Uni3D-LLM: Unifying Point Cloud Perception, Generation and Editing with Large Language Models

Dingning Liu[1,2]   Xiaoshui Huang[1]*   Yuenan Hou[1]   Zhihui Wang[2,*]

dutldn@mail.dlut.edu.cn   huangxiaoshui@pjlab.org.cn

Zhenfei Yin[1]   Yongshun Gong[3]   Peng Gao[1]   Wanli Ouyang[1]

[1]Shanghai Artificial Intelligence Laboratory   [2]Dalian University of Technology

[3]Shandong University

Abstract

*In this paper, we introduce **Uni3D-LLM**, a unified framework that leverages a Large Language Model (LLM) to integrate tasks of 3D perception, generation, and editing within point cloud scenes. This framework empowers users to effortlessly generate and modify objects at specified locations within a scene, guided by the versatility of natural language descriptions. **Uni3D-LLM** harnesses the expressive power of natural language to allow for precise command over the generation and editing of 3D objects, thereby significantly enhancing operational flexibility and controllability. By mapping point cloud into the unified representation space, Uni3D-LLM achieves cross-application functionality, enabling the seamless execution of a wide array of tasks, ranging from the accurate instantiation of 3D objects to the diverse requirements of interactive design. Through a comprehensive suite of rigorous experiments, the efficacy of Uni3D-LLM in the comprehension, generation, and editing of point cloud has been validated. Additionally, we have assessed the impact of integrating a point cloud perception module on the generation and editing processes, confirming the substantial potential of our approach for practical applications.*

1. Introduction

In recent years, multimodal large language models (MLLMs) have made significant strides in the fields of natural language processing and computer vision. The powerful language capabilities of MLLMs enable them to handle various textual and visual tasks. Extensive research work [3, 21, 22, 26, 29, 51, 55, 57] has demonstrated the ability of MLLMs to integrate natural language with multiple modalities, including images and point clouds. The advancements in this integration technique have brought tremendous potential to applications, such as precise spatial analysis, augmented interactive experiences in augmented reality, and automated design.

Currently, there have been some approaches to explore the integration of additional functionalities into 3D scenes using MLLMs. These works can be classified into three main categories, *i.e.*, direct embedding [48], 2D-to-3D mapping [18] and pre-alignment [16]. As to direct embedding, PointLLM [48] utilizes pre-trained encoders to directly embed the features of point clouds into the textual space, which facilitates detailed object descriptions and classification tasks. As for the second class of methods, 3D-LLM [18] reconstructs 3D features by generating rendered images from different perspectives. These reconstructed features are aligned with textual data and utilized for tasks such as visual question answering (VQA), planning, and dialogue in indoor scene settings. The third line of methods, as demonstrated in the work of Point-Bind [16], typically align point clouds with other modalities such as images in multimodal datasets. Subsequently, task-specific heads are employed for textual interaction and object generation. These exploratory methods provide valuable insights and practical approaches for integrating additional functionalities into 3D scenes.

Previous approaches, however, still exhibit certain limitations. Firstly, while aligning point clouds with text directly may seem straightforward, accurately recognizing and comprehending the spatial information of the point cloud poses a challenge for LLM. This limitation restricts its capacity to generate precise scene-level interpretations and descriptions. Secondly, although the conversion from 2D to 3D can reconstruct 3D features, it may encounter orthographic occlusion issues when dealing with images captured from different viewpoints. Consequently, this can lead to incomplete 3D scene reconstruction and underutilization of the original point cloud features. Thirdly, although Point-Bind [16] aligns point clouds with other modalities, it is limited to achieving object-level alignment. Aligning point
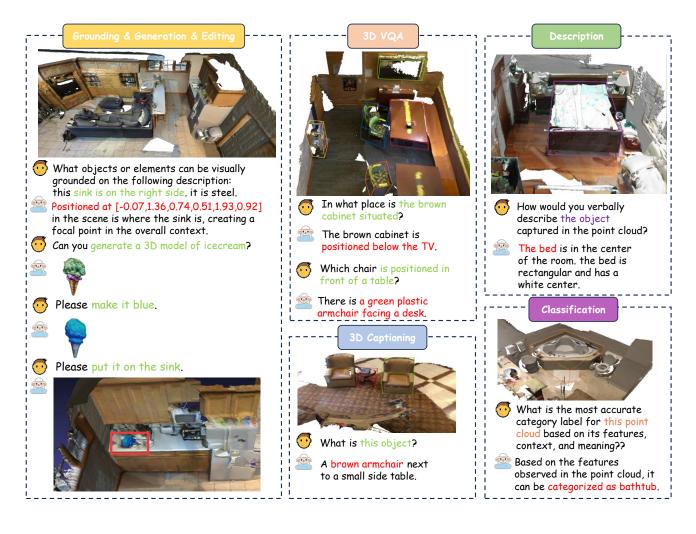
Grounding & Generation & Editing

👤 What objects or elements can be visually grounded on the following description: this sink is on the right side. it is steel.

🧑‍🦳 Positioned at [-0.07,1.36,0.74,0.51,1.93,0.92] in the scene is where the sink is, creating a focal point in the overall context.

👤 Can you generate a 3D model of icecream?

🧑‍🦳

👤 Please make it blue.

🧑‍🦳

👤 Please put it on the sink.

🧑‍🦳

3D VQA

👤 In what place is the brown cabinet situated?

🧑‍🦳 The brown cabinet is positioned below the TV.

👤 Which chair is positioned in front of a table?

🧑‍🦳 There is a green plastic armchair facing a desk.

3D Captioning

👤 What is this object?

🧑‍🦳 A brown armchair next to a small side table.

Description

👤 How would you verbally describe the object captured in the point cloud?

🧑‍🦳 The bed is in the center of the room. the bed is rectangular and has a white center.

Classification

👤 What is the most accurate category label for this point cloud based on its features, context, and meaning??

🧑‍🦳 Based on the features observed in the point cloud, it can be categorized as bathtub.

Figure 1. The example of Uni3D-LLM. Uni3D-LLM can complete various point cloud tasks in a unified framework, including 3D perception, point cloud generation and editing.

clouds at a scene level with other modalities presents a significant challenge. Moreover, Point-Bind treats LLM and the generation model as separate downstream tasks, overlooking the crucial influence of deep semantic understanding inherent in LLM on the controllability of the generative procedure.

Despite the progress achieved by several preliminary attempts utilizing large language models (LLMs) for point cloud tasks , none of them have considered integrating 3D perception, generation and editing into a unified framework. The existing methods are confronted with significant challenges in accomplishing this series of tasks, including fragmented processing workflows, low efficiency, and not utilizing the rich semantic knowledge of LLM to generate more freely. But these challenges involve not only enhancing the perception capabilities of 3D scenes but also bridging the gaps between different modalities and tasks, and effec-

tively integrating the rich linguistic information of Large Language Models (LLMs) into generation models. These shortcomings highlight the importance of unifying the application of LLMs in 3D perception, generation, and editing. A unified framework not only addresses the limitations of traditional 3D perception and generation tasks, but also greatly enhances collaborative work and overall efficiency. With a single training recipe, mutual enhancement across various scenarios becomes possible. In the text-to-3D generation, LLMs can leverage rich semantic information to guide the generation process. Additionally, their advancements in perception provide a solid foundation for more accurate generation and editing. Moreover, this unified framework promotes interaction between different tasks, enabling more efficient iteration and refinement in complex projects.

In this paper, we introduce a novel unified framework called Uni-3DLLM, which aims to enhance the understand-

ing and processing of 3D environments through the utilization of large language models (LLMs). Uni-3DLLM not only focuses on enabling LLMs to delve deeper into the details of 3D environments but also leverages their linguistic capabilities to guide the generation of 3D content. We propose an LLM-guided method for 3D generation and editing within this framework. By integrating scene point clouds and image information and aligning them with text, we accomplish the perception task of point clouds. To facilitate accurate generation, we design an information mapping module that transfers the rich semantic features of LLMs to the generation model. Subsequently, by iteratively updating the original 3D model using modified rendering images from various angles, we obtain a new, edited version. Our proposed model exhibits versatility across different scenarios and enables efficient mutual enhancement during both the generation and editing processes.

Specifically, within the Uni3D-LLM framework, to obtain better scene-level point cloud feature, we replace the original point cloud with a combination and integration of features from each object in the scene. In addition, We utilize various powerful image encoders to extract the top-down view features of the point cloud scene and embedd them into the textual space. In the generation task, in order to make the generation model understand the semantics of LLM, we introduce extra learnable generation tokens at the end of the language descriptions. These tokens are transformed through our mapping block, enabling the Generator to generate understandable signals. The world knowledge contained within the LLM empowers our method to generate and edit objects, even when user descriptions are rough or vague. And during the training phase, to connect perception, generation and editing, we adopt a two-stage approach. Firstly, we integrate the generation mapping module into the LLM and freeze the LLM. Then, we train the LLM to access the information features of point clouds using Parameter Efficient Fine Tuning (PEFT). This strategy effectively prevents catastrophic forgetting, ensuring robust and coherent learning throughout the entire training process.

The contributions of Uni3D-LLM are summarized as follows:

- **A Unified framework to Process Multiple 3D Tasks with LLM.** We make the first attempt to employ LLM to unify a wide array of 3D tasks, including 3D object generation, editing, 3D perception, 3D visual grounding to solve the disconnect between user intent, conveyed through language, and the execution of 3D tasks, providing a more natural and fluid interaction paradigm.
- **Multimodal Signal Alignment.** We pioneered the use of point cloud and additional image assisted by carefully designed, modality-specific projectors, to map heterogeneous text, image, and point cloud signals into a common

token space. The extracted multimodal tokens are fed into the LLM to generate rich semantic features, which are further sent to task-specific architectures to produce the desired outputs.
- **Multi-Task Synergy.** We conducted extensive experiments to verify the synergistic effects of unifying various 3D tasks and to pave the way for building 3D foundation models.

2. Related Work

2.1. Multi-modal Large Language Model

As the impact and accessibility of Large Language Models (LLMs) continue to grow, there is a growing body of research dedicated to extending these pretrained LLMs to handle multimodal comprehension tasks. Some studies have explored training models from scratch using a large amount of image-text pairs [3, 25, 26, 36] and applied them to downstream tasks such as visual question answering(VQA), captioning, and coarse/fine-grained understanding, followed by fine-tuning. Other researchers have connected pretrained visual models with pretrained LLMs, incorporating additional mapping modules like QFormers [26]. This approach leverages the perceptual abilities of pretrained visual models and the reasoning and generalization capabilities of LLMs. In our work, since many studies have demonstrated the strong perceptual capabilities of LLMs in images [15, 29, 56], we utilize two encoder(image and point cloud) to align with texts to assist in acquiring point cloud information and enhance the spatial understanding of LLMs for point clouds.

2.2. 3D Object Generation

3D generation is a task aimed at creating realistic and diverse 3D models from different inputs (such as text, images, sketches, or point clouds). This task is challenging and requires a deep understanding of the shape, structure, texture, and semantics of 3D objects. The main methods currently include parametric methods and non-parametric methods. Parametric methods use predefined templates or primitives to represent 3D shapes, such as voxels, meshes, point clouds, or implicit functions [38, 44]. These methods can generate high-resolution and high-fidelity smooth continuous 3D models. However, these methods also have some limitations, such as high computational cost, fixed topology, or difficulty in handling complex geometry. Non-parametric methods use generative models to learn the distribution of 3D shapes from data, such as Generative Adversarial Networks (GANs) [12], Variational Autoencoders (VAEs) [24], Normalizing Flows [50]. However, these methods also face some challenges, such as mode collapse, decoupling, or evaluation issues or . Recently, with the rise and rapid development of diffusion models in the 2D field,

more and more 3D generation research has begun to adopt diffusion models [27, 30, 31, 35, 40, 43]. But the mainstream issue now is that the generation time is not high enough or the generation quality is not good enough. Liu et al. use 3D Gaussian [23] to reconstruct the entire 3D scene rapidly. In our research, we use dreamgaussian [45], a model that utilizes 3D Gaussian for rapid 3D object modeling, as decoder to complete the generative network. The original dreamgaussian model uses CLIP [40] to embed text, and CLIP is trained on billions of text-image pairs. Therefore, when performing text-to-3D conversion, users can usually only generate the 3D objects they expect by providing relatively short text prompts, and cannot achieve more natural and descriptive words.

2.3. 3D Editing

3D Shape Editing is also a challenging task that requires a deep understanding of shapes. Traditional methods use explicit deformations, while recent years have seen the widespread adoption of CLIP, accelerating attempts to build language-guided editing systems for both images and 3D shapes. As CLIP is trained with pairs of images and texts, most recent efforts have focused on 2D image editing [5, 6]. For 3D, some works introduced a framework for synthesizing 3D shapes and scenes from texts [8]. However, these methods mainly focus on generating 3D shapes rather than editing, which requires language-shape alignment to resolve given edit descriptions. To achieve more intuitive and fine-grained 3D shape editing, some works have explored language-based 3D shape manipulation [1, 34]. These works use powerful vision-language models to generate mesh vertex deformations and colors or build a shape auto-encoder in a latent space and a neural listener to edit shapes according to text instructions. However, these works still have some limitations, such as relying on predefined parts or object localization. In contrast, we propose a MLLM-based 3D editing framework that uses instruct tuning to perform 3D object shape editing with more advanced natural language.

3. Method

In this part, we will present the model design of Uni3D-LLM and the detail of training strategy.

3.1. Model Design

The overall framework of Uni3D-LLM is presented in Fig. 2. In the this section, we will describe the details of the multimodal input alignment layer, the LLM-to-Generator mapping block and the generation-editing module.

3.1.1 Multi-modality alignment.

The multimodal alignment method is shown in the Fig. 3. Hong et al. [18] have established that the integration of 2D image data in the reconstruction of 3D information can be instrumental in augmenting the precision of point cloud recognition tasks. Consequently, in our approach, we have implemented a methodology that aligns point clouds with corresponding images data. To facilitate this cross-modal representation alignment, we have adopted modality-specific projector-based structures.

- **For the alignment of point cloud modality,** inspired by Octavius [10], we adopt a two-step approach to align the point cloud with the space of LLM. Firstly, we follow the detection method of rukhovich et al. [42] to extract the objects from the scene point cloud. Then, we employ a pretrained Point-Bert model [54] as the encoder for extracting point cloud features. Furthermore, the cognitive module LLaMA2 is utilized to facilitate the alignment process. It is important to note that for different tasks, there are distinct approaches. For object-level tasks, the point cloud data is typically mapped into the textual space through a mapping layer. This enables the fusion of visual and textual information for object-level understanding. However, for scene-level tasks, such as grounding, a different strategy is employed. In this case, for each individual scene point cloud, additional position encoding is introduced to preserve its original spatial information. Subsequently, these encoded point clouds are recombined to form a cohesive representation of the entire scene point cloud input. This methodology ensures that the model captures both the local details and the global context of the scene, facilitating accurate scene-level tasks.

- **For the alignment of image,** we directly incorporate the 2D representation extraction method from sphinx. We utilize multiple pre-trained encoders [14, 26, 32, 37, 40] to extract global and local features from the images. For a given scene point cloud, we consider the potential occlusion and limited visibility caused by different rendering pose. To address this, we employ a top-view representation as a representation of image modality. Similarly, we utilize two learnable special tokens to indicate the beginning and end of the inserted image. This allows us to effectively capture the essential visual information of the scene while mitigating the impact of occlusion and incomplete visibility. Once the features for both modalities are extracted, we concatenate align them at the beginning of the textual feature sequence. The image feature, serving as the global guiding feature, is placed at the front. special tokens indicating the start and end of the respective modalities). This concatenation allows for effective integration and alignment of the image and point cloud modalities within the overall textual feature representation.
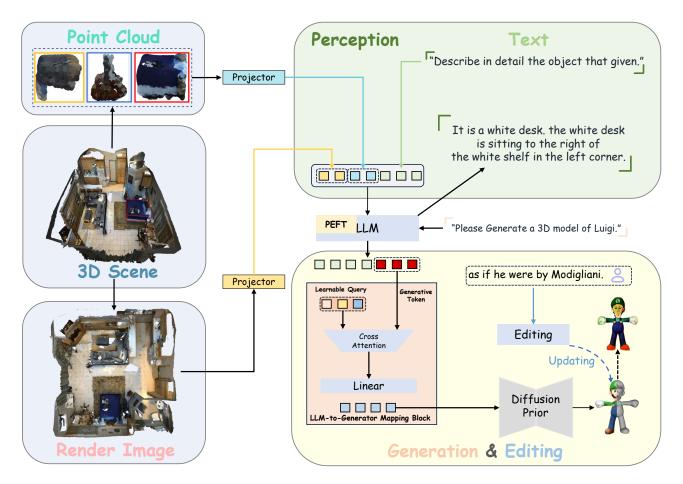
Figure 2. **The framework overview.** We first decompose the point cloud into sub point clouds using 3D detection algorithms and obtain its top-down view rendered image features. Both features added to complete various 3D perception tasks. Our instructions are passed to the generation and editing module through the mapping block.

This alignment facilitates the utilization of the complementary insights offered by both images and text, thus enhancing the granularity and overall accuracy of recognition tasks within point cloud scenes.

3.1.2 LLM-to-Generation mapping block.

To connect the language model output feature and the generation model, we establish a mapping block between them. During the training phase, when we input a generated text, we append 259 learnable generative tokens at the end, representing the desired image to be generated. In the final output feature, we extract the generative tokens and pass them through the mapping block to convert them into the corresponding generation features. The language model, with its rich semantic understanding, serves as a powerful control mechanism for the generation process. It consists of a learnable query, transformer layers and MLPs as the projector to map the text features to the signal that can be known by the generation model. DreamGaussian [45] primarily utilizes

Stable Diffusion [41] and SDS loss [39] to guide the generation process of gaussian splatting[23]. Our objective is to map our features as the text condition into Stable Diffusion. We aim to guide the diffusion process in a way that aligns with our desired outcomes. The overall process can be represented as follows:

$$F_{gen} = \Theta_{\text{map}}\left(q, \Theta_{\text{llm}}(\langle \text{Text}, Token_{gen}\rangle)\right) \in \mathbb{R}^{L \times D} \quad (1)$$

Where F_{gen} represents the text feature fed into the generation model, Θ represents various network parameters, q represents the learnable parameters, L represents the acceptable vector length of the generation model, and D represents the dimension.

3.1.3 Generation-to-editing module.

The overall editing process is shown in the Fig. 4. Once we obtain the generated 3D model, if we intend to modify the corresponding model, we can adopt a methodology similar
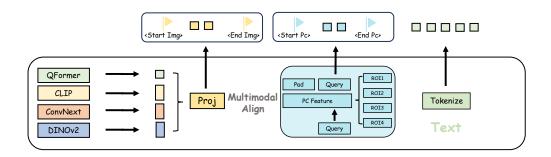
Figure 3. **Multimodal alignment method.** In which the images are re-connected together in the channel dimension after 3 encoder passes, in addition, the image is passing extra QFormer as a global feature concat with the other feature. the ROI is then recoded by Pointbert, with additional padding after cross attention in Query.

to instruct nerf2nerf [17]. We utilize the generated Gaussian splatting data as the initial data and select several rendered images of the 3D model from different poses to make sure the consistency. We leverage instruct-pix2pix [6] to generate the rendered images of the modified object. Subsequently, we gradually update the entire Gaussian splatting using the rendered images, ultimately completing the object editing process.

3.2. Training Strategy

Stage I. we first train our text-to-generation mapping block. In order to generate accurate images to guide 3D generation, the mapping feature F_{Gen} plays a crucial role as a condition during the denoising process. The mapping feature is expected to capture the relevant text features that effectively guide the latent diffusion model (LDM) in generating the desired ground truth image. To achieve this, we leverage the LDM training loss as a guiding mechanism during training. During training, the ground truth image is first encoded into a latent feature f using a pretrained VAE. Subsequently, we add t steps noise (ϵ) to the latent feature to obtain the noisy latent feature f_t. To calculate the conditional LDM loss, we utilize a pretrained U-Net model to predict the noise ϵ_{pred} added on the latent feature, which takes f_t as input. The conditional LDM loss can be expressed as follows:

$$L_{LDM} = \mathbb{E}_{\epsilon \sim \mathcal{N}(0,1),t}(\epsilon, \epsilon_{pred}(f_t, t, F_{gen})) \qquad (2)$$

Classifier-Free Guidance. To enhance the coherence between the text feature and the generator, we adopt the concept of Classifier-Free Guidance(CFG) for generation. We introduce a 10% probability of replacing the mapping feature with zero features and the zero features will be as the negative prompt at the inference stage.
Stage II. Once the mapping layer has been trained, we proceed to train the perception module.Parameter Efficient Fine Tuning(PEFT) [19, 20] plays a crucial role in the training of LLMs and MLLMs. [11, 52, 57] In this study, we employ

Lora on the LLM and during the training phase, the parameters of the whole large language model are frozen, and only the Lora layer is trained. By doing so, we can introduce new multimodal knowledge without losing the existing knowledge of the LLM. **Implementation details.** For the selection of large language models, we opted for sphinx[28], an MLLM that incorporates the image modality on llama2[46]. With this model, we can easily integrate the point cloud modality, thereby enhancing the performance of point cloud tasks by leveraging the joint information of point cloud and images. During training, all visual modality signals are embedded into the textual space as tokens of 259 length. The learning rate is set as 10^{-3} and the FuseAdam is chosen as the optimizer.

4. Experients

4.1. Experimental Setup

To explore the effectiveness of our framework in multimodal learning, we fine-tune Uni3D-LLM in two modality setups: i.) only point cloud modality and ii.) both image and point cloud modalities. We then evaluate the zero-shot and fine-tuned performance using these two fine-tuned models on various 3D downstream tasks.
Datasets. For the training of perception, we utilize an instruction dataset called "Scan2Inst" [10] whitch generated from ScanNet [13] which consists of tasks such as description and classification. In addition, this dataset is also including Scanqa(VQA)[4], and Scan2Cap(Cap)[9] for different task training data. Moreover, we have also integrated additional grounding data into our training dataset. For each scene, we captured 1,513 top-down view rendered images of the scenes, which are used as supplementary information across all tasks.
For the training of our LLM-to-generator mapping block, we initially trained using 2D image-text pairs from the MS-COCO[49] and LN-COCO[53] datasets. Considering the
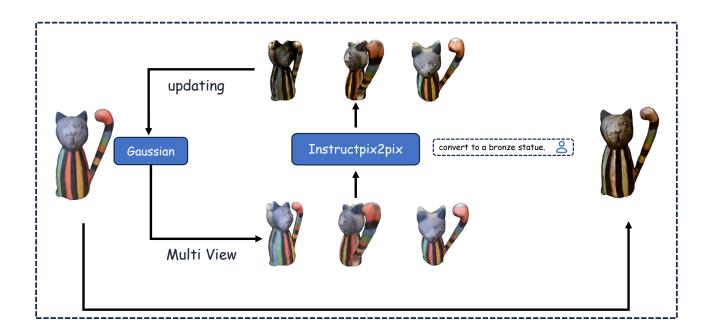
Figure 4. **The pipline of Editing method.** After we obtain a 3D model, we will also store the initialized 3D Gaussian. When the user is given the modification instruction, Generation-to-editing module will render multiple fixed perspective images and send them to Instructpix2pix and then update the 3D Gaussian gradually by updating the specified pose.

Models	VQA(Scanqa)	Classification(Acc@1)		Caption(Scan2Cap)	Grounding(Scannnet)
	BLEU-1	ShapeNet	Scannet	BLEU-1	mAP@0.5
3D-LLM (Flamingo)	30.30	-	-	-	-
Octavius w/ MoE	44.24	24.85	**48.80**	**35.94**	-
Ours(Only 3D)	43.26	20.45	46.90	33.60	Failed
Ours(3D+img)	**44.68**	**30.32**	47.10	34.70	**13.69**

Table 1. **Comparisons on 3D perception tasks.** we compared VQA, classification, caption task with 3D-LLM and Octavius. Among them, Scanqa, Scan2CAP, and Scannet are the results of fine-tuned.

target for 3D object generation, we also integrated pre-training with the Cap3d[33], a subset of 3D object-caption data extracted from Objaverse. This dataset includes 650k point cloud-different view render images-relevant description pairs. Additionally, to enable users to input descriptions as naturally as they would, we created a dataset called "Cap3descript" based on Cap3d. Since GPT cannot process point cloud data and the GPT4-V[36] API is not available, we used the open-source model PointBind-LLM[16] to generate a dataset with 10,000 detailed descriptions based on Cap3D. For each point cloud data, we generated a set of descriptions for eight view images and the original point cloud. Subsequently, we used GPT4 to integrate these nine descriptions into one paragraph, serving as the complete description of the object.

Implementation details. we employ LoRA and task-specific learning when training the perception part. The

rank of each LoRA is set to 32. We utilize the FusedAdam, an Adam optimizer[2], with a total batch size of 16 and the learning rate is 1×10^{-4} for 2 epochs. All experiments are performed using 8 NVIDIA A100 GPUs.

The images are resized to 224×224 and are processed by four different encoders, *i.e.*, CLIP[40], ConvNeXt[47], DINOv2[37], QFormer[26]. For point cloud data, we extract regions of interest (RoI) using FCAF3D[42] and sample 1024 points from each RoI. Each encoder is used for a pre-training weight. For each scene, we follow the settings that selecting N instances with a bounding box confidence higher than the threshold 0.3. In the multimodal fusion step, we employ 16 queries to obtain aligned 3D visual features. Additionally, we pad the output 3D visual features with masks to a size of 256, aligning with length of the image tokens.

Quantitative Results. The conclusions of all experiments

	FID ↓	CIIP Score		CLIP	R-precision		speed
		prompt	descript	R@1	R@5	R@10	
Shape-E(NeRF)	39.4	78.2	70.4	17.4	35.8	44.2	2̃min
Shape-E(STF)	34.7	78.8	71.1	19.6	39.4	47.5	2̃min
Uni3D-LLM	**33.4**	**79.7**	**71.3**	**22.3**	**42.6**	**52.4**	3̃min

Table 2. **Description-to-3D on Cap3descript.** Uni3D-LLM compared with Shap-e after fine-tuning on Cap3d and Cap3descript.

are shown in the Tab. 1, Tab. 2. For perception, we evaluate classification performance on ShapeNet [7], captioning performance on NR3D, vqa performance on ScanQA, and grounding on the test split of ScanNet.

For generation, We conducted two experiments: one is on the test set of Cap3d, and the other is on Cap3descript's test set which is curated test set of 100 samples extracted from the original test set, designed to assess the quality of generation under natural text descriptions. we evaluate the CLIP-Score on different fine-tuned 3D generation model to test the generation quality and time. Due to the time constraints for interactive use with LLM, excessively long generation times are not suitable. Therefore, we only fine-tuned the Shape-E and conducted a comparison.

For editing, it is challenging to have objective evaluation metrics to determine the quality. The assessment primarily relies on personal subjective judgment and feelings. Therefore, we do not evaluate the quality of our editing.

Based on Tab. 1, we can confirm that by introducing image auxiliary information, our model Uni3D-LLM is capable of performing grounding tasks. The primary reason for task failure when relying solely on point cloud information might be the sometimes inaccurate and incomplete capture of all objects in the Region of Interest (ROI), leading to fragmentation in the point cloud features. However, with the addition of complete image assistance, our model effectively handles questions and answers related to global scenes and specific objects. We also observed that in the classification tasks of Scannet and the caption tasks of Scan2Cap, even with the addition of image assistance, there was no significant improvement in the model's performance. We analyze that the main reason might be that these tasks focus on individual properties, and images, as global auxiliary information, are insufficient for significantly aiding these individual-level Q&A tasks. Therefore, to further explore the role of image information in different tasks, we fine-tuned our model on the Cap3d and conducted zero-shot testing on ShapeNet. In these tests, we provided front-view images (view 5) of the objects from the dataset. The results showed that in the task of object-level point cloud classification, providing image information related to the object significantly enhances the overall classification results. This finding emphasizes the importance of combining object-level image information to enhance model performance in

Model	Add perception(Lora)	CLIP Score
Ours	✘	79.5
Ours	✔	**79.7**

Table 3. **Ablation Study on Adding Perception.** In order to verify whether the multimodal sensing module interferes with the generation module, we use the test set and CLIP Score of the original Cap3d data set as the evaluation.

specific scenarios. During the testing of generation performance, we experimented with both short prompts and longer, more natural text on Tab. 2. Considering that the CLIP Score is not highly accurate for long texts and images, we aligned the object generated by natural texts with their original caption. The results indicate that the overall generation still meets the expected outcomes.

Ablation Study. To investigate whether the introduction of the perception module would have any negative impact on the generation module, we conducted an ablation study. The experiment shown on Tab. 3 was divided into two groups: the first group directly aligned the LLM using the LLM-to-Generator Mapping Block, while the second group performed the alignment after introducing the perception module, Lora. The results of the experiment show that the introduction of the perception module does not interfere with the generation results; in fact, it leads to a slight overall improvement in the performance metrics of the generation module.

5. Conclusion and Limitations

In this paper, we introduce Uni3D-LLM, the first attempt to integrate perception, generation, and editing for point clouds. By incorporating powerful image features as spatial assistance, we have overcome the perturbation issues in the original point cloud features that arise when solely inputting point cloud modality. Integrating multiple modalities as auxiliary information for point cloud tasks has proven beneficial for point cloud. However, further enhancing the positioning capability of point clouds remains a challenge for future research.Our generative-editing method also inherits many of the limitations of DreamGaussian and Instruct-Pix2Pix, such as the inability to generate large-scale spatial scenes and perform more freeform directive editing opera-

Data Source	CLIP Score
2D Data	79.1
2D+3D Data	**79.7**

Table 4. **Comparison of different data source effects.** We compared the 3D generation effect of training with normal 2D data and 2D+3D data.

tions. Our generative editing method also faces the same limitations as DreamGauss and InstructPix2Pix, such as the inability to generate large-scale spatial scenes and perform freeform editing. This also needs to be solved in the future.

6. Details of Cap3descript

In this section, We mainly introduce the detail for building Cap3descript. The overall process is shown in 5. We conducted a survey of 10 different MLLMs on the output of Cap3d render images and point cloud. We found that the output from Point-Bind[16] was relatively accurate and detailed. However, due to the issue of obstruction from different angles of render images, we acquired captions of each object from eight different perspectives and the point cloud. Subsequently, we input captions from all eight angles into GPT, allowing it to merge them into a single coherent passage.

In our experiments, we observed that using only 3D rendered images for training the LLM-to-generator mapping block resulted in poor performance. This phenomenon could be attributed to the large amounts of blank space surrounding the 3D rendered images, as Stable Diffusion did not incorporate a significant number of such samples in its training dataset. Therefore, during training, we not only introduced 3D caption-image and description-image data but also integrated 2D text-image data for joint training. As illustrated in Tab. 4, it is evident that the inclusion of 3D-related data significantly enhanced the overall generation effect.
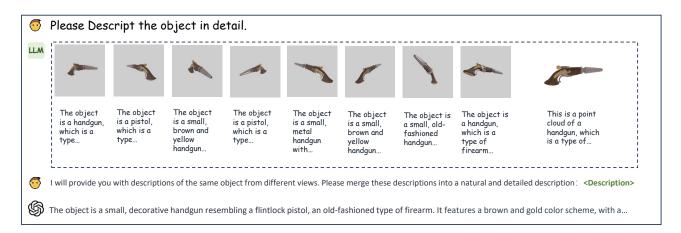
Figure 5. **The method of Cap3descript.** We input 8 perspectives render images of an object into MLLM to obtain different captions, and then use GPT for integration.
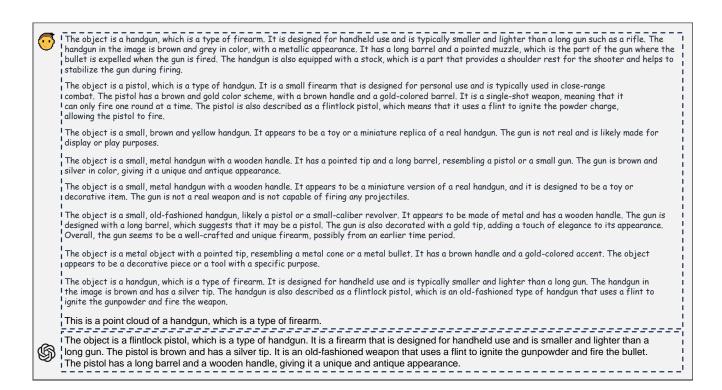

Figure 6. **A complete multi faceted description of an example.** Each angle caption is roughly same, but there may be some differences in details for specific angles.

References

[1] Panos Achlioptas, Ian Huang, Minhyuk Sung, Sergey Tulyakov, and Leonidas Guibas. Changeit3d: Language-assisted 3d shape edits and deformations. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 4

[2] Kingma DP Ba J Adam et al. A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 1412, 2014. 7

[3] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *Advances in Neural Information Processing Systems*, 35:23716–23736, 2022. 1, 3

[4] Daichi Azuma, Taiki Miyanishi, Shuhei Kurita, and Motoaki Kawanabe. Scanqa: 3d question answering for spatial scene understanding. In *proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 19129–19139, 2022. 6

[5] David Bau, Alex Andonian, Audrey Cui, YeonHwan Park, Ali Jahanian, Aude Oliva, and Antonio Torralba. Paint by word. *arXiv preprint arXiv:2103.10951*, 2021. 4

[6] Tim Brooks, Aleksander Holynski, and Alexei A Efros. Instructpix2pix: Learning to follow image editing instructions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18392–18402, 2023. 4, 6

[7] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015. 8

[8] Kevin Chen, Christopher B Choy, Manolis Savva, Angel X Chang, Thomas Funkhouser, and Silvio Savarese. Text2shape: Generating shapes from natural language by learning joint embeddings. In *Computer Vision–ACCV 2018: 14th Asian Conference on Computer Vision, Perth, Australia, December 2–6, 2018, Revised Selected Papers, Part III 14*, pages 100–116. Springer, 2019. 4

[9] Zhenyu Chen, Ali Gholami, Matthias Nießner, and Angel X Chang. Scan2cap: Context-aware dense captioning in rgb-d scans. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3193–3203, 2021. 6

[10] Zeren Chen, Ziqin Wang, Zhen Wang, Huayang Liu, Zhenfei Yin, Si Liu, Lu Sheng, Wanli Ouyang, Yu Qiao, and Jing Shao. Octavius: Mitigating task interference in mllms via moe, 2023. 4, 6

[11] Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E Gonzalez, et al. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality. 2023. *URL https://lmsys. org/blog/2023-03-30-vicuna*, 1(2):3. 6

[12] Antonia Creswell, Tom White, Vincent Dumoulin, Kai Arulkumaran, Biswa Sengupta, and Anil A Bharath. Generative adversarial networks: An overview. *IEEE signal processing magazine*, 35(1):53–65, 2018. 3

[13] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5828–5839, 2017. 6

[14] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 4

[15] Peng Gao, Jiaming Han, Renrui Zhang, Ziyi Lin, Shijie Geng, Aojun Zhou, Wei Zhang, Pan Lu, Conghui He, Xiangyue Yue, et al. Llama-adapter v2: Parameter-efficient visual instruction model. *arXiv preprint arXiv:2304.15010*, 2023. 3

[16] Ziyu Guo, Renrui Zhang, Xiangyang Zhu, Yiwen Tang, Xianzheng Ma, Jiaming Han, Kexin Chen, Peng Gao, Xianzhi Li, Hongsheng Li, et al. Point-bind & point-llm: Aligning point cloud with multi-modality for 3d understanding, generation, and instruction following. *arXiv preprint arXiv:2309.00615*, 2023. 1, 7, 9

[17] Ayaan Haque, Matthew Tancik, Alexei A Efros, Aleksander Holynski, and Angjoo Kanazawa. Instruct-nerf2nerf: Editing 3d scenes with instructions. *arXiv preprint arXiv:2303.12789*, 2023. 6

[18] 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. *arXiv preprint arXiv:2307.12981*, 2023. 1, 4

[19] Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for nlp. In *International Conference on Machine Learning*, pages 2790–2799. PMLR, 2019. 6

[20] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021. 6

[21] Wenlong Huang, Chen Wang, Ruohan Zhang, Yunzhu Li, Jiajun Wu, and Li Fei-Fei. Voxposer: Composable 3d value maps for robotic manipulation with language models. *arXiv preprint arXiv:2307.05973*, 2023. 1

[22] Xiaoshui Huang, Sheng Li, Wentao Qu, Tong He, Yifan Zuo, and Wanli Ouyang. Frozen clip model is efficient point cloud backbone. *arXiv preprint arXiv:2212.04098*, 2022. 1

[23] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics (ToG)*, 42(4):1–14, 2023. 4, 5

[24] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 3

[25] Bo Li, Yuanhan Zhang, Liangyu Chen, Jinghao Wang, Jingkang Yang, and Ziwei Liu. Otter: A multi-modal model with in-context instruction tuning. *arXiv preprint arXiv:2305.03726*, 2023. 3

[26] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with

frozen image encoders and large language models. *arXiv preprint arXiv:2301.12597*, 2023. 1, 3, 4, 7

[27] Chen-Hsuan Lin, Jun Gao, Luming Tang, Towaki Takikawa, Xiaohui Zeng, Xun Huang, Karsten Kreis, Sanja Fidler, Ming-Yu Liu, and Tsung-Yi Lin. Magic3d: High-resolution text-to-3d content creation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 300–309, 2023. 4

[28] Ziyi Lin, Chris Liu, Renrui Zhang, Peng Gao, Longtian Qiu, Han Xiao, Han Qiu, Chen Lin, Wenqi Shao, Keqin Chen, et al. Sphinx: The joint mixing of weights, tasks, and visual embeddings for multi-modal large language models. *arXiv preprint arXiv:2311.07575*, 2023. 6

[29] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *arXiv preprint arXiv:2304.08485*, 2023. 1, 3

[30] Ruoshi Liu, Rundi Wu, Basile Van Hoorick, Pavel Tokmakov, Sergey Zakharov, and Carl Vondrick. Zero-1-to-3: Zero-shot one image to 3d object. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9298–9309, 2023. 4

[31] Yuan Liu, Cheng Lin, Zijiao Zeng, Xiaoxiao Long, Lingjie Liu, Taku Komura, and Wenping Wang. Syncdreamer: Generating multiview-consistent images from a single-view image. *arXiv preprint arXiv:2309.03453*, 2023. 4

[32] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 4

[33] Tiange Luo, Chris Rockwell, Honglak Lee, and Justin Johnson. Scalable 3d captioning with pretrained models. *arXiv preprint arXiv:2306.07279*, 2023. 7

[34] Oscar Michel, Roi Bar-On, Richard Liu, Sagie Benaim, and Rana Hanocka. Text2mesh: Text-driven neural stylization for meshes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13492–13502, 2022. 4

[35] Alex Nichol, Heewoo Jun, Prafulla Dhariwal, Pamela Mishkin, and Mark Chen. Point-e: A system for generating 3d point clouds from complex prompts. *arXiv preprint arXiv:2212.08751*, 2022. 4

[36] OpenAI. Gpt-4 technical report, 2023. 3, 7

[37] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023. 4, 7

[38] Pablo Palafox, Aljaž Božič, Justus Thies, Matthias Nießner, and Angela Dai. Npms: Neural parametric models for 3d deformable shapes. in 2021 ieee. In *CVF International Conference on Computer Vision (ICCV)*, pages 12675–12685, 2021. 3

[39] Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv preprint arXiv:2209.14988*, 2022. 5

[40] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry,

Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 4, 7

[41] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2021. 5

[42] Danila Rukhovich, Anna Vorontsova, and Anton Konushin. Fcaf3d: Fully convolutional anchor-free 3d object detection. In *European Conference on Computer Vision*, pages 477–493. Springer, 2022. 4, 7

[43] Yichun Shi, Peng Wang, Jianglong Ye, Mai Long, Kejie Li, and Xiao Yang. Mvdream: Multi-view diffusion for 3d generation. *arXiv preprint arXiv:2308.16512*, 2023. 4

[44] Dmitriy Smirnov, Matthew Fisher, Vladimir G Kim, Richard Zhang, and Justin Solomon. Deep parametric shape predictions using distance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 561–570, 2020. 3

[45] Jiaxiang Tang, Jiawei Ren, Hang Zhou, Ziwei Liu, and Gang Zeng. Dreamgaussian: Generative gaussian splatting for efficient 3d content creation. *arXiv preprint arXiv:2309.16653*, 2023. 4, 5

[46] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023. 6

[47] Sanghyun Woo, Shoubhik Debnath, Ronghang Hu, Xinlei Chen, Zhuang Liu, In So Kweon, and Saining Xie. Convnext v2: Co-designing and scaling convnets with masked autoencoders. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16133–16142, 2023. 7

[48] 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. 1

[49] Tao Xu, Pengchuan Zhang, Qiuyuan Huang, Han Zhang, Zhe Gan, Xiaolei Huang, and Xiaodong He. Attngan: Fine-grained text to image generation with attentional generative adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1316–1324, 2018. 6

[50] Guandao Yang, Xun Huang, Zekun Hao, Ming-Yu Liu, Serge Belongie, and Bharath Hariharan. Pointflow: 3d point cloud generation with continuous normalizing flows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4541–4550, 2019. 3

[51] Qinghao Ye, Haiyang Xu, Guohai Xu, Jiabo Ye, Ming Yan, Yiyang Zhou, Junyang Wang, Anwen Hu, Pengcheng Shi, Yaya Shi, et al. mplug-owl: Modularization empowers large language models with multimodality. *arXiv preprint arXiv:2304.14178*, 2023. 1

[52] Zhenfei Yin, Jiong Wang, Jianjian Cao, Zhelun Shi, Dingning Liu, Mukai Li, Lu Sheng, Lei Bai, Xiaoshui Huang, Zhiyong Wang, et al. Lamm: Language-assisted multi-

modal instruction-tuning dataset, framework, and benchmark. *arXiv e-prints*, pages arXiv–2306, 2023. 6

[53] Jiahui Yu, Yuanzhong Xu, Jing Yu Koh, Thang Luong, Gunjan Baid, Zirui Wang, Vijay Vasudevan, Alexander Ku, Yinfei Yang, Burcu Karagol Ayan, et al. Scaling autoregressive models for content-rich text-to-image generation. *arXiv preprint arXiv:2206.10789*, 2(3):5, 2022. 6

[54] Xumin Yu, Lulu Tang, Yongming Rao, Tiejun Huang, Jie Zhou, and Jiwen Lu. Point-bert: Pre-training 3d point cloud transformers with masked point modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19313–19322, 2022. 4

[55] Renrui Zhang, Ziyu Guo, Wei Zhang, Kunchang Li, Xupeng Miao, Bin Cui, Yu Qiao, Peng Gao, and Hongsheng Li. Pointclip: Point cloud understanding by clip. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8552–8562, 2022. 1

[56] Renrui Zhang, Jiaming Han, Aojun Zhou, Xiangfei Hu, Shilin Yan, Pan Lu, Hongsheng Li, Peng Gao, and Yu Qiao. Llama-adapter: Efficient fine-tuning of language models with zero-init attention. *arXiv preprint arXiv:2303.16199*, 2023. 3

[57] Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *arXiv preprint arXiv:2304.10592*, 2023. 1, 6