Connecting NeRFs, Images, and Text

Francesco Ballerini Pierluigi Zama Ramirez Roberto Mirabella
Samuele Salti Luigi Di Stefano
University of Bologna
https://cvlab-unibo.github.io/clip2nerf

Abstract

Neural Radiance Fields (NeRFs) have emerged as a standard framework for representing 3D scenes and objects, introducing a novel data type for information exchange and storage. Concurrently, significant progress has been made in multimodal representation learning for text and image data. This paper explores a novel research direction that aims to connect the NeRF modality with other modalities, similar to established methodologies for images and text. To this end, we propose a simple framework that exploits pre-trained models for NeRF representations alongside multimodal models for text and image processing. Our framework learns a bidirectional mapping between NeRF embeddings and those obtained from corresponding images and text. This mapping unlocks several novel and useful applications, including NeRF zero-shot classification and NeRF retrieval from images or text.

1. Introduction

In the Neural Radiance Fields (NeRF) framework [27], a neural network is trained to construct a volumetric representation of a 3D environment from images. Once a NeRF is trained, it enables the generation of novel views of that environment through ray tracing. They have gained considerable popularity over recent years [29], emerging as a novel approach for 3D data representation. Representing a scene with a single NeRF decouples the actual memory occupation from the spatial resolution and the number of observations. Indeed, we can encode a hypothetically infinite number of images at arbitrary resolution into a finite number of network weights. This may potentially lead NeRFs to become a standard means of storing and exchanging 3D information, with entire databases of NeRFs residing on our hard drives in the future. Supporting this idea is the recent proliferation of various NeRF datasets [5, 15, 36].

Concurrently with the development of NeRFs, there has been notable progress in the field of Vision-Language Models (VLMs) [11, 34]. These models capitalize on large paired databases of images and text to extract rich multimodal representations. By combining modalities, these representations obtain a better overall comprehension of images and text, allowing for better performance in existing visual and textual tasks. Moreover, they unlocked many novel applications such as zero-shot classification, where unseen instances are classified based on textual descriptions, image retrieval by querying with both text and image prompts, visual question answering, and many others. In the scenario in which NeRFs are an additional input modality, an intriguing research direction is to understand whether and how it is possible to connect NeRFs to other input modalities, as it was for images and text. By bridging NeRFs with diverse input modalities, we might unlock new opportunities for innovative applications.

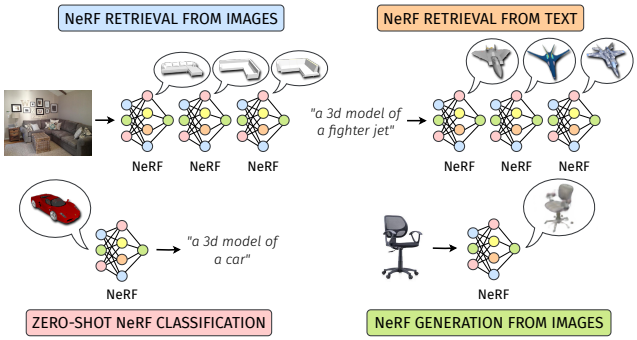

Figure 1. **Framework applications.** Examples of the possible tasks we can perform thanks to our framework that connects NeRFs, images, and text.

Unlike images and text, which are well-studied input formats, NeRFs present unique challenges as they are neural networks, making them less straightforward to process by conventional frameworks. One naive approach would involve rendering images of the object represented by a NeRF. However, this choice presents various challenges, including lengthy computation times, determining a viewpoint from which to render the object, or deciding on an appropriate

rendering resolution. Conversely, we would encounter none of these issues by processing the weights of the NeRF. The problem of how to process NeRF weights to enable downstream tasks has been the focus of the recently proposed framework `nf2vec` [52]. This work learns to encode the information contained within the network weights into a compact embedding that retains sufficient knowledge to be used as input for downstream tasks.

Thanks to the availability of general purpose pre-trained VLMs such as CLIP [34] and pre-trained NeRFs encoders such as `nf2vec` [52], this paper casts the problem of connecting NeRFs, images, and text as learning mapping functions between the latent space of the `nf2vec` encoder and the latent space of CLIP. In practice, we propose training two simple Multi-Layer Perceptrons (MLPs) to map a NeRF embedding into a CLIP embedding or vice versa. In this way, given an image or text input, we can discover the corresponding NeRF or vice versa. Notably, acquiring data for training such networks is straightforward, as we can directly leverage the renderings or ground-truth images of the NeRFs. Moreover, by exploiting a pre-trained model with multimodal text-image embeddings such as CLIP, we naturally learn the connection between NeRF and texts, avoiding the necessity of NeRF-text pairings.

Our framework unlocks many innovative and compelling applications, such as those depicted in Fig. 1. For instance, it is possible to classify NeRFs in a zero-shot manner based solely on their weights, or given one or more images depicting an object, we can retrieve the most closely matching NeRFs. Alternatively, textual queries can be used to search for NeRFs stored in our databases. We can even generate entirely new NeRFs from either images or text.

Despite the simplicity of the architecture, we observe that our framework effectively performs tasks such as NeRF zero-shot classification on par with baselines operating on images obtained from NeRFs without requiring to render even a single pixel. Moreover, leveraging recent text-to-image conditional generative approaches [53], we propose an adaptation technique to apply our method effectively to real images even when trained solely on synthetic data.

Briefly, our contributions are:

- We investigate for the first time the problem of connecting NeRFs with images and text.
- We propose the first framework to achieve this goal. Notably, this method is easy to train as it requires learning only two simple MLPs.
- Our idea unlocks many intriguing applications, such as zero-shot classification of NeRFs by solely processing their network weights and retrieving NeRFs from images or texts.
- We propose a technique to adapt our model to perform well on real images when trained solely on synthetic data.

2. Related work

Vision-Language Models. During the last few years, there has been a rapid advancement in visual-language modeling. The popularity of Vision Transformer (ViT) [8] has led to numerous studies that utilize ViT to simultaneously learn from vision-language data and achieve outstanding performance in downstream tasks [4, 24, 35, 41, 54]. Researchers have proposed efficient pretraining tasks to enhance the alignment between visual and language modalities. Contrastive learning is one of the most prominent methods widely adopted in many studies [3, 12, 19, 34]. Among these methods, CLIP [34] is one of the most popular. Additionally, there are emerging works that explore unified frameworks to address vision-language tasks [25, 39, 44, 45, 47, 48]. Recent works extend multimodal representation learning to other modalities such as audio and videos [11, 46, 49]. Our work employs CLIP to extract rich multimodal embeddings from images and text.

Neural Radiance Fields. NeRF [27] has emerged as a valuable tool for a variety of tasks, including view synthesis [26], generative media [33], robotics [51], and computational photography [28]. Initially, the base NeRF model employed an MLP to translate spatial coordinates into color and density. Recent advancements substitute or enhance MLPs with voxel grid-like data structures [2, 10, 42]. For instance, Instant NGP [30] utilizes a hierarchical arrangement of coarse and fine-grained grids stored using hashmaps. These structures facilitate the extraction of features, which are then processed by a compact MLP, resulting in significantly accelerated training processes. Our work employs NeRFs that follow the base formulation, *i.e.* a single MLP extracting density and color information for each 3D coordinate.

Deep Learning on Neural Networks. Multiple recent studies have delved into using neural networks to process other neural networks. Early works in the field focused on forecasting network properties such as accuracy and hyperparameters directly from their weights [16, 17, 23, 37, 43]. Recent studies handle networks implicitly representing data (INRs or Neural Fields). These works perform vision tasks directly using network weights as the input or output data. Functa [9] learns priors across an entire dataset using a shared network and subsequently encodes each sample into a compact embedding employed for downstream discriminative and generative tasks. The following approaches focus on processing networks representing individual data, *e.g.* a specific object or scene. The first framework doing it was `inr2vec` [6]. This approach encodes networks representing 3D shapes into compact embeddings, serving as input for subsequent tasks. `nf2vec` [36] extends `inr2vec`

to NeRFs, performing several tasks directly from NeRFs weights, such as classification, generation, or retrieval. [1] learns how to process neural fields represented as a hybrid tri-plane representation. Another research direction [31, 55–57], recognizing that MLPs exhibit weight space symmetries [13], proposes innovative architectures tailored for MLPs by leveraging network symmetries as an inductive bias. Other works [18, 21] exploit Graph Neural Networks to learn network representations. To improve the generalization of approaches processing neural networks, [38] explores various strategies for data augmentation directly in weight spaces. Our framework also processes neural networks representing individual objects. In particular, we employ `nf2vec` to extract rich embeddings from NeRFs.

3. Connecting NeRFs and CLIP

Our work aims to learn the connection between image, text, and NeRF [27] modalities. To achieve this goal, given rich multimodal representations extracted by Vision-Language Models such as CLIP [34] and compact embeddings extracted from NeRF weights by `nf2vec` [52], we learn how to map a `nf2vec` embedding into a plausible CLIP embedding and vice versa. In this section, we first report the relevant background knowledge: NeRF, `nf2vec`, and CLIP frameworks. Then, we describe our proposed framework depicted in Fig. 1.

3.1. Preliminaries

NeRF. Given images of a scene or an object, NeRF [27] allows for novel view synthesis from arbitrary vantage points. This is achieved by training a neural network, *i.e.* an MLP, on a set of sparse images collected from different viewpoints. We follow the base NeRF formulation [27] in which a single MLP parameterizes the radiance field of the scene as a function of continuous 3D coordinates in space $\mathbf{x} = (x, y, z)$. Such a function produces a 4D output $RGB\sigma$, encoding the RGB color and volume density σ of each 3D point in the scene. σ can be interpreted as the differential probability of a ray terminating at $\mathbf{x}$. Given a NeRF, we can render an image from an arbitrary viewpoint with a desired resolution through volume rendering [27]. In our paper, NeRFs are considered a standard data format and the input to our framework. We assume that each NeRF encodes a specific object or scene. We wish to avoid sampling any information from the NeRFs, such as rendering views, as it would require vast computational overhead and pose many challenges, such as the choice of the rendering viewpoint. This work aims to extract all information solely by processing MLP weights of the NeRF.

`nf2vec`. `nf2vec` [52] can extract compact embeddings from MLPs that parametrize NeRFs by processing only the network weights. These codes can then be processed using standard deep-learning pipelines to perform tasks such as classification or segmentation. `nf2vec` is a representation learning framework that comprises an encoder and a decoder. The encoder consists of a series of linear layers with batch normalization and ReLU non-linearity followed by a final max pooling. It processes each layer of the input MLP independently, obtaining one vector for each MLP layer. Then, the final max pooling compresses all the layer embeddings into one, obtaining the desired global latent vector representing the input MLP, *i.e.* the input NeRF. The decoder reproduces the original NeRF values given as input the embeddings produced by the encoder and a spatial coordinate $\mathbf{x}$. Our paper utilizes the pre-trained `nf2vec` encoder to embed NeRFs, keeping it frozen.

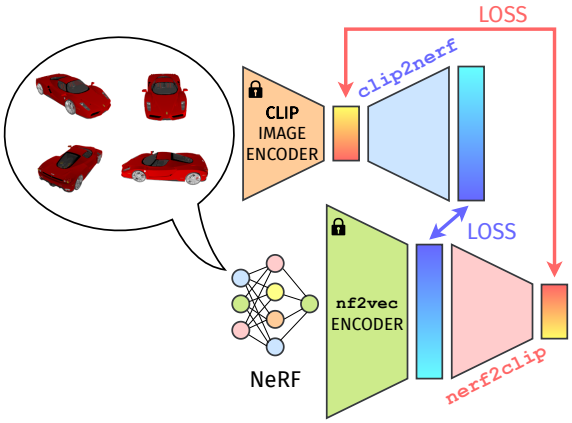

Figure 2. **Feature mapping network training.** `clip2nerf` is a feature mapping network trained to map image embeddings of NeRF views to NeRF embeddings. Conversely, `nerf2clip` computes the mapping in the opposite direction.

CLIP. CLIP (Contrastive Language-Image Pre-training) [34] is a pioneering visual language representation model. The CLIP architecture consists of an image and a text encoder such as ViT [8] and BERT [7], respectively. CLIP is trained using a contrastive learning objective on a large set of data, which encourages the model to assign similar embeddings to semantically related image-text pairs while maximizing the dissimilarity between embeddings of unrelated pairs. This procedure enforces a multimodal vision-language latent space in which images and corresponding textual prompts share the same embedding. In our framework, we employ pre-trained and frozen CLIP encoders.

3.2. Feature mapping networks

Architecture. To map CLIP embeddings to `nf2vec` embeddings, we use a simple MLP with GELU [14] activation function and layer dimensions $512 \rightarrow 768 \rightarrow 1024$, where 512 and 1024 are the sizes of CLIP embeddings and

nf2vec embeddings, respectively. We call this feature mapping network clip2nerf. Similarly, we use another MLP, dubbed nerf2clip, with GELU activation function and layer dimensions $1024 \rightarrow 768 \rightarrow 512$ to compute the mapping in the opposite direction.

Training. Given a NeRF and n views of an object, we extract the nf2vec embedding $\mathbf{v}$ of the NeRF and n CLIP embeddings $\mathbf{c}_i$ of its views. For each view embedding $\mathbf{c}_i$, we train clip2nerf to maximize the cosine similarity between its output 1024-dimensional vector $\hat{\mathbf{v}}_i$ and the embedding $\mathbf{v}$ of that NeRF. Formally, the clip2nerf loss for an object is:

$$\mathcal{L}_{\texttt{clip2nerf}} = \frac{1}{n}\sum_{i=1}^{n}\left(1 - \frac{\hat{\mathbf{v}}_i \cdot \mathbf{v}}{\|\hat{\mathbf{v}}_i\|\|\mathbf{v}\|}\right)$$

Instead, we train nerf2clip to map a NeRF embedding $\mathbf{v}$ to the *mean* embedding of the n views, $\mathbf{c} = \frac{1}{n}\sum_{i=1}^{n}\mathbf{c}_i$, as learning to map a $\mathbf{v}$ to every $\mathbf{c}_i$ would create a one-to-many correspondence, *i.e.* not a function. Specifically, we maximize the cosine similarity between the nerf2clip 512-dimensional output $\hat{\mathbf{c}}$ and $\mathbf{c}$. Formally:

$$\mathcal{L}_{\texttt{nerf2clip}} = 1 - \frac{\hat{\mathbf{c}} \cdot \mathbf{c}}{\|\hat{\mathbf{c}}\|\|\mathbf{c}\|}$$

During training, the n views can be the ground-truth images used to train the NeRF or those rendered from it.

4. Experimental Settings

NeRF framework and dataset. We use the NeRF [27] dataset of nf2vec [52]. The NeRF architecture consists of an MLP with ReLU activation function and 4 hidden layers with 256 features each. It applies a frequency encoding [27] $\text{enc}(\mathbf{x})$ to each input 3D coordinate $\mathbf{x}$. Each NeRF is trained on $N = 36$ views of a shape of the ShapeNetRender dataset [50]. The dataset consists of a NeRF for each ShapeNetRender shape, for a total of 38653 NeRFs, which we split into training (30946), validation (3847), and test (3860) sets. For each NeRF, we have access to the 36 synthetic images used for training and their depth maps.

Metrics. We evaluate our retrieval experiments (Sec. 6) with the recall@k metric, *i.e.* the percentage of queries q such that at least one among the first k neighbors of q share the same label as q. The top-k nearest neighbors of q are those with the highest cosine similarity with q, sorted from closest to furthest. We call them 1-NN, 2-NN, ..., k-NN. Our classification experiments use the standard multi-class accuracy (Sec. 5).

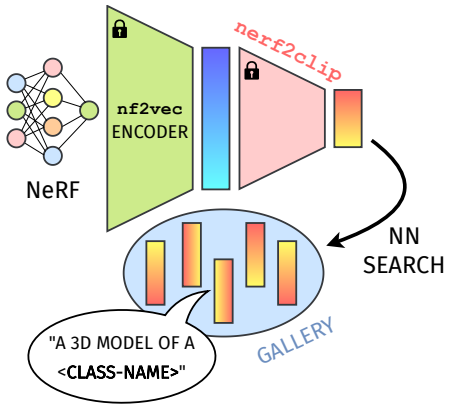

Figure 3. **Zero-shot NeRF classification method overview.**

Training details. Our feature mapping networks clip2nerf and nerf2clip are trained for 150 and 100 epochs, with learning rates 10^{-5} and 10^{-3}, respectively. Both are trained with the AdamW optimizer [22], one-cycle learning rate scheduler [40], weight decay 10^{-2}, and batch size 64. We perform all our experiments on a single NVIDIA RTX 3090 GPU.

5. Zero-shot NeRF classification

To perform zero-shot NeRF [27] classification, we build a gallery of CLIP [34] embeddings of sentences of form "A 3D model of <class-name>", where <class-name>" denotes a class label of ShapeNetRender [50]. In other words, the gallery contains one CLIP embedding for each ShapeNetRender class. We then take a test-set NeRF, encode it with the nf2vec [52] encoder, process the results with nerf2clip, and use the output embedding to query the gallery. Finally, the predicted label corresponds to the text of the 1-NN. This procedure is illustrated in Fig. 3. As in our retrieval experiments (Sec. 6), the 1-NN maximizes the cosine similarity with the query. Tab. 1 shows classification results for two variants of nerf2clip, one trained with ground-truth views ("nerf2clip GT" row) and the other with views rendered from the corresponding NeRF ("nerf2clip rendered" row). As an ablation, Tab. 2 shows the effect of training nerf2clip with an increasing number of views, *i.e.* training nerf2clip to map a NeRF embedding to the mean CLIP embedding of n NeRF views, where $n = 1, 2 \ldots N$. The model with the highest accuracy in Tab. 2 is the one reported in Tab. 1.

As a baseline to compare nerf2clip with, we query our gallery of CLIP class-label embeddings with the mean of the CLIP embeddings of n random rendered views of a test-set NeRF, where $n = 1, 2 \ldots N$. We do not use ground-truth views as queries, as we assume they are only available at training time. The results are shown in Tab. 1 ("CLIP" rows). nerf2clip, in both its versions, achieves higher

Method	Supervision	Rendering	Accuracy (%) ↑	Time (ms) ↓
CLIP 1 view	✗	✓	73.6	13
CLIP 2 views			77.7	25
CLIP 4 views			80.5	49
CLIP 8 views			81.7	97
CLIP 16 views			82.4	193
CLIP N views			82.4	433
`nerf2clip` rendered (ours)	✗	✗	83.0	**2**
`nerf2clip` GT (ours)			**84.0**	
`nf2vec` (oracle)	✓	✗	87.3	1

Table 1. **Zero-shot NeRF classification results.**

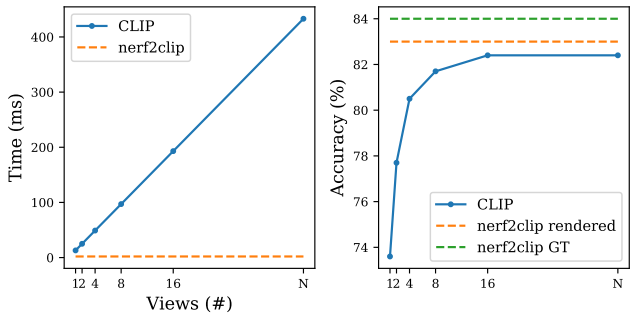

Figure 4. **Zero-shot NeRF classification results (plots).** Classification time and accuracy as a function of the number of views used for the query.

Method	Views	Accuracy (%) ↑
`nerf2clip` rendered	1	80.6
	2	81.4
	4	82.4
	8	82.3
	16	**83.0**
	N	82.7
`nerf2clip` GT	1	82.9
	2	83.2
	4	**84.0**
	8	83.8
	16	83.6
	N	82.7

Table 2. **`nerf2clip` training ablation.** Effect of the number of views used for the feature mapping network training on the classification accuracy.

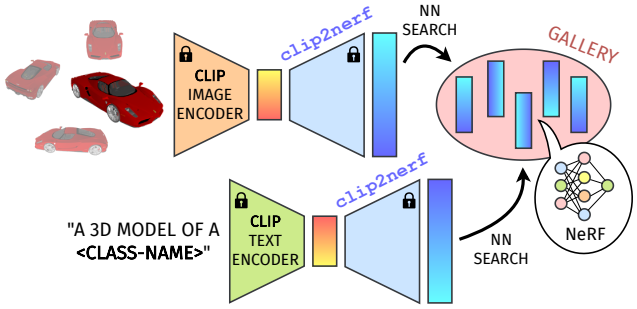

Figure 5. **NeRF retrieval method overview.** NeRF retrieval from images (top) and text (bottom).

accuracy than the baseline, which plateaus at 16 views. For comparison, we also report the classification accuracy of `nf2vec`. It is important to note, however, that `nf2vec` performs classification in a supervised manner, and therefore its accuracy should be regarded as an upper bound to our zero-shot classification results.

Furthermore, Tab. 1 compares the time required to perform zero-shot classification with `nerf2clip` vs. the CLIP baseline. For the baselines, we report the sum of view rendering time, CLIP inference time, and NN search time. The rendering and the CLIP inference times scale linearly with the number of views. For `nerf2clip`, we report the sum of `nf2vec` encoder inference time, `nerf2clip` inference time, and NN search time. Our method is even faster than the CLIP baseline using only 1 view.

The results of Tab. 1 are also depicted in Fig. 4, where we highlight the fact that, while time and accuracy are a function of the number of views used to query the gallery for the CLIP baseline, they are a constant for `nerf2clip` at inference time, as the query is the output of the trained model, thus requiring no rendering nor CLIP inference. Furthermore, as already pointed out, `nerf2clip` achieves the highest accuracy and the fastest inference time.

6. NeRF Retrieval

In the NeRF [27] retrieval application, we aim to identify NeRFs in a database that closely matches a given textual or image query. To achieve this, we first construct offline a gallery of NeRF embeddings obtained by `nf2vec` [52]. Then, we employ the image or text CLIP [34] encoders to generate the corresponding embedding during retrieval. This is processed by `clip2nerf`, yielding the predicted NeRF embedding. Finally, we employ the NN search to locate the closest embedding within the gallery. This procedure is illustrated in Fig. 5.

6.1. NeRF retrieval from images

The experiments reported here address the scenario in which a user takes one or more pictures of an object (*e.g.* a product in a store) and uses them to query a database of NeRF objects (*e.g.* the online store catalog). Thus, we always employ real or synthetic images as queries, *i.e.* no rendered images from NeRFs. In the experiments of this section, we build the gallery with the NeRF embeddings obtained by `nf2vec` on the test set of ShapeNetRender [50]. We exclude the queried NeRF from the gallery during retrieval.

Single-view query. In Tab. 3, we report the results using a single image as query (a random GT view of the test set for each object). We use the same random images across dif-

Method	Recall (%) ↑ @1	@5	@10	Time (ms) ↓	Memory (MB) ↓
CLIP [34] GT mean	81.4	93.9	**96.3**	24	8
CLIP [34] rendered mean	81.2	92.9	96.1		
CLIP [34] GT all	83.6	93.0	95.1	331	271
CLIP [34] rendered all	81.6	91.7	95.1		
`clip2nerf` GT (ours)	**86.1**	**94.0**	96.0	25	15
`clip2nerf` rendered (ours)	84.5	93.3	95.4		

Table 3. **NeRF retrieval from images (single-view query results).**

Figure 6. **Qualitative results of NeRF retrieval from images on ShapeNetRender [50].** For each NeRF, we visualize one view rendered from a vantage point.

ferent table rows to compare methods fairly. We report the results of our `clip2nerf` method, trained using either the ground-truth images (`clip2nerf` GT) or the rendered images from the NeRFs (`clip2nerf` rendered). In this way, we simulate the two possible scenarios in which the images used to train the NeRFs are available or not at training time. Moreover, we report the performance of four plausible baseline strategies that exploit different galleries built using CLIP embeddings from images. "CLIP GT mean": each gallery element is the average of $N = 36$ CLIP embeddings obtained from N object views. "CLIP rendered mean": the same as the previous one, yet we employ N rendered images from NeRF from the same N viewpoints. "CLIP GT all": for each object in the test set, we store N CLIP embeddings in the gallery, one for each ground-truth

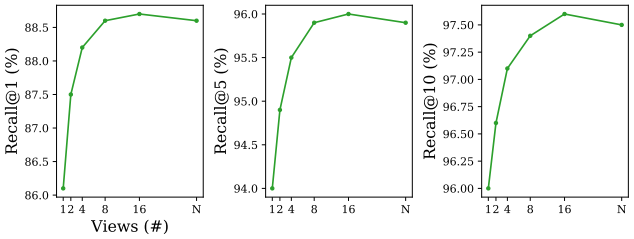

Figure 7. **NeRF retrieval from images (multi-view query results).** Performance of our `clip2nerf` feature mapping network as a function of the number of views used for the query.

Figure 8. **ControlNet generated views.** ShapeNetRender [50] views vs. their counterparts generated by ControlNet [53].

view. "CLIP rendered all": the same as the previous one, yet it employs the rendered views from NeRF instead of ground-truth images. As shown in Tab. 3, our model exhibits superior performance in terms of recall@1 compared to the baselines while maintaining comparable results in the other metrics. Finally, we show some retrieval examples in Fig. 6. We render a single reference view to visualize NeRFs. As we can see, the retrieved NeRFs belong to the same object class and resemble the input image in color and shape.

Multi-view query. We focus here on the case where a user can acquire multiple pictures of the same object. In the plots of Fig. 7, we show the retrieval recall@1, recall@5, and recall@10 results when varying the number of query images used for each object. The gallery is the same as in the single-view scenario, *i.e.* a NeRF embedding for each object. The query views are selected randomly among the ground-truth images in ShapeNetRender. We randomly choose only the additional views when increasing the number of queries. For instance, the experiment with 8 views includes the images used in the 4 views results. To retrieve the NeRF, we pass the n multi-view queries to the CLIP image encoder, obtaining n embeddings. Each is processed by `clip2nerf` (the model named `clip2nerf` GT in Tab. 3), and the resulting NeRF embeddings are averaged to get a reference embedding for that object. Then, we perform the NN search within the gallery. Interestingly, when the number of query images used for retrieval increases, the results improve significantly until 8 views, at which point

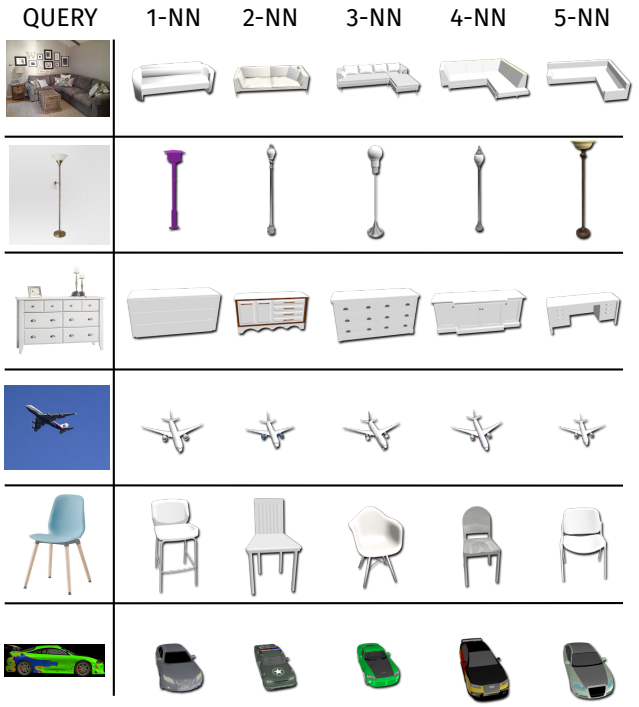

Figure 9. **Qualitative results of NeRF retrieval from real images.** Queries are real images from DomainNet [32]. For each NeRF, we visualize one view rendered from a vantage point.

Method	Recall (%) ↑		
	@1	@5	@10
CLIP [34] GT	75.5	**90.4**	93.7
CLIP [34] rendered	73.9	89.8	**93.8**
`clip2nerf` GT (ours)	67.9	80.7	85.6
`clip2nerf` GT syn2real (ours)	**79.9**	87.4	90.1

Table 4. **NeRF retrieval from real images (adaptation results).** Gallery of NeRFs from ShapeNetRender [50]. Queries from DomainNet [32].

performance plateaus. Thus, we can conclude that the information provided by the additional views can be valuable for retrieval.

Adaptation to real images. In the previous retrieval experiments, we employed solely synthetic query images. However, in a practical scenario, we would use real images acquired in the wild. Thus, we evaluated `clip2nerf` using the real split of the DomainNet [32] dataset, reporting results in Tab. 4. The gallery consists of the NeRF embeddings from ShapeNetRender. We note a performance drop compared to the case of testing on synthetic images, probably due to the domain-shift problem. Thus, we propose

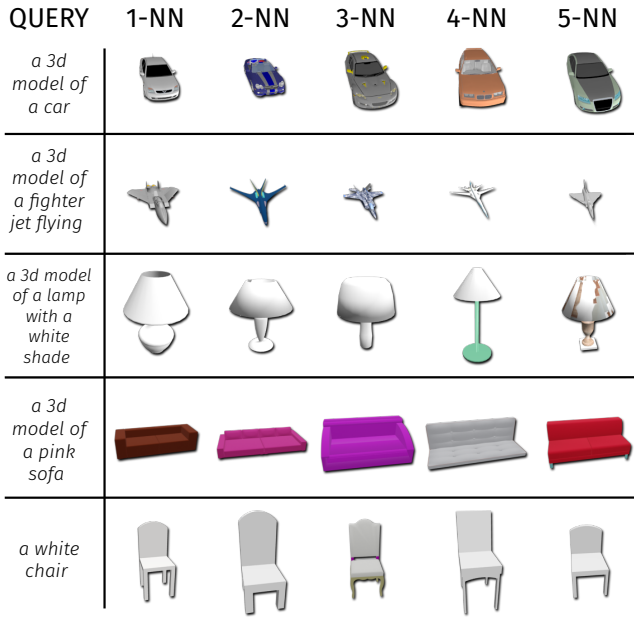

Figure 10. **Qualitative results of NeRF retrieval from text on ShapeNetRender [50].** For each NeRF, we visualize one view rendered from a vantage point.

Method	Recall (%) ↑		
	@1	@5	@10
CLIP GT	80.0	89.6	92.8
CLIP rendered	79.6	89.5	92.0
`clip2nerf` GT (ours)	63.2	75.2	79.0
`clip2nerf` GT multimodal (ours)	**85.6**	**91.7**	**93.3**

Table 5. **NeRF retrieval from text results.**

an adaptation protocol based on the recent diffusion-based conditional generative approach, ControlNet [53]. In particular, we generate a new *synthetic to real* datasets with ControlNet, using the synthetic object depth map as input to the generative network (see Fig. 8). These generated images are added to the synthetic ones to train `clip2nerf`. The augmented dataset contains 7 synthetic random views and 7 images generated by ControlNet for each object. By training on this dataset, we learned a feature mapping that can be applied effectively to real images. We report the performance of this network in the last row of Tab. 4, and we can observe a remarkable performance improvement w.r.t. the network trained without augmented data. Moreover, our method performs comparably to the baseline using the CLIP galleries obtained from images (rows 1 and 2 vs. 4). Finally, in Sec. 6.1, we show retrieval results using in-the-wild real queries. Remarkably, the retrieved NeRF resembles the geometry of the input image with high fidelity.

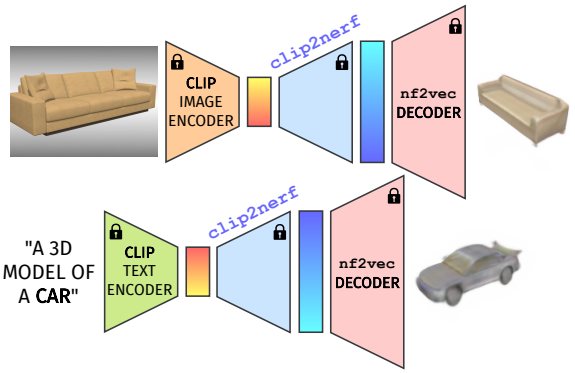

Figure 11. **NeRF generation method overview.** NeRF generation from images (top) and text (bottom).

6.2. NeRF retrieval from text

We experiment here with the retrieval of NeRFs from text. In this scenario, given a text prompt, we want to find the corresponding NeRF in our database.

We employ the same gallery of the single-view scenario of NeRF retrieval from images. To obtain a reasonable query text for the input images, we use the BLIP-2 [20] captioner. We report results on ShapeNetRender in Tab. 5. Our `clip2nerf` obtains lower performance than the baselines using the CLIP galleries. We relate this to the feature mapping function, which was never trained on the CLIP text embeddings. For this reason, we train a variant of our method using as input to `clip2nerf` either the CLIP embeddings obtained from an image or the CLIP embedding of an automatically generated caption with BLIP-2. As shown in Tab. 5, this multimodal training paradigm can even surpass the baselines by a moderate margin (row 1 and 2 vs. row 4).

Finally, we also visualize some qualitative results in Fig. 10. We note that we can retrieve NeRFs of the class described in the text, which contains details presented in the textual prompt, *e.g.* we correctly retrieve NeRFs of a jet fighter in the second row.

7. NeRF generation

Generation from images. Another application of our approach consists in, given a NeRF [27] view, *synthesize* new views of the object by leveraging the `nf2vec` [52] decoder. Specifically, this procedure works as follows: we embed the NeRF view with the CLIP [34] image encoder and give the resulting embedding as input to the trained `clip2nerf` network, which produces a NeRF embedding. The latter can be processed by the `nf2vec` decoder to render arbitrary views of the object. Thus, the embedding plus the decoder can be considered a NeRF architecture. This procedure is illustrated in Fig. 11 (top). Qualitative results are shown in Fig. 12, both with images from ShapeNetRender [50] (top) and real images from DomainNet [32] (bottom).

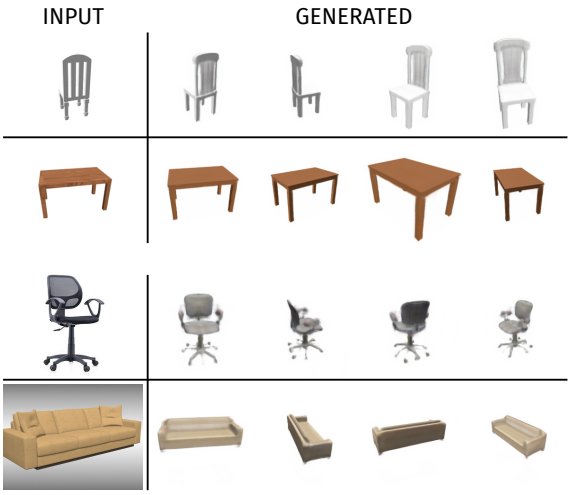

Figure 12. **Qualitative results of NeRF generation from images.** Synthetic images from ShapeNetRender [50] (top) and real images from DomainNet [32] (bottom).

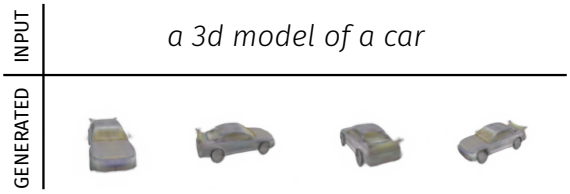

Figure 13. **Qualitative results of NeRF generation from text.**

Generation from text. Analogously, our framework allows to synthesize new NeRF views from text. Given the previous generation pipeline, we replace the CLIP image encoder with the CLIP text encoder (Fig. 11 bottom). Qualitative results are shown in Fig. 13.

8. Limitations and Conclusions

Our proposed framework effectively connects NeRF, images, and text. We have demonstrated its application in several novel tasks, including NeRF retrieval or generation from text and images, as well as zero-shot classification of NeRFs using only network weights.

However, our framework has its limitations. Firstly, the `nf2vec` encoder, trained on ShapeNetRender, limits our experiments to NeRFs of synthetic objects only. Additionally, the NeRF generation is constrained by the processing capabilities of the `nf2vec` decoder.

In the future, expanding our work to include NeRFs of real objects or scenes would be valuable. Learning a shared latent space for NeRFs, images, and text, *e.g.* by jointly training the vision, language, and NeRF encoders on larger datasets, could also be a promising direction. We plan to address these limitations by exploring these ideas in future studies and hope that our framework inspires further advancements in the field.

References

[1] 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. 3

[2] Anpei Chen, Zexiang Xu, Andreas Geiger, Jingyi Yu, and Hao Su. Tensorf: Tensorial radiance fields. In *European Conference on Computer Vision (ECCV)*, 2022. 2

[3] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020. 2

[4] Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. Uniter: Universal image-text representation learning. In *European conference on computer vision*, pages 104–120. Springer, 2020. 2

[5] Luca De Luigi, Damiano Bolognini, Federico Domeniconi, Daniele De Gregorio, Matteo Poggi, and Luigi Di Stefano. Scannerf: a scalable benchmark for neural radiance fields. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 816–825, 2023. 1

[6] 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. 2

[7] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 3

[8] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 2, 3

[9] 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. 2

[10] 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. 2

[11] 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. 1, 2

[12] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738, 2020. 2

[13] Robert Hecht-Nielsen. On the algebraic structure of feedforward network weight spaces. In *Advanced Neural Computers*, pages 129–135. Elsevier, 1990. 3

[14] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016. 3

[15] 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. 1

[16] Florian Jaeckle and M Pawan Kumar. Generating adversarial examples with graph neural networks. In *Uncertainty in Artificial Intelligence*, pages 1556–1564. PMLR, 2021. 2

[17] Boris Knyazev, Michal Drozdzal, Graham W. Taylor, and Adriana Romero. Parameter prediction for unseen deep architectures. In *Advances in Neural Information Processing Systems*, 2021. 2

[18] 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*, 2023. 3

[19] Junnan Li, Ramprasaath Selvaraju, Akhilesh Gotmare, Shafiq Joty, Caiming Xiong, and Steven Chu Hong Hoi. Align before fuse: Vision and language representation learning with momentum distillation. *Advances in neural information processing systems*, 34:9694–9705, 2021. 2

[20] 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. 8

[21] 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. 3

[22] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 4

[23] Jingyue Lu and M. Pawan Kumar. Neural network branching for neural network verification. In *International Conference on Learning Representations*, 2020. 2

[24] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. *Advances in neural information processing systems*, 32, 2019. 2

[25] Jiasen Lu, Christopher Clark, Rowan Zellers, Roozbeh Mottaghi, and Aniruddha Kembhavi. Unified-io: A unified model for vision, language, and multi-modal tasks. In *The Eleventh International Conference on Learning Representations*, 2022. 2

[26] Ricardo Martin-Brualla, Noha Radwan, Mehdi SM Sajjadi, Jonathan T Barron, Alexey Dosovitskiy, and Daniel Duck-

worth. 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. 2

[27] 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. 1, 2, 3, 4, 5, 8

[28] Ben Mildenhall, Peter Hedman, Ricardo Martin-Brualla, Pratul P Srinivasan, and Jonathan T Barron. Nerf in the dark: High dynamic range view synthesis from noisy raw images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16190–16199, 2022. 2

[29] Ansh Mittal. Neural radiance fields: Past, present, and future. *arXiv preprint arXiv:2304.10050*, 2023. 1

[30] 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. 2

[31] 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. 3

[32] Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1406–1415, 2019. 7, 8

[33] 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. 2

[34] 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. 1, 2, 3, 4, 5, 6, 7, 8

[35] 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. 2

[36] 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. 1, 2

[37] 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. 2

[38] Aviv Shamsian, Aviv Navon, David W Zhang, Yan Zhang, Ethan Fetaya, Gal Chechik, and Haggai Maron. Improved generalization of weight space networks via augmentations. *arXiv preprint arXiv:2402.04081*, 2024. 3

[39] Amanpreet Singh, Ronghang Hu, Vedanuj Goswami, Guillaume Couairon, Wojciech Galuba, Marcus Rohrbach, and Douwe Kiela. Flava: A foundational language and vision alignment model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15638–15650, 2022. 2

[40] Leslie N Smith and Nicholay Topin. Super-convergence: Very fast training of neural networks using large learning rates. In *Artificial intelligence and machine learning for multi-domain operations applications*, page 1100612. International Society for Optics and Photonics, 2019. 4

[41] Weijie Su, Xizhou Zhu, Yue Cao, Bin Li, Lewei Lu, Furu Wei, and Jifeng Dai. Vl-bert: Pre-training of generic visual-linguistic representations. *arXiv preprint arXiv:1908.08530*, 2019. 2

[42] 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. 2

[43] Thomas Unterthiner, Daniel Keysers, Sylvain Gelly, Olivier Bousquet, and Ilya O. Tolstikhin. Predicting neural network accuracy from weights. *arXiv*, abs/2002.11448, 2020. 2

[44] Jianfeng Wang, Zhengyuan Yang, Xiaowei Hu, Linjie Li, Kevin Lin, Zhe Gan, Zicheng Liu, Ce Liu, and Lijuan Wang. Git: A generative image-to-text transformer for vision and language. *Transactions on Machine Learning Research*, 2022. 2

[45] Peng Wang, An Yang, Rui Men, Junyang Lin, Shuai Bai, Zhikang Li, Jianxin Ma, Chang Zhou, Jingren Zhou, and Hongxia Yang. Ofa: Unifying architectures, tasks, and modalities through a simple sequence-to-sequence learning framework. In *International Conference on Machine Learning*, pages 23318–23340. PMLR, 2022. 2

[46] Peng Wang, Shijie Wang, Junyang Lin, Shuai Bai, Xiaohuan Zhou, Jingren Zhou, Xinggang Wang, and Chang Zhou. One-peace: Exploring one general representation model toward unlimited modalities. *arXiv preprint arXiv:2305.11172*, 2023. 2

[47] Wenhui Wang, Hangbo Bao, Li Dong, Johan Bjorck, Zhiliang Peng, Qiang Liu, Kriti Aggarwal, Owais Khan Mohammed, Saksham Singhal, Subhojit Som, et al. Image as a foreign language: Beit pretraining for vision and vision-language tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19175–19186, 2023. 2

[48] Zirui Wang, Jiahui Yu, Adams Wei Yu, Zihang Dai, Yulia Tsvetkov, and Yuan Cao. Simvlm: Simple visual language model pretraining with weak supervision. In *International Conference on Learning Representations*, 2021. 2

[49] Shengqiong Wu, Hao Fei, Leigang Qu, Wei Ji, and Tat-Seng Chua. NExt-GPT: Any-to-any multimodal LLM, 2024. 2

[50] 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. 4, 5, 6, 7, 8

[51] 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. 2

[52] 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. 2, 3, 4, 5, 8

[53] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3836–3847, 2023. 2, 6, 7

[54] Pengchuan Zhang, Xiujun Li, Xiaowei Hu, Jianwei Yang, Lei Zhang, Lijuan Wang, Yejin Choi, and Jianfeng Gao. Vinvl: Revisiting visual representations in vision-language models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5579–5588, 2021. 2

[55] 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. 3

[56] 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.

[57] Allan Zhou, Chelsea Finn, and James Harrison. Universal neural functionals. *arXiv preprint arXiv:2402.05232*, 2024. 3