LLaNA: Large Language and NeRF Assistant

Andrea Amaduzzi
andrea.amaduzzi4@unibo.it

Pierluigi Zama Ramirez
pierluigi.zama@unibo.it

Giuseppe Lisanti
giuseppe.lisanti@unibo.it

Samuele Salti
samuele.salti@unibo.it

Luigi Di Stefano
luigi.distefano@unibo.it

CVLAB, University of Bologna

https://andreamaduzzi.github.io/llana/

Abstract

Multimodal Large Language Models (MLLMs) have demonstrated an excellent understanding of images and 3D data. However, both modalities have shortcomings in holistically capturing the appearance and geometry of objects. Meanwhile, Neural Radiance Fields (NeRFs), which encode information within the weights of a simple Multi-Layer Perceptron (MLP), have emerged as an increasingly widespread modality that simultaneously encodes the geometry and photorealistic appearance of objects. This paper investigates the feasibility and effectiveness of ingesting NeRF into MLLM. We create LLaNA, the first general-purpose NeRF-language assistant capable of performing new tasks such as NeRF captioning and Q&A. Notably, our method directly processes the weights of the NeRF's MLP to extract information about the represented objects without the need to render images or materialize 3D data structures. Moreover, we build a dataset of NeRFs with text annotations for various NeRF-language tasks with no human intervention. Based on this dataset, we develop a benchmark to evaluate the NeRF understanding capability of our method. Results show that processing NeRF weights performs favourably against extracting 2D or 3D representations from NeRFs.

1 Introduction

Large Language Models (LLMs) [70, 1] have revolutionized the field of Natural Language Processing, demonstrating incredible text comprehension and generation capabilities. These results have fostered the development of Multimodal LLMs (MLLMs) [14, 82, 44, 11, 10], which can ingest various modalities such as images, videos, and audio, to generate text describing and reasoning about the content of such modalities. Recently, MLLMs have also been extended to 3D data [58, 78], primarily represented through colored point clouds, yielding remarkable results even in this scenario.

Beyond images and 3D data, another paradigm is emerging to represent objects and scenes: Neural Radiance Fields (NeRFs) [50]. NeRFs are coordinate-based neural networks, typically Multi-Layer Perceptrons (MLPs), designed to capture both the geometry and the photorealistic appearance of an object by learning a continuous radiance field at each 3D spatial location. After training, a NeRF model can be queried to render realistic images or to reconstruct the 3D surface of the encoded object. Therefore, capturing an object as a NeRF provides an interesting alternative to create a digital twin

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

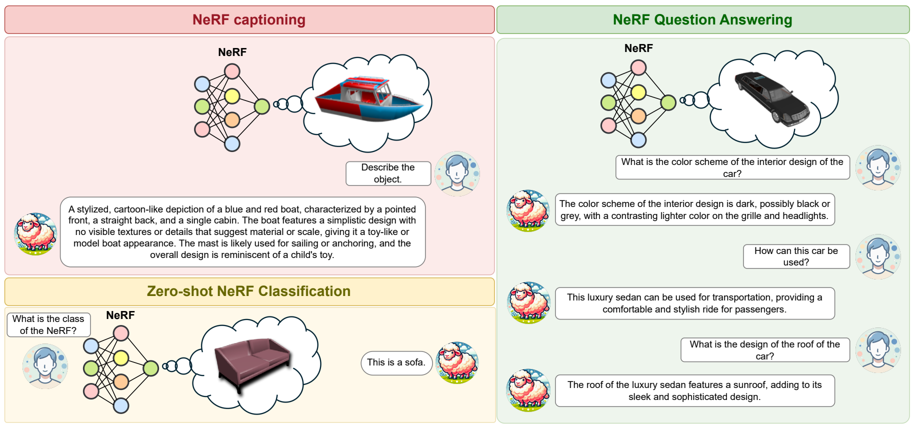

NeRF captioning
NeRF
Describe the object.
A stylized, cartoon-like depiction of a blue and red boat, characterized by a pointed front, a straight back, and a single cabin. The boat features a simplistic design with no visible textures or details that suggest material or scale, giving it a toy-like or model boat appearance. The mast is likely used for sailing or anchoring, and the overall design is reminiscent of a child's toy.
Zero-shot NeRF Classification
What is the class of the NeRF?
NeRF
This is a sofa.
NeRF Question Answering
NeRF
What is the color scheme of the interior design of the car?
The color scheme of the interior design is dark, possibly black or grey, with a contrasting lighter color on the grille and headlights.
How can this car be used?
This luxury sedan can be used for transportation, providing a comfortable and stylish ride for passengers.
What is the design of the roof of the car?
The roof of the luxury sedan features a sunroof, adding to its sleek and sophisticated design.

Figure 1: **LLaNA.** The first Multimodal Large Language Model that understands and reasons on an input NeRF. Our framework directly processes the NeRF weights and performs tasks such as captioning, Q&A, and zero-shot classification of NeRFs.

with respect to standard representations such as multi-view images or point clouds. For instance, thanks to its continuous formulation, from a single NeRF, one can generate an infinite number of photorealistic images at any resolution while storing only the weights of an MLP instead of the entire image set. See Appendix A.4 for more on the memory advantages of using NeRFs. Due to their advantages, NeRFs are effectively becoming a new modality stored and communicated independently, with datasets of NeRFs being made publicly available [25, 61] and companies providing digital twins of objects represented as NeRFs (e.g., `https://lumalabs.ai/`).

The increasing adoption of NeRFs and their appealing characteristics prompted us to the following research question: is it possible to build an MLLM able to ingest directly NeRFs? Inspired by recent studies on meta-networks that can process neural fields [81, 42], we answer this question in the positive by showing that it is possible to process the weights of a given NeRF with a meta-network encoder that projects the NeRF weights into the embedding space of a pre-trained LLM such as LLaMA 2 [70]. By doing so, we create the first MLLM for NeRFs, dubbed Large Language and NeRF Assistant (LLaNA), which can solve NeRF-language tasks such as NeRF captioning, Q&A and zero-shot NeRF classification (see Fig. 1).

We also introduce a new NeRF–language dataset, that we will make publicly available, to train LLaNA and test the capabilities of our assistant. To collect this dataset, we designed an automated annotation framework that leverages MLLMs to produce text annotations for NeRFs trained on Shapenet [8]. Using this dataset alongside an additional split containing manually curated textual descriptions [2], we establish a benchmark for NeRF textual assistants.

Since a straightforward way to create an assistant for NeRFs would be to render images or extract 3D point clouds out of it and provide them as input to existing MLLMs specifically designed to handle such modalities, we thoroughly compare LLaNA against these baselines on the proposed benchmark. We show how the resolution of the extracted 3D geometry or images, and for images also the vantage point used for rendering, negatively impact the quality of the MLLM's output. Important details might be lost by rendering from the wrong angle, or the extracted geometry might not be detailed enough. Vice versa, by operating directly on the MLP weights, we extract all the information they hold about the object without any other design decision. Our approach turns out to be the most effective way to create a NeRF assistant as it consistently outperforms MLLMs processing images or 3D geometries extracted by querying NeRFs. Our contributions can be summarized as follows:

• LLaNA, the first MLLM capable of performing tasks such as captioning and Q&A on NeRFs.

• We show that it is possible to build such an assistant by directly processing the NeRFs weights with a meta-encoder, which is faster and captures more information than rendering images or extracting 3D data.

• We automatically create a NeRF-language benchmark based on ShapeNet, and we thoroughly evaluate LLaNA on it, showing that it performs better than applying popular MLLMs on discrete representations obtained from NeRFs.

2 Related work

Multimodal Large Language Models (MLLMs). Significant advancements have been made by Large Language Models (LLMs) in language understanding, reasoning, and generalization capabilities [62, 1, 54, 70, 75, 60]. These models have been extended into Multimodal Large Language Models (MLLMs), which broaden their reasoning abilities by including other modalities like images [14, 82, 17, 19], audio [26], and videos [47, 10]. MLLMs generally align target features with textual ones and then integrate them into LLMs for various text inference tasks. Some MLLMs are trained entirely from scratch [27, 56], others utilize pretrained LLMs [37, 4, 44, 38, 11]. 3D MLLMs focus on understanding the 3D world typically represented as colored point clouds [58, 24, 86, 20, 78] or multi-view images [23]. Some of these models are trained using 2D images [24, 86, 23] while others directly align textual phrases with points [20, 78, 58].

Neural radiance fields. NeRF [50] have been applied in several visual tasks such as novel view synthesis [48], generative media [57], and robotics [79]. The base formulation employs MLPs to convert spatial coordinates into colors and densities. Recent advancements substitute or enhance MLPs with explicit data structures [9, 68, 16, 52] for faster training and inference.

Neural radiance fields and language. The interaction between NeRF and language has been recently investigated for several practical applications. Many works address the problem of generating geometrically consistent views of objects or scenes described by textual prompts [66, 49, 31, 65, 40, 36, 57]. Other approaches focus on editing the scene represented by a NeRF from text, e.g., by changing the appearance and shape of objects [73, 28, 67, 74, 69, 21, 80, 87], or by inserting/removing objects in the scene [3, 51]. Some techniques investigate new types of radiance fields that predict language features for each spatial location alongside density and color [32, 34]. By distilling knowledge from vision-language models into these models, the neural fields can be queried by textual prompts. LERF [32] extends the original radiance field formulation, considering functions which model density, color and language features at each spatial coordinate. Such *language fields* are parametrized by a neural network. Unlike all previous methods, Ballerini et al. [5] is the first to utilize the weights of a NeRF's MLP as an input modality. They aim to learn a mapping between the NeRF and CLIP [59] embedding spaces to perform tasks such as NeRF retrieval from textual or image queries. Differently, our goal is to develop an MLLM capable of reasoning about NeRFs.

Deep learning on neural networks. Several studies have explored using meta-networks, i.e. neural networks that analyze other neural networks. Initially, researchers concentrated on predicting network characteristics, such as accuracy and hyperparameters, by processing their weights [71, 64, 33, 30, 45]. Several recent works focus on processing networks implicitly representing data (Implicit Neural Representations or Neural Fields). These methods perform tasks such as classifying or segmenting the data by processing solely the weights of the input neural networks. Among these works, Functa [15] trains a shared network on the entire dataset and then learns a compact embedding for each sample for downstream tasks. Later works concentrate on processing networks representing individual data samples, e.g., a specific object. By leveraging a novel encoder architecture for MLP weights, `inr2vec` [12] extracts compact embeddings from INRs of 3D shapes, which are employed as inputs for downstream tasks. `nf2vec` [61] extends `inr2vec` to ingest the NeRF's network weights to classify, segment, or retrieve similar NeRFs. Cardace et al. [7] develop a strategy to process neural fields represented by a hybrid tri-plane structure. Other approaches [53, 84, 83, 85] develop equivariant architectures to handle MLPs by exploiting weight space symmetries [22] as an inductive bias. Also, Graph Neural Networks have been investigated to compute a network representation [35, 42]. Since we aim to process NeRFs directly from the network weights, we employ `nf2vec` as our meta-encoder due to its efficient and scalable architecture.

3 Methodology

This section describes the proposed Large Language and NeRF Assistant (LLaNA). We provide an overview of NeRFs and the meta-encoder that maps NeRF weights into a global embedding. Then, we present the overall LLaNA framework and discuss our training protocol.

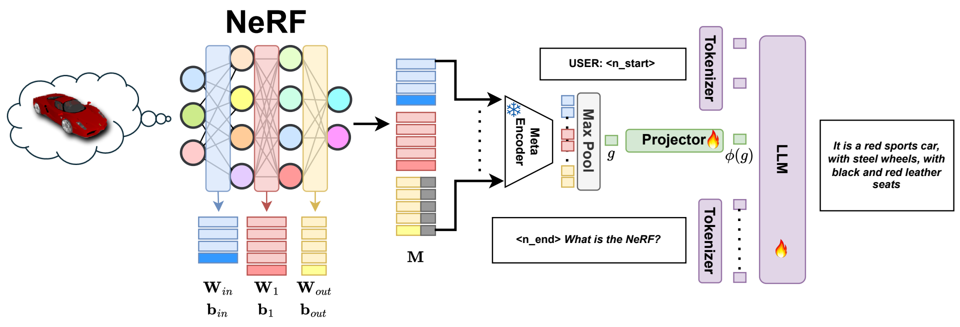

NeRF
$\mathbf{W}_{in}$ $\mathbf{b}_{in}$
$\mathbf{W}_1$ $\mathbf{b}_1$
$\mathbf{W}_{out}$ $\mathbf{b}_{out}$
M
Meta Encoder
Max Pool
g
Projector
$\phi(g)$
USER: <n_start>
<n_end> What is the NeRF?
Tokenizer
Tokenizer
LLM
It is a red sports car, with steel wheels, with black and red leather seats

Figure 2: **Framework overview. Example of NeRF captioning.**

Neural Radiance Fields (NeRF) Neural Radiance Field (NeRF) [50] is a framework that employs coordinate-based neural networks, typically MultiLayer Perceptrons (MLP) and is trained on a collection of images of an object or scene taken from various vantage points. The main application of NeRFs is the task of novel views synthesis, i.e., photorealistic rendering of images from viewpoints unseen at training time. In its base formulation, the MLP is a function of continuous 3D coordinates $\mathbf{p} = (x, y, z) \in \mathbb{R}^3$, that yields four-dimensional outputs, $RGB\sigma \in [0, 1]^4$. This output encodes the RGB color and the volume density σ of each 3D location in the scene. The volume density σ can be interpreted as the differential probability of a ray terminating at point $\mathbf{p}$. After training, a NeRF can render images from any desired viewpoint at arbitrary resolution by querying it for the values of RGB and σ at several points along the ray corresponding to each pixel and applying the volumetric rendering equation [50].

In this work, we realize NeRFs as MLPs composed of L hidden layers, an input layer, and an output layer. An example of MLP with 1 input, 1 output, and 1 hidden layer is shown in Fig. 2 (left). A layer is parameterized by a weight matrix plus a bias vector. More in detail, the hidden layers in our architecture have the same number of input and output neurons, H, thus having squared weight matrices $\mathbf{W}_l \in \mathbb{R}^{H \times H}$ for $l = 1, \ldots, L$ and H-dimensional biases $\mathbf{b}_l \in \mathbb{R}^H$. As input $\mathbf{p}$ goes through a 24-frequency encoding [50], the first layer has $\mathbf{W}_{in} \in \mathbb{R}^{144 \times H}$ and $\mathbf{b}_{in} \in \mathbb{R}^H$. The final one has $\mathbf{W}_{out} \in \mathbb{R}^{H \times 4}$ and $\mathbf{b}_{out} \in \mathbb{R}^4$. Refer to Appendix A for more details on NeRFs.

Meta-encoder In this work, we explore how a NeRF assistant can be realized by processing the NeRF weights directly. We expect the NeRF weights to contain comprehensive information about the represented object, such as its geometry and appearance. Thus, an encoder processing them might extract all the necessary information for downstream language tasks such as captioning and Q&A.

Inspired by the recent development of meta-networks capable of processing neural fields [42, 81], we employ as our meta-encoder architecture `nf2vec` [81]. It takes as input the weights of a NeRF and yields a global embedding that distills the content of the input NeRF. In particular, the weight matrices and biases of the input NeRF are stacked along the row dimension to form a matrix $\mathbf{M} \in \mathbb{R}^{S \times H}$, where the number of rows S depends on the number of hidden layers L, the number of units per hidden layer H, and the dimension of the input, which is a 144-dimensional array obtained by frequency encoding of the 3D coordinates. Before stacking, we pad the output layer weights $\mathbf{W}_{out}$ and biases $\mathbf{b}_{out}$ with zeros to obtain H columns (see Fig. 2, center).

The meta-encoder is parametrized as an MLP with batch normalization layers [29] and ReLU non-linearities. To scale gracefully with the input MLP dimensions, the encoder processes each row of $\mathbf{M}$ independently, extracting a total of S tokens, each of length G, from an input NeRF. They are then max-pooled to obtain a global representation $g \in \mathbb{R}^G$ of the NeRF, with $G = 1024$ in our experiments. The encoder is pre-trained using the self-training protocol of `nf2vec` [81], i.e., jointly with a decoder architecture that, given as input the NeRF global embedding, reconstructs the same images as the input NeRF from arbitrary viewpoints. More details in Appendix B.

Large language and NeRF assistant Inspired by recent approaches that created effective Multimodal Large Language Models, we build LLaNA by leveraging on a pre-trained LLM with a transformer backbone [72], in our experiments LLaMA 2 [70], and injecting the NeRF modality into its embedding input space, as proposed for images and 3D data [44, 78] (see Fig. 2, right). Thanks to the self-attention mechanism, the transformer can understand the contextual relationships between text and NeRF tokens, enabling it to generate responses based on both text and NeRF inputs.

We employ a trainable linear projection layer, ϕ, to project the embedding of the input NeRF computed by the meta-encoder into the LLaMA 2 embedding space. The projection layer has weights $\mathbf{W}_{proj} \in \mathbb{R}^{G \times T}$, where T is the word embedding dimension of the employed LLaMA model. This embedding is encapsulated between two special tokens, whose embeddings are learned end-to-end while training, namely <n_start> and <n_end>.

Then, given an input sequence of mixed NeRF and word tokens, (<n_start>, $\phi(g)$,<n_end>, $w_1, w_2, ..., w_k$), where k is the number of word tokens, the large language model returns a sequence of predicted word tokens $(\hat{w}_{k+1}, \hat{w}_{k+2}, \ldots, \hat{w}_{eos})$.

Training protocol Our framework is trained on the ShapeNeRF–Text dataset, described in detail in Sec. 4. This dataset is organized into a set of prompts from the user and expected ground-truth answers that are used to optimize the original auto-regressive objective of the LLM. For the meta-encoder, we employ the `nf2vec` encoder pre-trained on ShapeNet released by the authors [81], and we keep it frozen during training. We follow the two-stage training protocol delineated in Liu et al. [44]:

Stage1: projector training. In the first stage, we train the projector network ϕ to align the NeRF and the word embedding spaces while keeping the LLM weights fixed. We train on an instruction dataset of brief descriptions to learn the projection layer efficiently. We also train the embeddings of the special tokens used to encapsulate the NeRF one. We optimize the projector weights and the embeddings for 3 epochs with a learning rate of 0.002 and batch size of 64.

Stage2: instruction tuning. During the second stage, we train on complex instructions to help the model understand and reason about NeRF data. In this phase, we optimize both the projector and the LLM for 3 epochs on the detailed descriptions, single-round and multi-round Q&A conversations available in our dataset. For this phase, we employ a learning rate of 0.0002 and a batch size of 16.

Our model is implemented in PyTorch and trained on 4 NVIDIA A100 with 64GB of VRAM each. Completing both stages requires $\sim$1 day of training.

4 Benchmark

4.1 ShapeNeRF–Text dataset

To train and validate our NeRF assistant, we automatically created a dataset of conversations about NeRFs, the ShapeNeRF–Text dataset.

It features paired NeRFs and language annotations for ShapeNet objects [8], in particular for all the 40K NeRFs available in the `nf2vec` dataset [61]. We followed the structure defined in PointLLM [78] to create the textual annotations. More in detail, for each object, we generated a *brief description*, a *detailed description*, 3 *single-round Q&As*, and one *multi-round Q&A*. The brief descriptions are concise captions of the object, taking into account its global structure and appearance. The detailed descriptions are longer sentences that describe all the details of the object. The single-round Q&As consist of a question about the object and the corresponding ground-truth answer. Finally, the multi-round Q&As are longer conversations formed by 3 questions and the relative answers. The automatic data annotation pipeline is inspired by Cap3D [46] and is shown in Fig. 3. First, multiple views of each ShapeNet object have been rendered from different perspectives. Then, each view has been provided as input to LLaVA (LLaVA2-13b) [44] to get a detailed description of the object from that point of view. Afterward, starting from the captions generated by LLaVA, LLaMA 3 (LLaMA3-8B-chat) was used to generate the final ground-truth text data (brief and

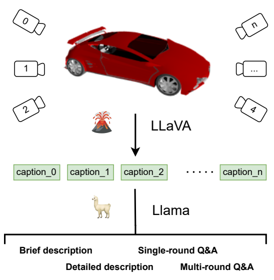


Figure 3: **Automatic annotation pipeline.** Given a 3D model, N views are rendered and processed by a VLM (LLaVA) to generate view-specific captions. These are aggregated by an LLM (LLaMA) for final descriptions and Q&A.

detailed descriptions, single and multi-round Q&As). Both the frozen LLMs employed to create our benchmark (LLaVA2-13b, LLaMA3-8b-chat) are equipped with safeguards.

When building the ground-truth data, to ensure diversity in the language annotations, each brief and detailed description has been associated with a question randomly sampled from 30 instructions for each kind of description. Such instructions, together with the carefully engineered request prompts for LLaVA and LLaMA, are reported in Appendix C.1.

ShapeNeRF–Text provides 30939, 3846 and 3859 objects for the train, validation and test sets, respectively. Overall, the dataset features 13 object classes, and the train, validation and test splits are obtained by randomly sampling objects within each class, i.e., holding out a fixed percentage of objects per class (80%, 10%, and 10% for the sets, respectively). Appendix C.2 provides more dataset statistics. As quantitatively proven in Appendix C.3 and Appendix D.1, many of the questions belonging to the Q&A set require a holistic 3D understanding of the object, to be answered correctly.

4.2 Language tasks and metrics

We evaluate NeRF assistants on three different language tasks, given an input NeRF: brief captioning, detailed captioning, and single-round Q&A. We evaluate all tasks on the objects from the ShapeNeRF–Text test set. For brief captioning, we additionally evaluate the methods on the GPT2Shape Human Shape Text (HST) dataset [2], a subset of ShapeNet for which human-curated brief descriptions are publicly available. To generate the dialogues for HST, we randomly pair each of its captions with one of the 30 instructions requesting a brief description, used in ShapeNeRF–Text and reported in Appendix C.1. We employ standard language similarity metrics to evaluate these methods. We compute the cosine similarity between the global embeddings of the generated and ground-truth sentences provided by the pre-trained encoders Sentence-BERT [63] and SimCSE [18]. These metrics based on learned networks are the most effective at measuring the quality of the generated output. We also include standard handcrafted metrics based on n-gram statistics, like BLEU-1 [55], ROUGE-L [43], and METEOR [6].

5 Experiment results

5.1 Foundation models as baselines

As our method is the first to investigate language tasks on NeRF, there are no baselines in the literature. However, given a NeRF, a straightforward way to create an assistant for it could be to render an image and use an MLLM capable of ingesting images. Alternatively, we could extract the 3D shape from the NeRF and use one of the recent 3D MLLMs. Hence, in a first set of experiments, we use MLLMs as off-the-shelf foundation models, trained on hundreds of thousands of shapes or millions of images, without performing any fine-tuning on the training set of ShapeNeRF–Text, and consider such pipelines as natural baselines. Specifically, we use LLaVA (v1.6) [44] and BLIP-2 [39] for images, as well as PointLLM [78] and GPT4Point [58] for colored point clouds. Since NeRFs can render arbitrary viewpoints after training, we also include the evaluation of LLaVA [44] in a multi-view scenario. More in detail, we render images from N viewpoints randomly sampled between the set of camera poses used to train each NeRF; then, we concatenate tokens from these N images and fed them into LLaVA alongside text instructions. We set N=3 because the model cannot process a higher number of images correctly. In addition, we test 3D-LLM [24] to compare its performance to LLaNA. We employ the official code and pre-trained models released by the respective authors for such evaluations [1]. We note that the only official GPT4Point weights available at submission time were those obtained from fine-tuning OPT-2.7B on Cap3D [46]. In Tabs. 1 to 5, we present the performance of all methods under the more realistic scenario where NeRFs are treated as the only input data to the assistant. Hence, images and point clouds can only be extracted from NeRFs. Details on the extraction procedure are provided in Appendix A.3. As for 3D-LLM, we extract colored 3D meshes from the NeRFs of ShapeNeRF–Text and process such data with the official 3D-LLM code to render images from multiple views and compute both the 2D and 3D features required by the model

[1]LLaVA: `https://github.com/haotian-liu/LLaVA` BLIP-2: `https://github.com/salesforce/LAVIS/tree/main/projects/blip2` PointLLM: `https://github.com/OpenRobotLab/PointLLM` GPT4Point: `https://github.com/Pointcept/GPT4Point` 3D-LLM: `https://github.com/UMass-Foundation-Model/3D-LLM`

Table 1: **NeRF brief captioning on ShapeNeRF-Text. Frozen baselines.**
Best results are in **bold**, runner-up is underlined. (FV: front-view, BV: back-view, MV: multi-view)

Model	Modality	S-BERT	SimCSE	BLEU-1	ROUGE-L	METEOR
LLaVA-vicuna-13b	Image (FV)	61.00	61.16	14.30	20.00	23.31
LLaVA-vicuna-13b	Image (BV)	54.35	56.09	21.94	21.67	22.09
LLaVA-vicuna-13b	Image (MV)	59.64	61.01	**22.84**	22.17	23.08
LLaVA-vicuna-7b	Image (FV)	59.85	62.35	22.67	23.24	23.35
LLaVA-vicuna-7b	Image (BV)	55.68	58.46	21.97	22.46	22.50
BLIP-2 FlanT5-xxl	Image (FV)	56.13	58.21	5.46	18.69	9.67
BLIP-2 FlanT5-xxl	Image (BV)	52.48	54.05	5.67	18.20	9.50
PointLLM-7b	Point cloud	49.59	48.84	16.74	17.92	14.56
GPT4Point-Opt-2.7b	Point cloud	41.85	40.22	11.76	16.54	11.63
3D-LLM	Mesh + MV	59.46	56.42	12.69	21.49	14.32
LLaNA-7b	NeRF	**68.63**	**70.54**	20.64	**28.33**	**31.76**

Table 2: **NeRF brief captioning on the HST dataset. Frozen baselines.**
Best results are in **bold**, runner-up is underlined. (FV: front-view, BV: back-view, MV: multi-view)

Model	Modality	S-BERT	SimCSE	BLEU-1	ROUGE-L	METEOR
LLaVA-vicuna-13b	Image (FV)	55.62	55.56	6.56	11.81	14.52
LLaVA-vicuna-13b	Image (BV)	50.00	50.79	9.39	12.76	14.46
LLaVA-vicuna-13b	Image (MV)	54.25	55.56	9.78	14.13	14.99
LLaVA-vicuna-7b	Image (FV)	54.31	56.28	10.08	14.71	14.53
LLaVA-vicuna-7b	Image (BV)	51.75	52.29	8.13	13.96	14.18
BLIP-2 FlanT5-xxl	Image (FV)	57.11	59.43	8.21	18.02	12.14
BLIP-2 FlanT5-xxl	Image (BV)	54.11	56.37	9.09	17.38	11.79
PointLLM-7b	Point cloud	43.40	44.50	8.53	11.64	9.97
GPT4Point-Opt-2.7B	Point cloud	43.15	42.22	12.02	18.73	13.69
3D-LLM	Mesh + MV	56.07	52.13	**15.94**	**20.71**	15.22
LLaNA-7b	NeRF	**59.20**	**61.66**	9.47	14.94	**17.06**

Table 3: **NeRF detailed captioning on ShapeNeRF-Text. Frozen baselines.**
Best results are in **bold**, runner-up is underlined. (FV: front-view, BV: back-view, MV: multi-view)

Model	Modality	S-BERT	SimCSE	BLEU-1	ROUGE-L	METEOR
LLaVA-vicuna-13b	Image (FV)	59.08	58.87	23.63	23.55	22.55
LLaVA-vicuna-13b	Image (BV)	50.09	50.33	13.77	21.36	13.18
LLaVA-vicuna-13b	Image (MV)	60.21	59.51	15.07	32.16	14.64
LLaVA-vicuna-7b	Image (FV)	57.55	57.68	14.99	22.82	14.36
LLaVA-vicuna-7b	Image (BV)	53.11	54.46	14.73	22.47	14.05
BLIP-2 FlanT5-xxl	Image (FV)	41.27	40.69	0.18	7.83	2.60
BLIP-2 FlanT5-xxl	Image (BV)	38.49	37.89	0.19	7.72	2.58
PointLLM-7b	Point cloud	59.02	58.30	10.28	19.26	10.55
GPT4Point-Opt-2.7b	Point cloud	42.44	38.33	3.72	9.21	5.13
3D-LLM	Mesh + MV	60.00	53.91	1.58	14.40	5.28
LLaNA-7b	NeRF	**77.43**	**79.81**	**41.32**	**36.18**	**32.39**

at inference time. Moreover, in Appendix E, we report the results dealing with the images used to train the NeRF or the original 3D point cloud from ShapeNet, which confirms the methods' ranking. When rendering an image, a non-obvious design decision for the pipeline is from which vantage point to render it. ShapeNet artificially simplifies this task since all objects have been canonically aligned to a common reference frame, but this may not be the case in a general setting. To show the vantage point's effect on the assistant's results, we report results processing a frontal or back view.

5.2 NeRF captioning

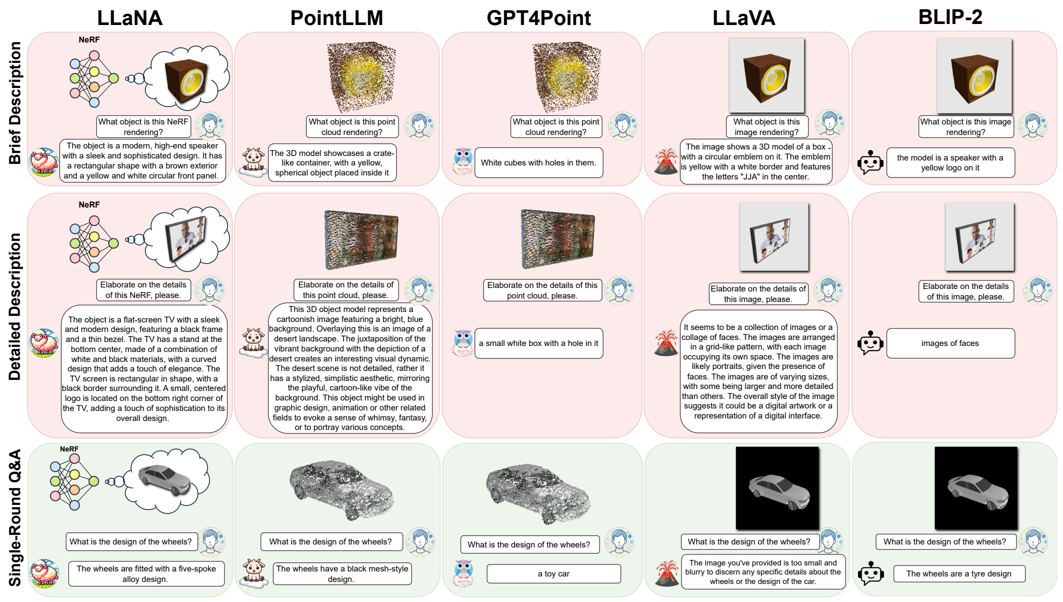
We test the assistants' ability to describe the NeRF content in the captioning experiments. We prompt them with the NeRF, or the image/cloud extracted from it, followed by the question which has been paired with its ground-truth description, as detailed in Section 4.2, e.g. *"What's the content of this NeRF/image/cloud?"*. We then collect the answers generated by the models and compare them with the ground-truth description according to the selected metrics.

Brief description. We report results for the brief description tasks on ShapeNeRF–Text and the HST dataset in Tab. 1 and Tab. 2, respectively. Comparing LLaNA with the baselines described in Sec. 5.1, we appreciate how LLaNA achieves the best performance in most metrics, often by large margins against runner-ups. For instance, for the Sentence-BERT similarity on the ShapeNeRF–Text dataset, LLaNA achieves 68.63, 7.63 points more than LLaVA-vicuna13b, even if LLaNA uses a smaller LLM. Results on the HST dataset, which provides ground-truth descriptions validated by humans, are generally lower for all methods. Yet, LLaNA provides again the best performance according to most metrics. The difference in the quality of the brief description provided by LLaNA compared to the baselines is showcased by the qualitative result reported in the first row of Fig. 4, where the description provided by LLaNA is the most accurate.

A clear trend in both tables and qualitative results is that image-based models tend to perform better than models processing point clouds. This is likely due to the larger amount of data used during training of the modality encoder, i.e. millions of images versus hundreds of thousands of shapes, which enhances their generalization ability, as well as the capability of images to capture more details than point clouds at the input resolutions required by image-based MLLMs versus 3D MLLMs. Nonetheless, our method, which operates on NeRFs, benefits from a holistic view of the object and provides the most accurate descriptions. Remarkably, in LLaNA, all the necessary information for this

Table 4: **NeRF single-round Q&A on ShapeNeRF-Text. Frozen baselines.** Best results are in **bold**, runner-up is <u>underlined</u>. (FV: front-view, BV: back-view, MV: multi-view)

Model	Modality	S-BERT	SimCSE	BLEU-1	ROUGE-L	METEOR
LLaVA-vicuna-13b	Image (FV)	71.61	70.98	20.19	30.42	32.53
LLaVA-vicuna-13b	Image (BV)	68.25	69.06	20.03	29.84	32.27
LLaVA-vicuna-13b	Image (MV)	71.84	71.16	20.04	30.20	33.46
LLaVA-vicuna-7b	Image (FV)	71.79	71.96	25.79	34.04	34.86
LLaVA-vicuna-7b	Image (BV)	70.88	70.93	25.17	33.30	34.22
BLIP-2 FlanT5-xxl	Image (FV)	45.20	47.92	11.50	20.16	13.49
BLIP-2 FlanT5-xxl	Image (BV)	45.06	47.66	11.50	19.98	13.44
PointLLM-7b	Point cloud	<u>74.70</u>	<u>74.40</u>	<u>36.81</u>	<u>44.41</u>	<u>39.76</u>
GPT4Point-Opt-2.7b	Point cloud	27.62	31.41	6.26	9.38	5.41
3D-LLM	Mesh + MV	69.62	67.55	32.19	40.95	35.83
LLaNA-7b	NeRF	**81.03**	**81.56**	**46.16**	**53.17**	**50.15**



What object is this NeRF rendering?
The object is a modern, high-end speaker with a sleek and sophisticated design. It has a rectangular shape with a brown exterior and a yellow and white circular front panel.
What object is this point cloud rendering?
The 3D model showcases a crate-like container, with a yellow, spherical object placed inside it
What object is this point cloud rendering?
White cubes with holes in them.
What object is this image rendering?
The image shows a 3D model of a box - with a circular emblem on it. The emblem is yellow with a white border and features the letters "JJA" in the center.
What object is this image rendering?
the model is a speaker with a yellow logo on it

Elaborate on the details of this NeRF, please.
The object is a flat-screen TV with a sleek and modern design, featuring a black frame and a thin bezel. The TV has a stand at the bottom center, made of a combination of white and black materials, with a curved design that adds a touch of elegance. The TV screen is rectangular in shape, with a black border surrounding it. A small, centered logo is located on the bottom right corner of the TV, adding a touch of sophistication to its overall design.
Elaborate on the details of this point cloud, please.
This 3D object model represents a cartoonish image featuring a bright, blue background. Overlaying this is an image of a desert landscape. The juxtaposition of the vibrant background with the depiction of a desert creates an interesting visual dynamic. The desert scene is not detailed, rather it has a stylized, simplistic aesthetic, mirroring the playful, cartoon-like vibe of the background. This object might be used in graphic design, animation or other related fields to evoke a sense of whimsy, fantasy, or to portray various concepts.
Elaborate on the details of this point cloud, please.
a small white box with a hole in it
Elaborate on the details of this image, please.
It seems to be a collection of images or a collage of faces. The images are arranged in a grid-like pattern, with each image occupying its own space. The images are likely portraits, given the presence of faces. The images are of varying sizes, with some being larger and more detailed than others. The overall style of the image suggests it could be a digital artwork or a representation of a digital interface.
Elaborate on the details of this image, please.
images of faces

What is the design of the wheels?
The wheels are fitted with a five-spoke alloy design.
What is the design of the wheels?
The wheels have a black mesh-style design.
What is the design of the wheels?
a toy car
What is the design of the wheels?
The image you've provided is too small and blurry to discern any specific details about the wheels or the design of the car.
What is the design of the wheels?
The wheels are a tyre design

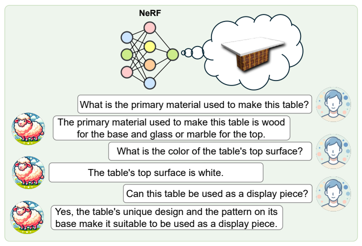
Figure 4: **Qualitative results of NeRF captioning and Q&A.** Results on ShapeNeRF–Text. From top to bottom: brief and detailed descriptions, single-round Q&A

language task can be extracted from a single global embedding obtained by directly processing the NeRF weights. It is also worth pointing out that, while LLaNA directly processes weights and thus is independent by design from spatial resolution, the baselines face a computational overhead growing with the desired resolution due to the necessity of extracting spatial data from NeRF (Appendix A.3). Results show that 3D-LLM performs better than the point-based models and comparably to image-based models. Comparing the results of image-based MLLMs when processing front versus back views, we can see that the vantage point has a non-negligible effect on the performance of such baselines, with SentenceBERT and SimCSE metrics diminishing by about 4 points in all baselines. In a dataset without canonical poses for objects, this would be a relevant limitation that processing NeRF weights seamlessly sidesteps. Finally, we observe that the multi-view setup of LLaVA provides similar performance to the single-view counterpart. In Appendix G, additional qualitative examples are provided.

Detailed description. We evaluate the performance for the detailed description tasks on the proposed ShapeNeRF–Text, reporting the results in Tab. 3. For this task, the point-based model PointLLM [78] performs similarly to the image-based one, LLaVA [44]. However, we appreciate that LLaNA achieves the best performance in all metrics by large margins. For instance, for the Sentence-BERT metric, LLaNA achieves 77.43, notably 18.35 points more than LLaVA-vicuna-13b single-view and 17.22 for the LLaVA-vicuna-13b multi-view setup. These large improvements indicate that, while individual or aggregated images may be sufficient for brief descriptions, they may lack all the details needed to provide a comprehensive description. Moreover, the dependency

Table 5: **Zero-Shot NeRF Classification. Frozen baselines.**
Best results are in **bold**, runner-up is underlined. (FV: front-view, BV: back-view, MV: multi-view)

Model	Modality	Accuracy (%)
LLaVA-vicuna-13b	Image (FV)	66.13
LLaVA-vicuna-13b	Image (BV)	63.90
LLaVA-vicuna-13b	Image (MV)	**73.45**
LLaVA-vicuna-7b	Image (FV)	60.25
LLaVA-vicuna-7b	Image (BV)	57.00
BLIP-2 FlanT5-xxl	Image (FV)	63.67
BLIP-2 FlanT5-xxl	Image (BV)	61.47
PointLLM-7b	Point cloud	50.14
GPT4Point-Opt-2.7b	Point cloud	41.93
3D-LLM	Mesh + MV	60.55
LLaNA-7b	NeRF	67.14



What is the primary material used to make this table?

The primary material used to make this table is wood for the base and glass or marble for the top.

What is the color of the table's top surface?

The table's top surface is white.

Can this table be used as a display piece?

Yes, the table's unique design and the pattern on its base make it suitable to be used as a display piece.

Figure 5: **NeRF multi-round Q&A example.**

Table 6: **NeRF brief captioning on ShapeNeRF-Text. Trained baselines.**
Best results are in **bold**, runner-up is underlined. (FV: front-view)

Model	Modality	S-BERT	SimCSE	BLEU-1	ROUGE-L	METEOR
LLaVA-vicuna-13b	Image (FV)	42.86	43.22	15.56	13.74	15.27
PointLLM-7b	Point cloud	55.48	57.28	**21.67**	25.84	24.54
GPT4Point-Opt-2.7b	Point cloud	37.96	39.00	21.33	22.29	24.88
LLaNA-7b	NeRF	**68.63**	**70.54**	20.64	**28.33**	**31.76**

Table 7: **NeRF brief captioning on the HST dataset. Trained baselines.**
Best results are in **bold**, runner-up is underlined. (FV: front-view)

Model	Modality	S-BERT	SimCSE	BLEU-1	ROUGE-L	METEOR
LLaVA-vicuna-13b	Image (FV)	33.79	42.66	**10.28**	**13.22**	12.19
PointLLM-7b	Point cloud	44.65	44.68	8.91	12.33	12.64
GPT4Point-Opt-2.7B	Point cloud	30.50	31.08	8.12	12.35	11.62
LLaNA-7b	NeRF	**55.62**	**55.56**	6.56	11.81	**14.52**

of the output quality on the selected vantage points remains strong. Contrarily, the NeRF weights contain detailed and complete information about the object, which is fundamental for more granular description tasks, with the additional advantage of not requiring tuning such hyperparameters. The ability of NeRF to capture holistic information about the object is also shown in the second row of Fig. 4, where only the direct processing of NeRF weights lets LLaNA understand that the object is a TV. PointLLM and LLaVA provide detailed but wrong descriptions, likely because of the need to extract the intermediate discrete representation as a point cloud or an image, losing information. Indeed, in both cases, it is hard even for a human observer to provide the right description from the intermediate modalities shown in the figure. More qualitative examples of this task are shown in Appendix G.

5.3 NeRF single-round Q&A

In the single-round Q&A experiment, we test the ability of the assistants to provide precise answers to specific questions about the object instead of open-ended general descriptions. We prompt the models with the NeRF, or the image/cloud extracted from it, followed by one of the questions in the single-round Q&A annotations associated with the NeRF. We then collect the answer generated by the model and compare it against the ground-truth answer with the selected metrics. Results are reported in Tab. 4. Interestingly, PointLLM [78] performs better than LLaVA [44] in this task, likely because it has been specifically trained to answer detailed questions about objects represented as point clouds. Nevertheless, similarly to the detailed description results, LLaNA is the top-performing method across all metrics, again by large margins. This result suggests that the meta-encoder and the projector can extract fine-grained information from the NeRF, even if they are processing directly NeRF weights. Remarkably, the amount of information they can extract lets LLaNA answer more precisely than when images or point clouds are extracted from the NeRF. Indeed, as shown in the third row of Fig. 4 which reports a qualitative example, the only assistant able to answer correctly to a precise question about the appearance of the tyres of the car is LLaNA. In Appendix G, additional qualitative examples of this task are provided. Finally, another qualitative result confirming the ability of LLaNA to provide high-quality answers to specific questions, in this case in a multi-round Q&A experiment, is reported in Fig. 5.

5.4 Zero-shot NeRF classification

Finally, we compare assistants on the task of zero-shot classification. We query the models with the sentence *"What is the class of the NeRF/image/cloud? Choose among these: <Shapenet_classes>"*

Table 8: **NeRF detailed captioning on ShapeNeRF-Text. Trained baselines.** Best results are in **bold**, runner-up is underlined. (FV: front-view)

Model	Modality	S-BERT	SimCSE	BLEU-1	ROUGE-L	METEOR
LLaVA-vicuna-13b	Image (FV)	44.69	42.31	10.08	23.46	12.70
PointLLM-7b	Point cloud	67.30	59.56	15.39	21.42	11.37
GPT4Point-Opt-2.7b	Point cloud	41.33	40.52	14.48	19.15	13.80
LLaNA-7b	NeRF	**77.43**	**79.81**	**41.32**	**36.18**	**32.39**

Table 9: **NeRF single-round Q&A on ShapeNeRF-Text. Trained baselines.** Best results are in **bold**, runner-up is underlined. (FV: front-view)

Model	Modality	S-BERT	SimCSE	BLEU-1	ROUGE-L	METEOR
LLaVA-vicuna-13b	Image (FV)	56.29	62.36	26.87	29.55	30.49
PointLLM-7b	Point cloud	79.24	80.38	46.00	52.60	42.36
GPT4Point-Opt-2.7b	Point cloud	22.22	28.66	8.76	13.46	14.19
LLaNA-7b	NeRF	**81.03**	**81.56**	**46.16**	**53.17**	**50.15**

where <*Shapenet_classes*> are the 10 ShapeNet classes available in our dataset. We consider the answer correct only if the ground truth class appears in the response. We report results in Tab. 5 on the ShapeNeRF–Text dataset. Using multiple views boosts the zero-shot classification performance of LLaVA, which turns out to be the best model for this task, followed by LLaNA.

5.5 Training baselines on ShapeNeRF–Text

Tabs. 6 to 9 report results on language tasks of several baselines trained on ShapeNeRF–Text, while Tab. 13 of the appendix, shows zero-shot NeRF classification performance of such models. We employed those baselines on ShapeNeRF–Text, for which we were able to run the official training code. Accordingly, we followed their protocol, which, for all of them, keeps the modality-specific encoder frozen and trains an adaptor and the LLM in two steps. We notice that the trained baselines exhibit different behaviors to their frozen counterparts, with LLaVA performing significantly worse and PointLLM showing clear improvements. As for GPT4Point, we observe greater variability across metrics; however, overall, it shows no significant benefit from training on ShapeNeRF–Text. LLaNA yields the best performance compared to all baselines, either frozen or trained on ShapeNeRF–Text. Finally, Appendix F shows the generalization performance on Objaverse of LLaNA and the trained baselines.

6 Limitations and future directions

Despite the promising results of our framework, it is the first study in this direction and several limitations are yet to be addressed. First, the pre-trained `nf2vec` encoder, having been trained exclusively on synthetic data from ShapeNet, may not generalize well to real-world objects. To address this, future work should create a NeRF–Text dataset including a more diverse set of objects, like the ones provided by Objaverse [13] and OmniObject3D [76]. Another limitation is that `nf2vec` currently processes only MLPs, restricting our model to MLP-only NeRFs. However, with the rapid advancements in meta-networks, it may become very soon possible to extend LLaNA to more complex NeRF architectures, such as InstantNGP [52]. For instance, the approach by Lim et al. [42] suggests the feasibility of processing various input architectures, although it is currently limited to small networks. Finally, our framework has been tested solely on object-centric NeRFs. Expanding its application to NeRFs representing entire scenes would be a compelling direction for future research.

7 Concluding remarks

This paper addressed the novel task of creating a language assistant for NeRF. We have tackled this problem by leveraging recent advances in MLLMs and meta-networks processing neural fields. We have shown that it is feasible and effective to directly process the weights of a NeRF to project it into the input embedding space of an LLM. We have built and made publicly available a dataset of textual annotations of NeRFs and have shown that our approach compares favourably with respect to several MLLMs used as baselines for the novel tasks of brief and detailed captioning, question answering, and zero-shot classification of NeRFs.

Acknowledgements

We acknowledge the CINECA award under the ISCRA initiative, for the availability of high-performance computing resources and support.

References

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[2] Andrea Amaduzzi, Giuseppe Lisanti, Samuele Salti, and Luigi Di Stefano. Looking at words and points with attention: a benchmark for text-to-shape coherence. In *2023 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, pages 2860–2869. IEEE Computer Society, 2023.

[3] Haotian Bai, Yuanhuiyi Lyu, Lutao Jiang, Sijia Li, Haonan Lu, Xiaodong Lin, and Lin Wang. Componerf: Text-guided multi-object compositional nerf with editable 3d scene layout. *arXiv preprint arXiv:2303.13843*, 2023.

[4] Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen-vl: A frontier large vision-language model with versatile abilities. *arXiv preprint arXiv:2308.12966*, 2023.

[5] Francesco Ballerini, Pierluigi Zama Ramirez, Roberto Mirabella, Samuele Salti, and Luigi Di Stefano. Connecting nerfs, images, and text. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2024.

[6] Satanjeev Banerjee and Alon Lavie. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pages 65–72, 2005.

[7] Adriano Cardace, Pierluigi Zama Ramirez, Francesco Ballerini, Allan Zhou, Samuele Salti, and Luigi di Stefano. Neural processing of tri-plane hybrid neural fields. In *The Twelfth International Conference on Learning Representations*, 2024.

[8] 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.

[9] Anpei Chen, Zexiang Xu, Andreas Geiger, Jingyi Yu, and Hao Su. Tensorf: Tensorial radiance fields. In *European Conference on Computer Vision (ECCV)*, 2022.

[10] Guo Chen, Yin-Dong Zheng, Jiahao Wang, Jilan Xu, Yifei Huang, Junting Pan, Yi Wang, Yali Wang, Yu Qiao, Tong Lu, et al. Videollm: Modeling video sequence with large language models. *arXiv preprint arXiv:2305.13292*, 2023.

[11] Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale N Fung, and Steven Hoi. Instructblip: Towards general-purpose vision-language models with instruction tuning. *Advances in Neural Information Processing Systems*, 36, 2024.

[12] Luca De Luigi, Adriano Cardace, Riccardo Spezialetti, Pierluigi Zama Ramirez, Samuele Salti, and Luigi Di Stefano. Deep learning on implicit neural representations of shapes. In *International Conference on Learning Representations (ICLR)*, 2023.

[13] Matt Deitke, Dustin Schwenk, Jordi Salvador, Luca Weihs, Oscar Michel, Eli VanderBilt, Ludwig Schmidt, Kiana Ehsani, Aniruddha Kembhavi, and Ali Farhadi. Objaverse: A universe of annotated 3d objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13142–13153, 2023.

[14] Danny Driess, Fei Xia, Mehdi SM Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, et al. Palm-e: An embodied multimodal language model. In *International Conference on Machine Learning*, pages 8469–8488. PMLR, 2023.

[15] Emilien Dupont, Hyunjik Kim, SM Ali Eslami, Danilo Jimenez Rezende, and Dan Rosenbaum. From data to functa: Your data point is a function and you can treat it like one. In *International Conference on Machine Learning*, pages 5694–5725. PMLR, 2022.

[16] Sara Fridovich-Keil, Alex Yu, Matthew Tancik, Qinhong Chen, Benjamin Recht, and Angjoo Kanazawa. Plenoxels: Radiance fields without neural networks. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

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

[18] Tianyu Gao, Xingcheng Yao, and Danqi Chen. Simcse: Simple contrastive learning of sentence embeddings. In *2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021*, pages 6894–6910. Association for Computational Linguistics (ACL), 2021.

[19] Rohit Girdhar, Alaaeldin El-Nouby, Zhuang Liu, Mannat Singh, Kalyan Vasudev Alwala, Armand Joulin, and Ishan Misra. Imagebind: One embedding space to bind them all. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15180–15190, 2023.

[20] 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.

[21] Ayaan Haque, Matthew Tancik, Alexei A. Efros, Aleksander Holynski, and Angjoo Kanazawa. Instruct-nerf2nerf: Editing 3d scenes with instructions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 19740–19750, 2023.

[22] Robert Hecht-Nielsen. On the algebraic structure of feedforward network weight spaces. In *Advanced Neural Computers*, pages 129–135. Elsevier, 1990.

[23] Yining Hong, Chunru Lin, Yilun Du, Zhenfang Chen, Joshua B. Tenenbaum, and Chuang Gan. 3d concept learning and reasoning from multi-view images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9202–9212, 2023.

[24] 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. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[25] Benran Hu, Junkai Huang, Yichen Liu, Yu-Wing Tai, and Chi-Keung Tang. Nerf-rpn: A general framework for object detection in nerfs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23528–23538, 2023.

[26] Rongjie Huang, Mingze Li, Dongchao Yang, Jiatong Shi, Xuankai Chang, Zhenhui Ye, Yuning Wu, Zhiqing Hong, Jiawei Huang, Jinglin Liu, et al. Audiogpt: Understanding and generating speech, music, sound, and talking head. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 23802–23804, 2024.

[27] Shaohan Huang, Li Dong, Wenhui Wang, Yaru Hao, Saksham Singhal, Shuming Ma, Tengchao Lv, Lei Cui, Owais Khan Mohammed, Barun Patra, et al. Language is not all you need: Aligning perception with language models. *Advances in Neural Information Processing Systems*, 36, 2024.

[28] Sungwon Hwang, Junha Hyung, Daejin Kim, Min-Jung Kim, and Jaegul Choo. Faceclipnerf: Text-driven 3d face manipulation using deformable neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 3469–3479, 2023.

[29] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. pmlr, 2015.

[30] Florian Jaeckle and M Pawan Kumar. Generating adversarial examples with graph neural networks. In *Uncertainty in Artificial Intelligence*, pages 1556–1564. PMLR, 2021.

[31] Kyungmin Jo, Gyumin Shim, Sanghun Jung, Soyoung Yang, and Jaegul Choo. Cg-nerf: Conditional generative neural radiance fields for 3d-aware image synthesis. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 724–733, 2023.

[32] Justin* Kerr, Chung Min* Kim, Ken Goldberg, Angjoo Kanazawa, and Matthew Tancik. Lerf: Language embedded radiance fields. In *International Conference on Computer Vision (ICCV)*, 2023.

[33] Boris Knyazev, Michal Drozdzal, Graham W. Taylor, and Adriana Romero. Parameter prediction for unseen deep architectures. In *Advances in Neural Information Processing Systems*, 2021.

[34] Sosuke Kobayashi, Eiichi Matsumoto, and Vincent Sitzmann. Decomposing nerf for editing via feature field distillation. In *Advances in Neural Information Processing Systems*, pages 23311–23330. Curran Associates, Inc., 2022.

[35] Miltiadis Kofinas, Boris Knyazev, Yan Zhang, Yunlu Chen, Gertjan J Burghouts, Efstratios Gavves, Cees GM Snoek, and David W Zhang. Graph neural networks for learning equivariant representations of neural networks. In *The Twelfth International Conference on Learning Representations*, 2024.

[36] Han-Hung Lee and Angel X. Chang. Understanding pure clip guidance for voxel grid nerf models, 2022.

[37] Bo Li, Yuanhan Zhang, Liangyu Chen, Jinghao Wang, Fanyi Pu, Jingkang Yang, Chunyuan Li, and Ziwei Liu. Mimic-it: Multi-modal in-context instruction tuning. *arXiv preprint arXiv:2306.05425*, 2023.

[38] 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 *International conference on machine learning*, pages 19730–19742. PMLR, 2023.

[39] 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.

[40] Jianhui Li, Shilong Liu, Zidong Liu, Yikai Wang, Kaiwen Zheng, Jinghui Xu, Jianmin Li, and Jun Zhu. Instructpix2neRF: Instructed 3d portrait editing from a single image. In *The Twelfth International Conference on Learning Representations*, 2024.

[41] Ruilong Li, Hang Gao, Matthew Tancik, and Angjoo Kanazawa. Nerfacc: Efficient sampling accelerates nerfs. *arXiv preprint arXiv:2305.04966*, 2023.

[42] Derek Lim, Haggai Maron, Marc T. Law, Jonathan Lorraine, and James Lucas. Graph metanetworks for processing diverse neural architectures. In *The Twelfth International Conference on Learning Representations*, 2024.

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

[44] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36, 2024.

[45] Jingyue Lu and M. Pawan Kumar. Neural network branching for neural network verification. In *International Conference on Learning Representations*, 2020.

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

[47] Muhammad Maaz, Hanoona Rasheed, Salman Khan, and Fahad Shahbaz Khan. Video-chatgpt: Towards detailed video understanding via large vision and language models. *arXiv preprint arXiv:2306.05424*, 2023.

[48] Ricardo Martin-Brualla, Noha Radwan, Mehdi SM Sajjadi, Jonathan T Barron, Alexey Dosovitskiy, and Daniel Duckworth. Nerf in the wild: Neural radiance fields for unconstrained photo collections. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7210–7219, 2021.

[49] Gal Metzer, Elad Richardson, Or Patashnik, Raja Giryes, and Daniel Cohen-Or. Latent-nerf for shape-guided generation of 3d shapes and textures. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12663–12673, 2023.

[50] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *European conference on computer vision*, pages 405–421. Springer, 2020.

[51] Ashkan Mirzaei, Tristan Aumentado-Armstrong, Marcus A. Brubaker, Jonathan Kelly, Alex Levinshtein, Konstantinos G. Derpanis, and Igor Gilitschenski. Reference-guided controllable inpainting of neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 17815–17825, 2023.

[52] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Trans. Graph.*, 41(4):102:1–102:15, 2022.

[53] Aviv Navon, Aviv Shamsian, Idan Achituve, Ethan Fetaya, Gal Chechik, and Haggai Maron. Equivariant architectures for learning in deep weight spaces. In *International Conference on Machine Learning*, 2023.

[54] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.

[55] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318, 2002.

[56] Zhiliang Peng, Wenhui Wang, Li Dong, Yaru Hao, Shaohan Huang, Shuming Ma, and Furu Wei. Kosmos-2: Grounding multimodal large language models to the world. *arXiv preprint arXiv:2306.14824*, 2023.

[57] Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. In *The Eleventh International Conference on Learning Representations*, 2022.

[58] Zhangyang Qi, Ye Fang, Zeyi Sun, Xiaoyang Wu, Tong Wu, Jiaqi Wang, Dahua Lin, and Hengshuang Zhao. Gpt4point: A unified framework for point-language understanding and generation. In *CVPR*, 2024.

[59] 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.

[60] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67, 2020.

[61] Pierluigi Zama Ramirez, Luca De Luigi, Daniele Sirocchi, Adriano Cardace, Riccardo Spezialetti, Francesco Ballerini, Samuele Salti, and Luigi Di Stefano. Deep learning on 3d neural fields. *arXiv preprint arXiv:2312.13277*, 2023.

[62] Partha Pratim Ray. Chatgpt: A comprehensive review on background, applications, key challenges, bias, ethics, limitations and future scope. *Internet of Things and Cyber-Physical Systems*, 3:121–154, 2023.

[63] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*, 2019.

[64] Konstantin Schürholt, Dimche Kostadinov, and Damian Borth. Self-supervised representation learning on neural network weights for model characteristic prediction. In *Advances in Neural Information Processing Systems*, 2021.

[65] Bipasha Sen, Gaurav Singh, Aditya Agarwal, Rohith Agaram, Madhava Krishna, and Srinath Sridhar. Hyp-nerf: Learning improved nerf priors using a hypernetwork. In *Advances in Neural Information Processing Systems*, pages 51050–51064. Curran Associates, Inc., 2023.

[66] Hoigi Seo, Hayeon Kim, Gwanghyun Kim, and Se Young Chun. Ditto-nerf: Diffusion-based iterative text to omni-directional 3d model, 2023.

[67] Hyeonseop Song, Seokhun Choi, Hoseok Do, Chul Lee, and Taehyeong Kim. Blending-nerf: Text-driven localized editing in neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 14383–14393, 2023.

[68] Cheng Sun, Min Sun, and Hwann-Tzong Chen. Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5459–5469, 2022.

[69] Chunyi Sun, Yanbin Liu, Junlin Han, and Stephen Gould. Nerfeditor: Differentiable style decomposition for 3d scene editing. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 7306–7315, 2024.

[70] 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.

[71] Thomas Unterthiner, Daniel Keysers, Sylvain Gelly, Olivier Bousquet, and Ilya O. Tolstikhin. Predicting neural network accuracy from weights. *arXiv*, abs/2002.11448, 2020.

[72] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2017.

[73] Can Wang, Menglei Chai, Mingming He, Dongdong Chen, and Jing Liao. Clip-nerf: Text-and-image driven manipulation of neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3835–3844, 2022.

[74] Can Wang, Ruixiang Jiang, Menglei Chai, Mingming He, Dongdong Chen, and Jing Liao. Nerf-art: Text-driven neural radiance fields stylization. *IEEE Transactions on Visualization and Computer Graphics*, pages 1–15, 2023.

[75] Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. Finetuned language models are zero-shot learners. In *International Conference on Learning Representations*, 2021.

[76] Tong Wu, Jiarui Zhang, Xiao Fu, Yuxin Wang, Liang Pan Jiawei Ren, Wayne Wu, Lei Yang, Jiaqi Wang, Chen Qian, Dahua Lin, and Ziwei Liu. Omniobject3d: Large-vocabulary 3d object dataset for realistic perception, reconstruction and generation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[77] Qiangeng Xu, Weiyue Wang, Duygu Ceylan, Radomir Mech, and Ulrich Neumann. Disn: Deep implicit surface network for high-quality single-view 3d reconstruction. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2019.

[78] 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.

[79] Lin Yen-Chen, Pete Florence, Jonathan T Barron, Tsung-Yi Lin, Alberto Rodriguez, and Phillip Isola. Nerf-supervision: Learning dense object descriptors from neural radiance fields. In *2022 international conference on robotics and automation (ICRA)*, pages 6496–6503. IEEE, 2022.

[80] Yingchen Yu, Rongliang Wu, Yifang Men, Shijian Lu, Miaomiao Cui, Xuansong Xie, and Chunyan Miao. Morphnerf: Text-guided 3d-aware editing via morphing generative neural radiance fields. *IEEE Transactions on Multimedia*, pages 1–13, 2024.

[81] Pierluigi Zama Ramirez, Luca De Luigi, Daniele Sirocchi, Adriano Cardace, Riccardo Spezialetti, Francesco Ballerini, Samuele Salti, and Luigi Di Stefano. Deep learning on 3D neural fields. *arXiv preprint arXiv:2312.13277*, 2023.

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

[83] Allan Zhou, Kaien Yang, Kaylee Burns, Adriano Cardace, Yiding Jiang, Samuel Sokota, J Zico Kolter, and Chelsea Finn. Permutation equivariant neural functionals. *Advances in neural information processing systems*, 37, 2023.

[84] Allan Zhou, Kaien Yang, Yiding Jiang, Kaylee Burns, Winnie Xu, Samuel Sokota, J Zico Kolter, and Chelsea Finn. Neural functional transformers. *Advances in neural information processing systems*, 37, 2023.

[85] Allan Zhou, Chelsea Finn, and James Harrison. Universal neural functionals. *arXiv preprint arXiv:2402.05232*, 2024.

[86] Ziyu Zhu, Xiaojian Ma, Yixin Chen, Zhidong Deng, Siyuan Huang, and Qing Li. 3d-vista: Pre-trained transformer for 3d vision and text alignment. *ICCV*, 2023.

[87] Jingyu Zhuang, Chen Wang, Liang Lin, Lingjie Liu, and Guanbin Li. Dreameditor: Text-driven 3d scene editing with neural fields. In *SIGGRAPH Asia 2023 Conference Papers*, pages 1–10, 2023.

A Details on NeRFs

We report here some details regarding the NeRFs of the ShapeNeRF–Text dataset, which were trained by Zama Ramirez et al. [81]. The NeRF code is implemented leveraging the `NerfAcc` library [41].

A.1 Architecture

An instance of the employed NeRFs consists of a multi-layer perceptron (MLP) that contains three hidden layers, each with 64 neurons. The ReLU activation function is applied between all layers except for the last one, which calculates the density and RGB values directly without any activation function. A frequency encoding [50] is applied to the input 3D coordinates, in order to improve the NeRF reconstruction quality. NeRFs do not take as input the view direction. The MLP processes an input coordinate $\mathbf{p} \in \mathbb{R}^3$, to produce a 4-dimensional vector containing $RGB\sigma$.

A.2 Training

Training a NeRF consists of minimizing the error between the rendered images from the NeRF and the ground truth images. Our NeRFs were trained using an L_1 loss between the predicted and ground truth RGB pixel intensities, weighting background pixels less than foreground pixels (0.8 foregrounds vs. 0.2 background). Image rendering involves querying the neural network by feeding it 3D coordinates to obtain RGB color values and density estimates. By integrating these outputs along camera rays using volumetric rendering techniques [50], colors and opacities are accumulated to produce the final rendered image. Each NeRF is trained until it reaches a good reconstruction quality, approximately for 2000 steps.

A.3 Generating images and point clouds from NeRFs

To compare with 2D and 3D MLLMs on the new tasks of NeRF captioning and NeRF Q&A, we need to render images or reconstruct point clouds from the NeRF. To render images, we employ the same volumetric rendering procedure used during the NeRF's training. In order to extract a point cloud, the marching cubes algorithm is first applied to the volumetric density field derived from the NeRF. This process generates a mesh by identifying isosurfaces within the density field. The mesh is then converted into a point cloud by considering only the mesh vertices, uniformly distributed in the 3D space. We sample RGB values from NeRF for each point coordinate to approximate point cloud colors. An example of data extracted from NeRF is depicted in Fig. 6.

Generating images and point clouds requires the user to make some decisions, the effects of which on the assistant's performance are not easy to anticipate. When dealing with images, it is difficult to select the rendering viewpoint. It might happen that the object is not clearly visible from the chosen viewpoint or that important elements are missing. Another decision is the resolution of the generated image, which, if too coarse, may prevent the identification of fine-grained details. The same concerns regarding the resolution also apply to point clouds. Yet, the modality encoder may not handle large resolutions or may greatly increase the processing time. Another important point is the additional computational time required to extract data from NeRF. For instance, extracting point clouds from NeRF with only 8192 points requires approximately $620\texttt{ms}$. Moreover, the time for sampling the MLP and running a marching cube algorithm scales cubically with the desired spatial resolution. On the other hand, the time required to process the MLP weights is independent of the spatial resolution.

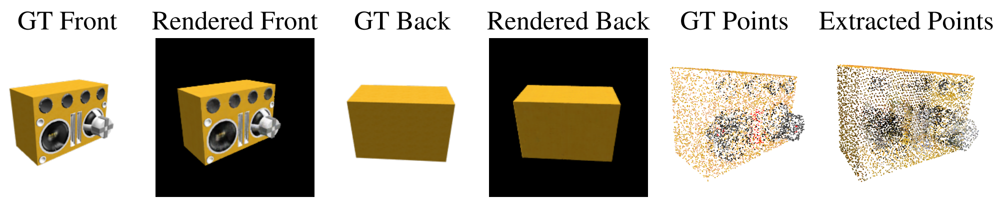


Figure 6: **Example of data extracted from NeRF.** From left to right: GT front view, rendered front view, GT back view, rendered back view, GT point cloud, extracted point cloud.

A.4 NeRF memory occupation compared to images or point clouds

An important benefit of using NeRF to represent objects is that memory consumption is decoupled from spatial resolution. In Fig. 7, we analyze the number of parameters needed for point clouds and images compared to neural fields by altering the spatial resolution of the data. We account for all variables required by an explicit representation in their parameter count. For instance, each point in a point cloud has six parameters corresponding to its coordinates (x, y, z) and color (R, G, B), while each pixel has only three channels (R, G, B). The orange line represents the parameters of the NeRF MLP, while the blue lines indicate the parameters for 3D points (Fig. 7-left) and image pixels (Fig. 7-right).

We observe that the space occupied by the NeRF MLP is comparable to that used by point clouds in our experiments (i.e., 8192 points, the data size used in GPT4Point [58] and PointLLM [78]). However, NeRF becomes advantageous for representing data as soon as the point cloud size is greater than 8621 points. This is crucial, considering real datasets may contain point clouds or meshes with significantly more points or faces; for example, Objaverse [13] features meshes with over 10^7 polygons.

The advantages are even more pronounced for images, where a single NeRF MLP corresponds to 36 images at a resolution of 22×22. Storing the 36 pictures from ShapeNetRender at 256×256 resolution, used to train our NeRF on a single object, requires substantially more memory.

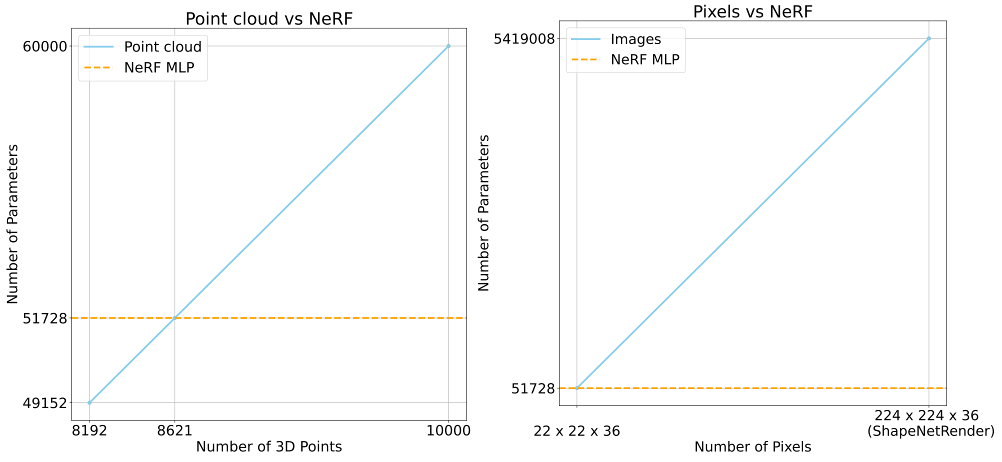


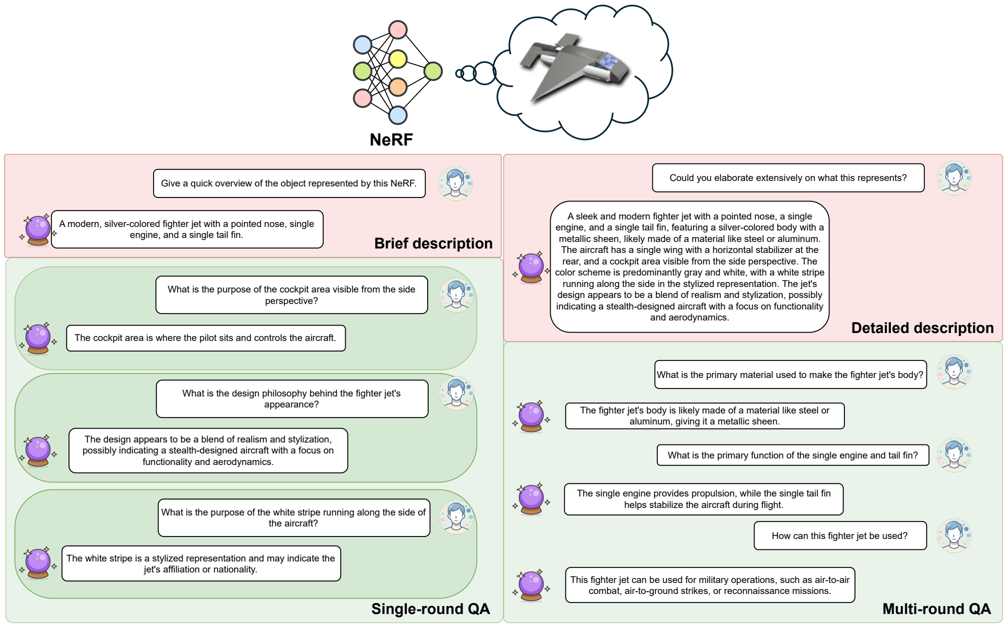
Figure 7: **Memory usage of NeRF compared to images or point clouds.** Left: NeRF vs point clouds. Right: NeRF vs pixels.

B Details on the Meta-Encoder

We employ `nf2vec` [81] as the meta-encoder of LLaNA. Thus, in the following paragraphs, we describe the details of the `nf2vec` architecture and training protocol.

B.1 Architecture

The `nf2vec` encoder consists of 4 linear layers with 512, 512, 512, and 1024 output neurons each. The final embedding extracted for a given MLP is a 1024-dimensional array. Each linear layer is followed by a batch normalization and a ReLU activation function. The weight matrices and biases of the input NeRF are stacked along the row dimension to form a matrix $\mathbf{M} \in \mathbb{R}^{S \times H}$, where $S = 144 + 1 + L * (H + 1) + H + 1 = L * H + L + H + 146$ and L is the number of units per hidden layer, while H is the dimension of the input.



Give a quick overview of the object represented by this NeRF.

A modern, silver-colored fighter jet with a pointed nose, single engine, and a single tail fin.

Could you elaborate extensively on what this represents?

A sleek and modern fighter jet with a pointed nose, a single engine, and a single tail fin, featuring a silver-colored body with a metallic sheen, likely made of a material like steel or aluminum. The aircraft has a single wing with a horizontal stabilizer at the rear, and a cockpit area visible from the side perspective. The color scheme is predominantly gray and white, with a white stripe running along the side in the stylized representation. The jet's design appears to be a blend of realism and stylization, possibly indicating a stealth-designed aircraft with a focus on functionality and aerodynamics.

What is the purpose of the cockpit area visible from the side perspective?

The cockpit area is where the pilot sits and controls the aircraft.

What is the design philosophy behind the fighter jet's appearance?

The design appears to be a blend of realism and stylization, possibly indicating a stealth-designed aircraft with a focus on functionality and aerodynamics.

What is the purpose of the white stripe running along the side of the aircraft?

The white stripe is a stylized representation and may indicate the jet's affiliation or nationality.

What is the primary material used to make the fighter jet's body?

The fighter jet's body is likely made of a material like steel or aluminum, giving it a metallic sheen.

What is the primary function of the single engine and tail fin?

The single engine provides propulsion, while the single tail fin helps stabilize the aircraft during flight.

How can this fighter jet be used?

This fighter jet can be used for military operations, such as air-to-air combat, air-to-ground strikes, or reconnaissance missions.

Figure 8: **Example of data sample from ShapeNeRF–Text dataset.**

B.2 Training

We employ the official code and weights of `nf2vec`[2] pre-trained on an augmented version of ShapeNetRender [77]. The encoder was trained in an end-to-end manner together with an implicit decoder. The decoder takes in input 3D coordinates after a frequency encoding and the global 1024-dimensional output of the encoder. It consists of 5 linear layers with 1024 neurons each, followed by ReLU activations except for the last layer. It yields a 4-dimensional output $RGB\sigma$, similar to the NeRF given in input to the encoder. The framework supervision comes from the pixel-wise rendering L_1 error computed between the ground truth RGB image and the predicted image, which is obtained through volumetric rendering after encoding and decoding the NeRF.

C Details on ShapeNeRF–Text dataset

The proposed ShapeNeRF–Text dataset consists of 40K paired NeRFs and language annotations for ShapeNet objects [8]. In particular, for every 3D model, multiple annotations have been provided: a brief description, a detailed description, 3 single-round Q&As, and one multi-round Q&A. Figure 8 shows an example of such annotations. These annotations have been obtained by exploiting LLaVA 2 and LLaMA 3 as described in section 4 of the main paper.

C.1 Instruction prompts and ground-truth questions

In this section, we provide the instruction prompts used to generate the ground-truth answers of ShapeNeRF–Text and the list of questions used to build the ground-truth questions for the brief and detailed descriptions.

Instruction prompts for LLaVA and LLaMA to generate the dataset For constructing ShapeNerf–Text, first, descriptive captions for multiple views of each object have been obtained using the following input request to LLaVA:

[2]`https://cvlab-unibo.github.io/nf2vec/`

- *"USER:<image>\nYou will be provided the image of an object, seen from the <view_point>. Describe the object in detail. Include as much information as possible, but do not infer anything that is not in the image. Avoid describing the background. Generate an answer with a maximum length of 30 words.\nASSISTANT:"*

The placeholder <view_point> was replaced with "back", "side", or "front" according to the viewpoint of the image provided as input. To expedite computation and leverage the high symmetry of ShapeNet objects, 7 views have been employed for this process.

After obtaining the captions for each view, LLaMA was queried to aggregate these single-view captions into comprehensive descriptions and Q&A rounds. The input provided to LLaMA was:

- *You will be shown 7 different descriptions of an object, obtained from different points of view. Please provide two descriptions, which aggregates all these ones. The first description must be concise, the second one will be more descriptive. Both these description must refer to the same subject. Avoid repetitions. Important: The output descriptions must be followed by the string "Final concise description:" and "Final more detailed description:". Notice: There are errors in some descriptions, due to occlusion and improper angle. You need to combine all the descriptions and eliminate possible wrong details (please fix the errors directly, do not tell me). Input descriptions:* [list of the single-view captions generated by LLaVA]

The detailed description was then used to generate multiple Q&A rounds, through the following request:

- *Given this description of an object, generate 6 short Q&A dialogues regarding diverse aspects of the object described, ensuring logical relevance between the questions and answers. Include always a question about how this object can be used. Question begins with 'Q'. Answer begins with 'A'. IMPORTANT: Do not mention size, background. Do not mention "how many". Do not add text after the last answer.".*

From the 6 generated Q&A pairs, 3 were randomly sampled to build the sequence of multi-round Q&A, while the remaining pairs were used as single-round Q&A.

Ground-truth questions for the brief and detailed descriptions Tab. 10 and Tab. 11 provide the list of questions used to build the ground-truth data of ShapeNeRF–Text, as explained in Sec. 4.1.

Table 10: List of questions to prompt the model to produce brief descriptions. An instruction from the list is randomly selected and coupled with a ShapeNeRF–Text brief caption to form a ground-truth data sample.

- Summarize the 3D object briefly.
- What kind of object is depicted by this NeRF?
- Provide a short explanation of this object.
- What does this NeRF represent?
- Can you give a brief overview of this object?
- Characterize the object this NeRF is illustrating.
- Share a brief interpretation of this NeRF.
- Provide an outline of this 3D shape's characteristics.
- What object is this NeRF rendering?
- Deliver a quick description of the object represented here.
- How would you describe the 3D form shown in this NeRF?
- What is the nature of the object this NeRF is representing?
- Present a compact account of this 3D object's key features.
- What can you infer about the object from this NeRF?
- Offer a clear and concise description of this object.
- How would you summarize this 3D data?
- Give a brief explanation of the object that this NeRF represents.
- What kind of structure does this NeRF depict?
- Could you delineate the object indicated by this NeRF?
- Express in brief, what this NeRF is representing.

• Give a quick overview of the object represented by this NeRF.
• Convey a summary of the 3D structure represented in this NeRF.
• What kind of object is illustrated by this NeRF?
• Describe the object that this NeRF forms.
• How would you interpret this NeRF?
• Can you briefly outline the shape represented by this NeRF?
• Give a concise interpretation of the 3D data presented here.
• Explain the object this NeRF depicts succinctly.
• Offer a summary of the 3D object illustrated by this NeRF.

Table 11: List of questions to prompt the model to produce detailed descriptions. An instruction from the list is randomly selected and paired with a ShapeNeRF–Text detailed caption to form a ground-truth data sample.

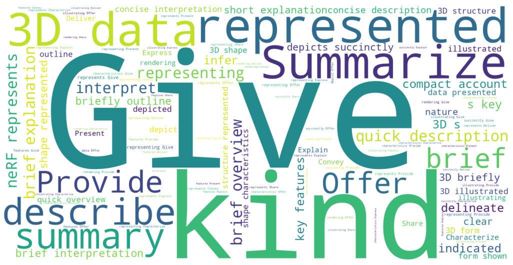
• Can you tell me more about this?
• What does this represent?
• Can you describe this in more detail?
• I'm interested in this. Can you explain?
• Could you provide more information about this?
• What exactly am I looking at here?
• What is this?
• Could you describe the detailed structure of this?
• This looks interesting. Can you expand on it?
• Can you explain more about this form?
• What can you tell me about the shape of this object?
• Could you delve deeper into this?
• I want to know more about this. Can you help?
• Can you walk me through the details of this object?
• Can you provide a comprehensive account of this object?
• Offer a detailed interpretation of this NeRF.
• Please elucidate on the characteristics of this form.
• Could you provide an in-depth description of this structure?
• What does this NeRF represent in its entirety?
• Elaborate on the details of this NeRF, please.
• Kindly furnish me with more information about this object.
• Please expand on the intricate structure of this form.
• Provide a meticulous explanation of what this NeRF represents.
• Provide a detailed explanation of what this NeRF represents.
• I request a detailed breakdown of this structure.
• Give a thorough rundown of this NeRF.
• Can you offer a complete analysis of this object?
• I would like a comprehensive explanation of this form.
• Please detail the specific features of this NeRF.
• Could you elaborate extensively on what this represents?

C.2 ShapeNeRF–Text statistics

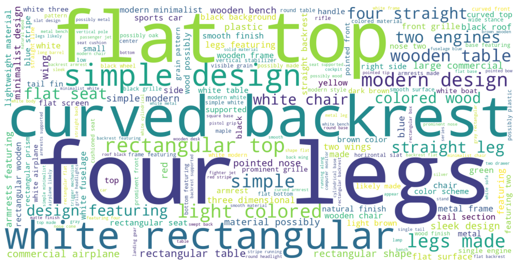
The average lengths in words of the instructions/responses are 8.51/22.76 for brief descriptions, 7.82/77.90 for detailed descriptions, 8.81/14.25 for single-round QAs and 8.80/14.14 (per round) for multi-round QAs. Fig. 9 and Fig. 10 report instruction/response length histograms and the word clouds obtained after removing generic words like "model", "object" and "NeRF", emphasizing frequent words in the ground-truth instructions and responses.

Figure 9: **ShapeNeRF-Text statistics for ground-truth brief and detailed descriptions.**

Brief Descriptions - Word clouds

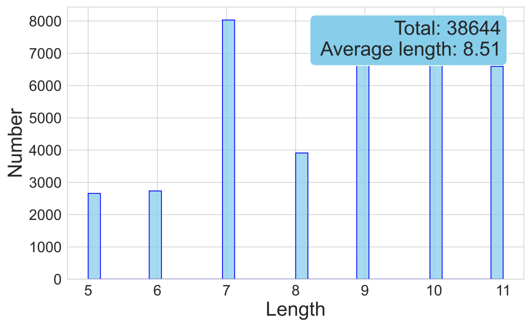


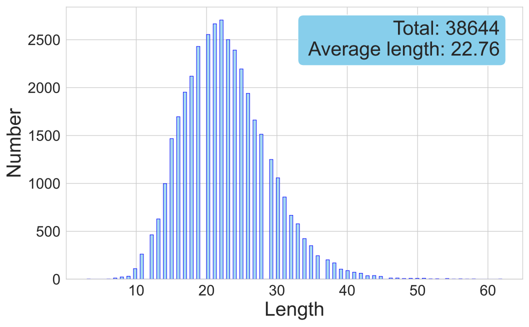
Instructions



Responses

Brief Descriptions - Lengths (Words)

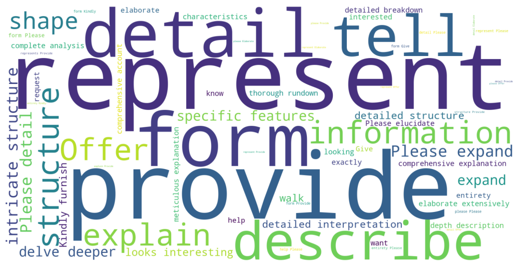




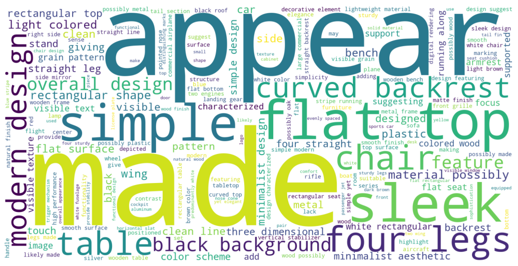
Instructions

Responses

Detailed Descriptions - Word clouds

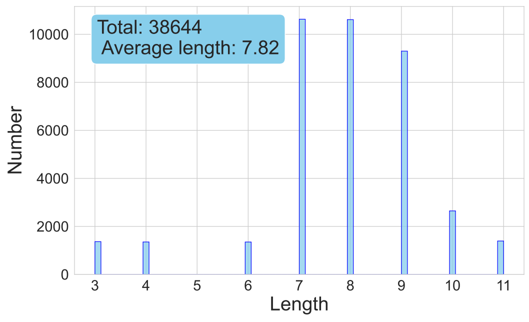


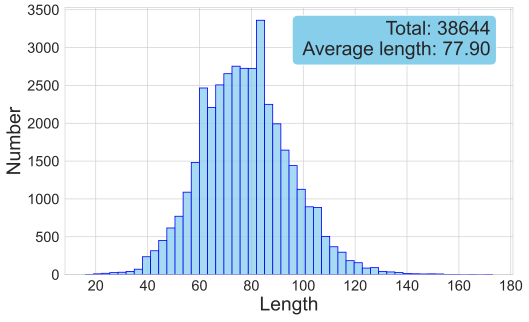
Instructions



Responses

Detailed Descriptions - Lengths (Words)



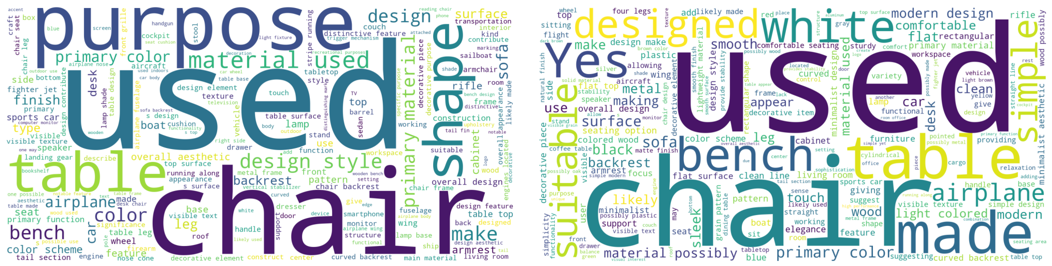


Instructions

Responses

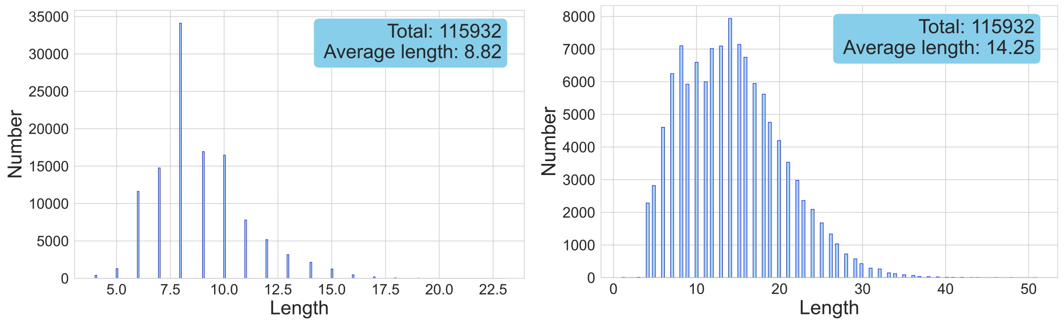
Figure 10: **ShapeNeRF-Text statistics for ground-truth single-round and multi-round Q&A.**

Single-round Q&A - Word clouds



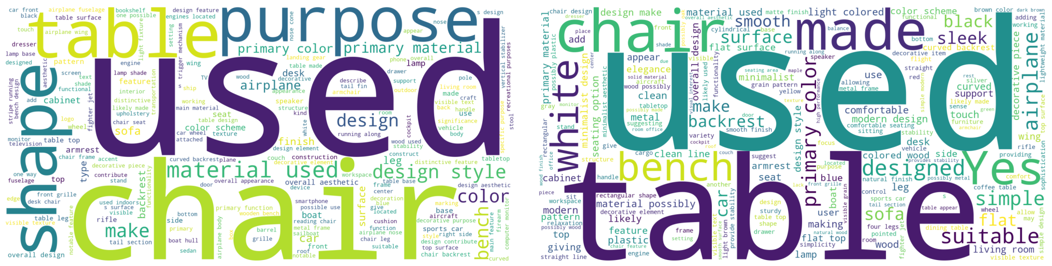
Instructions Responses

Single-round Q&A - Lengths (Words)



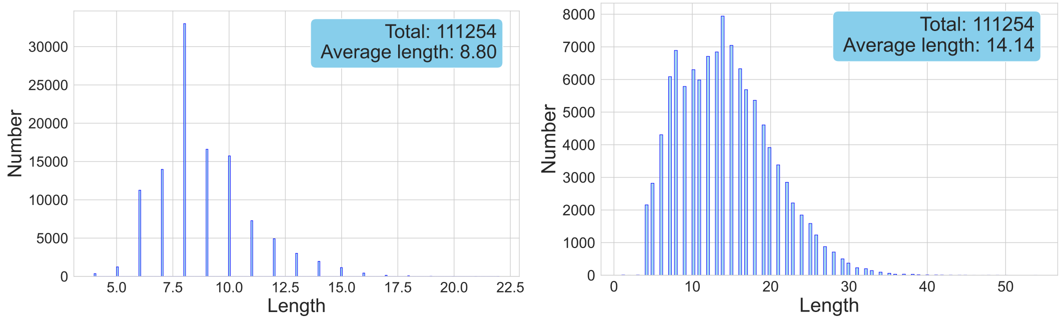
Instructions Responses

Multi-round Q&A - Word clouds



Instructions Responses

Multi-round Q&A - Lengths (Words)



Instructions Responses

C.3 ShapeNeRF–Text quality analysis

We have carried out several experiments to assess the quality of the questions in ShapeNeRF–Text. More specifically, the purpose of this analysis was to understand how many questions referred to a detail that is visible only from a specific viewpoint of the object. First, we evaluated our dataset questions with a language-only model, LLaMA3. For each question Q, we asked LLaMA3:
Is a random viewpoint of the object enough to answer this question?
<Q>
If so, reply "YES"; if a specific viewpoint is needed, answer "NO".

By doing so, we obtained 5163 "YES" and 5847 "NO", highlighting that most questions refer to some details which are visible only from a point of view.

Second, we ran a Vision-Language model, LLaVA-1.6-13b, on each question of the single-round Q&A dataset, on the front and back views of objects. Then, we selected only the LLaVA responses where the answer for the front or back view achieves a SimCSE score higher than 80%, i.e., likely correct answers, which selects approximately 45% of the answers. Among these correct responses, we calculated the percentage of those where the front and back answers are extremely different (i.e., a difference in SimCSE scores > 10). Remarkably, 26% of such answers are correct from one point of view but wrong from the other: these questions would have required multi-view information to be answered correctly. We report two qualitative examples in Fig. 11. In the first row, the Mercedes-Benz logo cannot be recognized from the back view. In the second row, from the rear viewpoint, the monitor seems turned off, and thus it is not possible to identify correctly the helicopter displayed on the screen. Similarly, Fig. 14 of the Appendix shows other examples of this kind of cases.

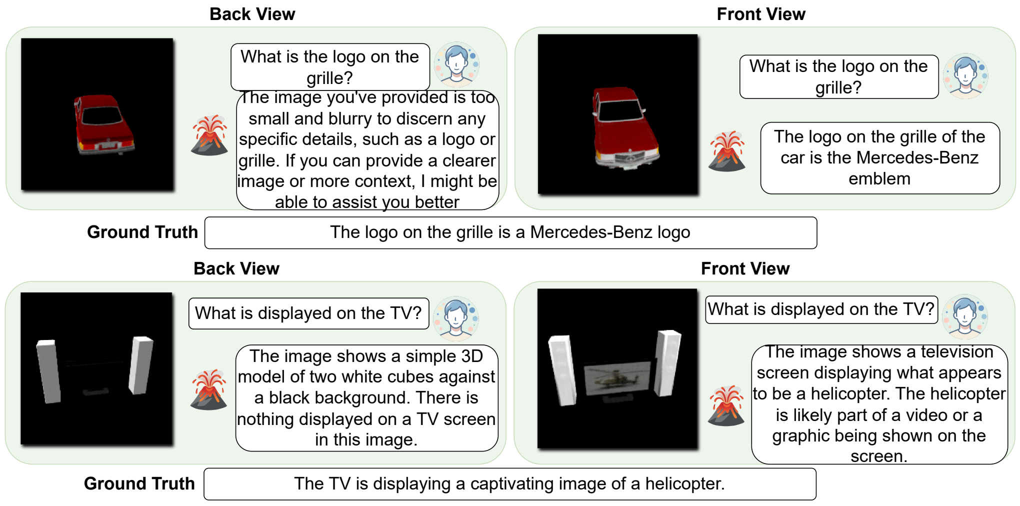

Back View
What is the logo on the grille?
The image you've provided is too small and blurry to discern any specific details, such as a logo or grille. If you can provide a clearer image or more context, I might be able to assist you better
Front View
What is the logo on the grille?
The logo on the grille of the car is the Mercedes-Benz emblem
Ground Truth
The logo on the grille is a Mercedes-Benz logo
Back View
What is displayed on the TV?
The image shows a simple 3D model of two white cubes against a black background. There is nothing displayed on a TV screen in this image.
Front View
What is displayed on the TV?
The image shows a television screen displaying what appears to be a helicopter. The helicopter is likely part of a video or a graphic being shown on the screen.
Ground Truth
The TV is displaying a captivating image of a helicopter.

Figure 11: **Front vs back results with LLaVA.** The dataset contains many view-dependent questions.

D Additional baselines results and details

D.1 Language-only baseline

To assess potential spurious patterns in the question-answer relationships, we evaluate the performance of LLaMA 2, the LLM on which LLaNA relies, fine-tuned on ShapeNeRF-Text. In this training and evaluation protocol, the LLM is provided with questions belonging to the dataset and must return the correct answers without having access to the NeRF data. Therefore, the predicted answers may be generated only based on the textual patterns present in the training set. Results are shown in Tab. 12. A significant performance gap exists between LLaMA 2 and LLaNA, highlighting that our dataset consists of questions that can only be answered with access to information about 3D objects.

Table 12: **Language-only baseline.**

	Model	Modality	Sentence-BERT	SimCSE	BLEU-1	ROUGE-L	METEOR
Brief	LLaMA2	Text-only	42.62	40.70	**25.14**	24.53	25.53
	LLaNA-7b	NeRF	**68.63**	**70.54**	20.64	**28.33**	**31.76**
Detailed	LLaMA2	Text-only	49.73	47.68	15.15	23.27	14.78
	LLaNA-7b	NeRF	**77.43**	**79.81**	**41.32**	**36.18**	**32.39**
SingleQA	LLaMA2	Text-only	68.37	68.46	44.07	51.15	48.00
	LLaNA-7b	NeRF	**81.03**	**81.56**	**46.16**	**53.17**	**50.15**

D.2 Zero-Shot NeRF classification of trained baselines

We report in Tab. 13 the results obtained on zero-shot NeRF classification task by the baselines trained on ShapeNeRF-Text. Results follow the same trend as the other language tasks, reported in the main paper.

Table 13: **Zero-shot NeRF classification on ShapeNeRF-Text. Trained baselines.** Best results are in **bold**, runner-up is <u>underlined</u>. (FV: front-view)

Model	Modality	Accuracy (%)
LLaVA-vicuna-13b	Image (FV)	36.49
PointLLM-7b	Point cloud	<u>49.69</u>
GPT4Point-Opt-2.7b	Point cloud	26.30
LLaNA-7b	NeRF	**67.14**

E Ground-truth images and point clouds

This section presents the results of an experiment in which the baseline 2D and 3D MLLMs have been provided with ground-truth input images and point clouds extracted from the original 3D meshes in the dataset rather than from the NeRFs. This scenario estimates an upper bound for the performance of such approaches when used as NeRF assistants, by simulating perfect extraction of images or point clouds from the NeRFs. In other words, it simulates the ideal scenario in which the encoding of information inside a NeRF is lossless, a non-realisitc situation in which the baselines can achieve their best performance. Tab. 14, Tab. 15, and Tab. 16 show the results of this experiments on the tasks of brief description, detailed description, and single-round Q&A, respectively. For brevity, the best-performing 2D model, i.e., LLaVA [44] (on front views) and the best-performing 3D model, i.e., PointLLM [78], have been tested in this scenario. The results demonstrate that, even in this idealized and most favorable scenario for the baselines, LLaNA outperforms them.

Model	Modality	Sentence-BERT	SimCSE	BLEU-1	ROUGE-L	METEOR
LLaVA-vicuna-13b	Image (FV)	68.61	67.99	17.48	23.08	27.03
PointLLM-7b	Point cloud	51.99	51.70	17.19	18.63	15.03
LLaNA-7b	NeRF	**68.63**	**70.54**	**20.64**	**28.33**	**31.76**

Table 14: **NeRF brief captioning on ShapeNeRF–Text dataset.** Frozen baseline results obtained on data extracted from ShapeNet mesh data. Best results in **bold**. Runner-up <u>underlined</u>. (FV: front-view)

F Generalization experiments

We conducted an experiment to probe the generalization capabilities of LLaNA against the trained baselines. We evaluate the models on the subset of 200 Objaverse [13] objects with human-annotated captions used as a test set by PointLLM [78]. This evaluation protocol sets forth a challenging out-of-domain and open-set experiment (164 out of 200 Objaverse objects belong to categories not

Model	Modality	Sentence-BERT	SimCSE	BLEU-1	ROUGE-L	METEOR
LLaVA-vicuna-13b	Image (FV)	68.32	67.35	27.46	26.62	24.40
PointLLM-7b	Point cloud	61.87	61.77	10.65	19.90	10.93
LLaNA-7b	NeRF	**77.43**	**79.81**	**41.32**	**36.18**	**32.39**

Table 15: **NeRF detailed captioning on ShapeNeRF–Text dataset.** Frozen baseline results obtained on data extracted from ShapeNet mesh data. Best results in **bold**. Runner-up underlined.

Model	Modality	Sentence-BERT	SimCSE	BLEU-1	ROUGE-L	METEOR
LLaVA-vicuna-13b	Image (FV)	78.40	75.68	22.65	33.04	35.70
PointLLM-7b	Point cloud	74.98	74.90	36.93	44.60	39.87
LLaNA-7b	NeRF	**81.03**	**81.56**	**46.16**	**53.17**	**50.15**

Table 16: **NeRF single-round Q&A on ShapeNeRF–Text dataset.** Frozen baseline results obtained on data extracted from ShapeNet mesh data. Best results in **bold**. Runner-up underlined.

present in ShapeNeRF-Text). To test LLaNA, we fit NeRFs for all the objects of the test set. Then, we extracted colored point clouds and rendered front views from NeRFs, in order to test the baselines. In Tab. 17 we can observe that the scores of all models are significantly lower compared to Tab. 6, which hints at all models struggling when evaluated on objects very different from those belonging to the training domain. LLaNA achieves the second-best generalization performance after PointLLM. Yet, it is worth highlighting that the frozen modality-specific encoder of PointLLM (and GPT4Point) is PointBERT, which was pre-trained on Objaverse. In contrast, LLaNA meta-encoder, nf2vec, has been trained only on ShapeNet, meaning it has never encountered objects outside the ShapeNet categories.

Table 17: **Generalization results on Objaverse.**

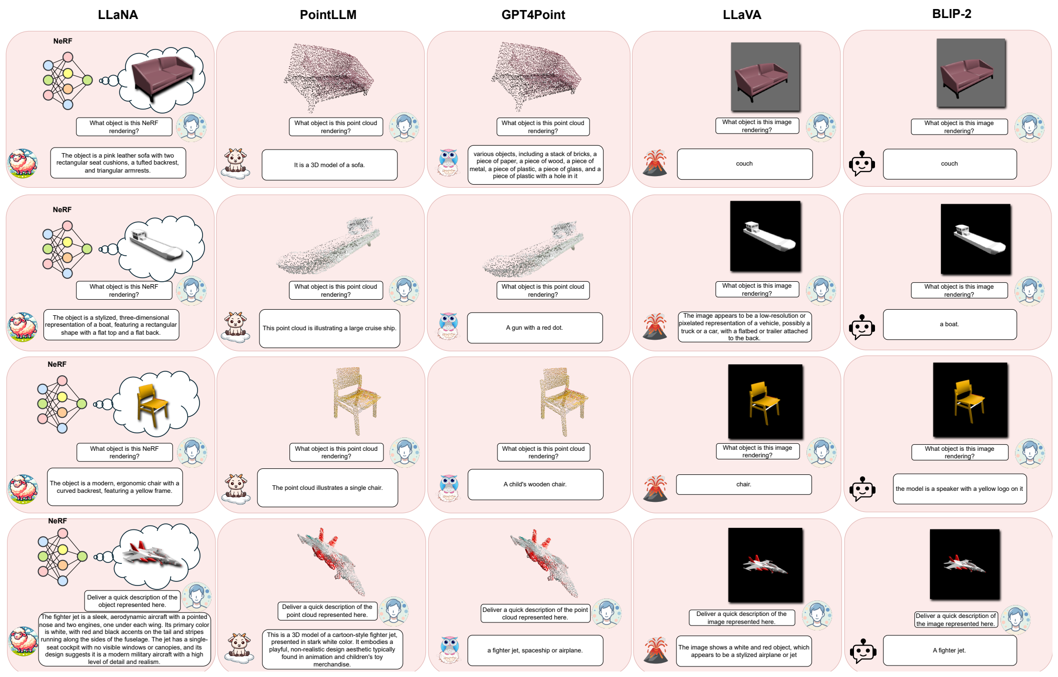
		NeRF captioning				
Model	**Modality**	**Sentence-BERT**	**SimCSE**	**BLEU-1**	**ROUGE-L**	**METEOR**
LLaVA-vicuna-13b	Image (FV)	27.07	26.82	4.41	6.81	9.77
PointLLM-7b	Point cloud	**33.88**	**33.04**	**5.37**	**8.14**	**12.28**
GPT4Point-Opt-2.7b	Point cloud	25.94	29.04	4.25	7.99	10.42
LLaNA-7b	NeRF	30.07	28.17	4.93	7.47	11.15

G Additional qualitative examples

This section provides additional qualitative comparisons between the proposed method, i.e. LLaNA which directly processes NeRF, and the baselines that take as input images [44, 39] or 3D representations [78, 58]. In particular, Figs. 12 to 14 show additional brief descriptions, detailed descriptions, and single-round Q&A provided as output by the different methods. Many examples, such as the white speaker in the third row of Fig. 13, are not described properly by MLLMs operating on point clouds. Indeed, due to the input point cloud containing only 8192 points, these methods cannot perceive the object details, such as the curved surface of the speaker, therefore they predict that the object is a "cubic white object" or an "ice cube". In other examples, such as the white screen sample in the last row of Fig. 14, the LLM operating on images cannot give the right answer to the question on the button location as it is not visible from the given viewpoint. Contrarily, by operating directly on the holistic representation provided by NeRFs, LLaNA provides the right answer in these situations.

H Information about datasets, models, and source code licenses

This section provides details about the datasets, models, and source code licenses used in the paper, ensuring proper credit to the creators or original owners, and adherence to license terms.



What object is this NeRF rendering?

The object is a pink leather sofa with two rectangular seat cushions, a tufted backrest, and triangular armrests.

What object is this point cloud rendering?

It is a 3D model of a sofa.

What object is this point cloud rendering?

various objects, including a stack of bricks, a piece of paper, a piece of wood, a piece of metal, a piece of plastic, a piece of glass, and a piece of plastic with a hole in it

What object is this image rendering?

couch

What object is this image rendering?

couch

What object is this NeRF rendering?

The object is a stylized, three-dimensional representation of a boat, featuring a rectangular shape with a flat top and a flat back.

What object is this point cloud rendering?

This point cloud is illustrating a large cruise ship.

What object is this point cloud rendering?

A gun with a red dot.

What object is this image rendering?

The image appears to be a low-resolution or pixelated representation of a vehicle, possibly a truck or a car, with a flatbed or trailer attached to the back.

What object is this image rendering?

a boat.

What object is this NeRF rendering?

The object is a modern, ergonomic chair with a curved backrest, featuring a yellow frame.

What object is this point cloud rendering?

The point cloud illustrates a single chair.

What object is this point cloud rendering?

A child's wooden chair.

What object is this image rendering?

chair.

What object is this image rendering?

the model is a speaker with a yellow logo on it

Deliver a quick description of the object represented here.

The fighter jet is a sleek, aerodynamic aircraft with a pointed nose and two engines, one under each wing. Its primary color is white, with red and black accents on the tail and stripes running along the sides of the fuselage. The jet has a single-seat cockpit with no visible windows or canopies, and its design suggests it is a modern military aircraft with a high level of detail and realism.

Deliver a quick description of the point cloud represented here.

This is a 3D model of a cartoon-style fighter jet, presented in stark white color. It embodies a playful, non-realistic design aesthetic typically found in animation and children's toy merchandise.

Deliver a quick description of the point cloud represented here.

a fighter jet, spaceship or airplane.

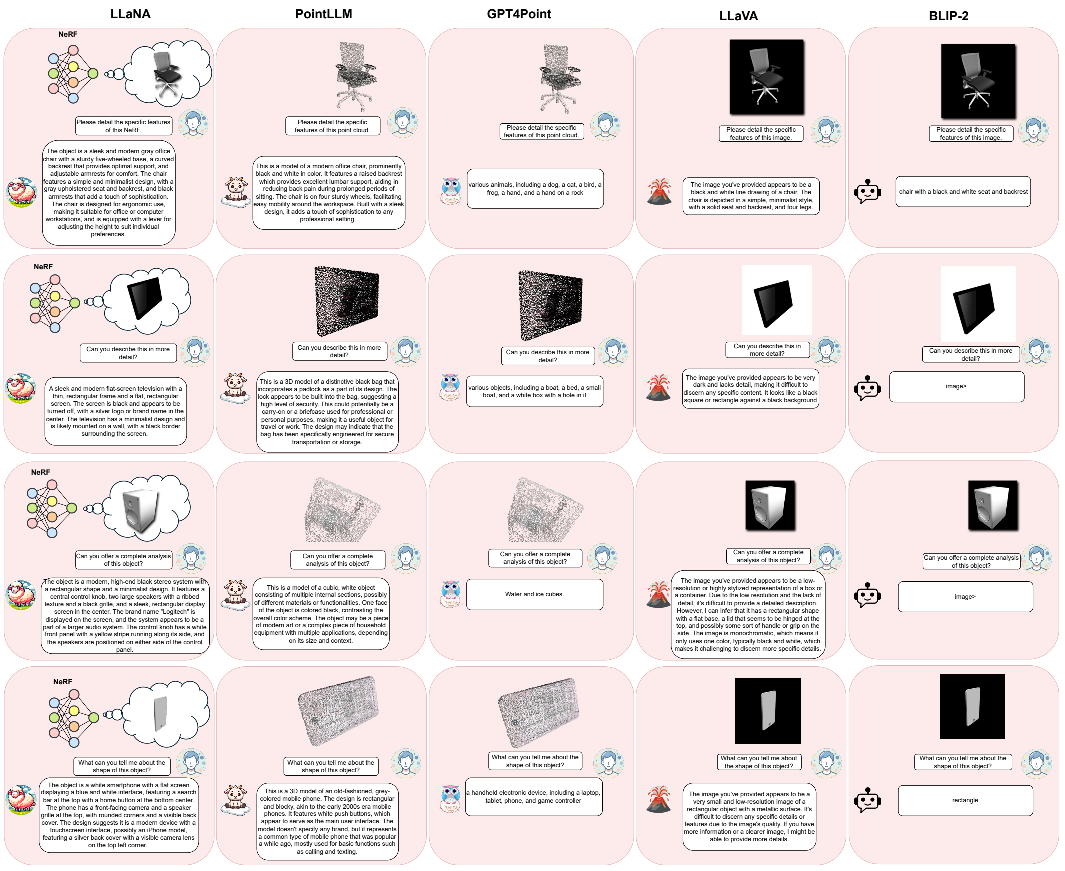
Deliver a quick description of the image represented here.

The image shows a white and red object, which appears to be a stylized airplane or jet

Deliver a quick description of the image represented here.

A fighter jet.

Figure 12: **Additional qualitative examples for the brief description task.**



Please detail the specific features of this NeRF.

The object is a sleek and modern gray office chair with a sturdy five-wheeled base, a curved backrest that provides optimal support, and adjustable armrests for comfort. The chair features a simple and minimalist design, with a gray upholstered seat and backrest, and black armrests that add a touch of sophistication. The chair is designed for ergonomic use, making it suitable for office or computer workstations, and is equipped with a lever for adjusting the height to suit individual preferences.

Please detail the specific features of this point cloud.

This is a model of a modern office chair, prominently black and white in color. It features a raised backrest which provides excellent lumbar support, aiding in reducing back pain during prolonged periods of sitting. The chair is on four sturdy wheels, facilitating easy mobility around the workspace. Built with a sleek design, it adds a touch of sophistication to any professional setting.

Please detail the specific features of this point cloud.

various animals, including a dog, a cat, a bird, a frog, a hand, and a hand on a rock

Please detail the specific features of this image.

The image you've provided appears to be a black and white line drawing of a chair. The chair is depicted in a simple, minimalist style, with a solid seat and backrest, and four legs.

Please detail the specific features of this image.

chair with a black and white seat and backrest

Can you describe this in more detail?

A sleek and modern flat-screen television with a thin, rectangular frame and a flat, rectangular screen. The screen is black and appears to be turned off, with a silver logo or brand name in the center. The television has a minimalist design and is likely mounted on a wall, with a black border surrounding the screen.

Can you describe this in more detail?

This is a 3D model of a distinctive black bag that incorporates a padlock as a part of its design. The lock appears to be built into the bag, suggesting a high level of security. This could potentially be a carry-on or a briefcase used for professional or personal purposes, making it a useful object for travel or work. The design may indicate that the bag has been specifically engineered for secure transportation or storage.

Can you describe this in more detail?

various objects, including a boat, a bed, a small boat, and a white box with a hole in it

Can you describe this in more detail?

The image you've provided appears to be very dark and lacks detail, making it difficult to discern any specific content. It looks like a black square or rectangle against a black background

Can you describe this in more detail?

image>

Can you offer a complete analysis of this object?

The object is a modern, high-end black stereo system with a rectangular shape and a minimalist design. It features a central control knob, two large speakers with a ribbed texture and a black grille, and a sleek, rectangular display screen in the center. The brand name "Logitech" is displayed on the screen, and the system appears to be a part of a larger audio system. The control knob has a white front panel with a yellow stripe running along its side, and the speakers are positioned on either side of the control panel.

Can you offer a complete analysis of this object?

This is a model of a cubic, white object consisting of multiple internal sections, possibly of different materials or functionalities. One face of the object is colored black, contrasting the overall color scheme. The object may be a piece of modern art or a complex piece of household equipment with multiple applications, depending on its size and context.

Can you offer a complete analysis of this object?

Water and ice cubes.

Can you offer a complete analysis of this object?

The image you've provided appears to be a low-resolution or highly stylized representation of a box or a container. Due to the low resolution and the lack of detail, it's difficult to provide a detailed description. However, I can infer that it has a rectangular shape with a flat base, a lid that seems to be hinged at the top, and possibly some sort of handle or grip on the side. The image is monochromatic, which means it only uses one color, typically black and white, which makes it challenging to discern more specific details.

Can you offer a complete analysis of this object?

image>

What can you tell me about the shape of this object?

The object is a white smartphone with a flat screen displaying a blue and white interface, featuring a search bar at the top with a home button at the bottom center. The phone has a front-facing camera and a speaker grille at the top, with rounded corners and a visible back cover. The design suggests it is a modern device with a touchscreen interface, possibly an iPhone model, featuring a silver back cover with a visible camera lens on the top left corner.

What can you tell me about the shape of this object?

This is a 3D model of an old-fashioned, grey-colored mobile phone. The design is rectangular and blocky, akin to the early 2000s era mobile phones. It features white push buttons, which appear to serve as the main user interface. The model doesn't specify any brand, but it represents a common type of mobile phone that was popular a while ago, mostly used for basic functions such as calling and texting.

What can you tell me about the shape of this object?

a handheld electronic device, including a laptop, tablet, phone, and game controller

What can you tell me about the shape of this object?

The image you've provided appears to be a very small and low-resolution image of a rectangular object with a metallic surface. It's difficult to discern any specific details or features due to the image's quality. If you have more information or a clearer image, I might be able to provide more details.

What can you tell me about the shape of this object?

rectangle

Figure 13: **Additional qualitative examples for the detailed description task.**

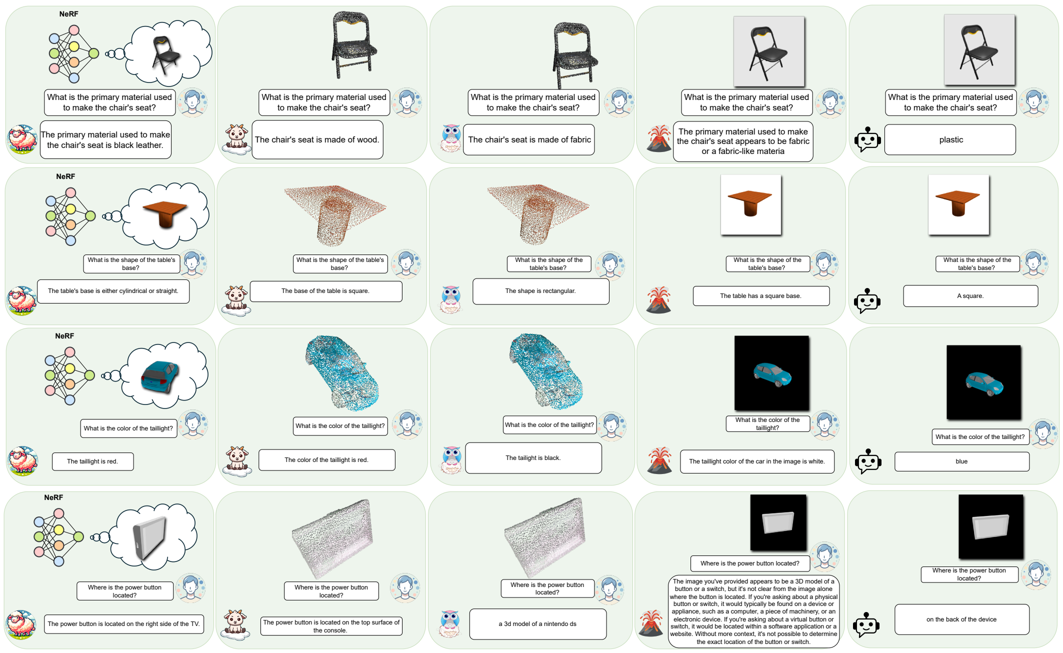

NeRF
What is the primary material used to make the chair's seat?
The primary material used to make the chair's seat is black leather.
What is the primary material used to make the chair's seat?
The chair's seat is made of wood.
What is the primary material used to make the chair's seat?
The chair's seat is made of fabric
What is the primary material used to make the chair's seat?
The primary material used to make the chair's seat appears to be fabric or a fabric-like materia
What is the primary material used to make the chair's seat?
plastic
NeRF
What is the shape of the table's base?
The table's base is either cylindrical or straight.
What is the shape of the table's base?
The base of the table is square.
What is the shape of the table's base?
The shape is rectangular.
What is the shape of the table's base?
The table has a square base.
What is the shape of the table's base?
A square.
NeRF
What is the color of the taillight?
The taillight is red.
What is the color of the taillight?
The color of the taillight is red.
What is the color of the taillight?
The tailight is black.
What is the color of the taillight?
The taillight color of the car in the image is white.
What is the color of the taillight?
blue
NeRF
Where is the power button located?
The power button is located on the right side of the TV.
Where is the power button located?
The power button is located on the top surface of the console.
Where is the power button located?
a 3d model of a nintendo ds
Where is the power button located?
The image you've provided appears to be a 3D model of a button or a switch, but it's not clear from the image alone where the button is located. If you're asking about a physical button or switch, it would typically be found on a device or appliance, such as a computer, a piece of machinery, or an electronic device. If you're asking about a virtual button or switch, it would be located within a software application or a website. Without more context, it's not possible to determine the exact location of the button or switch.
Where is the power button located?
on the back of the device

Figure 14: **Additional qualitative examples for the single-round Q&A task.**

Datasets: the datasets employed in our work and the relative licenses are listed below:

- **ShapeNet**: licensed under GNU Affero General Public License v3.0.
- **GPT2Shape HST**: licensed under Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License.

Models: the models used in all our experiments and their relative licenses are detailed in the following:

- **nf2vec**: licensed under MIT License.
- **PointLLM**: licensed under Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License.
- **GPT4Point**: licensed under Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License.
- **LLAMA-2**: licensed under META LLAMA 2 COMMUNITY LICENSE AGREEMENT[3].
- **LLAMA-3**: licensed under META LLAMA 3 COMMUNITY LICENSE AGREEMENT[4].
- **LLAVA**: licensed under Apache License 2.0.

Proper care has been taken to ensure that all licenses and terms of use are explicitly mentioned and respected throughout this paper.

[3]https://ai.meta.com/llama/license/
[4]https://ai.meta.com/llama/license/

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: the claims made in the article are also highlighted in the abstract and introduction.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: the limitations of the proposed approach are detailed in Sec. 6.

   Guidelines:

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

3. **Theory Assumptions and Proofs**

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

   Answer: [NA] .

Justification: the article does not introduce new theoretical results.

Guidelines:

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

4. **Experimental Result Reproducibility**

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

   Answer: [Yes]

   Justification: we provide all the information related to the training protocol used in our experiments and the implementation details in Sec. 3. In addition, Sec. 4 extensively details the experimental setup used for all the evaluated tasks. Moreover, we provide additional details in the appendix.

   Guidelines:

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

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

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

Answer: [Yes]

Justification: all our experiments have been conducted on publicly available data, i.e., ShapeNet, HST and Objaverse datasets. Our newly introduced benchmark, the source code and the weights for all our models will be publicly released in case of acceptance.

Guidelines:

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

6. **Experimental Setting/Details**

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

   Answer: [Yes]

   Justification: all implementation details and data splits are detailed in Sec. 3, Sec. 4 and in the appendix.

   Guidelines:

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

7. **Experiment Statistical Significance**

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

   Answer: [No]

   Justification: we did not conduct multiple trials for each model training and evaluation due to the large computational requirements needed to fine-tune the LLMs employed in our approach.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

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

8. **Experiments Compute Resources**

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

   Answer: [Yes]

   Justification: we provide details about the training time and computational resources required by our approach in Sec. 3.

   Guidelines:

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

9. **Code Of Ethics**

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

   Answer: [Yes]

   Justification: we reviewed the guidelines listed in the NeurIPS Code of Ethics and we confirm that our approach does not violate them.

   Guidelines:

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

10. **Broader Impacts**

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

    Answer: [NA]

    Justification: we do not foresee any direct path to using our solution for negative applications as it pertains describing digital twins of single objects.

    Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

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

    Answer: [Yes]

    Justification: the LLaNA model will be released under the following terms of use, reported on the Github page of the released code: "By using this service, users are required to agree to the following terms: The service is a research preview intended for non-commercial use only. It only provides limited safety measures and may generate offensive content. It must not be used for any illegal, harmful, violent, racist, or sexual purposes. The service may collect user dialogue data for future research."

    Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: in Appendix H of the supplemental material, we provide details about the licenses for: *(i)* the large language models used in our approach, *(ii)* the employed source codes and *(iii)* the datasets used in all our experiments.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.

- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

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

    Answer: [Yes]

    Justification: the dataset defined for our benchmark will be made publicly available, in case of acceptance, together with the documentation required for reproducing the experiments. Moreover, in case of acceptance, we will also release the source code of our model.

    Guidelines:

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

14. **Crowdsourcing and Research with Human Subjects**

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

    Answer: [NA]

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

    Guidelines:

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

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

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

    Answer: [NA]

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

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.