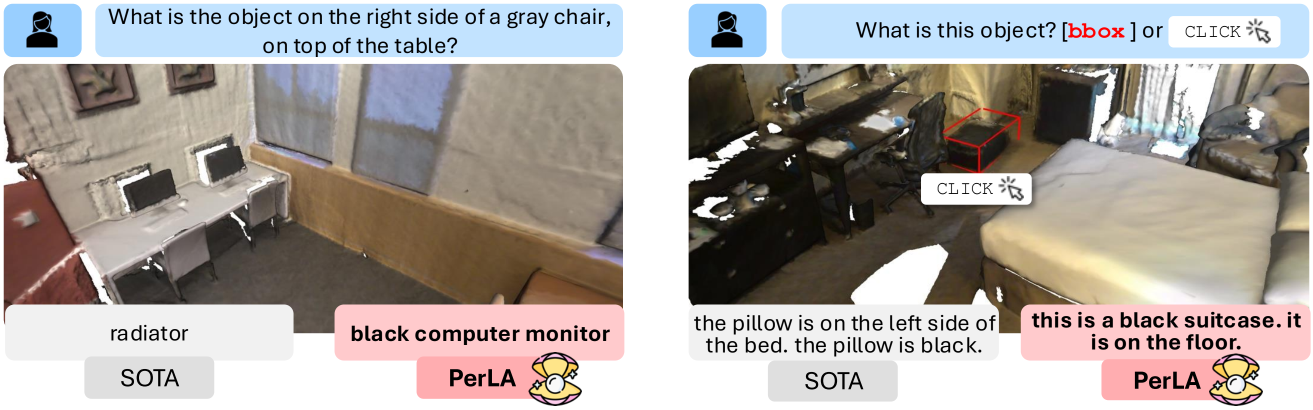

What is the object on the right side of a gray chair, on top of the table?
radiator
SOTA
black computer monitor
PerLA
What is this object? [bbox] or CLICK
CLICK
the pillow is on the left side of the bed. the pillow is black.
SOTA
this is a black suitcase. it is on the floor.
PerLA

Figure 1. PerLA is a 3D language assistant that integrates local details with global context to learn informative representations of 3D scenes, whereas state-of-the-art (SOTA) 3DLAs focus solely on global context information. PerLA can provide more accurate responses, correctly distinguishing between objects such as a "black computer monitor" and a "black suitcase," where SOTA models instead fail with hallucinated responses. Examples in figures show cases where capturing details from the point cloud matters for accurate output captions.

Abstract

Enabling Large Language Models (LLMs) to understand the 3D physical world is an emerging yet challenging research direction. Current strategies for processing point clouds typically downsample the scene or divide it into smaller parts for separate analysis. However, both approaches risk losing key local details or global contextual information. In this paper, we introduce PerLA, a 3D language assistant designed to be more perceptive to both details and context, making visual representations more informative for the LLM. PerLA captures high-resolution (local) details in parallel from different point cloud areas and integrates them with (global) context obtained from a lower-resolution whole point cloud. We present a novel algorithm that preserves point cloud locality through the Hilbert curve and effectively aggregates local-to-global information via cross-attention and a graph neural network. Lastly, we introduce a novel loss for local representation consensus to promote training stability. PerLA outperforms state-of-the-art 3D language assistants, with gains of up to +1.34 CiDEr on ScanQA for question answering, and +4.22 on ScanRefer and +3.88 on Nr3D for dense captioning. Project page: https://gfmei.github.io/PerLA

1. Introduction

3D language assistants (3DLAs) jointly process natural language and 3D data to achieve 3D scene understanding, such as recognizing object categories, locations, appearances, and relationships, without requiring specialized models for each recognition task [9, 11, 21]. These capabilities are primarily powered by Large Language Models (LLMs) trained in large text corpora [54]. These approaches can aggregate multi-view features [21] or process point clouds [9] to generate 3D representations, which are in turn converted to tokens for the LLM [54, 66]. However, extracting multi-view representations is computationally costly and often fails to capture essential geometric properties [21]. Directly processing point clouds can yield more accurate results, yet it is even more computationally costly than handling multi-view data, as point clouds typically have rather large cardinalities [60]. To address this, the point cloud cardinality can be reduced via downsampling [8, 9]. However, as with images [36], downsampling can compromise downstream task performance due to the reduced model's ability to perceive fine details of 3D scenes [47].

Fig. 1 illustrates two cases where our approach can accurately capture details and describe small objects within large scenes, while a state-of-the-art method hallucinates object details [9]. Although fine-grained attributes are critical for performance, directly extracting detailed information from high-resolution 3D data for 3DLAs remains underexplored.

In this work, we aim to enhance 3DLAs' ability to perceive finer details in point clouds in order to execute downstream tasks more accurately. While increasing the number of visual tokens as the point cloud grows in size seems the straightforward solution, our empirical study (Tab. 3) shows that this solution has limited effectiveness in capturing scene details and it just increases computational burden. To address this, we propose PerLA, a novel 3DLA with a perceptive 3D scene encoder that captures detailed point cloud information, allowing the language model to generate *more accurate responses without processing additional tokens*. PerLA first divides the complete 3D scene into non-overlapping local parts to be processed in parallel, then integrates this local information with the global context obtained from a lower-resolution representation of the entire point cloud. Although dividing visual input has gained popularity in image processing [35], it has not yet been applied to point clouds. Processing point clouds presents unique challenges beyond those of images, as it requires handling an unordered set of points rather than rasterized pixels. To address these challenges, we serialize and partition the point cloud before encoding using a Hilbert curve approach [50], which efficiently preserves locality. We then combine local and global information through an efficient Hilbert curve-based k-NN search and aggregate this information via a novel cross-attention module and Graph Convolutional Network (GCN) to generate highly informative point-level representations for the LLM. Lastly, we train PerLA with a novel loss function designed to promote consensus on local representation, addressing the issue of divergent representations during local-to-global aggregation. We validate our approach on the question answering benchmark ScanQA [2], and the 3D dense captioning benchmarks ScanRefer [6] and Nr3D [1] to demonstrate the effectiveness of PerLA. PerLA demonstrates high transferability on both tasks of 3D question answering and 3D dense captioning, achieving state-of-the-art performance. In summary, our contributions are:

- We present a novel perceptive 3D encoder to preserve local and global information for 3D language assistant.
- We introduce an efficient approach based on Hilbert curve k-NN search and cross attention to aggregate local and global information at point level.
- We propose a novel algorithm based on Graph Neural Network to refine and enhance aggregated information.
- We introduce a novel loss objective to enable local representation consensus for local-to-global information aggregation.

2. Related work

3D language understanding involves understanding 3D scenes by describing or answering to scene-relevant questions in the format of natural language. Examples of typical downstream tasks include *3D Dense Captioning* [8, 12, 56], and *3D Question Answering* [2, 39, 65, 67]. *3D Dense Captioning* consists of 3D localization of object instances, and textual descriptions of each object instance [4, 8, 10, 12]. *3D Question Answering* requires the model equipped with a language decoder to answer questions regarding the visual context in the given 3D scene [2, 44]. Several works focus on addressing the problems of 3D-language pre-alignment [13, 26], or designing adapter layers [9, 21], or constructing 3D synthetic data [64]. In contrast, our work focuses on designing an encoding approach that can enhance 3D-language understanding by capturing fine-grained details.

3D language assistants (3DLAs) can be roughly categorized into two types: object-level 3DLAs [23, 46, 53, 61] and scene-level 3DLAs [9, 21, 59, 69]. Object-level 3DLAs learn from large 3D object datasets [15] to connect object-level 3D representation with language models. However, they underperform in compositional reasoning in complex 3D scenes with numerous objects. Scene-level 3DLAs, such as 3D-LLM [21], LLaVA-3D [69], Chat-3D [59], and LL3DA [9] enable scene understanding via the interaction with objects. Because 3D data is orders of magnitude less than 2D data, existing 3DLAs address such lack of data by leveraging pre-trained 2D Large Multimodal Models (LMMs) [21, 69], or data-efficient training recipes [9, 59]. 3D-LLM [21] leverages 2D pre-trained representations of rendered multi-view images to construct 3D representations and 2D VLMs as backbones. LLaVA-3D [69] adapts LLaVA [36] for 3D scene understanding by associating 2D patch representations with their positions in 3D space. Chat-Scene [22] improves the referencing and grounding capability and models scene representations as a sequence of object-level representations. LL3DA [9] extracts point-level representations from a downsampled 3D scene, and includes both interaction prompts and textual instructions to resemble human interactions with the 3D environment. Orthogonal to the efforts of aligning 3D-language with limited 3D data in existing 3DLAs, PerLA aims to improve the capability of 3DLAs in perceiving scene details.

Visual perception enhancement on multimodal models. Multi-granularity representation learning has been explored in 2D multimodal models [5, 18, 20, 29, 49, 51, 63], showing that combining local and global views yields more informative representations than relying on a single global view. Mini-Gemini [32] introduces a dual vision encoder setup using CLIP ViT [48] and ConvNeXt [37] to process low- and high-resolution views of an input image. Models in the LLaVA-Next series [30, 35, 62] and InternLM-XComposer2 [17] use an additional branch to handle view partitions, increasing the number of visual tokens and preserving more fine-grained visual details compared to language-vision models with a single global view branch. In the 3D domain, Scene-LLM [19] and Segment3D [24] improve segmentation accuracy by transferring semantic details from multi-view images to point clouds. While extensive work has been done to enhance perception in 2D models and transfer multi-view image information into 3D, methods for preserving detailed representations from point clouds in 3DLAs remain underexplored. Our method aims to address this gap by preserving both global and local visual information to enhance 3D perception.

3. Perceptive 3D language assistant (PerLA)

PerLA takes as inputs *i)* a text prompt in natural language, *ii)* the 3D scene in the form of a point cloud, and *iii)* a visual prompt provided as either a user click or a bounding box. The text prompt is processed by a *text prompt encoder* to produce text representations, which are then input to both the *Large Language Model* (LLM) and the *multimodal adapter* (MMA). The text encoder is a transformer based on BLIP-2 [31]. The point cloud is processed by our *perceptive scene encoder*, which generates scene representations that feed into both the MMA and the subsequent encoder. We will detail the perceptive scene encoder in the next sections. The visual prompt is processed by the *visual prompt encoder*, which, by combining the perceptive scene encoder's representations, outputs scene representations that are further processed by the MMA. For more details on visual prompts, please refer to Supp. Mat. The MMA takes as input these multimodal representations and outputs tokens for the LLM. The MMA is implemented as a Q-former [31]. MMA's output is projected into the LLM's representation space through linear projector. Lastly, these projected representations are processed by the LLM to generate the output response. We train PerLA using data provided by [21] and finetune it for each downstream task. Fig. 2 illustrates our approach.

Formally, let $\mathcal{P}=\{\left(\boldsymbol{p}_i\in\mathbb{R}^3, \boldsymbol{f}_i\in\mathbb{R}^{d_0}\right) \mid i = 1, 2, \ldots, N\}$ denote PerLA's input point cloud, where $\boldsymbol{p}_i$ is a point coordinate, $\boldsymbol{f}_i$ is a d_0-dimensional feature vector (e.g., color or normal vector) corresponding to $\boldsymbol{p}_i$, and N is the number of points (cardinality). Let $\mathcal{I}^t$ and $\mathcal{I}^v$ denote the input text prompt and visual prompt, respectively. Based on these inputs, PerLA (denoted as Φ) generates a free-form natural language response $\mathcal{O}^t$ for various 3D-related tasks.

3.1. Perceptive scene encoder

To compute point-level representations through the scene encoder, $\mathcal{P}$ is typically downsampled into super-points using Farthest Point Sampling (FPS) [8, 9]. However, downsampling and using only $\mathcal{P}$ may hinder the encoder to capture fine-grained details, thus affecting performance in downstream 3D scene understanding tasks. Our proposed perceptive scene encoder is designed to preserve such scene details without increasing the number of tokens or the representation dimensions of the 3D scene. We propose to split $\mathcal{P}$ into parts, and employ a pre-trained 3D scene encoder to encode these parts and the whole point cloud separately. We then aggregate the output representations from these different pieces into a single (highly-informative) representation using a cross-attention-based module and Graph Convolutional Network (GCN).

Hilbert-based scene partitioning. We partition $\mathcal{P}$ into L equally-sized parts, each containing the same number of points, $\lfloor \frac{N}{L} \rfloor$, where $\lfloor \cdot \rfloor$ denotes the greatest integer less than or equal to its argument. Enforcing equal cardinality across parts allows for spatially smaller parts in highly structured regions (areas with more semantic information) and spatially larger parts in less structured regions (areas with less semantic information). To achieve this, we adopt a Hilbert curve approach [50] to efficiently serialize the point cloud and partition it into parts [33, 57, 60].

Global and partial scene encoding. Both full-scene point cloud and the partial-scene point clouds are encoded separately by the same pre-trained 3D scene encoder ϕ. The 3D scene encoder ϕ downsamples the input point cloud into super-points using Farthest Point Sampling (FPS) [41] and produces a representation for each super-point. The representation of the full scene, *i.e.*, the global representation, encode the overall scene context, while the representations of the partial scenes, *i.e.*, the local representations, encode scene details. Let $\mathcal{P}^g = \{\boldsymbol{p}_i^g \in \mathbb{R}^3\}_{i=1}^M$ denote the downsampled full-scene point cloud composed of M points, and $\mathcal{F}^g = \{\boldsymbol{f}_i^g \in \mathbb{R}^d\}_{i=1}^M$ the associated global representations. Let $\mathcal{P}^l = \{\boldsymbol{p}_i^l \in \mathbb{R}^3\}_{i=1}^{L\cdot M}$ denote the set of downsampled partial-scene point clouds, and $\mathcal{F}^l = \{\boldsymbol{f}_i^l \in \mathbb{R}^d\}^{L\cdot M}$ the associated local representations. Note that, we aggregate all the points of the downsampled partial-scene point clouds into the single set $\mathcal{P}^l$, and both the full-scene and the partial-scenes are downsampled to the same number of super-points M. The representations $\mathcal{F}^l$ provide a more detailed view of the scene, as they are derived from the same number of points downsampled from smaller, localized regions, resulting in higher resolution compared to the global versions. Yet, their semantic visibility is focused within each local part. In the following steps, we aggregate local and global information, to enrich the representations with details within the scene context.

Hilbert-based nearest-neighbor search. In order to enhance global representations through local representations, we need to first find the correspondences among them, *i.e.*, which super-points of partial-scene point clouds are spatially neighbors with super-points of the full-scene point

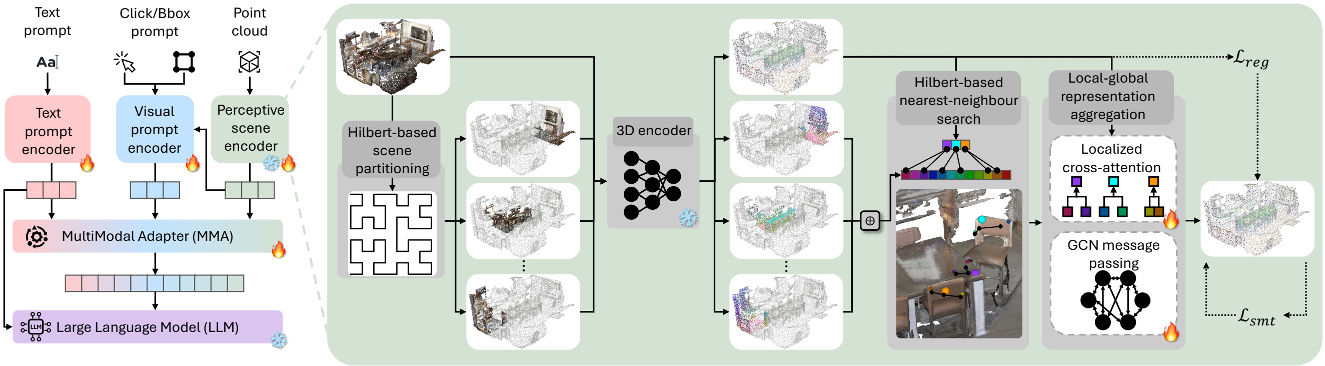


Figure 2. Overview of PerLA. (Left): The overall pipeline of PerLA, which begins by extracting interaction-aware 3D scene representations. These representations are then projected onto the prefix of textual instructions via MMA, serving as input to a frozen language model (LLM). (Right): The detailed design of PerLA. First, the 3D scene is divided into spatially compact regions using Hilbert-based scene serialization [50]. Next, an efficient k-NN algorithm associates each point-level global representation with its detail-enriched local representations, creating a comprehensive scene representation through a Graph Convolutional Network (GCN). Finally, smoothness and regularization losses are applied to promote stable learning for the proposed perceptive scene encoder.

cloud. This can be done via k-NN search. However, traditional k-NN searches can be computationally intensive on large-scale or high-resolution point clouds and cannot ensure that the identified neighbors maintain geometric consistency. To address these limitations, we approximate k-NN with an efficient neighbor mapping technique using Hilbert serialized point clouds, which improves query speed while reducing computational complexity. With serialized point clouds, we can leverage point indices to perform k-NN searches with $O(1)$ complexity [60].

Specifically, we first serialize the union $\mathcal{P}^g \bigcup \mathcal{P}^l$ using the Hilbert curve ordering. We then apply geometric partitioning [27] on the original point clouds to generate geometric labels $\mathcal{Y}^g \bigcup \mathcal{Y}^l$. The center points of these labels are incorporated as high-order bits (by bit shifting) in the serialized index, ensuring that points with different labels occupy non-overlapping index ranges. Next, for each $\boldsymbol{p}_i^g \in \mathcal{P}^g$, we identify the $\mathtt{k}$ nearest super-points within $\mathcal{P}^l$ based on the serialized indexes. These geometric labels guarantee that $\boldsymbol{p}_i^g$ and its nearest local super-points all originate from the same instance. Let $\mathcal{P}^l_{\mathtt{k}_i}$ denote the set of $\mathtt{k}$ nearest local super-points to the global super-point $\boldsymbol{p}_i^g$. Consequently, $\mathcal{F}^l_{\mathtt{k}_i}$ are the representations of the super-points in $\mathcal{P}^l_{\mathtt{k}_i}$.

Local-global representation aggregation. We make global representations more informative by combining them with information from local representations. To enable this, we present a novel two-step aggregation technique. In the first step, we employ a novel cross-attention algorithm between local and global representations using neighborhood information to update global representations, which we term as *localized cross-attention*. In the second step, we refine global representations through a Graph Convolutional Network based formulation based on message passing, which we term as *GCN message passing* (Fig. 2).

Localized cross-attention operates with a learning-based weighting mechanism that is a function of the representations and relative positions of global super-points and their associated local neighbors. We constrain cross-attention to local neighborhood regions because *i)* limiting the number of points reduces computational complexity, and *ii)* neighbors of a given point are likely to belong to the same object. Specifically, for each point $\boldsymbol{p}_i^g$ and its nearest neighbors $\mathcal{P}^l_{\mathtt{k}_i}$, their relative position embeddings $\mathcal{R}_i$ are extracted as

$$\mathcal{R}_i = \left\{\mathcal{R}_{ij} = \mathrm{pos}\left(\left(\boldsymbol{p}_i^g - \boldsymbol{p}_j^l\right)/\sigma\right) \in \mathbb{R}^d\right\}_{j=1}^{K^l}, \quad (1)$$

where $\sigma > 0$ is a learnable parameter that controls the relative position scaling, and $\mathrm{pos}(\cdot)$ is the 3D Fourier positional embedding [52] operation, which satisfies:

$$\mathrm{pos}(\boldsymbol{x}) = \left[\sin\left(2\pi\boldsymbol{x}\cdot B\right);\cos\left(2\pi\boldsymbol{x}\cdot B\right)\right], \quad (2)$$

where $B \in \mathbb{R}^{3\times(d/2)}$ is a learnable matrix. We update the global representations through cross-attention as follows

$$\begin{aligned}\boldsymbol{s}_{ij} &= \left(W_q \boldsymbol{f}_i^g\right)^\top \left(W_k\left(\boldsymbol{f}_j^l + W_r \mathcal{R}_{ij}\right)\right)/\sqrt{d},\\ \boldsymbol{s}_i &= \{\boldsymbol{s}_{ij}\}_{i=1}^{K^l}, \boldsymbol{w}_i = \mathrm{softmax}\left(\boldsymbol{s}_i\right),\\ \hat{\boldsymbol{f}}_i^g &= \boldsymbol{f}_i^g + \boldsymbol{w}_i\left(W_v\left(\mathcal{F}^l_{\mathcal{K}_i} + \mathcal{R}_i\right)\right),\end{aligned} \quad (3)$$

where $\hat{\boldsymbol{f}}_i^g$ are the updated representations, and W_q, W_k, W_v, W_r are projection parameter matrices. Let $\hat{\mathcal{F}}^g = \{\hat{\boldsymbol{f}}_i^g\}_{i=1}^M$ denote the set of $\hat{\boldsymbol{f}}_i^g$.

GCN message passing further refines $\hat{\mathcal{F}}^g$ by aggregating information from neighboring global super-points. Let $\mathcal{G}^g = \{\mathcal{P}^g, \mathcal{W}^g\}$ be a k-NN graph, where $\mathcal{P}^g$ is the set of the vertices, *i.e.*, global super-points, and $\mathcal{W}^g \in \mathbb{R}^{N\times N}$ is the adjacency matrix defining the edges. The construction of the adjacency matrix uses the similarity between super-point representations $\mathcal{F}^G$. Note that we use the (original) global representations to define the adjacency matrix because we empirically experienced more stability in training as opposed to using the updated global representation, likely due to the fact that they change value during training. The adjacency matrix $\mathcal{W}^g$ with elements $\mathcal{W}^g_{ij}$ is defined as

$$\mathcal{W}^g_{ij}=\begin{cases}\frac{\boldsymbol{f}_i^{g\top}\boldsymbol{f}_j^g}{\|\boldsymbol{f}_i^g\|\|\boldsymbol{f}_j^g\|}\cdot\left(\boldsymbol{f}_i^{g\top}\boldsymbol{f}_j^g>0\right), & \text{if } p_j^g\in k\text{-NN}(p_i^g),\\ 0, & \text{otherwise.}\end{cases}$$

Lastly, we perform one-time GCN message passing (MP) on the graph $\mathcal{G}^g$ to obtain an updated feature matrix $\hat{\mathcal{F}}^g$.

Let $\tilde{\mathcal{W}}^g$ be the symmetrically normalized adjacency matrix of $\mathcal{W}^g$ by $\tilde{\mathcal{W}}^g = \mathbf{D}^{-\frac{1}{2}}\mathcal{W}^g\mathbf{D}^{-\frac{1}{2}} \in \mathbb{R}^{M\times M}$ where $\mathbf{D}$ is the diagonal degree matrix of $\mathcal{W}^g$.

$$\hat{\mathcal{F}}^g \leftarrow \text{ReLU}\left(W_m\left(\tilde{\mathcal{W}}^g\hat{\mathcal{F}}^g\right)+W_s\hat{\mathcal{F}}^g\right), \tag{4}$$

where W_m and W_s are trainable parameters. This MP layer structure allows each node to aggregate information from its neighbors while also preserving its own feature representation (via the skip connection).

3.2. Learning with local representation consensus

The main learning objective encourages the model to generate outputs that are close to the target responses in the training dataset. This objective is typically expressed as a per-token cross-entropy loss $\mathcal{L}_{pred}$ [9, 21] (see details in our Supp. Mat.). We found that adding a consensus term to $\mathcal{L}_{pred}$ that encourages local regularization of representations improves training stability. Firstly, points belonging to the same object (*i.e.*, local neighborhoods) should share similar aggregated representations. Secondly, aggregated super-point representations should remain close to their original point representations to preserve context knowledge. Our novel consensus loss $\mathcal{L}_{\text{con}}$ is defined as

$$\begin{aligned}\mathcal{L}_{\text{con}} &= \mathcal{L}_{\text{smt}} + \mu\mathcal{L}_{\text{reg}},\\ \mathcal{L}_{\text{smt}} &= \sum_{i=1}^{N}\sum_{j=1}^{N}\mathcal{W}^g_{ij}\left\|\frac{\hat{\boldsymbol{f}}^g_i}{\sqrt{d_i}}-\frac{\hat{\boldsymbol{f}}^g_j}{\sqrt{d_j}}\right\|,\\ \mathcal{L}_{\text{reg}} &= \sum_{i=1}^{N}\left\|\hat{\boldsymbol{f}}^g_i-\boldsymbol{f}^g_i\right\|,\end{aligned} \tag{5}$$

where $d_k = \sum_l \mathcal{W}^g_{kl}$ is the diagonal matrix representing row-wise sums of $\mathcal{W}^g$. The first term $\mathcal{L}_{\text{smt}}$ enforces spatial connectivity by promoting similarity among fused representations of points within the same object, addressing the first attribute. $\mathcal{L}_{\text{reg}}$ is a regularization term that promotes the learned representations stay close to the original global representations. μ is a hyperparameter. The overall loss $\mathcal{L}$, is a weighted sum of $\mathcal{L}_{pred}$ and $\mathcal{L}_{con}$

$$\mathcal{L} = \lambda\mathcal{L}_{con} + \mathcal{L}_{pred}, \tag{6}$$

where λ is the balance hyperparameter. By jointly training with these two loss terms, we encourage the model to learn more object-aware representations and enhance its ability to extract finer details. This approach also preserves global context information, which is essential for capturing spatial relationships between different objects.

During inference, we use beam search to predict response $\mathcal{O}^t$ that maximizes the following objective:

$$\mathcal{O}^t = \arg\max_{\mathcal{O}}\Phi\left(\mathcal{O}\mid\mathcal{P},\mathcal{I}^t,\mathcal{I}^v\right). \tag{7}$$

where we set a beam size of 4.

4. Experiments

We evaluate PerLA on 3D Question Answering and 3D Dense Captioning downstream tasks, and compare its performance with state-of-the-art methods from the literature.

Datasets. We conduct experiments using the ScanNet dataset [14], which encompasses 1,201 training and 312 validation scenes, featuring diverse and complex indoor 3D environments. Language annotations are from ScanQA [2], ScanRefer [6], Nr3D [1], and the ScanNet subset of 3D-LLM [21], collectively supporting a range of tasks including instance and scene descriptions, conversations, embodied planning, and question answering. For additional data statistics, please refer to the supplementary materials.

Metrics. We follow LL3DA's evaluation protocol [9] to evaluate the quality of output responses. We use the abbreviations C, B4, M and R for CiDEr [55], BLEU-4 [43], METEOR [3], and Rouge-L [34], respectively.

Implementation details. As in [9, 12], we input 40,000 randomly sampled points from each 3D scene. We divide the point cloud into partitions L=6 and choose `k`=4L=24 neighbors to enhance global representations. We set λ=μ=0.1 in the loss. We use the pre-trained OPT-1.3B [66] language model, kept frozen and loaded in `float16` precision. We use the AdamW [38] optimizer with a weight decay of 0.1, applying a cosine annealing scheduler that decays the learning rate from 10^{-4} to 10^{-6} over approximately 100,000 iterations. All tasks are trained with a total batch size of 16. Each training process is completed within two days using up to two NVIDIA H100 P0 (96GB) GPUs. For each evaluation, we fine-tune the model's parameters on the respective task for about 30,000 iterations.

4.1. Results

3D question answering is a task that involves answering questions about a 3D scene. It allows a model to provide information about the objects, relationships, and attributes within a 3D environment based on a given question. Tab. 1 shows the results of ScanQA's validation and test sets. Classification-based methods (CLS) select responses from a predefined answer set. Generation-based approaches (GEN) generate the entire textual response. PerLA consistently outperforms existing approaches across all evaluation sets and metrics, in particular with +1.34 CiDEr score over LL3DA. We also compare against our reproduced version of LL3DA, where PerLA scores +3.76 CiDEr.

Table 1. Comparative results for 3D Question Answering on ScanQA [2] benchmark. CLS and GEN denote classification-based and generation-based methods, respectively. LL3DA (repr.) is results of LL3DA we reproduced. PerLA outperforms all the other methods.

Method	Type	Validation				Test w/ object				Test w/o object			
		C↑	B4↑	M↑	R↑	C↑	B4↑	M↑	R↑	C↑	B4↑	M↑	R↑
ScanQA[2]	CLS	64.86	10.08	13.14	33.33	67.29	12.04	13.55	34.34	60.24	10.75	12.59	31.09
Clip-Guided[44]	-	-	-	-	-	69.53	14.64	13.94	35.15	62.83	11.73	13.28	32.41
Multi-CLIP[16]	CLS	-	-	-	-	68.70	12.65	13.97	35.46	63.20	12.87	13.36	32.61
3D-VLP[26]	CLS	66.97	11.15	13.53	34.51	70.18	11.23	14.16	35.97	63.40	15.84	13.13	31.79
3D-VisTA[70]	-	-	-	-	-	68.60	10.50	13.80	35.50	55.70	8.70	11.69	29.60
3D-LLM [21]	GEN	69.40	12.00	14.50	35.70	69.60	11.60	14.90	35.30	-	-	-	-
LL3DA[9]	GEN	76.79	13.53	15.88	37.31	78.16	13.97	16.38	38.15	70.29	12.19	14.85	35.17
LL3DA (repr.)	GEN	74.37	13.50	15.09	36.31	-	-	-	-	-	-	-	-
PerLA	GEN	**78.13**	**14.49**	**17.44**	**39.60**	**80.91**	**17.21**	**16.49**	**40.71**	**74.82**	**14.97**	**15.23**	**38.18**
Δ w.r.t. LL3DA[9]	-	+1.34	+0.96	+1.56	+2.29	+2.75	+3.24	+0.11	+2.56	+4.53	+2.78	+0.38	+3.01

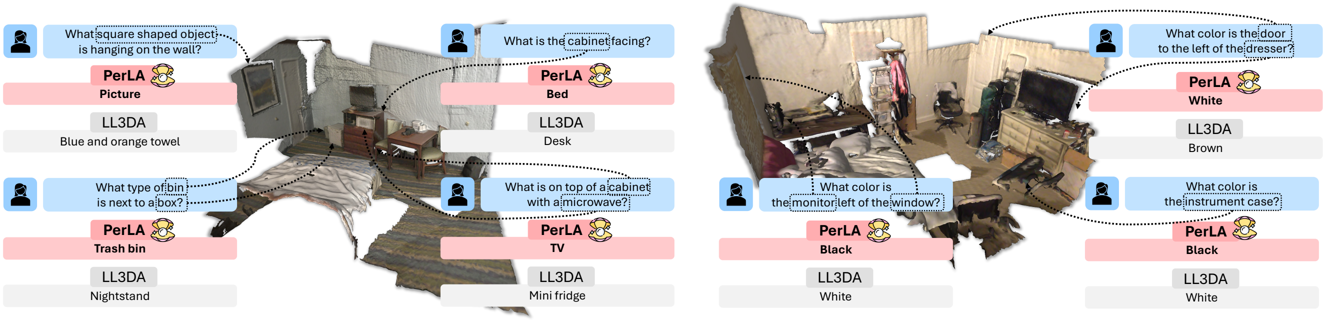

What square shaped object is hanging on the wall?
PerLA
Picture
LL3DA
Blue and orange towel
What type of bin is next to a box?
PerLA
Trash bin
LL3DA
Nightstand
What is the cabinet facing?
PerLA
Bed
LL3DA
Desk
What is on top of a cabinet with a microwave?
PerLA
TV
LL3DA
Mini fridge
What color is the monitor left of the window?
PerLA
Black
LL3DA
White
What color is the door to the left of the dresser?
PerLA
White
LL3DA
Brown
What color is the instrument case?
PerLA
Black
LL3DA
White

Figure 3. The qualitative comparison between our method, PerLA, and LL3DA [9] on the ScanQA [2] dataset shows that our approach achieves higher accuracy in responding to "what"-related questions.

Fig. 3 provides qualitative comparisons between LL3DA and PerLA on the ScanQA benchmark [2]. It highlights PerLA's accuracy in answering questions regarding object attributes and spatial relationships within a 3D scene. PerLA provides precise answers, correctly identifying objects such as a"picture", "trash bin", and "TV" along with their specific colors and types, while LL3DA often yields incorrect or less specific responses. In particular, PerLA accurately identifies the colors of objects within the scene, such as the door, the monitor, and the instrument case, correctly answering the questions about these attributes. In contrast, LL3DA frequently misidentifies colors, labeling objects as "White" instead of "Black," for example. This illustrates PerLA's superior ability to capture fine-grained details and contextual information, delivering more accurate answers in complex environments.

3D dense captioning involves localizing and describing each 3D instance within complex 3D environments. Tab. 2 shows the results on ScanRefer [6] and Nr3D [1] benchmarks. Following previous works [9, 21], we use the m@kIoU metric, where $m \in \{\text{C, B4, M, R}\}$ and k denote the IoU threshold. As in [12, 25], we report C@0.25 and C@0.5 for ScanRefer, and C@0.5 for Nr3D. UniT3D [13], 3DJCG [4], and 3D-VLP [26] are pre-trained on multiple 3D vision and language tasks from ScanNet scenes. UniT3D uses image caption models and multi-view images to generate extra instance captions for pre-training. For fair comparison, we report results from models trained using standard per-word cross-entropy loss without additional 3D scenes. Box annotations are estimated with Vote2CapDETR and used as visual prompts. PerLA consistently outperforms comparison methods on both benchmarks. PerLA significantly outperforms LL3DA by scoring +3.75 and +4.22 CiDEr on ScanRefer, and +3.88 CiDEr on Nr3D. These results highlight the effectiveness of PerLA in 3D dense captioning tasks.

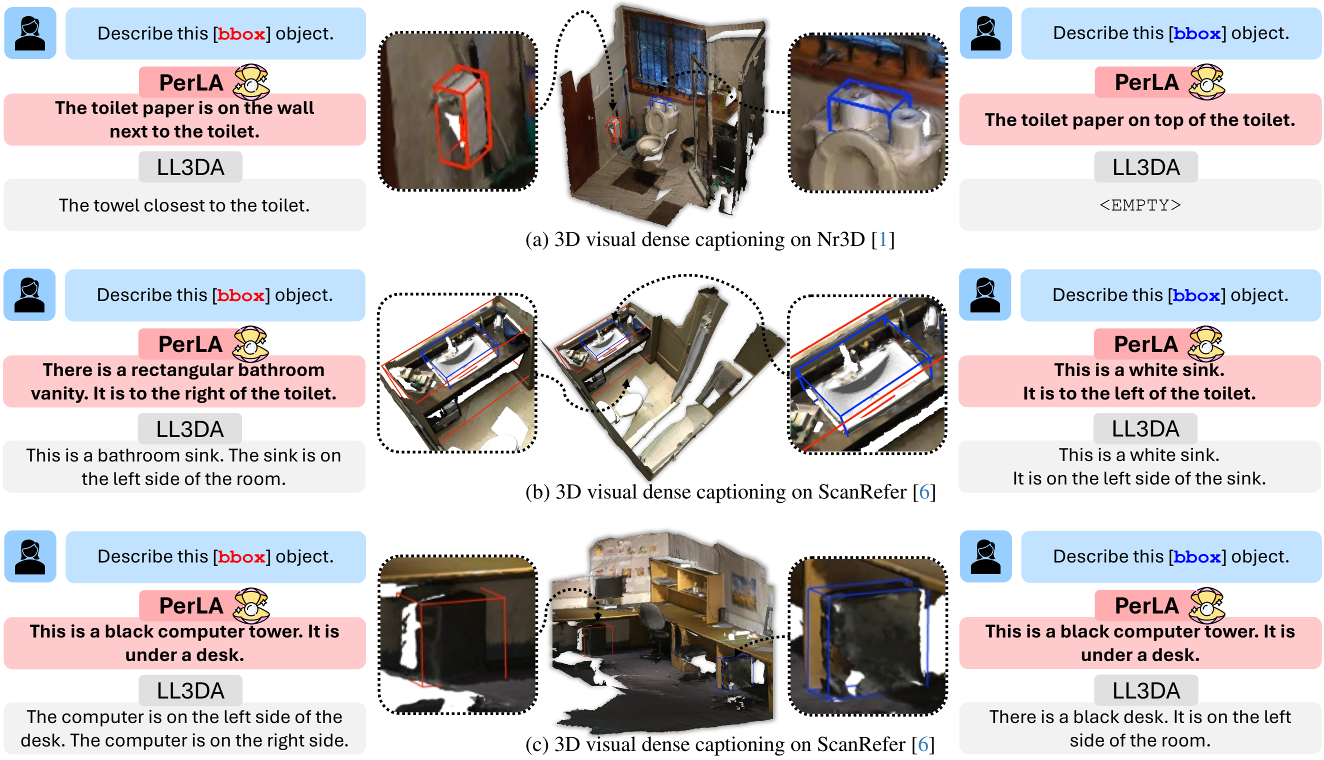
Fig. 4 provides examples of qualitative results of LL3DA and PerLA on Nr3D and ScanRefer. PerLA produces more accurate and detailed descriptions, effectively capturing spatial relationships and fine-grained attributes, such as the positioning of objects relative to surrounding elements.

4.2. Ablation study

Local, global, and GCN representations. To investigate the impact of local and global representations, we design two variants of PerLA: one using only local representations and the other using only global representations. We also report an extended version of LL3DA, termed LL3DA†, in which we increase the number of query tokens used to encode local partitions. Representations are first extracted from each partition, and FPS is applied to sample 1,024 points with their corresponding representations from the

Table 2. Comparative results for 3D Dense Captioning on ScanRefer [6] and Nr3D [1] benchmarks. LL3DA (repr.) is results of LL3DA we reproduced. Generally, PerLA outperforms all other methods on both benchmarks.

Method	ScanRefer@0.25				ScanRefer@0.5				Nr3D@0.5			
	C↑	B4↑	M↑	R↑	C↑	B4↑	M↑	R↑	C↑	B4↑	M↑	R↑
Scan2Cap[12]	56.82	34.18	26.29	55.27	39.08	23.32	21.97	44.78	27.47	17.24	21.80	49.06
MORE[25]	62.91	36.25	26.75	56.33	40.94	22.93	21.66	44.42	-	-	-	-
SpaCap3D[56]	-	-	-	-	-	44.02	25.26	22.33	33.71	19.92	22.61	50.50
REMAN[40]	62.01	36.37	26.76	56.25	45.00	26.31	23.13	46.96	34.81	20.37	22.71	50.90
D3Net[7]	-	-	-	-	-	51.67	-	-	35.26	20.42	22.77	53.38
Contextual[68]	-	-	-	-	-	46.07	23.40	23.95	-	-	-	-
UniT3D[13]	-	-	-	-	46.69	27.52	21.91	45.98	-	-	-	-
3DJCG[4]	64.70	40.17	27.63	59.23	49.48	31.63	24.36	50.80	38.06	22.82	23.77	52.99
3D-VLP[26]	70.73	41.03	28.14	**59.72**	54.94	32.31	24.83	51.51	-	-	-	-
3D-VisTA*[70]	-	-	-	-	61.60	34.10	26.80	55.00	-	-	-	-
Vote2CapDETR[8]	71.45	39.34	28.25	59.63	61.81	34.46	26.22	54.40	43.84	26.68	25.41	54.43
LL3DA[9]	74.17	41.41	27.76	59.53	65.19	36.79	25.97	55.06	51.18	28.75	25.91	56.61
LL3DA (repr.)	71.86	39.57	27.29	58.37	63.79	35.67	25.94	54.56	48.38	28.36	25.72	55.66
PerLA	**77.92**	**43.41**	**28.97**	59.69	**69.41**	**38.02**	**29.07**	**56.80**	**55.06**	**31.24**	**28.52**	**59.13**
Δ w.r.t. LL3DA[9]	+3.75	+2.00	+1.21	+0.16	+4.22	+1.23	+2.27	+1.74	+3.88	+2.49	+2.61	+2.52


Describe this [bbox] object.

PerLA: The toilet paper is on the wall next to the toilet.

LL3DA: The towel closest to the toilet.

Describe this [bbox] object.

PerLA: The toilet paper on top of the toilet.

LL3DA: <EMPTY>

(a) 3D visual dense captioning on Nr3D [1]

Describe this [bbox] object.

PerLA: There is a rectangular bathroom vanity. It is to the right of the toilet.

LL3DA: This is a bathroom sink. The sink is on the left side of the room.

Describe this [bbox] object.

PerLA: This is a white sink. It is to the left of the toilet.

LL3DA: This is a white sink. It is on the left side of the sink.

(b) 3D visual dense captioning on ScanRefer [6]

Describe this [bbox] object.

PerLA: This is a black computer tower. It is under a desk.

LL3DA: The computer is on the left side of the desk. The computer is on the right side.

Describe this [bbox] object.

PerLA: This is a black computer tower. It is under a desk.

LL3DA: There is a black desk. It is on the left side of the room.

(c) 3D visual dense captioning on ScanRefer [6]

Figure 4. Qualitative comparisons on the dense captioning task across the Nr3D [1] and ScanRefer [6]. We compare the results of our PerLA with LL3DA [9]. PerLA generates accurate descriptions, effectively capturing fine-grained object attributes and spatial relationships.

union of these partitions. These representations are then passed through a multimodal adapter to produce 32 global tokens. LL3DA generates an additional 32 global tokens from the point cloud of the entire scene, resulting in a total of 64 tokens, which are concatenated and fed into the LLM for response generation. (For details of LL3DA† please refer to the Supp. Mat.). Tab. 3 shows that the combination of local and global representations yields a much better performance than the two independently used. Tab. 3 also shows that LL3DA† does not reach PerLA performance, underlying the importance of information exchange between local and global representations. To evaluate the effectiveness of GCN, we also report the results without using GCN (w/o GCN). Compared to our PerLA, which incorporates GCN, it consistently improves performance across the three datasets, confirming its effectiveness.

Table 3. Ablation study of our local-to-global, GCN representation aggregation algorithms on ScanQA, ScanRefer[6] and Nr3D[1] benchmarks. LL3DA† denotes an extended version of LL3DA with an increased number of query tokens.

Method	ScanQA				ScanRefer@0.5				Nr3D@0.5			
	C↑	B4↑	M↑	R↑	C↑	B4↑	M↑	R↑	C↑	B4↑	M↑	R↑
LL3DA†	74.54	12.89	15.11	36.96	62.25	34.50	25.55	53.84	48.70	28.24	25.72	55.59
Global	74.49	13.50	15.16	36.55	63.79	35.67	25.94	54.56	48.38	28.36	25.72	55.66
Local	73.55	12.98	14.95	36.07	62.79	34.78	25.94	54.16	49.09	28.20	25.80	55.89
w/o GCN	77.61	13.87	15.93	39.28	69.04	37.43	28.71	55.59	54.30	29.78	27.41	58.32
PerLA	**78.13**	**14.49**	**17.44**	**39.60**	**69.41**	**38.02**	**29.07**	**56.80**	**55.06**	**31.24**	**28.52**	**59.13**

Table 4. Ablation study on the impact of constrained attention on ScanQA validation dataset.

cross-attention				mean pooling				max pooling			
C↑	B4↑	M↑	R↑	C↑	B4↑	M↑	R↑	C↑	B4↑	M↑	R↑
78.1	**14.5**	**17.4**	**39.6**	74.4	12.5	15.2	36.5	74.0	13.0	14.8	35.7

Table 5. Ablation study on the impact of joint learning by different loss combination on ScanQA validation dataset.

Loss			ScanQA (Validation)			
$\mathcal{L}_{\text{pre}}$	$\mathcal{L}_{\text{smt}}$	$\mathcal{L}_{\text{reg}}$	C↑	B4↑	M↑	R↑
✓			75.31	13.83	15.90	37.48
✓	✓		76.62	13.96	16.99	38.14
✓		✓	76.47	14.07	16.72	38.79
✓	✓	✓	**78.13**	**14.49**	**17.44**	**39.60**

Table 6. Ablation study on the impact of number of parts on ScanQA validation dataset.

4				6				8			
C↑	B4↑	M↑	R↑	C↑	B4↑	M↑	R↑	C↑	B4↑	M↑	R↑
76.9	13.6	16.1	37.3	**78.1**	**14.5**	**17.4**	**39.6**	78.0	14.5	17.2	39.4

Localized cross-attention. To evaluate the effectiveness of our novel localized cross-attention module, we tested max pooling and mean pooling as alternative methods. Tab. 4 shows that localized cross-attention significantly outperforms both mean pooling and max pooling across all metrics. These results indicate that localized cross-attention enhances ability of model to generate more informative scene representations, thereby improving performance on 3D question answering tasks. This improvement comes from the incorporation of positional information and semantic similarity in cross-attention, which helps to exclude neighboring points that do not belong to the same object.

Loss function. To evaluate the effectiveness of our novel loss term $\mathcal{L}_{\text{con}}$, we train PerLA using different combinations of $\mathcal{L}_{\text{con}}$'s components (Eq. 5). Since our task is intended for downstream applications, each combination also includes the task-specific loss $\mathcal{L}_{\text{pre}}$. Tab. 5 shows that adding $\mathcal{L}_{\text{smt}}$ to $\mathcal{L}_{\text{pre}}$ improves all metrics (C, B4, M, and R). Incorporating the loss of regularization $\mathcal{L}_{\text{reg}}$ further enhances performance, particularly in metrics C, M, and R. These results show the effectiveness of learning jointly with both semantic awareness and regularization losses.

Number of partitions. To evaluate the effectiveness of different number of partitions, we test PerLA by splitting the point cloud into 4, 6, and 8 partitions. Tab. 6 shows that increasing the number of partitions generally improves performance, with 6 partitions yielding the best results. Although 8 partitions outperform 4, they show slightly lower performance than 6. This suggests that 6 partitions provide a better balance between granularity and effectiveness, capturing relevant scene details with lower computational complexity. More ablation studies, including an analysis of performance in different training strategies, refer to Supp. Mat..

5. Conclusions

We presented PerLA, a perceptive 3D language assistant capable of capturing both detailed and contextual information to enhance visual representations for LLMs. PerLA features a dual-branch architecture: the global branch processes superpoints from the whole point cloud via downsampling, while the local branch focuses on partitioned regions. We demonstrated that by integrating representations from both branches, PerLA effectively captures scene details, reducing hallucinations. Moreover, we employ a Graph Convolutional Network to facilitate information exchange among neighboring local and global superpoints, and introduce a novel loss term for local representation consensus to promote training stability. Experiments on the ScanQA, ScanRefer, and Nr3D benchmarks highlight the effectiveness of our approach, setting a new state-of-the-art performance in 3D question answering and dense captioning.

Limitations & Future Work. PerLA primarily focuses on enhancing performance using point cloud input, but integrating it with optimized modules, such as token merging, presents a promising direction to extend its capabilities. Although PerLA shows strong performance on standard benchmarks, future work could explore its robustness and generalizability in more complex 3D scenarios, broadening its applicability in diverse real-world settings.

Acknowledgement. This work was supported by PNRR FAIR - Future AI Research (PE00000013) and ICSC National Research Centre for HPC, Big Data and Quantum Computing (CN00000013), funded by NextGeneration EU.

References

[1] Panos Achlioptas, Ahmed Abdelreheem, Fei Xia, Mohamed Elhoseiny, and Leonidas Guibas. Referit3d: Neural listeners for fine-grained 3d object identification in real-world scenes. In *ECCV*, pages 422–440. Springer, 2020. 2, 5, 6, 7, 8, 14, 15, 16, 17, 18

[2] Daichi Azuma, Taiki Miyanishi, Shuhei Kurita, and Motoaki Kawanabe. Scanqa: 3d question answering for spatial scene understanding. In *CVPR*, 2022. 2, 5, 6, 14, 15, 16, 17, 18

[3] Satanjeev Banerjee and Alon Lavie. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *ACL Workshop*, pages 65–72, 2005. 5, 16

[4] Daigang Cai, Lichen Zhao, Jing Zhang, Lu Sheng, and Dong Xu. 3djcg: A unified framework for joint dense captioning and visual grounding on 3d point clouds. In *CVPR*, pages 16464–16473, 2022. 2, 6, 7

[5] Chun-Fu Richard Chen, Quanfu Fan, and Rameswar Panda. Crossvit: Cross-attention multi-scale vision transformer for image classification. In *ICCV*, pages 357–366, 2021. 2

[6] Dave Zhenyu Chen, Angel X Chang, and Matthias Nießner. Scanrefer: 3d object localization in rgb-d scans using natural language. In *ECCV*, 2020. 2, 5, 6, 7, 8, 14, 15, 16, 17, 18

[7] Dave Zhenyu Chen, Qirui Wu, Matthias Nießner, and Angel X Chang. D 3 net: A unified speaker-listener architecture for 3d dense captioning and visual grounding. In *ECCV*, pages 487–505. Springer, 2022. 7

[8] Sijin Chen, Hongyuan Zhu, Xin Chen, Yinjie Lei, Gang Yu, and Tao Chen. End-to-end 3d dense captioning with vote2cap-detr. In *CVPR*, pages 11124–11133, 2023. 1, 2, 3, 7, 12

[9] Sijin Chen, Xin Chen, Chi Zhang, Mingsheng Li, Gang Yu, Hao Fei, Hongyuan Zhu, Jiayuan Fan, and Tao Chen. Ll3da: Visual interactive instruction tuning for omni-3d understanding reasoning and planning. In *CVPR*, 2024. 1, 2, 3, 5, 6, 7, 12, 14, 16, 17, 18

[10] Sijin Chen, Hongyuan Zhu, Mingsheng Li, Xin Chen, Peng Guo, Yinjie Lei, YU Gang, Taihao Li, and Tao Chen. Vote2cap-detr++: Decoupling localization and describing for end-to-end 3d dense captioning. *IEEE TPAMI*, 2024. 2

[11] Yilun Chen, Shuai Yang, Haifeng Huang, Tai Wang, Ruiyuan Lyu, Runsen Xu, Dahua Lin, and Jiangmiao Pang. Grounded 3d-llm with referent tokens. *arXiv preprint arXiv:2405.10370*, 2024. 1

[12] Zhenyu Chen, Ali Gholami, Matthias Nießner, and Angel X Chang. Scan2cap: Context-aware dense captioning in rgb-d scans. In *CVPR*, pages 3193–3203, 2021. 2, 5, 6, 7

[13] Zhenyu Chen, Ronghang Hu, Xinlei Chen, Matthias Nießner, and Angel X Chang. Unit3d: A unified transformer for 3d dense captioning and visual grounding. In *ICCV*, pages 18109–18119, 2023. 2, 6, 7

[14] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *CVPR*, 2017. 5, 14, 15

[15] Matt Deitke, Ruoshi Liu, Matthew Wallingford, Huong Ngo, Oscar Michel, Aditya Kusupati, Alan Fan, Christian Laforte, Vikram Voleti, Samir Yitzhak Gadre, et al. Objaverse-xl: A universe of 10m+ 3d objects. *NeurIPS*, 36, 2024. 2

[16] Alexandros Delitzas, Maria Parelli, Nikolas Hars, Georgios Vlassis, Sotirios Anagnostidis, Gregor Bachmann, and Thomas Hofmann. Multi-clip: Contrastive vision-language pre-training for question answering tasks in 3d scenes. *arXiv preprint arXiv:2306.02329*, 2023. 6

[17] Xiaoyi Dong, Pan Zhang, Yuhang Zang, Yuhang Cao, Bin Wang, Linke Ouyang, Songyang Zhang, Haodong Duan, Wenwei Zhang, Yining Li, et al. Internlm-xcomposer2-4khd: A pioneering large vision-language model handling resolutions from 336 pixels to 4k hd. *arXiv preprint arXiv:2404.06512*, 2024. 3

[18] Haoqi Fan, Bo Xiong, Karttikeya Mangalam, Yanghao Li, Zhicheng Yan, Jitendra Malik, and Christoph Feichtenhofer. Multiscale vision transformers. In *ICCV*, pages 6824–6835, 2021. 2

[19] Rao Fu, Jingyu Liu, Xilun Chen, Yixin Nie, and Wenhan Xiong. Scene-llm: Extending language model for 3d visual understanding and reasoning. *arXiv preprint arXiv:2403.11401*, 2024. 3

[20] Ritwik Gupta, Shufan Li, Tyler Zhu, Jitendra Malik, Trevor Darrell, and Karttikeya Mangalam. xt: Nested tokenization for larger context in large images. *arXiv preprint arXiv:2403.01915*, 2024. 2

[21] Yining Hong, Haoyu Zhen, Peihao Chen, Shuhong Zheng, Yilun Du, Zhenfang Chen, and Chuang Gan. 3d-llm: Injecting the 3d world into large language models. *NeurIPS*, 2023. 1, 2, 3, 5, 6, 14, 16, 17

[22] Haifeng Huang, Yilun Chen, Zehan Wang, Rongjie Huang, Runsen Xu, Tai Wang, Luping Liu, Xize Cheng, Yang Zhao, Jiangmiao Pang, et al. Chat-scene: Bridging 3d scene and large language models with object identifiers. *NeurIPS*, 2024. 2

[23] Jiangyong Huang, Silong Yong, Xiaojian Ma, Xiongkun Linghu, Puhao Li, Yan Wang, Qing Li, Song-Chun Zhu, Baoxiong Jia, and Siyuan Huang. An embodied generalist agent in 3d world. In *ICML*, 2024. 2

[24] Rui Huang, Songyou Peng, Ayca Takmaz, Federico Tombari, Marc Pollefeys, Shiji Song, Gao Huang, and Francis Engelmann. Segment3d: Learning fine-grained class-agnostic 3d segmentation without manual labels. In *ECCV*, pages 278–295. Springer, 2025. 3

[25] Yang Jiao, Shaoxiang Chen, Zequn Jie, Jingjing Chen, Lin Ma, and Yu-Gang Jiang. More: Multi-order relation mining for dense captioning in 3d scenes. In *ECCV*, pages 528–545. Springer, 2022. 6, 7

[26] Zhao Jin, Munawar Hayat, Yuwei Yang, Yulan Guo, and Yinjie Lei. Context-aware alignment and mutual masking for 3d-language pre-training. In *CVPR*, pages 10984–10994, 2023. 2, 6, 7

[27] Loic Landrieu and Martin Simonovsky. Large-scale point cloud semantic segmentation with superpoint graphs. In *CVPR*, pages 4558–4567, 2018. 4, 14

[28] Hugo Laurençon, Léo Tronchon, Matthieu Cord, and Victor Sanh. What matters when building vision-language models? *Advances in Neural Information Processing Systems*, 37:87874–87907, 2024. 15

[29] Youngwan Lee, Jonghee Kim, Jeffrey Willette, and Sung Ju Hwang. Mpvit: Multi-path vision transformer for dense prediction. In *CVPR*, pages 7287–7296, 2022. 2

[30] Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Yanwei Li, Ziwei Liu, and Chunyuan Li. Llava-onevision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326*, 2024. 3

[31] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *ICML*, pages 19730–19742. PMLR, 2023. 3, 12, 13

[32] Yanwei Li, Yuechen Zhang, Chengyao Wang, Zhisheng Zhong, Yixin Chen, Ruihang Chu, Shaoteng Liu, and Jiaya Jia. Mini-gemini: Mining the potential of multi-modality vision language models. *arXiv preprint arXiv:2403.18814*, 2024. 2

[33] Dingkang Liang, Xin Zhou, Wei Xu, Xingkui Zhu, Zhikang Zou, Xiaoqing Ye, Xiao Tan, and Xiang Bai. Pointmamba: A simple state space model for point cloud analysis. In *NeurIPS*, 2024. 3

[34] Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81, 2004. 5, 16

[35] Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. Llava-next: Improved reasoning, ocr, and world knowledge, 2024. 2, 3

[36] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *NeurIPS*, 2024. 1, 2

[37] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *CVPR*, pages 11976–11986, 2022. 2

[38] Ilya Loshchilov, Frank Hutter, et al. Fixing weight decay regularization in adam. *arXiv preprint arXiv:1711.05101*, 5, 2017. 5

[39] Xiaojian Ma, Silong Yong, Zilong Zheng, Qing Li, Yitao Liang, Song-Chun Zhu, and Siyuan Huang. Sqa3d: Situated question answering in 3d scenes. *arXiv preprint arXiv:2210.07474*, 2022. 2

[40] Aihua Mao, Zhi Yang, Wanxin Chen, Ran Yi, and Yong-jin Liu. Complete 3d relationships extraction modality alignment network for 3d dense captioning. *IEEE TVCG*, 2023. 7

[41] Carsten Moenning and Neil A Dodgson. Fast marching farthest point sampling. Technical report, University of Cambridge, ECCV Laboratory, 2003. 3

[42] Bongki Moon, Hosagrahar V Jagadish, Christos Faloutsos, and Joel H. Saltz. Analysis of the clustering properties of the hilbert space-filling curve. *IEEE TKDE*, 13(1):124–141, 2001. 13

[43] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *ACL*, pages 311–318, 2002. 5, 16

[44] Maria Parelli, Alexandros Delitzas, Nikolas Hars, Georgios Vlassis, Sotirios Anagnostidis, Gregor Bachmann, and Thomas Hofmann. Clip-guided vision-language pre-training for question answering in 3d scenes. In *CVPR*, pages 5607–5612, 2023. 2, 6

[45] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *NeurIPS*, 30, 2017. 12

[46] Zekun Qi, Runpei Dong, Shaochen Zhang, Haoran Geng, Chunrui Han, Zheng Ge, Li Yi, and Kaisheng Ma. Shapellm: Universal 3d object understanding for embodied interaction. In *ECCV*, pages 214–238, 2025. 2

[47] Shi Qiu, Saeed Anwar, and Nick Barnes. Semantic segmentation for real point cloud scenes via bilateral augmentation and adaptive fusion. In *CVPR*, pages 1757–1767, 2021. 1

[48] A. Radford, J. 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 *ICML*, 2021. 2

[49] Chaitanya Ryali, Yuan-Ting Hu, Daniel Bolya, Chen Wei, Haoqi Fan, Po-Yao Huang, Vaibhav Aggarwal, Arkabandhu Chowdhury, Omid Poursaeed, Judy

Hoffman, et al. Hiera: A hierarchical vision transformer without the bells-and-whistles. In *ICML*, pages 29441–29454. PMLR, 2023. 2

[50] Hans Sagan. Hilbert's space-filling curve. In *Space-filling curves*, pages 9–30. Springer, 1994. 2, 3, 4

[51] Baifeng Shi, Ziyang Wu, Maolin Mao, Xin Wang, and Trevor Darrell. When do we not need larger vision models? In *ECCV*, pages 444–462. Springer, 2025. 2

[52] Matthew Tancik, Pratul Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. *NeurIPS*, 33: 7537–7547, 2020. 4

[53] Yuan Tang, Xu Han, Xianzhi Li, Qiao Yu, Yixue Hao, Long Hu, and Min Chen. Minigpt-3d: Efficiently aligning 3d point clouds with large language models using 2d priors. *ACM'MM*, 2024. 2

[54] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023. 1, 17

[55] Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. Cider: Consensus-based image description evaluation. In *CVPR*, pages 4566–4575, 2015. 5, 16

[56] Heng Wang, Chaoyi Zhang, Jianhui Yu, and Weidong Cai. Spatiality-guided transformer for 3d dense captioning on point clouds. *arXiv preprint arXiv:2204.10688*, 2022. 2, 7

[57] Peng-Shuai Wang. Octformer: Octree-based transformers for 3d point clouds. *ACM TOG*, 42(4):1–11, 2023. 3

[58] Qing Wang et al. Divergence-augmented policy optimization. *NeurIPS*, 32, 2019. 15

[59] Zehan Wang, Haifeng Huang, Yang Zhao, Ziang Zhang, and Zhou Zhao. Chat-3d: Data-efficiently tuning large language model for universal dialogue of 3d scenes. *arXiv preprint arXiv:2308.08769*, 2023. 2

[60] Xiaoyang Wu, Li Jiang, Peng-Shuai Wang, Zhijian Liu, Xihui Liu, Yu Qiao, Wanli Ouyang, Tong He, and Hengshuang Zhao. Point transformer v3: Simpler faster stronger. In *CVPR*, pages 4840–4851, 2024. 1, 3, 4

[61] Runsen Xu, Xiaolong Wang, Tai Wang, Yilun Chen, Jiangmiao Pang, and Dahua Lin. Pointllm: Empowering large language models to understand point clouds. *arXiv preprint arXiv:2308.16911*, 2023. 2

[62] Ruyi Xu, Yuan Yao, Zonghao Guo, Junbo Cui, Zanlin Ni, Chunjiang Ge, Tat-Seng Chua, Zhiyuan Liu, Maosong Sun, and Gao Huang. Llava-uhd: an lmm perceiving any aspect ratio and high-resolution images. *arXiv preprint arXiv:2403.11703*, 2024. 3

[63] Jianwei Yang, Chunyuan Li, Pengchuan Zhang, Xiyang Dai, Bin Xiao, Lu Yuan, and Jianfeng Gao. Focal attention for long-range interactions in vision transformers. *NeurIPS*, 34:30008–30022, 2021. 2

[64] Jianing Yang, Xuweiyi Chen, Nikhil Madaan, Madhavan Iyengar, Shengyi Qian, David F. Fouhey, and Joyce Chai. 3d-grand: A million-scale dataset for 3d-llms with better grounding and less hallucination. *arXiv preprint 2406.05132*, 2024. 2

[65] Shuquan Ye, Dongdong Chen, Songfang Han, and Jing Liao. 3d question answering. *arXiv preprint arXiv:2112.08359*, 2021. 2

[66] Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*, 2022. 1, 5, 17

[67] Lichen Zhao, Daigang Cai, Jing Zhang, Lu Sheng, Dong Xu, Rui Zheng, Yinjie Zhao, Lipeng Wang, and Xibo Fan. Toward explainable 3d grounded visual question answering: A new benchmark and strong baseline. *IEEE TCSVT*, 33(6):2935–2949, 2022. 2

[68] Yufeng Zhong, Long Xu, Jiebo Luo, and Lin Ma. Contextual modeling for 3d dense captioning on point clouds. *arXiv preprint arXiv:2210.03925*, 2022. 7

[69] Chenming Zhu, Tai Wang, Wenwei Zhang, Jiangmiao Pang, and Xihui Liu. Llava-3d: A simple yet effective pathway to empowering lmms with 3d-awareness. *arXiv preprint arXiv:2409.18125*, 2024. 2

[70] Ziyu Zhu, Xiaojian Ma, Yixin Chen, Zhidong Deng, Siyuan Huang, and Qing Li. 3d-vista: Pre-trained transformer for 3d vision and text alignment. In *ICCV*, pages 2911–2921, 2023. 6, 7