LEAP: LIBERATE SPARSE-VIEW 3D MODELING FROM CAMERA POSES

Hanwen Jiang **Zhenyu Jiang** **Yue Zhao** **Qixing Huang**
Department of Computer Sciences, University of Texas at Austin
Project page: https://hwjiang1510.github.io/LEAP/

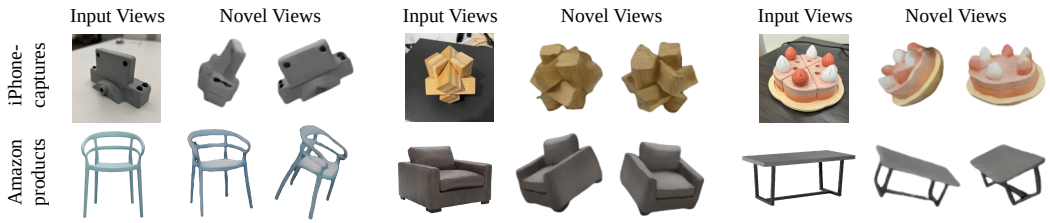

Figure 1: LEAP performs 3D modeling from **sparse views without camera pose information**. We show the capability of LEAP on real-world cases with **three unposed image inputs**. We show one of the inputs.

ABSTRACT

Are camera poses necessary for multi-view 3D modeling? Existing approaches predominantly assume access to accurate camera poses. While this assumption might hold for dense views, accurately estimating camera poses for sparse views is often elusive. Our analysis reveals that noisy estimated poses lead to degraded performance for existing sparse-view 3D modeling methods. To address this issue, we present LEAP, a novel *pose-free* approach, therefore challenging the prevailing notion that camera poses are indispensable. LEAP discards pose-based operations and learns geometric knowledge from data. LEAP is equipped with a neural volume, which is shared across scenes and is parameterized to encode geometry and texture priors. For each incoming scene, we update the neural volume by aggregating 2D image features in a feature-similarity-driven manner. The updated neural volume is decoded into the radiance field, enabling novel view synthesis from any viewpoint. On both object-centric and bounded scene-level datasets, we show that LEAP significantly outperforms prior methods when they employ predicted poses from state-of-the-art pose estimators. Notably, LEAP performs on par with prior approaches that use ground-truth poses while running $400\times$ faster than PixelNeRF. We show LEAP generalizes to novel object categories and scenes, and learns knowledge closely resembles epipolar geometry.

1 INTRODUCTION

In 3D vision, camera poses offer powerful explicit geometric priors to connect 3D points and 2D pixels (Zisserman, 2001). Its effectiveness has been verified across a spectrum of 3D vision tasks (Goesele et al., 2006; Geiger et al., 2011), enabling high-quality 3D modeling (Mildenhall et al., 2020; Wang et al., 2021a). However, accurate camera poses are not always available in the real world, and inaccurate poses lead to degraded performance (Lin et al., 2021). To obtain accurate camera poses, one solution is capturing *dense* views and applying structure-from-motion techniques (Schönberger & Frahm, 2016). Nevertheless, in real-world scenarios, like product images in online stores, we usually observe *sparse* images captured by wide-baseline cameras. For sparse views, estimating accurate camera poses is still challenging (Zhang et al., 2022a). Then a question arises: is using noisy estimated camera poses still the best choice for **3D modeling from sparse and unposed views**?

In this paper, we present LEAP, which champions a *pose-free* paradigm. Instead of pursuing a more accurate camera pose estimator, LEAP challenges the prevailing notion that camera poses are indispensable for 3D modeling. LEAP abandons any operations that explicitly use camera poses, e.g. projection, and learns the pose-related geometric knowledge/representations from data. Thus, LEAP is entirely liberated from camera pose errors during inference, leading to better performance.

LEAP specifically represents each scene as a neural radiance field, which is predicted in a single feed-forward step. To initialize the radiance field, we introduce a neural volume, which is shared across all scenes. Each voxel grid of the volume is parameterized to learn the geometry and texture priors from data. For any incoming scene, the neural volume queries input 2D image features and gets updated through aggregation. Instead of using camera poses to identify source 2D pixels to aggregate (Yu et al., 2021), LEAP leverages attention to aggregate all 2D image features with adaptive weights. Subsequently, LEAP performs spatial-aware attention on the updated neural volume to capture long-range geometry dependency. We iterate the process of aggregating and 3D reasoning, resulting in a refined neural volume. The refined neural volume is then decoded into the radiance field.

An important issue is which reference 3D coordinate frame we should use to define the neural volume. A good choice of this 3D coordinate frame can significantly stabilize and enhance learning (Qi et al., 2017; Deng et al., 2021). As the world coordinate frame for an unposed image set is not well-defined, we instead use a local coordinate frame. Specifically, we choose an arbitrary input image as the canonical view, and the neural volume is defined in its corresponding local camera coordinate. The camera pose of the canonical view is fixed, e.g. as an identity pose, in the local camera coordinate frame. To enable the model aware of the choice of canonical view, we find the key is making 2D image features of non-canonical views consistent with the canonical view. Thus, we design a multi-view encoder to improve the consistency by capturing cross-view 2D correlations.

During training, the canonical view is randomized among all input views. We train LEAP with 2D rendering loss of the input views, using ground-truth camera poses to render them. Note that these ground-truth camera poses are only used during training to learn the mapping from input images to the neural volume. During inference, LEAP predicts the radiance field without reliance on any poses.

We perform a thorough evaluation of LEAP on a diverse array of object-centric (Wu et al., 2023; Jiang et al., 2022; Deitke et al., 2022) and scene-level (Jensen et al., 2014) datasets. This assessment spans multiple data scales and incorporates both synthetic and real images. Experimental results highlight LEAP's four interesting properties: i) **Superior performance**. LEAP consistently synthesizes novel views from $\mathbf{2} \sim \mathbf{5}$ unposed images. It surpasses prior generalizable NeRFs when they use camera poses predicted by SOTA pose estimators. It performs on par with methods using ground-truth camera poses. ii) **Fast inference speed**. LEAP constructs the radiance field in a feed-forward manner without optimization, running within one second on a single consumer-grade GPU. iii) **Strong generalization capability**. LEAP models novel-category objects accurately. The model learned on large object-centric datasets transfer well to the scene-level DTU dataset. iv) **Interpretable learned priors**. While LEAP does not explicitly use camera poses by design, it acquires priors consistent with epipolar geometry. We are committed to *releasing code* for reproducibility and future research.

2 Related Work

NeRF from sparse views with ground-truth camera poses. NeRF variants that work on sparse view inputs can be categorized into two genres. The first is **scene-specific NeRFs**. Following the original NeRF setting, these methods optimize the radiance field for each scene from scratch. They use additional information to regularize NeRF optimization, e.g., normalization flow (Niemeyer et al., 2022) and frequency regularization (Yang et al., 2023). The second is **generalizable NeRF variants** (Yu et al., 2021; Wang et al., 2021b; Chen et al., 2021a), which predict the radiance field in a feed-forward manner. The key is making the radiance fields conditioned on the 2D image features. Typically, these approaches project the 3D points on the input images using camera poses, and information is aggregated from the image features at projected 2D locations. Thus, they are generalizable and transferable to novel scenes by training on curated datasets. However, these methods lack reasoning of correlations between 3D points and assume the access of ground-truth camera poses. In contrast, LEAP has 3D reasoning ability, which works on images without poses.

Sparse-view camera pose estimation. Estimating the camera poses from sparse views presents a significantly greater challenge than from dense views. The complexity arises from the minimal or absent overlap between images, which hampers the formation of cross-view correspondence cues, vital for accurate camera pose estimation (Zisserman, 2001). RelPose (Zhang et al., 2022a) highlights the limitations of conventional dense-based camera pose estimation techniques, e.g., COLMAP (Schönberger & Frahm, 2016), in sparse view contexts. In response, it introduces an energy-based model to handle multi-modal solutions indicative of potential camera poses. A subsequent method (Lin et al., 2023) further develops this approach by harnessing multi-view information to

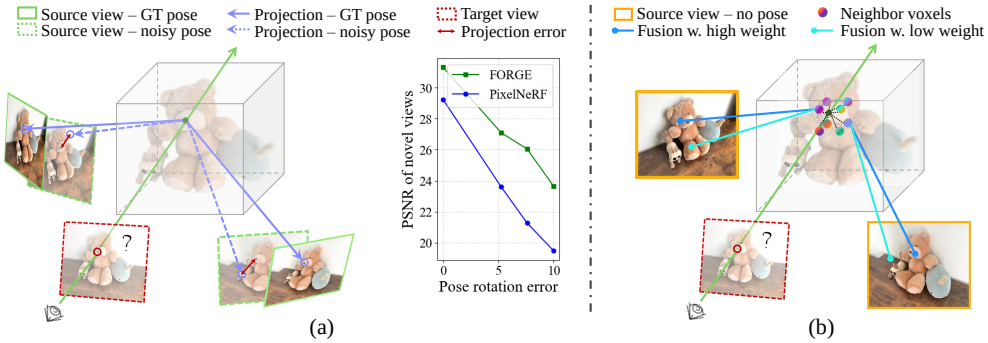

Figure 2: **Comparing LEAP with pose-based generalizable NeRFs (Yu et al., 2021).** (a) For any 3D point along the casting ray from a target view, pose-based methods project these 3D points onto input source views and aggregate 2D image features at these projected locations. These methods are sensitive to slight pose inaccuracies, as the errors cause misaligned 3D points and 2D pixels. (b) In contrast, LEAP offers a pose-free method. It defines a parametrized 3D neural volume to encapsulate geometry and texture priors. For any scene, each voxel grid aggregates information from all 2D image pixel, and their feature similarity determines the fusion weight. Features for an inquired 3D point are interpolated from the neighboring voxel grids. This sidestep of pose-dependent operations allows for direct inference on unposed images.

enhance pose estimation accuracy. Concurrently, SparsePose employs a pre-trained foundational model, namely DINO (Caron et al., 2021), to iteratively refine the predictions of noisy camera poses. Besides, researchers also have explored directional representation (Chen et al., 2021b), stronger image matching techniques (Sun et al., 2021) or using image matching priors (Rockwell et al., 2022), and co-visibility (Hutchcroft et al., 2022) to improve the performance. In contrast, our LEAP operates without dedicated camera pose estimation modules.

NeRF with imperfect or no camera poses. Building NeRF from images without precise camera poses is challenging, given that many NeRF variants rely on pose-based geometric operations. To tackle this problem, scene-specific NeRFs (Lin et al., 2021; Wang et al., 2021c; Xia et al., 2022; Bian et al., 2022; Zhang et al., 2022b; Meng et al., 2021) treat camera poses as modifiable parameters, jointly optimizing them alongside the radiance field. Yet, these methods require dense views, and they either require reasonably accurate initial poses or assume small-baseline cameras to work. SPARF (Truong et al., 2022) leverages dense 2D image correspondences derived from existing models to augment radiance field optimization. Nevertheless, its efficacy heavily hinges on the precision of both dense correspondences and initial poses. For generalizable NeRF variants, SRT (Sajjadi et al., 2022b) proposes a pose-free paradigm building a 2D latent scene representation, but SRT is not 3D-aware, and its novel view synthesis quality is limited. RUST further deals with the necessity of target camera pose by prompting the model with a partial target image (Sajjadi et al., 2022a). FORGE (Jiang et al., 2022) jointly estimates camera poses and predicts the radiance field, leveraging their synergy to improve the performance of both. However, the performance is sensitive to pose estimation precision and training FORGE in multi-stages is non-trivial. In contrast, our proposed LEAP benefits from the 3D-aware designs and leans into a feature-similarity-driven 2D-3D information mapping. This approach eliminates reliance on camera poses during inference, yielding results more closely aligned with using ground-truth poses.

3 Overview

We focus on novel view synthesis from **sparse views** of a scene **without camera poses**. Prior approaches have adjusted NeRF for sparse views under the assumption of accurate camera poses (Yu et al., 2021; Chen et al., 2021a; Niemeyer et al., 2022; Yang et al., 2023). Concurrently, enhanced camera pose estimation methods for these sparse images have emerged (Zhang et al., 2022a). However, preliminary results for combining the efforts indicate a potential incompatibility; minor pose estimation inaccuracies can significantly degrade the quality of synthesized views in NeRF (Truong et al., 2022; Jiang et al., 2022; Sinha et al., 2022).

We first diagnose the limitations of existing approaches. As illustrated in Fig. 2, the existing generalizable NeRF approaches (Yu et al., 2021; Wang et al., 2021b) rely on camera poses for performing 2D-3D information mapping. Specifically, these methods *project a 3D point* to its single corresponding 2D locations in each of the input images *based on camera poses*, and aggregate features at these projected locations. Consequently, any pose inaccuracies distort this 3D-2D association, leading to compromised 3D point features, which are used to predict the radiance.

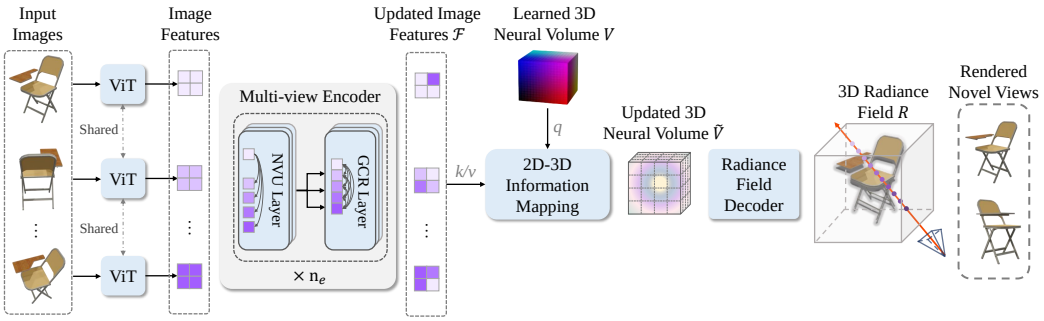

Figure 3: **LEAP overview.** LEAP extracts image features of all inputs using a ViT backbone. The first image in the image set is selected as the canonical view. The neural volume is defined in the local camera coordinate of the canonical view, which has learnable parameters to encode geometry and texture priors. To make LEAP aware of the choice of canonical view, we use a Multi-View Encoder to propagate information from the canonical view to the non-canonical views, making the 2D representations more coherent across views. Then the neural volume is updated by querying the 2D image features of all images using a 2D-3D Information Mapping module. We decode the neural volume into the radiance field and render novel views at inference time.

In contrast, LEAP proposes a novel *pose-free* paradigm, eliminating the influence of any pose inaccuracies. At its core, LEAP establishes the 3D-2D association based on *feature similarities*, enabling a 3D point to aggregate information from all pixels, rather than its 2D projection only. For each 3D voxel grid, the pose-free aggregation will learn to adaptively assign larger weights to its corresponding 2D pixels. We introduce the details of LEAP architecture in the following section.

4 METHOD

We present the task formulation and the overview of LEAP. Given a set of k 2D image observations of a scene, represented as $\{I_i | i = 1, ..., k\}$, LEAP predicts a neural radiance field (Mildenhall et al., 2020), which enables synthesizing a 2D image from an arbitrary target viewpoint. Note that in our setup of *sparse source views* captured by wide-baseline cameras, the number k is typically less than 5. Moreover, these views are presented *without any associated camera pose information* at inference.

4.1 MODEL ARCHITECTURE

As illustrated in Fig. 3, LEAP initiates by extracting 2D image features from all views. We use a DINOv2-initialized ViT (Oquab et al., 2023; Dosovitskiy et al., 2020) as the feature extractor since it demonstrates strong capability in modeling cross-view correlations (Zhang et al., 2023). We denote the image features of I_i as $f_i \in \mathbb{R}^{h \times w \times c}$, and the resulting features set for all input views as $\{f_i | i = 1, ..., k\}$. Due to the unawareness of the world coordinate frame on unposed images, we perform 3D modeling in a local camera coordinate frame. Specifically, we designate one image as the canonical view, where the neural volume and radiance field are defined in its local coordinate frame. During training, we randomly select a canonical view and denote it as I_1 for notation clarity.

To make LEAP aware of the choice of the canonical view, we find the key is to make the features of the non-canonical views consistent with the canonical view. We propose a multi-view image encoder to improve the feature consistency. Then, LEAP introduces a learnable neural volume, which is shared across scenes, to encode the geometric and textural prior. The neural volume serves as the initial 3D representation for all scenes. For each incoming scene, by querying the multi-view features, LEAP maps the 2D information to the 3D domain, represented by an updated neural volume. Finally, LEAP predicts the radiance field from the updated neural volume. We describe each step as follows.

Multi-view Image Encoder makes LEAP aware of the choice of the canonical view, by performing multi-view information reasoning. It takes in image features of all views and refines them by capturing cross-view correlations. It consists of n_e blocks, and each block has two layers: a Non-canonical View Update (NVU) layer and a Global Consensus Reasoning (GCR) layer. The NVU layer updates each of the non-canonical view features by aggregating the canonical view features. It is denoted as $\bar{f}_j = \text{NVU}(f_j, f_1)$, where $j \neq 1$ and $\bar{f}$ denotes the updated features. The GCR layer performs joint reasoning on all views for a global consensus, which leverages the correlation between all views.

We implemented the two layers with Transformer layers (Vaswani et al., 2017). Specifically, the NVU layer is modeled as a Transformer layer with cross-attention, where the non-canonical view

features are queries, and the canonical view features are keys/values. It is formulated as

$$\left[\bar{f}_2, ..., \bar{f}_k\right] = \text{FFN}(\text{CrossAttention}([f_2, ..., f_k], f_1)), \tag{1}$$

where FFN is a feed-forward network, and $[\cdot]$ denotes the concatenating operation over tokenized image features. For clarity, $\left[\bar{f}_2, ..., \bar{f}_k\right]$ is in $\mathbb{R}^{(k-1)hw \times c}$ and f_1 is flattened into $\mathbb{R}^{hw \times c}$.

Similarly, the GCR layer is instantiated by a Transformer layer with self-attention, where the query, key, and value are the 2D image features of all views. It is formulated as

$$\left[\tilde{f}_1, ..., \tilde{f}_k\right] = \text{FFN}(\text{SelfAttention}(\left[f_1, \bar{f}_2, ..., \bar{f}_k\right])). \tag{2}$$

Specifically, $\left[\tilde{f}_1, ..., \tilde{f}_k\right]$ is in $\mathbb{R}^{khw \times c}$. For simplicity, we denote the final output $\left[\tilde{f}_1, ..., \tilde{f}_k\right]$ as $\mathbf{F}$.

2D-3D Information Mapping. LEAP introduces a 3D latent neural volume $\mathbf{V} \in \mathbb{R}^{H \times W \times D \times c}$ to encode the geometry and texture priors, where H, W, D are the spatial resolution of the volume. It is defined in the local camera coordinate of the canonical view. The neural volume is shared across different scenes and gets updated by mapping the 2D image information to the 3D domain.

To perform the 2D-3D information mapping, we use n_m Transformer Decoder blocks (Vaswani et al., 2017), each of which consists of a cross-attention layer and a self-attention layer. In the cross-attention layer, we use the 3D latent volume $\mathbf{V}$ as the query and use $\mathbf{F}$ as the key/value. The updated 3D neural volume $\bar{\mathbf{V}}$ is defined as $\bar{\mathbf{V}} = \text{FFN}(\text{CrossAttention}(\mathbf{V}, \mathbf{F}))$. Intuitively, for each 3D point belonging to the neural volume, we compute its feature similarity with all 2D image features and use the similarity to get the weighted average of 2D image features. Subsequently, the self-attention layer performs refinement on the 3D volume features, capturing long-range geometry correlation. With multiple 2D-3D information lifting blocks, LEAP learns to update the latent volume with a fixed resolution in a coarse-to-fine manner. We denote the updated neural volume as $\tilde{\mathbf{V}} \in \mathbb{R}^{H \times W \times D \times c}$.

Neural Rendering. With the updated neural volume $\tilde{\mathbf{V}}$, LEAP predicts a volume-based neural radiance field (Yu et al., 2022; Jiang et al., 2022). The radiance field is denoted as $\mathbf{R} := (\mathbf{R}_\sigma, \mathbf{R}_f)$, where $\mathbf{R}_\sigma$ and $\mathbf{R}_f$ are the density and features of the radiance field. $\mathbf{R}_\sigma \in \mathbb{R}^{H' \times W' \times D'}$ and $\mathbf{R}_f \in \mathbb{R}^{H' \times W' \times D' \times C}$, where H', W', and D' are spatial resolutions. Both $\mathbf{R}_\sigma$ and $\mathbf{R}_f$ are predicted from $\tilde{\mathbf{V}}$ using 3D convolution layers. We read out the rendered image $\hat{I}$ and object mask $\hat{\sigma}$ using volumetric rendering techniques (Mildenhall et al., 2020). In detail, we first render a feature map and predict the rendered image using 2D convolutions. Formally, $(\hat{I}, \hat{\sigma}) = \Pi(\mathbf{R}, \Phi)$, where Π denotes the volumetric rendering process, and Φ is the target camera pose.

4.2 Training and Inference of LEAP

Training. LEAP is trained with the photo-metric loss between the rendering results and the inputs without any 3D supervision. We first define the loss L_I applied on the RGB images, where $L_I = \Sigma_i L_{mse}(\hat{I}_i, I_i) + \Sigma_i \lambda_p L_p(\hat{I}_i, I_i)$. The L_{mse} is the MSE loss, I_i and $\hat{I}_i$ are the original and rendered input images, λ_p is a hyper-parameter used for balancing losses, and L_p is the perceptual loss (Johnson et al., 2016). We then define the loss L_M applied on the density masks, as $L_M = \Sigma_i L_{mse}(\hat{\sigma}_i, \sigma_i)$, where $\hat{\sigma}_i$ and σ_i are original and rendered density masks. The final loss is defined as $L = L_I + \lambda_m L_M$, where λ_m is the weight balancing hyperparameter. We only use L_I if the masks are not available. We use ground-truth camera poses of training scenes to render the predicted inputs.

Inference and Evaluation. During inference, LEAP predicts the radiance field without reliance on any poses. To evaluate the novel view synthesis quality, we use the testing camera poses to render the radiance field under specific viewpoints. We use the relative pose system for novel view synthesis.

5 Experiment

We introduce our evaluation results on diverse and large-scale datasets, including both object-centric and scene-level datasets, for comparison with prior arts.

Implementation Details. We consider the number of views to be $k = 5$, with image resolution 224. We set $\lambda_p = 0.1$ and $\lambda_m = 5.0$. We set the peak learning rate as $2e-5$ (for the backbone) and $2e-4$ (for other components) with a warmup for 500 iterations using AdamW optimizer (Loshchilov & Hutter, 2017). We train the model for 150k iterations and use a linear learning rate scheduler, where

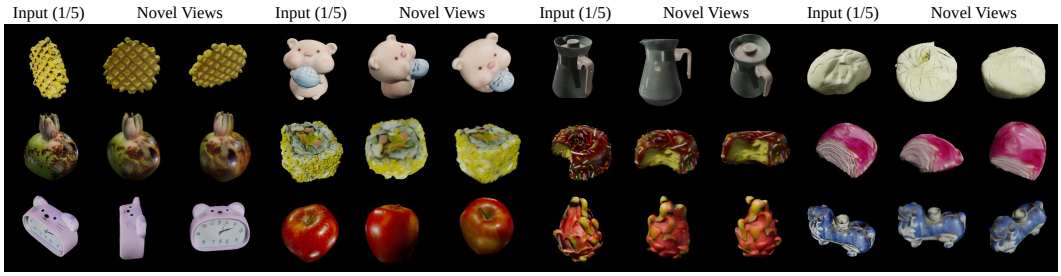

Figure 4: **Visualization of LEAP results on the object-centric Omniobject3D dataset.** For each instance, we include one out of five (denoted as "1/5") input views and two novel views. See more results in the supplementary.

the batch size is 32. LEAP has $n_e = 2$ multi-view encoder blocks and $n_m = 4$ 2D-3D mapping blocks. The resolution of the 3D neural volume and the volume-based radiance fields are 16^3 and 64^3, respectively. We sample 64 points on each ray for rendering.

Datasets. We train LEAP on each of the following datasets and test its capability to model the 3D object/scenes on each dataset that has different properties. We note that these datasets are captured by **wide-baseline** cameras, with randomly sampled or fixed camera poses that are far from each other.

- **OmniObject3D** (Wu et al., 2023) contains daily objects from 217 categories. We use a subset with 4800 instances for training and 498 instances for testing. OmniObject3D contains objects with complicated and realistic textures.
- **Kubric-ShapeNet** (Jiang et al., 2022) is a synthetic dataset generated using Kubric (Greff et al., 2022). Its training set has 1000 instances for each of 13 ShapeNet (Chang et al., 2015) categories, resulting in 13000 training samples. Its test set is composed of two parts: i) 1300 object instances from training categories; ii) 1000 object instances from 10 novel object categories. The two subsets are used to test the reconstruction quality and generalization ability of models. This dataset contains objects with complicated geometry but simple textures.
- **Objaverse** (Deitke et al., 2022) is one of the largest object-centric datasets. We use a subset of $200k$ and $2k$ instances for training and testing, used to validate LEAP on large-scale data.
- **DTU dataset** (Jensen et al., 2014) is a real scene-level dataset. DTU is on a small scale, containing only 88 scenes for training, which tests the ability of LEAP to fit small-scale data.

Metrics. Following previous works, we use standard novel view synthesis metrics, including PSNR (in dB), SSIM (Wang et al., 2004) and LPIPS (Zhang et al., 2018).

Baselines. We compare LEAP with the following baselines. We note that we train each baseline model (except zero123) on each of the datasets using the same setting with LEAP for fair comparisons. We use official or officially verified implementations of all baselines.

- **PixelNeRF** (Yu et al., 2021) is a generalizable NeRF variant, using camera poses to correlate 3D points and 2D pixels. We experiment with both ground-truth poses and predicted poses (with ground-truth translations) from a state-of-the-art pose estimator RelPose (Zhang et al., 2022a).
- **FORGE** (Jiang et al., 2022) is a generalizable NeRF variant with test-time optimization, which jointly predicts camera poses and the neural radiance field, and leverages their synergy to improve the performance of both. We experiment with FORGE using ground-truth and its predicted poses.
- **SPARF** (Truong et al., 2022) is a scene-specific NeRF variant that jointly optimizes the camera poses and the radiance field. It requires reasonable pose initialization and is dependent on dense visual correspondences predicted from off-the-shelf methods.
- **SRT** (Sajjadi et al., 2022b) uses only 2D representation to perform novel view synthesis. It is trained and tested on unposed image sets.
- **Zero123** (Liu et al., 2023) is a novel view synthesis method using diffusion models. We note that Zero123 takes a single image as input, which is different from LEAP and other baselines. We test Zero123 to compare LEAP with large-scale 2D models.

5.1 Comparisons with State-of-the-Art

Object-centric Results. The results are shown in Table 1. On all four test sets, LEAP outperforms all prior pose-free works and pose-based works (with estimated poses). The results demonstrate the success of LEAP for modeling objects with different geometry and texture properties. In detail, LEAP improves over the next-best baseline (FORGE) by about 3 dB PSNR and 50% LPIPS relatively

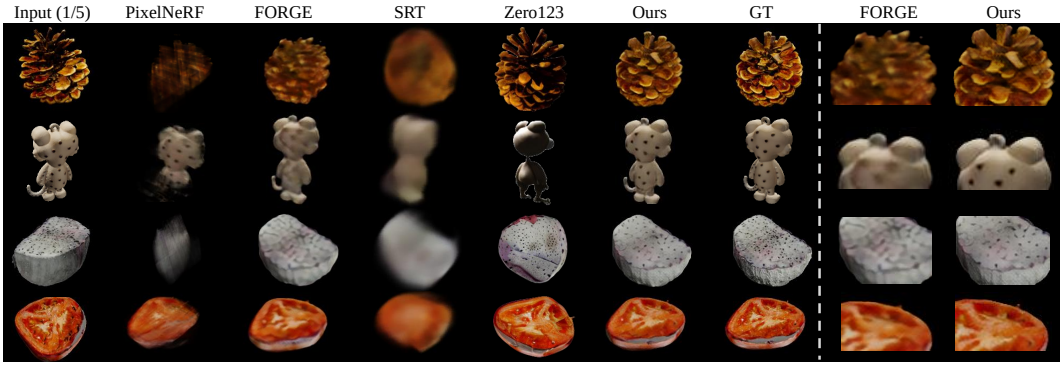

Figure 5: **Comparison with prior arts.** The performance of PixelNeRF degenerates dramatically with the state-of-the-art pose estimator. FORGE benefits from its joint optimization of shape and pose, but the high-frequency details are lost. SRT can only recover noisy results. Zero123 can synthesize high-quality images, while the content is not consistent with the inputs. In contrast, LEAP reliably recovers the details and the novel views match the ground-truth target view well. We also include zoom-in results on the right for a clearer comparison.

Table 1: **Evaluation on four object-centric test sets.** We include the inference time of each method. ✗ means the method is pose-free. For experiments without using perfect poses, We highlight the best and second-best results. For experiments with perfect poses, we also highlight the best GT pose result if it is better than ours.

			Omniobject3D *217 Ctg. / 498 Inst.*			Kubric-ShapeNet-seen *13 Ctg. / 1300 Inst.*			Kubric-ShapeNet-novel *10 Ctg. / 1000 Inst.*			Objaverse *Open-vocab. Ctg.*		
Model	Pose	Inf. Time	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
PixelNeRF	GT	2 min	26.97	0.888	0.123	29.25	0.893	0.127	29.37	0.906	0.122	26.21	0.871	0.133
	Pred.	2 min	18.87	0.810	0.199	21.36	0.836	0.188	21.22	0.851	0.174	20.97	0.819	0.191
FORGE	GT	0.05 sec	28.93	0.913	0.087	31.32	0.938	0.053	31.17	0.946	0.058	27.76	0.896	0.100
	Pred.	15 min	26.56	0.889	0.108	26.61	0.896	0.106	25.57	0.898	0.107	23.67	0.856	0.226
SRT	✗	0.4 sec	20.22	0.786	0.303	22.62	0.802	0.267	22.46	0.793	0.284	20.41	0.798	0.312
Zero123	✗	27 sec	16.77	0.812	0.147	14.42	0.803	0.174	15.51	0.837	0.152	19.59	0.862	0.110
LEAP	✗	**0.3 sec**	**29.10**	**0.919**	**0.057**	**29.86**	**0.929**	**0.067**	**28.22**	**0.924**	**0.070**	**26.77**	**0.887**	**0.103**

on all datasets. Furthermore, without the need for any test-time refinement, the running speed of LEAP is significantly faster than FORGE (0.3-second v.s. 15 min). Besides, LEAP demonstrates strong generalization capability, as the model trained on the Kubric ShapeNet dataset of only 13 categories is able to work on novel ShapeNet categories with nearly no gap.

Interestingly, when compared with prior pose-based methods using ground-truth poses, LEAP exhibits a comparable or even better performance. This result verifies our proposition that camera poses may not be necessary for 3D modeling, at least in the sparse-view setting. We present a visualization of our results in Fig. 4 and the comparison with prior works in Fig. 5.

Scene-level Results. The results are shown in Table 2. Since the DTU dataset is too small to train a usable pose estimator, we follow SPARF to use different levels of noisy poses. Our method outperforms PixelNeRF with noisy poses and achieves comparable results with SPARF. We note that SPARF is a scene-specific NeRF variant that takes much longer time to optimize the radiance field and requires additional inputs, i.e., accurate dense visual correspondences between input views. We include a qualitative comparison in Fig. 6.

Besides, we also observe that a compelling phenomenon - a LEAP model pre-trained on the large-scale object-centric datasets largely improves its performance on the scene-level evaluation. The reason is that as a pose-free method with our geometric inductive bias, LEAP requires learning the knowledge on larger data compared with pose-based works. Training from scratch on the small-scale DTU dataset, which only contains 88 scenes for training, leads to unsatisfying performance. On the other hand, the effectiveness of the pre-training demonstrates the capability of LEAP to learn general-purpose 3D knowledge, which is generalizable and can be transferred to novel domains.

5.2 Ablation Study and Insights

We present an ablation study on each block of LEAP to study their impacts.

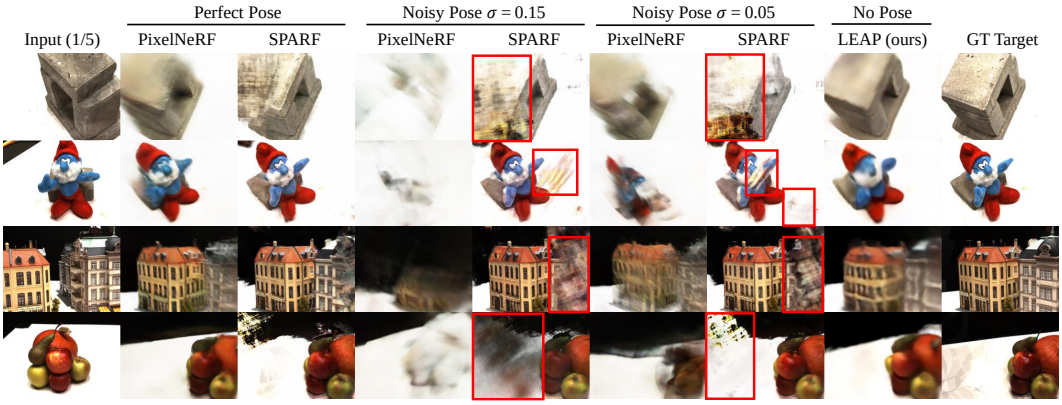

Figure 6: **Comparison with prior arts on DTU dataset.** PixelNeRF collapses under noisy poses. SPARF recovers high-frequency details well, but it degenerates when the correspondences are not accurate and demonstrates strong artifacts (shown in red boxes). LEAP reliably recovers the geometry well but lacks texture details. The result implies that LEAP, as a pose-free method, requires larger training datasets.

Table 2: **Evaluation on the DTU dataset.** LEAP performs on par with SPARF which requires slow per-scene optimization and additional dense image correspondence inputs. Numbers with $*$ are after SPARF optimization.

Method	Generalizable	Image-only	Pose Noise	Rot Err.	Trans Err.	PSNR$_\uparrow$	SSIM$_\uparrow$	LPIPS$_\downarrow$	Inference Time
PixelNeRF	✓	✓	GT	–	–	19.60	0.720	0.295	2 min
			σ=0.05	5.03	0.17	14.42	0.486	0.463	
			σ=0.15	14.31	0.42	10.78	0.432	0.538	
SPARF	✗	✗	GT	–	–	19.79	0.749	0.275	1 day
			σ=0.05	1.31*	0.04*	**18.57**	**0.682**	**0.336**	
			σ=0.15	1.93*	0.06*	18.03	0.668	0.361	
LEAP	✓	✓	–	–	–	15.37	0.535	0.478	**0.3 sec**
LEAP-pretrain						18.07	0.671	0.371	

Coordinate Frame. We study the importance of the local camera coordinate frame by using the world coordinate frame instead. We use the category-level coordinate as the world coordinate, where objects have aligned rotation and are zero-centered. As shown in Table 3 (a), the model demonstrates better performance on the seen object categories (with 31.23 v.s. 29.10 PSNR) but generalizes worse on novel categories. We conjecture the reason is that the aligned rotation/translation world coordinate frame makes it easier to perform 2D-3D information mapping for training categories. However, it also limits the performance of novel categories, as their category-level coordinate frames are not learned by LEAP. This result matches our intuition of using the local camera coordinate frame to define the neural volume, which enables LEAP to generalize to any objects/scenes.

Multi-view Encoder. We explore the impact of using the multi-view encoder to make LEAP aware of the choice of the canonical view. We test the following alternatives: i) LEAP without the multi-view encoder; ii) LEAP that only has the global consensus reasoning (GCR) layers; iii) LEAP that only has the non-canonical view update (NVU) layers. As shown in Table 3 (b)-(d), without the multi-view encoder, we observe a significant performance drop. The reason is that the inconsistent features across views hamper the 2D-3D information mapping. Similarly, only with the GCR layers, LEAP struggles to determine which view is the canonical view. When only using the NVU layers, it achieves a slightly worse performance than the full model. The experiments show the effectiveness of using the multi-view encoder to make the model aware of the choice of canonical view.

The 2D-3D Information Mapping Layer. As shown in Table 3 (e), using two mapping layers (the default has four layers) slightly degenerates the performance, which shows its efficacy.

Interpreting LEAP. We perform visualization to understand what knowledge LEAP learns to handle the absence of camera poses. As shown in Fig. 7, LEAP adaptively assigns weights to reasonable 2D regions to perform 2D-2D reasoning and 2D-3D information mapping. The neural volume is updated in a coarse-to-fine manner during the process. Moreover, we test how the learned knowledge is related to explicit pose-based operations. As shown in Fig. 8, we input images of a small dot. We find that LEAP lifts the 2D pixel of the dot into the 3D space as a line segment. The location of the line segment projected in another view corresponds to its epipolar line. The phenomenon reveals that LEAP lifts a 2D point as its reprojection ray, and it leverages multiview

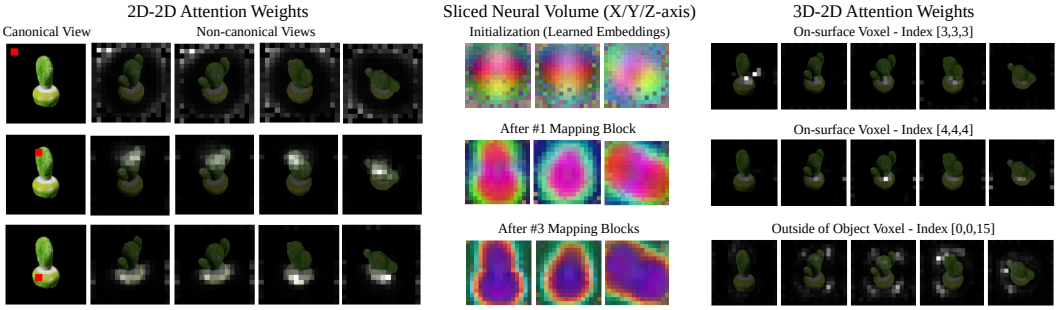

Figure 7: **Visualization of LEAP working mechanism.** (Left) We show the 2D-2D attention weights of the multi-view encoder. For each query pixel (in red) in the canonical view, it assigns larger weights to the corresponding regions in the Non-canonical views. The attention of query points in the background diffuses. (Middle) We visualize the learned neural volume by using PCA and slicing along the three axes. As our 3D modeling happens in the local coordinate frame, which is not axis-aligned, the learned embeddings show isotropic properties. The neural volume is refined in a coarse-to-fine manner, where the object boundary becomes more compact after more mapping layers. (Right) We show the attention map between 3D-2D. As shown in the top two rows, the neighbor on-surface voxels have similar attention patterns on a specific 2D object region. The attention of an out-of-object voxel diffuses on the background 2D region. See more details in the supplementary.

Table 3: **Ablation study on the Kubric dataset.** We ablate on a) using category-level world canonical space rather than the camera space. b)-d) the multi-view encoder designs for enabling the model aware of canonical view choice. And e) using less (#2/4) mapping layers. See visualization in supplementary.

	ShapeNet-novel		
	PSNR↑	SSIM↑	LPIPS↓
LEAP (full)	**28.22**	**0.924**	**0.072**
a) use world frame	24.62	0.865	0.116
b) no multi-view enc.	21.68	0.710	0.431
c) only GCR layer	22.98	0.770	0.359
d) only NVU layer	27.62	0.907	0.085
e) two mapping layers	27.99	0.910	0.080

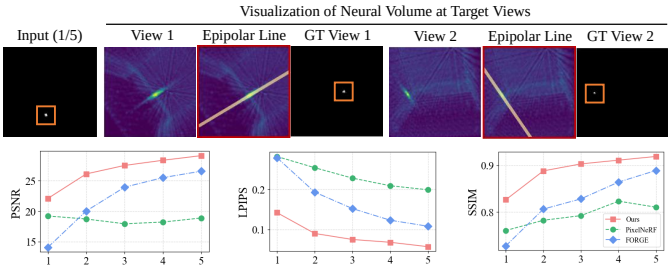

Figure 8: **Interpret LEAP.** Top: We input images of a small dot (in orange boxes), and the visualization of the reconstructed neural volume shows consistency with the epipolar lines of the small dot on target views. This implies LEAP mapps a 2D point as its 3D reprojection ray segment even though there are no reprojection operations. It leverages the multi-view information to resolve the depth ambiguity of the ray. Bottom: The performance with different numbers of inputs on Omniobject3D. Note that we only train the model with 5 images and it is directly tested.

information to resolve the ambiguity of the ray to determine the depth of the 2D point. Besides, we show LEAP performance with different numbers of input images. The results show that LEAP reliably reconstructs the object with two to five images, and its performance drops slightly with fewer inputs. However, we observe a big drop when we decrease the number of inputs from two images to one image. These results validate the effectiveness of LEAP in using multi-view information to perform the 3D modeling.

6 Conclusion

We propose LEAP, a pose-free approach for 3D modeling from a set of unposed sparse-view images. By appropriately setting the 3D coordinate and aggregating 2D image features, LEAP demonstrates satisfying novel view synthesis quality. In our experiments, spanning from both object-centric to scene-level, from synthetic images to real images, and from small-scale to large-scale data, LEAP consistently demonstrates better performance compared with prior pose-based works that use estimated poses or noisy poses. LEAP also achieve comparable results with the versions of prior works that use ground-truth poses. Besides, LEAP showcases a strong generalization capability, fast inference speed, and interpretable learned knowledge.

Limitations. LEAP adopts the neural volume representation where the 3D voxel grids span uniformly in the 3D space, and the physical size of the volume is bounded. Designing better 3D representation, e.g. incorporating techniques from prior works to enable it to work on unbounded scenes, will further benefit the application of LEAP.

REFERENCES

Wenjing Bian, Zirui Wang, Kejie Li, Jiawang Bian, and Victor Adrian Prisacariu. Nope-nerf: Optimising neural radiance field with no pose prior. *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4160–4169, 2022.

Mathilde Caron, Hugo Touvron, Ishan Misra, Herv'e J'egou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9630–9640, 2021.

Angel X. Chang, Thomas A. Funkhouser, Leonidas J. Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, L. Yi, and Fisher Yu. Shapenet: An information-rich 3d model repository. *ArXiv*, abs/1512.03012, 2015.

Anpei Chen, Zexiang Xu, Fuqiang Zhao, Xiaoshuai Zhang, Fanbo Xiang, Jingyi Yu, and Hao Su. Mvsnerf: Fast generalizable radiance field reconstruction from multi-view stereo. *ICCV*, pp. 14104–14113, 2021a.

Kefan Chen, Noah Snavely, and Ameesh Makadia. Wide-baseline relative camera pose estimation with directional learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3258–3268, 2021b.

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. *ArXiv*, abs/2212.08051, 2022.

Congyue Deng, Or Litany, Yueqi Duan, Adrien Poulenard, Andrea Tagliasacchi, and Leonidas J. Guibas. Vector neurons: A general framework for so(3)-equivariant networks. *ICCV*, pp. 12180–12189, 2021.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *ArXiv*, abs/2010.11929, 2020.

Laura Downs, Anthony Francis, Nate Koenig, Brandon Kinman, Ryan Michael Hickman, Krista Reymann, Thomas Barlow McHugh, and Vincent Vanhoucke. Google scanned objects: A high-quality dataset of 3d scanned household items. *2022 International Conference on Robotics and Automation (ICRA)*, pp. 2553–2560, 2022.

Andreas Geiger, Julius Ziegler, and Christoph Stiller. Stereoscan: Dense 3d reconstruction in real-time. *2011 IEEE Intelligent Vehicles Symposium (IV)*, pp. 963–968, 2011.

Michael Goesele, Brian Curless, and Steven M Seitz. Multi-view stereo revisited. In *CVPR*, volume 2, pp. 2402–2409. IEEE, 2006.

Klaus Greff, Francois Belletti, Lucas Beyer, Carl Doersch, Yilun Du, Daniel Duckworth, David J. Fleet, Dan Gnanapragasam, Florian Golemo, Charles Herrmann, Thomas Kipf, Abhijit Kundu, Dmitry Lagun, Issam Hadj Laradji, Hsueh-Ti Liu, Henning Meyer, Yishu Miao, Derek Nowrouzezahrai, Cengiz Oztireli, Etienne Pot, Noha Radwan, Daniel Rebain, Sara Sabour, Mehdi S. M. Sajjadi, Matan Sela, Vincent Sitzmann, Austin Stone, Deqing Sun, Suhani Vora, Ziyu Wang, Tianhao Wu, Kwang Moo Yi, Fangcheng Zhong, and Andrea Tagliasacchi. Kubric: A scalable dataset generator. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3739–3751, 2022.

Will Hutchcroft, Yuguang Li, Ivaylo Boyadzhiev, Zhiqiang Wan, Haiyan Wang, and Sing Bing Kang. Covispose: Co-visibility pose transformer for wide-baseline relative pose estimation in 360° indoor panoramas. In *European Conference on Computer Vision*, pp. 615–633. Springer, 2022.

Rasmus Ramsbøl Jensen, A. Dahl, George Vogiatzis, Engil Tola, and Henrik Aanæs. Large scale multi-view stereopsis evaluation. *CVPR*, pp. 406–413, 2014.

Hanwen Jiang, Zhenyu Jiang, Kristen Grauman, and Yuke Zhu. Few-view object reconstruction with unknown categories and camera poses. *ArXiv*, abs/2212.04492, 2022.

Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *ECCV*, 2016.

Amy Lin, Jason Y. Zhang, Deva Ramanan, and Shubham Tulsiani. Relpose++: Recovering 6d poses from sparse-view observations. *ArXiv*, abs/2305.04926, 2023.

Chen-Hsuan Lin, Wei-Chiu Ma, Antonio Torralba, and Simon Lucey. Barf: Bundle-adjusting neural radiance fields. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 5721–5731, 2021.

Ruoshi Liu, Rundi Wu, Basile Van Hoorick, Pavel Tokmakov, Sergey Zakharov, and Carl Vondrick. Zero-1-to-3: Zero-shot one image to 3d object. *ICCV*, 2023.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2017.

Quan Meng, Anpei Chen, Haimin Luo, Minye Wu, Hao Su, Lan Xu, Xuming He, and Jingyi Yu. Gnerf: Gan-based neural radiance field without posed camera. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 6331–6341, 2021.

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. *ArXiv*, abs/2003.08934, 2020.

Michael Niemeyer, Jonathan T. Barron, Ben Mildenhall, Mehdi S. M. Sajjadi, Andreas Geiger, and Noha Radwan. Regnerf: Regularizing neural radiance fields for view synthesis from sparse inputs. *CVPR*, pp. 5470–5480, 2022.

Maxime Oquab, Timoth'ee Darcet, Th'eo Moutakanni, Huy Q. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russ Howes, Po-Yao (Bernie) Huang, Shang-Wen Li, Ishan Misra, Michael G. Rabbat, Vasu Sharma, Gabriel Synnaeve, Huijiao Xu, Hervé Jégou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision. *ArXiv*, abs/2304.07193, 2023.

C. Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. *CVPR*, pp. 77–85, 2017.

Chris Rockwell, Justin Johnson, and David F Fouhey. The 8-point algorithm as an inductive bias for relative pose prediction by vits. In *2022 International Conference on 3D Vision (3DV)*, pp. 1–11. IEEE, 2022.

Mehdi S. M. Sajjadi, Aravindh Mahendran, Thomas Kipf, Etienne Pot, Daniel Duckworth, Mario Lucic, and Klaus Greff. Rust: Latent neural scene representations from unposed imagery. *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 17297–17306, 2022a. URL https://api.semanticscholar.org/CorpusID:254017732.

Mehdi S. M. Sajjadi, Henning Meyer, Etienne Pot, Urs M. Bergmann, Klaus Greff, Noha Radwan, Suhani Vora, Mario Lucic, Daniel Duckworth, Alexey Dosovitskiy, Jakob Uszkoreit, Thomas A. Funkhouser, and Andrea Tagliasacchi. Scene representation transformer: Geometry-free novel view synthesis through set-latent scene representations. *CVPR*, 2022b.

Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-Motion Revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

Samarth Sinha, Jason Y. Zhang, Andrea Tagliasacchi, Igor Gilitschenski, and David B. Lindell. Sparsepose: Sparse-view camera pose regression and refinement. *ArXiv*, abs/2211.16991, 2022.

Jiaming Sun, Zehong Shen, Yuang Wang, Hujun Bao, and Xiaowei Zhou. Loftr: Detector-free local feature matching with transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 8922–8931, 2021.

Prune Truong, Marie-Julie Rakotosaona, Fabian Manhardt, and Federico Tombari. Sparf: Neural radiance fields from sparse and noisy poses. *ArXiv*, abs/2211.11738, 2022.

Ashish Vaswani, Noam M. Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NIPS*, 2017.

Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. In *NeurIPS*, 2021a.

Qianqian Wang, Zhicheng Wang, Kyle Genova, Pratul P. Srinivasan, Howard Zhou, Jonathan T. Barron, Ricardo Martin-Brualla, Noah Snavely, and Thomas A. Funkhouser. Ibrnet: Learning multi-view image-based rendering. *CVPR*, pp. 4688–4697, 2021b.

Zhou Wang, Alan Conrad Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13:600–612, 2004.

Zirui Wang, Shangzhe Wu, Weidi Xie, Min Chen, and Victor Adrian Prisacariu. Nerf-: Neural radiance fields without known camera parameters. *ArXiv*, abs/2102.07064, 2021c.

Tong Wu, Jiarui Zhang, Xiao Fu, Yuxin Wang, Jiawei Ren, Liang Pan, 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. *CVPR*, 2023.

Yitong Xia, Hao Tang, Radu Timofte, and Luc Van Gool. Sinerf: Sinusoidal neural radiance fields for joint pose estimation and scene reconstruction. In *British Machine Vision Conference*, 2022.

Jiawei Yang, Marco Pavone, and Yue Wang. Freenerf: Improving few-shot neural rendering with free frequency regularization. *CVPR*, 2023.

Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelnerf: Neural radiance fields from one or few images. *CVPR*, pp. 4576–4585, 2021.

Alex Yu, Sara Fridovich-Keil, Matthew Tancik, Qinhong Chen, Benjamin Recht, and Angjoo Kanazawa. Plenoxels: Radiance fields without neural networks. *cvpr*, pp. 5491–5500, 2022.

Jason Y. Zhang, Deva Ramanan, and Shubham Tulsiani. Relpose: Predicting probabilistic relative rotation for single objects in the wild. 2022a.

Jiahui Zhang, Fangneng Zhan, Rongliang Wu, Yingchen Yu, Wenqing Zhang, Bai Song, Xiaoqin Zhang, and Shijian Lu. Vmrf: View matching neural radiance fields. *Proceedings of the 30th ACM International Conference on Multimedia*, 2022b.

Junyi Zhang, Charles Herrmann, Junhwa Hur, Luisa Polania Cabrera, Varun Jampani, Deqing Sun, and Ming Yang. A tale of two features: Stable diffusion complements dino for zero-shot semantic correspondence. *ArXiv*, abs/2305.15347, 2023.

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 586–595, 2018.

Andrew Zisserman. Multiple view geometry in computer vision. *Künstliche Intell.*, 15:41, 2001.

Appendices

In the supplementary, we first include a detailed analysis to understand the novel pose-free framework of LEAP by visualizing the intermediate results. Then, we include more novel view synthesis visualization results. Finally, we include the details of our model and the baselines, as well as the evaluation details on each dataset.

A Model Details

Novel View Synthesis Coordinate System. We use **relative** camera poses for both training and inference. The relative camera pose specifies the rotation and translation of a target view with respect to the canonical view (first image) Jiang et al. (2022); Lin et al. (2023). Thus, during testing, you can control the target NVS viewpoints by specifying small or large relative camera transformations. This definition of target NVS camera pose will not be influenced by the different definitions or absence of "absolute" camera pose of input views, as the reconstructed radiance field is defined in the local frame of canonical view.

Reconstruction Scale. We perform all experiments (both training and evaluation) with a normalized scale. This is a common setting in RGB-based reconstruction methods Schönberger & Frahm (2016). As scaling up/down the size of the scenes as well as the camera translations jointly will lead to the same RGB observations at these camera viewpoints, it requires depth information to resolve the scale ambiguity. It is an ill-posed problem to recover the scale when only RGB images are available (if without any category-level prior, e.g. human category average height).

Improved Speed. The fast inference speed of LEAP originates from its ability to predict the radiance field in a single feed-forward step during inference, without any test-time optimization. In contrast, most prior works require some per-scene optimization at test time (SPARF, RelPose, and FORGE), which are slow to converge.

B More Analysis

In this section, we dive deep into the pose-free framework and visualize the intermediate layer results and attention weights to understand the working mechanism of LEAP. Following the sequence of modules of LEAP, we perform analyses on the Multi-view Encoder and the 2D-to-3D information mapping module.

Multi-view Encoder. We first understand the Non-canonical View Update layers. We visualize the cross-attention weights between the canonical views and the non-canonical views. The results are shown in Fig. 9.

The Learned Neural Volume Embeddings. We then visualize the learned neural volume embeddings and the updated neural volume through the 2D-to-3D information mapping layers. The results are shown in Fig. 10.

The 2D-to-3D Information Mapping Module. We then dive deeper into the 2D-to-3D information mapping module to understand how the neural volume aggregates information from the 2D image features. The results are shown in Fig. 11.

C More Results and Ablations

Working on more views. We further experiment with increasing the number of input views. As shown in Fig. 8 (bottom), LEAP is able to work with fewer images (2-4 views) when the model is trained with 5 views. We further evaluate the results using 7 and 10 images (as shown below), which reach the upper bound of the number of views in prior sparse-view research Zhang et al. (2022a) As shown in Table 4, LEAP demonstrates better performance compared with the best baseline method FORGE. In detail, the performance of the best baseline FORGE drops with more than 7 views. We conjecture that the reason is the compounding pose estimation error. In contrast, LEAP continually demonstrates better performance with more inputs.

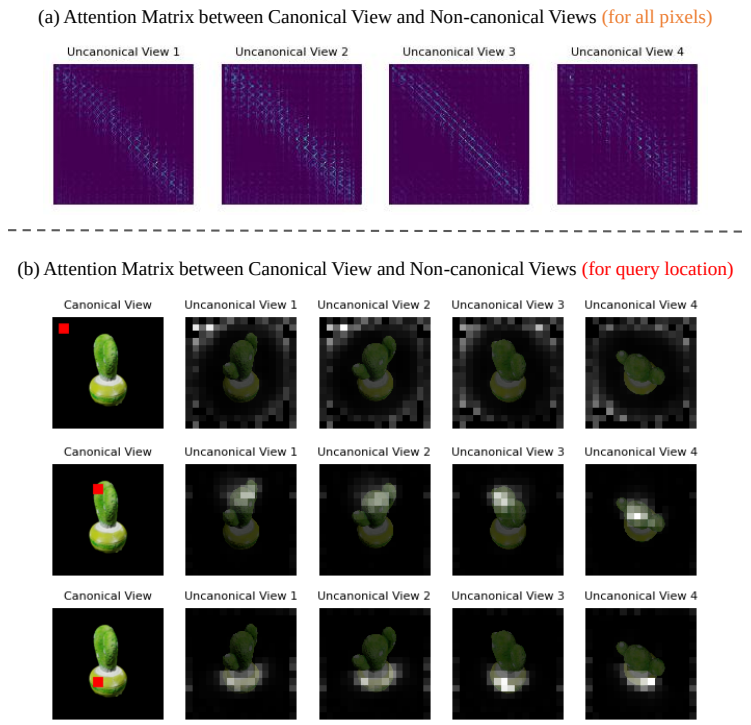

Figure 9: **Visualization of cross-attention weight in the Non-canonical View Update layer in the Multi-view Encoder.** (a) We show the attention matrix on the entire image, where the shape of the attention matrix is $\mathbb{R}^{hw \times hw}$ and h, w are the resolution of the image features. The visualization demonstrates clear patterns of the correlation between canonical view and non-canonical view image features. (b) We further show detailed attention weights for each query pixel in the canonical view. The visualization demonstrates that the attention captures the cross-view correlations well. In detail, for a query pixel in the background, the corresponding regions are also in the background of the non-canonical views. When the query pixel is located on the object, the attention focuses on small corresponding regions in the non-canonical views. The results demonstrate that the image features of the multi-view images are coherent.

Incremental Reconstruction. we provide a study on incremental reconstruction using 5 existing views with 1 incoming view. Our solution is i) reconstructing the neural volume using the 5 existing views as an initialization, ii) using the multiview encoder to propagate information between the canonical view and the incoming view, and iii) refining the existing neural volume using the image features of the incoming and canonical view by performing 2D-3D information mapping attentions. Using the refined neural volume for predicting the radiance field demonstrates 0.37 PSNR novel view synthesis improvement, compared with the results from the existing 5 views. For reference, direct inference on 6 images observes a 0.45 PSNR improvement. The slight gap shows the feasibility of the solution and the potential of LEAP in handling the incremental reconstruction problem.

More Cross-dataset Generalization. We study the cross-data transfer capability as shown in Table 1 of the paper, where we train on 13 ShapeNet categories (Kubric-ShapeNet-seen) and test its generalization on 10 novel categories (Kubric-ShapeNet-novel). LEAP demonstrates robust generalization in this test with about 3 dB PSNR higher than the next-best method FORGE. To further address your concern, we include another cross-domain evaluation, where the model is tested on Google Scanned Objects (GSO) Downs et al. (2022), collected from another domain. LEAP demonstrates better performance than FORGE in this setting (24.12 v.s. 22.78 PSNR).

Geometry Reconstruction Quality. We compare with FORGE on the Kubric-ShapeNet-dataset. Our method LEAP decreases the depth rendering error significantly compared with the best baseline method FORGE (**0.11** v.s. 0.16).

LEAP for pose estimation. We experiment with using image features of LEAP for estimating poses on Omniobject3D. As shown in Table 5, the fine-tuned DINO backbone of LEAP demonstrates better performance than the original DINO. Moreover, the features after the multiview encoder demonstrate even better performance, showing that LEAP learns cross-view correspondence cues. We use a concatenation-regression pose estimator for this experiment.

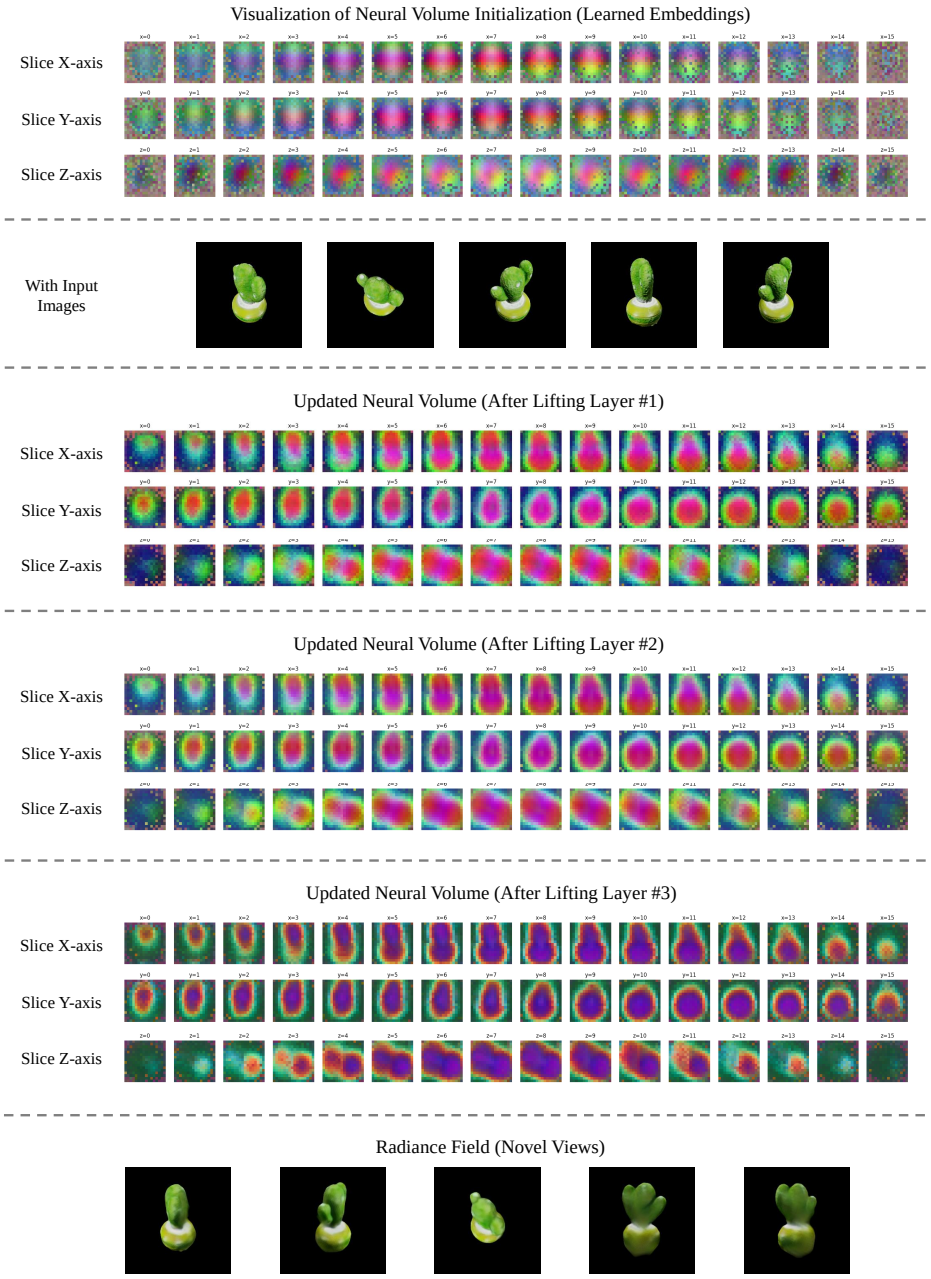

Figure 10: **Visualization of learned neural volume embeddings and the updated neural volume.** We visualize the neural volume by slicing it from the three axes. (Top) As our 3D modeling happens in the local camera coordinate which is not axis-aligned, the learned embeddings show isotropic properties, i.e. like a sphere which is able to model objects with different orientations. (Bottom) After lifting information from the images, the neural volume is turned into the shape of the input object. With more 2D-to-3D information mapping layers, the boundary of the object becomes more sharp.

More Results. We show more visualization of novel view synthesis on the evaluation datasets in Fig. 12 (Omniobject3D) and Fig. 13 (Kurbic and Objaverse). To better understand the performance of LEAP, we also include failure cases in Fig. 14.

Besides, we also include the results for ablation experiments, including the results without using the multi-view encoder (Fig. 15) and using different numbers of inputs (Fig. 16). The results verify the significance of the multi-view encoder and the strong capability of LEAP for modeling objects from only two views.

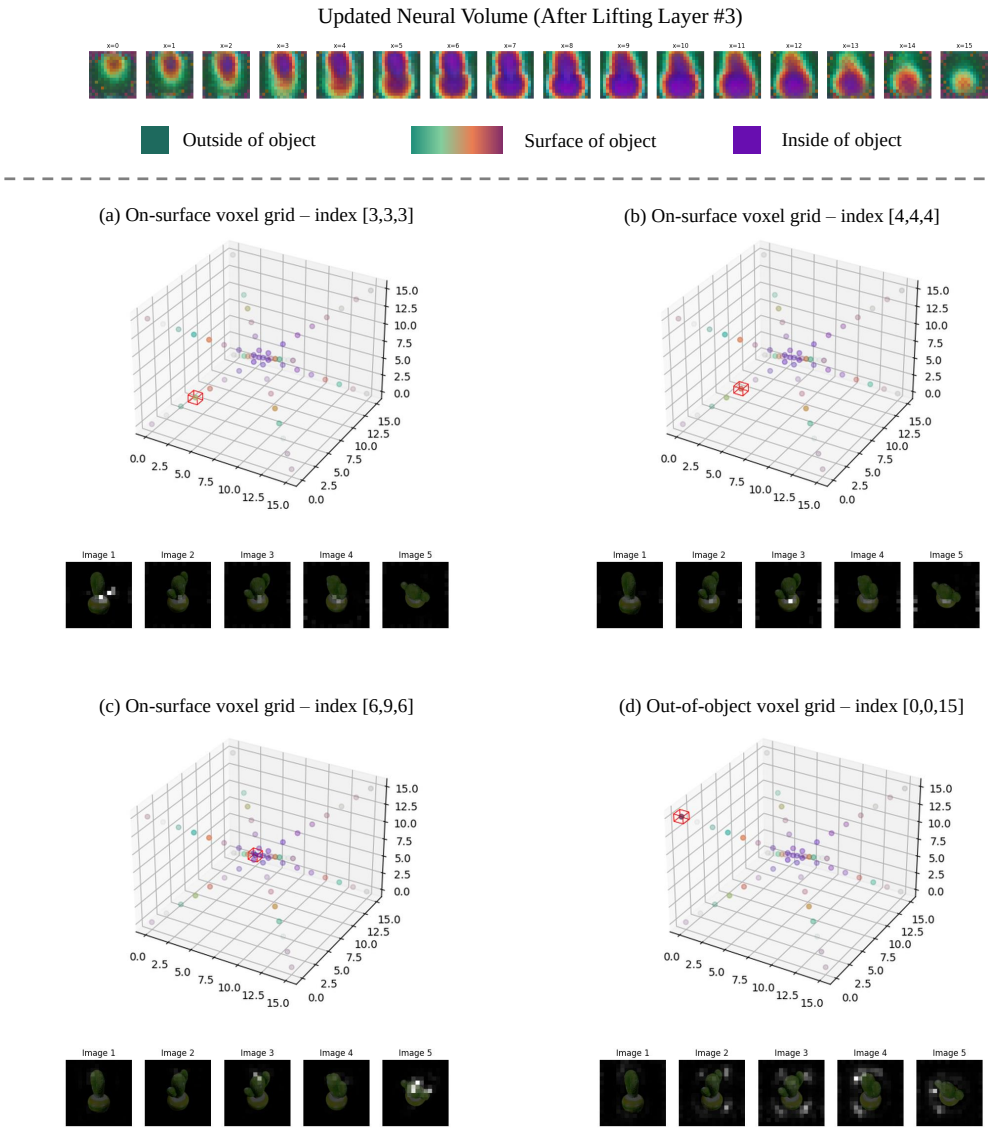

Figure 11: **Visualization of cross-attention weights between the neural volume and the image features in the 2D-to-3D information mapping layer.** (Top) We specify the visualized color for voxel grids outside, on the surface of, and inside the object. (Bottom) Due to the complexity of the 3D volume, we visualize the attention weights for voxel grids on the diagonal of the volume to ensure the traversal of the object surface. The query voxel grids are contained by the red bounding box. (a)-(b) The on-surface voxels attend 2D image features from the same image regions across views. Moreover, the two neighbor voxel grids demonstrate similar attention patterns, showing the smoothness of the learned mapping function. (c) Another example for an on-surface voxel grid. (d) The attention diffuses for a non-surface voxel grid.

	5 Views	7 Views	10 Views
FORGE	26.56	27.16	27.08
LEAP	**29.10**	**29.75**	**30.11**

Table 4: Comparison with FORGE on using more views as inputs.

	DINOv2	DINOv2-LEAP	LEAP-MV Encoder
rot. err.	16.53	14.17	**9.85**

Table 5: Evaluation on LEAP for estimating camera poses.

Figure 12: **Visualization of the Omniobject3D dataset.** For each example, we include three images, where the first is one out of five input views, and the last two are rendered novel views.

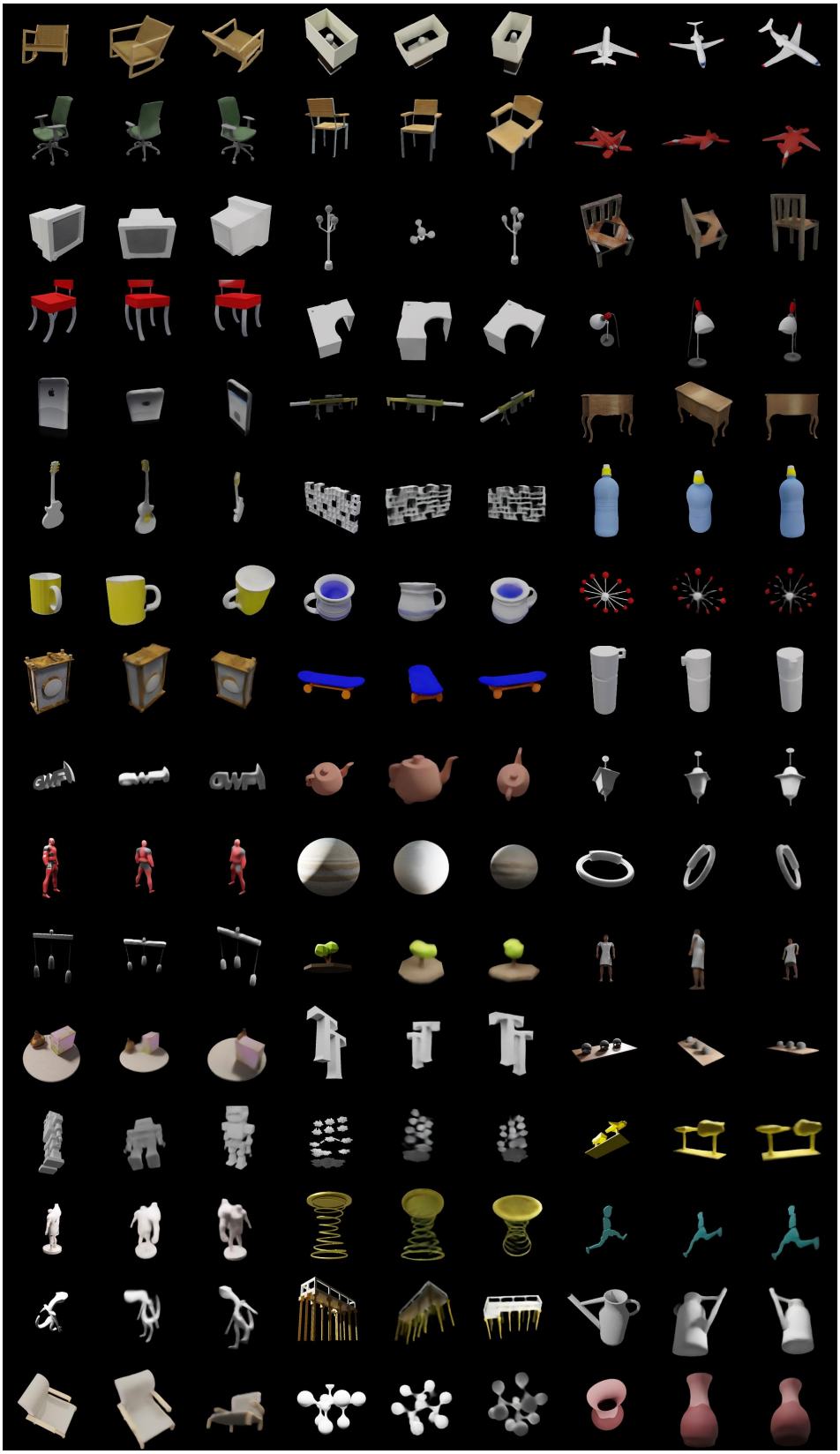

Figure 13: **Visualization of the Kubric (top 8 rows) and Objaverse (bottom 8 rows) dataset.** For each example, we include three images, where the first is one out of five input views, and the last two are rendered novel views.

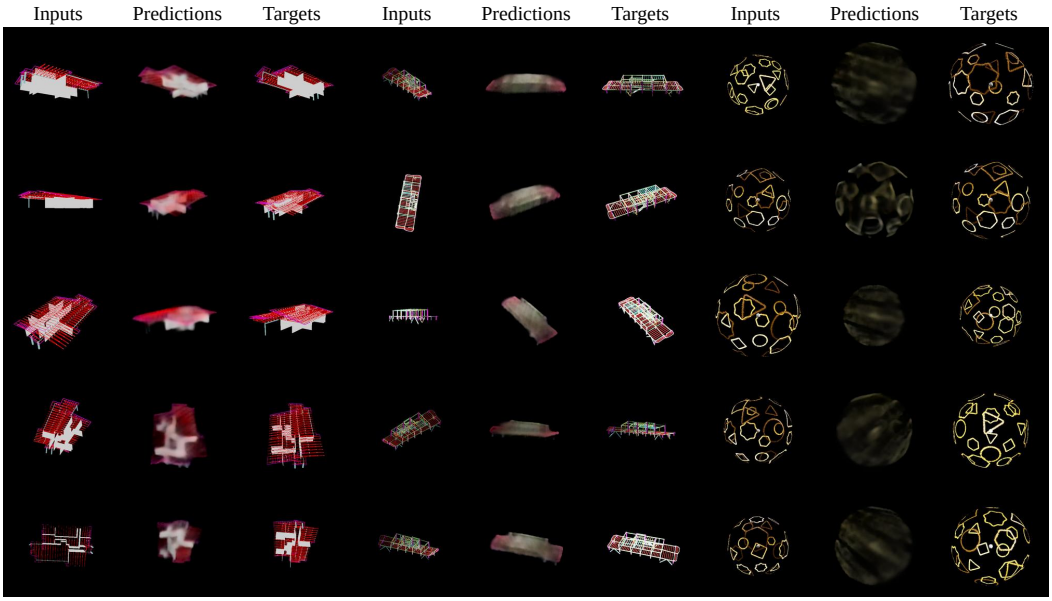

Figure 14: **Visualization of failure cases.** We observe that the performance on objects with very fine-grained geometry details is still limited.

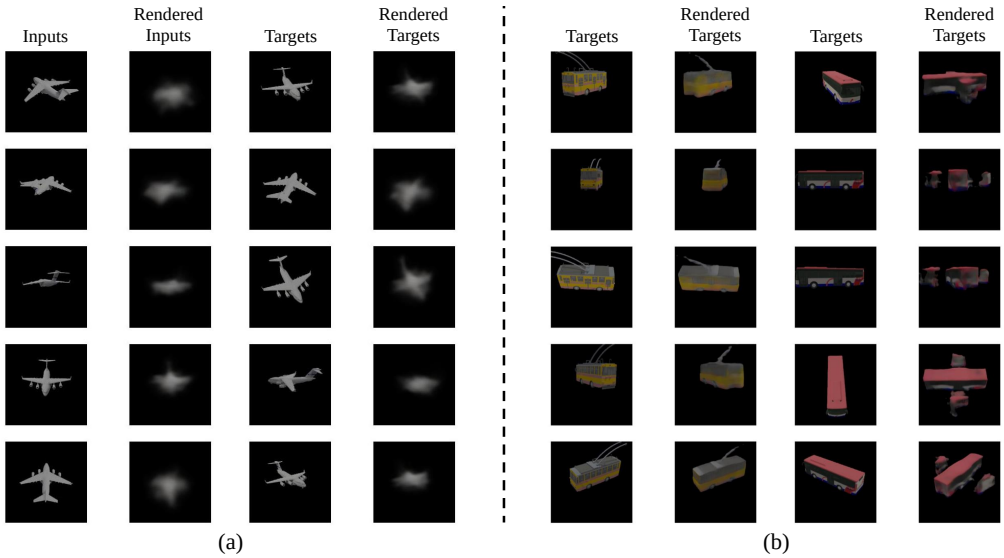

Figure 15: **Visualization of ablation experiments.** (a) Without the multi-view encoder, LEAP can only reconstruct noisy results. (b) With using the category-specific coordinate as the world coordinate, LEAP degenerates on novel categories. We show two examples, where LEAP successfully maps the information in a shared reconstruction space for the first example, but it fails in the second case.

D DETAILS

In this section, we introduce the details for training LEAP and baselines.

We note that as the neural volume is defined in the local camera coordinate of the canonical view, we render novel views using the relative camera poses rather than the absolute camera poses for training and testing LEAP. We keep the same setting for FORGE, SRT, and PixelNeRF. As the definition of world coordinate will not influence the performance of other baseline methods, i.e. SPARF, we use the absolute poses as performed in its official code. Besides, Zero123 does not have the concept of a 3D coordinate system. For FORGE, we perform 5,000-step optimization.

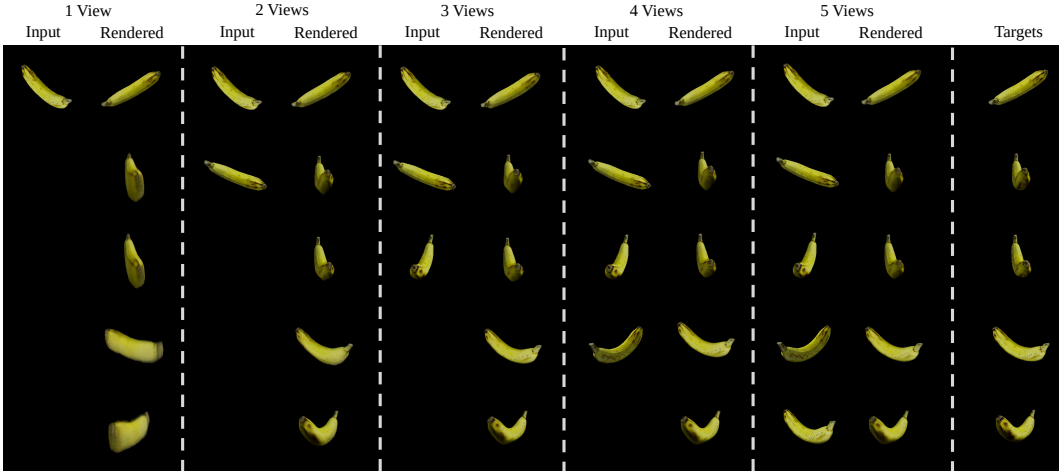

Figure 16: **Visualization of results with different number of input views.** LEAP fails to model the object with only one input view, while the performance with two views and five views is close enough.

The datasets have different numbers of views for each scene. The object-centric datasets have 100 (OmniObject3D dataset), 10 (Kubric dataset), and 12 views (Objaverse dataset). All views are sampled randomly. During testing, we use the first 5 images of each scene (with index 0-4) as inputs, and the other 5 images (with index 5-9) for evaluation. For the scene-level DTU dataset, we use images with indexes of 5, 15, 25, 35, and 45 as inputs and use images with indexes of 0, 10, 20, 30, and 40 for testing. This configuration ensures the coverage of the input and evaluation views. For training, we use 5 randomly sampled image sets from each scene as the inputs as well as targets. We keep the setting for training and testing all baseline methods the same as LEAP. During the evaluation, we perform multiple time inference, by setting each input view as canonical. We select the result having the largest PSNR on the input views. Note that the inference speed is calculated for the whole process rather than a single canonical index inference.