# FeatureNeRF: Learning Generalizable NeRFs by Distilling Foundation Models

Jianglong Ye[1] Naiyan Wang[2] Xiaolong Wang[1]
[1]UC San Diego [2]TuSimple

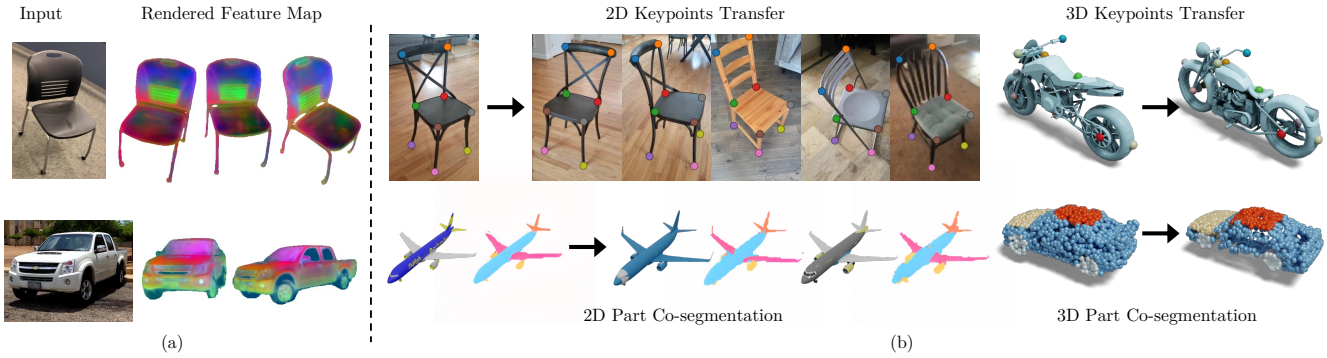



Figure 1: While most generalizable NeRFs focus on novel-view synthesis, we propose a framework named FeatureNeRF to learn 3D semantic representations by distilling vision foundation models. After distillation, FeatureNeRF allows to render novel-view feature maps given a single input image (a), which can be leveraged to various downstream tasks. Here, we show how we propagate part segmentation labels and keypoints to different views and instances in both 2D and 3D domains (b).

## Abstract

*Recent works on generalizable NeRFs have shown promising results on novel view synthesis from single or few images. However, such models have rarely been applied on other downstream tasks beyond synthesis such as semantic understanding and parsing. In this paper, we propose a novel framework named FeatureNeRF to learn generalizable NeRFs by distilling pre-trained vision foundation models (e.g., DINO, Latent Diffusion). FeatureNeRF leverages 2D pre-trained foundation models to 3D space via neural rendering, and then extract deep features for 3D query points from NeRF MLPs. Consequently, it allows to map 2D images to continuous 3D semantic feature volumes, which can be used for various downstream tasks. We evaluate FeatureNeRF on tasks of 2D/3D semantic keypoint transfer and 2D/3D object part segmentation. Our extensive experiments demonstrate the effectiveness of FeatureNeRF as a generalizable 3D semantic feature extractor. Our project page is available at* https://jianglongye.com/featurenerf/.

## 1. Introduction

Neural fields have emerged as a compelling paradigm for representing a variety of visual signals [8, 10, 31, 32, 37]. In particular, the Neural Radiance Fields (NeRF [32]), which implicitly encodes density and color via Multi-Layer Perceptrons (MLPs), has shown high quality novel view synthesis results from dense input images. A body of follow-up works [7, 27, 45, 54, 66] further reduce the dependency on dense inputs and generalizes NeRF to unseen objects by learning priors from large-scale multi-view image datasets. With the remarkable abilities on reconstruction and view synthesis of generalizable NeRFs, we ask the question: Can we adapt such models to learn 3D representations as foundations for general 3D applications (e.g., recognition, matching) beyond view synthesis?

Recent years have witnessed the rise of vision foundation models [5, 40, 43, 67] that are pre-trained on web-scale image datasets and demonstrate generalization capabilities across massive vision tasks (e.g., CLIP [40], DINO [5], Latent Diffusion [47]). The feature space constructed by foundation models captures rich semantic and structural information of 2D visual data and make it possible to identify object categories, parts and correspondences even without extra supervisions [1, 5, 30]. Motivated by these works, our goal is to leverage the powerful 2D foundations models to obtain generalizable 3D features.

In this paper, we present FeatureNeRF, a unified framework for learning generalizable NeRFs from distilling pre-trained 2D vision foundation models. Unlike previous gen-

eralizable NeRFs [45, 66], which utilize 2D encoder solely for novel view synthesis, FeatureNeRF explores the use of deep features extracted from NeRFs as generalizable 3D visual descriptors. We show that distilling 2D foundation models into 3D space via neural rendering equips the NeRF features with rich semantic information. As a result, FeatureNeRF allows to predict a continuous 3D semantic feature volume from a single or a few images, which can be applied to various downstream tasks such as semantic keypoint transfer and object part co-segmentation. Examples of these applications are shown in Fig. 1.

Specifically, we adopt an encoder to map 2D images to corresponding 3D NeRF volume similar to previous generalizable NeRFs. Apart from density and color, we propose to extract deep features of the query 3D points from the intermediate layers of NeRF MLP. To enrich semantic information of the NeRF features, we further transfer knowledge from the foundation models to the encoder via neural rendering during training: The rendered feature outputs should be consistent with the feature extracted from the foundation models, which is enforced by a distillation loss.

To evaluate FeatureNeRF, we tackle the tasks of 2D/3D semantic keypoint transfer and object part segmentation. To the best of our knowledge, our work is the first to resolve these 3D semantic understanding tasks without 3D supervision. We validate our framework with two foundation models: (i) DINO [5], a self-supervised vision transformer aware of object correspondences, and (ii) Latent Diffusion [47], a diffusion-based model that achieves state-of-the-art text-to-image generation performance. Our extensive experiments demonstrate the effectiveness of FeatureNeRF as a generalizable 3D semantic feature extractor.

## 2. Related Work

**Generalizable NeRFs.** In the past few years, neural fields have gained significant attention and led to rapid progress in representing various visual signals [8, 10, 25, 31, 32, 36, 37, 50, 52, 61]. In particular, NeRF [32] achieves photo-realistic results on novel view synthesis by mapping 3D coordinates and 2D viewing directions to density and color via MLPs. However, the original NeRF requires enormous posed images and time-consuming optimization for each single scene. To address these issues, a large number of follow-up methods [9, 23, 27, 45, 46, 54, 56, 66] propose to learn generalizable NeRFs from large-scale multi-view image datasets. For example, PixelNeRF [66] employs an image encoder to condition NeRF on image features, which enables novel views synthesis from a single image and generalizes NeRF to unseen objects. CodeNeRF [23] learns to disentangle shape and texture by learning separate embeddings, allowing shape and texture editing by varying the latent codes. Recent work TransINR [9] is proposed to infer NeRF parameters directly with a vision transformer to overcome the information bottleneck of encoder-decoder architecture. However, most of these works focus only on view synthesis. Our work differs from them by learning general-purpose 3D representations for multiple downstream tasks.

**Vision Foundation Models.** The term foundation model is introduced in [3] to refer to the model pre-trained from data at scale and capable of generalizing to a wide range of downstream tasks. After demonstrating huge impacts in NLP [4, 15, 41], a large family of vision foundation models [5, 40, 42, 43, 47, 58, 62, 67] have been proposed and effectively transferred to various vision tasks. For example, the CLIP model [40] is trained from large-scale image-text data using contrastive learning, and it is shown to be transferable to multiple tasks in a zero-shot manner. The DINO model [5] has shown object segment can emerge automatically with only self-supervision, and the learned feature can be applied in a wide range of visual correspondence and recognition tasks [1, 12, 30, 57]. Besides contrastive learning, recent text-conditioned generative model such as diffusion models [16, 21, 35, 47] have been introduced and shown astonishing performance on image generation. Subsequently, feature spaces learned by these generative models have also been used for recognition tasks such as semantic segmentation [2, 59]. In contrast to the success of 2D foundation models, the 3D counterparts are still suffering from the lack of large-scale annotated datasets and effective architectures [18, 49, 60]. In this paper, we propose to distill the features from 2D foundation models to 3D space via the generalizable NeRFs.

**Feature Distillation.** For the purpose of model compression and knowledge transfer, distillation has been widely studied by the community. After the pioneering work by Hinton et al. [20], which matches the softmax output distribution of the teacher model to that of the student, numerous methods have been proposed to tackle various tasks [19, 22, 38, 39, 53, 64]. Recently, researchers also propose to distill features from 2D models to 3D space by optimizing neural feature fields [26, 55]. Multiple editing tasks are shown as applications. However, these methods not only require test-time optimization for each single scene, but the learned features are also not generalizable to unseen objects, which makes them unsuitable for general semantic understanding tasks and differ from our work fundamentally.

**Semantic Correspondences.** Given a pair of visual observation, semantic correspondences learning aims to find corresponding points between them. Several supervised [13, 24, 51] and self-supervised [1, 11, 28] methods have been proposed to resolve this task in 2D and 3D domain respectively. In particular, Amir et al. [1] show that utilizing pre-trained 2D foundation models in a zero-shot manner can achieve competitive results with supervised methods on semantic correspondences. Cheng et al. [11] propose to learn point cloud correspondences via

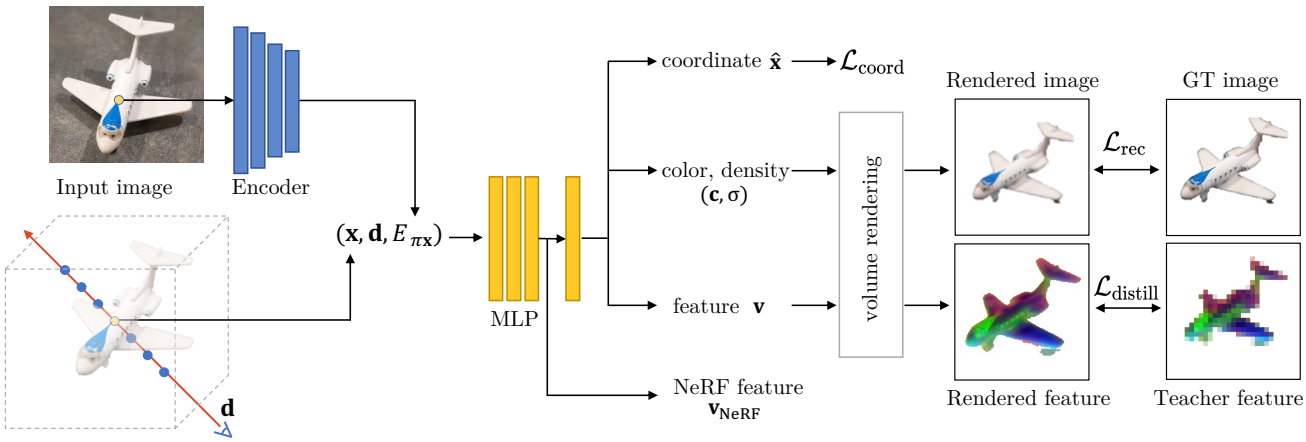



Figure 2: **Pipeline of FeatureNeRF**. Given a single image $I$ as input, FeatureNeRF adopts an encoder to extract the image feature $E_{\pi(\mathbf{x})}$, and then concatenate it with the query point $\mathbf{x}$ as well as the view direction $\mathbf{d}$ as the inputs for NeRF MLPs. Apart from density $\sigma$ and color $\mathbf{c}$, we add two MLP branches to predict the feature vector $\mathbf{v}$ and coordinate $\hat{\mathbf{x}}$, which are supervised by two novel loss terms $\mathcal{L}_{\text{distill}}$ and $\mathcal{L}_{\text{coord}}$ respectively. Consequently, we distill knowledge from 2D vision foundation models to FeatureNeRF. Besides, we propose to extract internal NeRF feature $\mathbf{v}_{\text{NeRF}}$ as 3D-consistent feature representation.

self-reconstruction and cross-reconstruction of 3D shapes. To the best of our knowledge, our work is the first to address semantic correspondences in both 2D and 3D space with only 2D observations.

## 3. Method

We present FeatureNeRF, a unified framework for learning generalizable NeRF from vision foundation models. We give an overview of generalizable NeRFs in Sec. 3.1 and elaborate our feature distillation process in Sec. 3.2. Then we introduce how to learn internal NeRF features for 3D semantic understanding in Sec. 3.3 and downstream applications in Sec. 3.4. The overall pipeline of FeatureNeRF is illustrated in Fig. 2.

### 3.1. Preliminary: Generalizable NeRF

Neural Radiance Fields (NeRF [32]) consists of two functions: $\sigma(\mathbf{x}) : \mathbb{R}^3 \mapsto \mathbb{R}_+$ that maps a 3D point $\mathbf{x}$ to the density $\sigma$ and $\mathbf{c}(\mathbf{x}, \mathbf{d}) : \mathbb{R}^{3\times3} \mapsto \mathbb{R}^3$ that maps a 3D point as well as a unit viewing direction $\mathbf{d}$ to color. The radiance field can be rendered and optimized via differentiable volume rendering [29]. Given a pixel's camera ray $\mathbf{r}(t) = \mathbf{o} + t\mathbf{d}$, which is defined by the camera origin $\mathbf{o} \in \mathbb{R}^3$, view direction $\mathbf{d}$ and depth $t$ with bounds $[t_n, t_f]$, the estimated color of the ray can be calculated by:

$$\hat{\mathbf{C}}(\mathbf{r}) = \int_{t_n}^{t_f} T(t)\sigma\left(\mathbf{r}(t)\right)\mathbf{c}\left(\mathbf{r}(t), \mathbf{d}\right)\mathrm{d}t, \tag{1}$$

where $T(t) = \exp\left(-\int_{t_n}^{t}\sigma(s)ds\right)$. In practice, the integral is approximated with numerical quadrature by sampling points along the ray. NeRF is optimized to a single scene with multi-view posed images by minimizing the following reconstruction loss:

$$\mathcal{L}_{\text{rec}} = \sum_{\mathbf{r}\in\mathcal{R}} \left\| \mathbf{C}(\mathbf{r}) - \hat{\mathbf{C}}(\mathbf{r}) \right\|_2^2, \tag{2}$$

where $\mathbf{C}(\mathbf{r})$ is the ground truth color of the ray and $\mathcal{R}$ is the set of rays generated from camera poses.

In order to generalize to novel scenes, the NeRF model can be conditioned on the input image $I \in \mathbb{R}^{H\times W\times 3}$:

$$\begin{aligned} \sigma(\mathbf{x}, I) &= g_\sigma\left(\mathbf{x}, f(I)_{\pi(\mathbf{x})}\right) \\ \mathbf{c}(\mathbf{x}, \mathbf{d}, I) &= g_{\mathbf{c}}\left(\mathbf{x}, \mathbf{d}, f(I)_{\pi(\mathbf{x})}\right), \end{aligned} \tag{3}$$

where $g_\sigma$ and $g_{\mathbf{c}}$ are MLPs that predict density and color respectively, $f$ is an image encoder and $\pi$ is the projection function.

As shown in the left part of Fig. 2, the image $I$ is firstly passed to an encoder $f_{\text{enc}}$ (blue blocks in the figure) to obtain a feature map $E = f(I)$. The query point $\mathbf{x}$ is projected onto the image plane using known pose and intrinsics to extract the corresponding feature vector $E_{\pi(\mathbf{x})}$. Then the feature vectors are concatenated with the positional-encoded point $\mathbf{x}$ and direction $\mathbf{d}$, and passed to subsequent MLPs $g_\sigma$ and $g_{\mathbf{c}}$ (yellow blocks in the figure) to predict appearance and geometry. When multi-view images are available, feature vectors from different views are aggregated with the average pooling before passing to MLPs.

### 3.2. Feature Distillation from Foundation Models

While most generalizable NeRFs only predict density $\sigma$ and color $\mathbf{c}$, it's possible to extend NeRF to predict other

quantities of interests. For example, SemanticNeRF [68] and PanopticNeRF [17] propose to add a branch to predict segmentation labels to achieve a 3D-consistent semantic segmentation of a scene. However, these methods require expensive semantic labels during optimization, which is impractical for general cases. In this paper, we aim to transfer knowledge from a pre-trained foundation model $f_{\text{teacher}}$ to our generalizable NeRFs to perform 3D semantic understanding. To this end, we add a branch to output a high-dimensional feature vector $\mathbf{v} \in \mathbb{R}^D$ for the query point $\mathbf{x}$, where $D$ is the feature channels. Similar to the color rendering (Eq. 1), we can aggregate the feature vectors along a ray as follows:

$$\begin{aligned} \hat{\mathbf{V}}(\mathbf{r}, I) &= \int_{t_n}^{t_f} T(t)\sigma\left(\mathbf{r}(t), I\right)\mathbf{v}\left(\mathbf{r}(t), \mathbf{d}, I\right)\mathrm{d}t \\ \mathbf{v}\left(\mathbf{x}, \mathbf{d}, I\right) &= g_{\mathbf{v}}\left(\mathbf{x}, \mathbf{d}, f(I)_{\pi(\mathbf{x})}\right), \end{aligned} \tag{4}$$

where $g_{\mathbf{v}}$ is the MLP that predicts feture vectors. Note that our model is still conditioned on the image $I$ in the above equation.

We then minimize the difference between rendered pixel feature vector $\hat{\mathbf{V}}$ and the teacher's feature $\mathbf{V} = f_{\text{teacher}}(I)_{\pi(\mathbf{x})}$. In this way, we distill the teacher network into our generalizable NeRFs via neural rendering. We add a distillation loss to penalize the difference:

$$\mathcal{L}_{\text{distill}} = \sum_{\mathbf{r}\in\mathcal{R}} \left\| \mathbf{V}(\mathbf{r}) - \hat{\mathbf{V}}(\mathbf{r}) \right\|_2^2. \tag{5}$$

FeatureNeRF can be trained jointly for image reconstruction and feature distillation by combing two losses (Eq. 2 and Eq. 5). We show both color rendering and feature rendering processes in the right part of Fig. 2. We emphasize that, after distillation, FeatureNeRF can obtain a 3D semantic feature function $\mathbf{v}$ in a single forward pass, which can be used for downstream applications. Our experiments show that the feature function $\mathbf{v}$, learned with only 2D observation, contains accurate 3D semantic information.

Our framework can be built on top of any foundation model, and in this work we employ DINO [5] and Latent Diffusion [47] as teacher networks. DINO is a vision transformer trained with self-distillation, and we simply extract features from the deepest layer as teacher features. Latent Diffusion firstly transforms input image $I$ to the latent space, and utilizes a U-Net [48] architecture to estimate the noise for the backward diffusion process. Besides, the denoising module can be conditioned on inputs like text and segmentation maps. We add noise to the original image, condition the pre-trained model with fixed language prompts (e.g. "Car", "Chair") and extract features from the intermediate layers of U-Net. Distilling these foundation models that pre-trained on large-scale image datasets brings open-world knowledge to our generalizable 3D representation.

### 3.3. Learning Internal NeRF Features for 3D Semantic Understanding

Although we have distilled features from foundation models, it is still questionable whether the final feature output is best suitable for 3D semantic understanding. Here we explore using internal NeRF features as view-independent representations for 3D semantic understanding and introduce a new coordinate loss for learning spatial-aware NeRF features.

While previous works only utilize final outputs of MLPs, we explore whether we can use features from intermediate layers as continuous 3D visual descriptors. Given a input image $I$, we firstly learn a function $\mathbf{v}_{\text{NeRF}}(\mathbf{x}, I)$ to predict NeRF features and utilize several shallow MLPs to predict other quantities:

$$\begin{aligned} \mathbf{v}_{\text{NeRF}}(\mathbf{x}, I) &= g_{\text{NeRF}}\left(\mathbf{x}, f(I)_{\pi(\mathbf{x})}\right) \\ \sigma(\mathbf{x}, I) &= g_{\sigma}\left(\mathbf{v}_{\text{NeRF}}(\mathbf{x}, I)\right) \\ \mathbf{c}(\mathbf{x}, \mathbf{d}, I) &= g_{\mathbf{c}}\left(\mathbf{d}, \mathbf{v}_{\text{NeRF}}(\mathbf{x}, I)\right) \\ \mathbf{v}(\mathbf{x}, \mathbf{d}, I) &= g_{\mathbf{v}}\left(\mathbf{d}, \mathbf{v}_{\text{NeRF}}(\mathbf{x}, I)\right), \end{aligned} \tag{6}$$

where $g_{\text{NeRF}}$ is MLP that predicts NeRF features. Note that the proposed function $\mathbf{v}_{\text{NeRF}}$ can be learned without the feature distillation introduced in Sec. 3.2, therefore it can be applied to all generalizable NeRFs. The feature extraction process is demonstrated in the bottom of Fig. 2. We compare the performance of NeRF features with and without feature distillation in our experiments.

Even for the feature distillation version, since teacher's features $\mathbf{V}$ are always not 3D-consistent, using a view-independent representation $\mathbf{v}_{\text{NeRF}}$ and modeling view-dependent effect using another MLP $g_{\mathbf{v}}$ that conditioned on view direction $\mathbf{d}$ further boosts the performance (See Sec. 4.4 for the ablation study).

The supervision of most generalizable NeRFs are RGB values, which do not contain spatial information. To enhance the spatial perception of the NeRF feature, we propose to utilize another MLP branch $g_{\mathbf{x}}$ (shown at the top of Fig. 2) to regress the input coordinates $\mathbf{x}$ given the NeRF feature $\mathbf{v}_{\text{NeRF}}$:

$$\hat{\mathbf{x}} = g_{\mathbf{x}}\left(\mathbf{v}_{\text{NeRF}}(\mathbf{x}, I)\right). \tag{7}$$

We add a cycle-consistent loss to penalize the difference:

$$\mathcal{L}_{\text{coord}} = \sum_{\mathbf{r}\in\mathcal{R}} \sum_{t=t_n}^{t_f} \left\| \mathbf{r}(t) - g_{\mathbf{x}}\left(\mathbf{v}_{\text{NeRF}}\left(\mathbf{r}(t), I\right)\right) \right\|_2^2. \tag{8}$$

The final loss function is the weighted sum of all three losses: $\mathcal{L} = \mathcal{L}_{\text{rec}} + \lambda_{\text{distill}}\mathcal{L}_{\text{distill}} + \lambda_{\text{coord}}\mathcal{L}_{\text{coord}}$, where $\lambda_{\text{distill}}$ and $\lambda_{\text{coord}}$ are weights for different losses. All losses and their forward flows are shown in Fig. 2.

### 3.4. Applications of FeatureNeRF

We demonstrate the effectiveness of the learned 3D semantic NeRF feature function $\mathbf{v}_{\text{NeRF}}$ on various downstream applications: 2D/3D semantic keypoint transfer and object part segmentation. We deliberately apply simple, zero-shot methodologies on NeRF features, without any fine-tuning or post-process, to validate the proposed representations.

**2D Tasks.** Given a single image $I$, we can render its NeRF feature map $F \in \mathbb{R}^{H\times W\times D}$ using Eq. 4. Then we can render novel-view feature map $F'$ from other viewpoints. For feature vectors $\mathbf{V}$ and $\mathbf{V}'$ of two pixels from $F$ and $F'$, we use cosine similarity to measure their distance in the feature space: $D(\mathbf{V},\mathbf{V}') = \frac{\mathbf{V}\cdot\mathbf{V}'}{\|\mathbf{V}\|_2\cdot\|\mathbf{V}'\|_2}$. For the part co-segmentation task, for each pixel feature $\mathbf{V}'$ in $F'$, we take the segmentation label of its closest pixel feature $\mathbf{V}$ in $F$ as the predicted segmentation label. For the keypoint transfer task, we adopt a similar process, for the feature $\mathbf{V}$ of each keypoint in $F$, we take the pixel location of its closest feature in $F'$ as the predicted keypoint.

Since FeatureNeRF is a generalizable NeRF conditioned on the input image, it can be further applied to cross-instance tasks in addition to novel-view tasks for a single instance. Given images $I_1$ and $I_2$ of two instances, we can render their feature maps $F_1$ and $F_2$. Then we can resolve semantic correspondence tasks in a process similar to the novel-view tasks.

**3D Tasks.** The FeatureNeRF model learned with only 2D observations can also be leveraged to 3D tasks. Given images $I_1$ and $I_2$ of two instances, we can construct two continuous 3D feature fields. For feature vectors $\mathbf{v}_1$ and $\mathbf{v}_2$ of two 3D points $\mathbf{x}_1$ and $\mathbf{x}_2$, we still utilize cosine similarity to measure their distance: $D(\mathbf{v}_1,\mathbf{v}_2) = \frac{\mathbf{v}_1\cdot\mathbf{v}_2}{\|\mathbf{v}_1\|_2\cdot\|\mathbf{v}_2\|_2}$. Then we can resolve semantic correspondence tasks in a process similar to the 2D tasks.

## 4. Experiments

### 4.1. Experimental Setting

**Datasets.** Our experiments are mainly conducted on 6 categories from the ShapeNet [6] dataset: Chair, Car, Airplane, Table, Bottle and Motorcycle. We evaluate our model using annotations from from KeypointNet [65], ShapeNet part dataset [63] and PartNet [34]. In addition, we train our model on the real-world CO3D [45] dataset and evaluate its keypoint transfer performance on the Spair [33] dataset. For ShapeNet, we split each category into training (70%), validation (10%), and testing (20%) splits. All shapes are normalized so that the longest edges of the bounding box are equal. For the training set, 50 random camera poses from the upper hemisphere are sampled. For validation and testing sets, 50 fixed camera poses are used. We employ blender [14] to render RGB images, with a resolution of $128\times128$, the same as in PixelNeRF [66]. For PartNet, we use level-1 annotations. All annotations are in 3D and can be used for the evaluation of 3D tasks directly. For the evaluation of 2D tasks, we employ PyTorch3D [44] rasterizer to render 2D part segmentation labels and 2D keypoints.

**Baselines.** We mainly compare FeatureNeRF quantitatively and qualitatively to two foundation models DINO [5] and Latent Diffusion [47]. We extend PixelNeRF [66] with the mechanisms mentioned in Sec. 3.3 to make it possible for semantic understanding tasks and report its performance as "NeRF feature" in all results. In addition, for the 2D co-segmentaion task, we re-implement a one-shot generalizable SemanticNeRF* [68] by adding a semantic branch in PixelNeRF, and train it with the source segmentation labels.

**Implementation Details.** Following PixelNeRF [66], we employ a ResNet-34 model pre-trained on ImageNet as the image encoder $f$. The batch size is 4 (objects) and 1024 rays per object. We train a single model for each object category for 500k steps. The weights for different losses are $\lambda_{\text{distill}} = 0.25$ and $\lambda_{\text{coord}} = 0.25$. The dimension of the internal NeRF feature is 512. MLP for NeRF feature $g_{\text{NeRF}}$ is 4-layer, all other shallow MLPs for final outputs ($g_\sigma$, $g_{\mathbf{c}}$, $g_{\mathbf{v}}$ and $g_{\mathbf{x}}$) are one-layer.

**Teacher Networks.** We employ DINO [5] and Latent Diffusion [47] as teacher networks, which are pre-trained and publicly available. The patch size of DINO is 8. The final feature map has dimension $32\times32\times384$. For Latent Diffusion, we extract the outputs of the 4th layer in U-Net as teacher features. The language prompts used to condition denoising modules are class names (e.g. "Chair", "Car") from ShapeNet. The final feature map has dimension $128\times128\times960$. The weights of teacher networks are fixed during distillation.

### 4.2. 2D Semantic Understanding Tasks

As mentioned in Sec. 3.4, we evaluate FeatureNeRF on tasks of 2D keypoints transfer and part segmentation labels transfer under two settings: (i) *Cross-instance*: given images of two instances, we transfer keypoints/segmentation labels from the source image to the target image and compare the transferred labels in the target image with the ground truth. (ii) *Novel-view*: given a single image of an instance, we render a novel-view feature map, transfer labels from the source viewpoint to the target viewpoint and compare the transferred labels in the target viewpoint with the ground truth.

**Metrics.** For the task of 2D keypoints transfer, we report the percentage of predicted keypoints whose distances from their corresponding ground truths are below thresholds of $(2.5, 5.0, 7.5, 10.0)$ pixels in the target image. We denote this percentage as *Correspondence Accuracy* in the following. For the task of part segmentation labels trans-

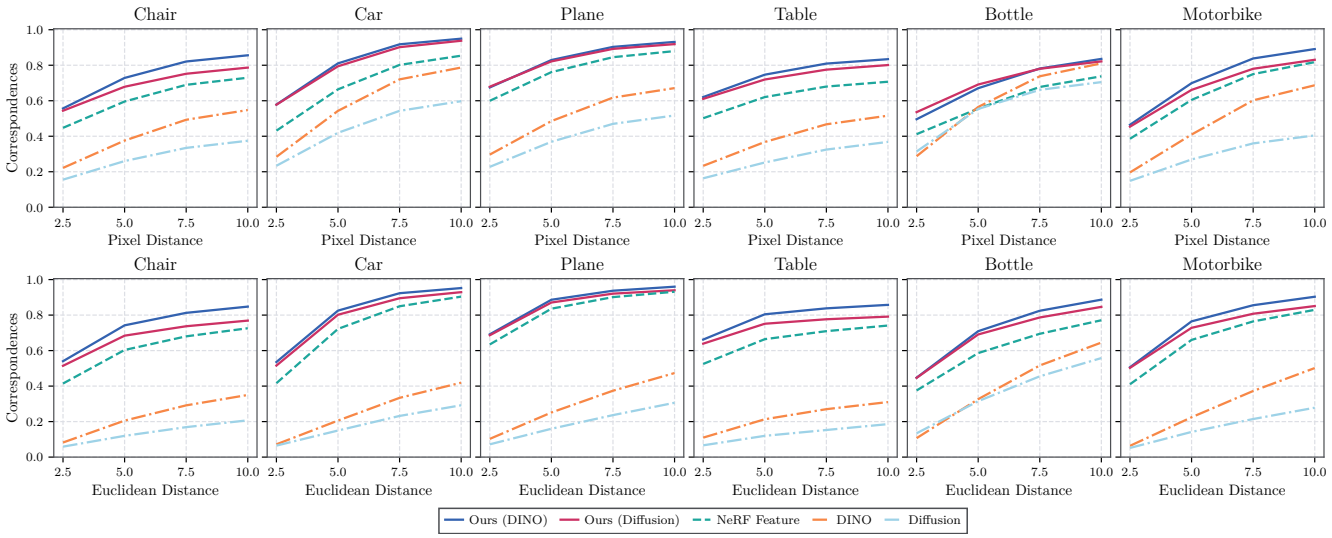
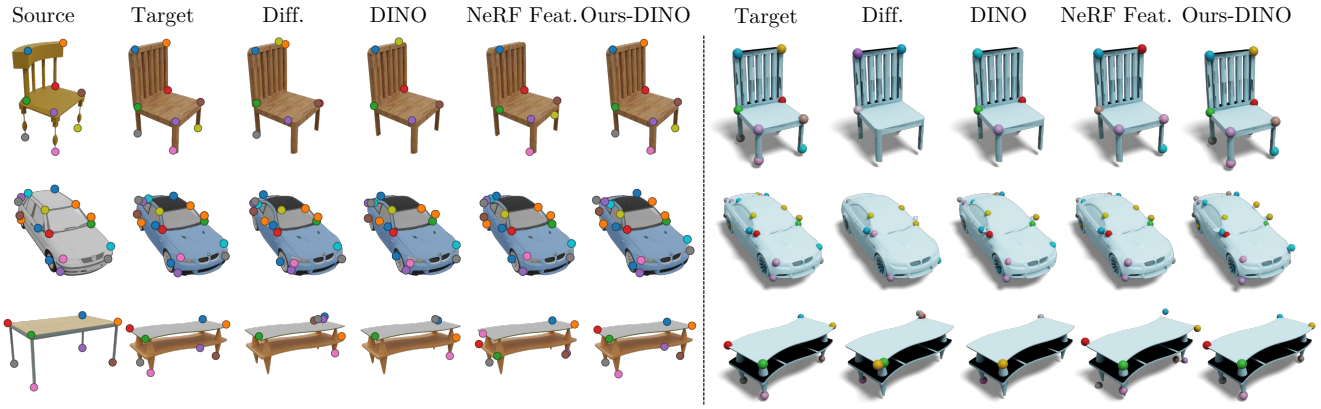



Figure 3: **Correspondence accuracy for cross-instance semantic keypoints transfer**. The first row is for 2D keypoints transfer and the second row is for 3D. Our approach distilled with different features consistently outperforms baselines for all categories in both 2D and 3D domains.




(a) 2D results (b) 3D results

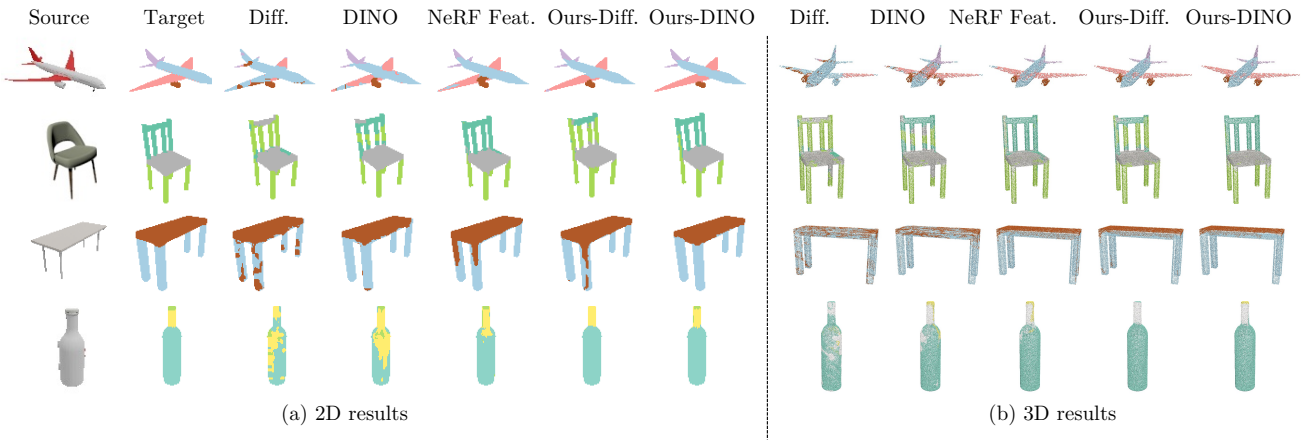
Figure 4: **Qualitative results for cross-instance semantic keypoints transfer**. Both 2D (a) and 3D (b) results are presented here. Each row contains a source image with keypoints annotations and its pairwise transfer results.

fer, we calculate the mean intersection over union (*mIoU*) over every part category for each object class. For both settings, we randomly generate 1000 combinations per category. While generating, we make sure that the source viewpoints/instances and target viewpoints/instances have intersecting keypoints/segmentation labels.

**2D Keypoints Transfer.** We report correspondence accuracy of 2D cross-instance keypoints transfer in the first row of Fig. 3. We show that for all 6 categories, keypoints transferred via our proposed method are more accurate than baselines. The performances of distilling diffusion and DINO features are similar. Using NeRF feature without distilling foundation models (denoted as "NeRF feature" in the figure) also achieves a reasonable performance compared to 2D foundation model baselines. Fig. 4 (a) shows the qualitative results for 2D keypoints transfer. It can be seen that despite various appearances and structures of instances, our method can successfully transfer keypoints based on semantic understanding, while baselines often fails.

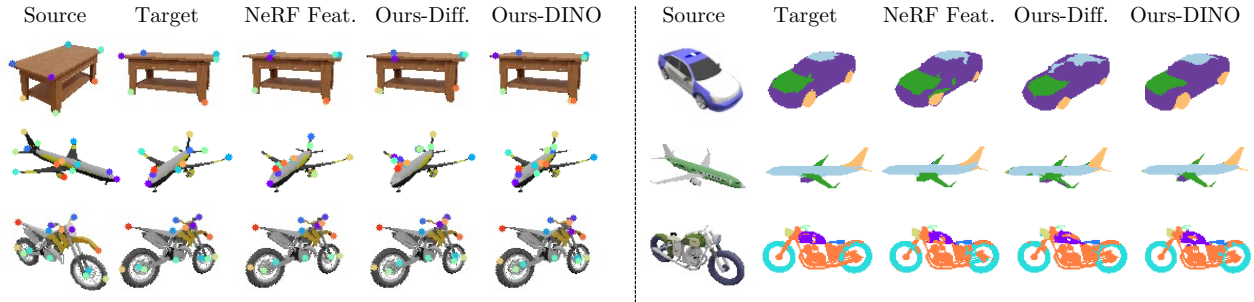
FeatureNeRF learns a 3D representation from 2D observations, which allows to synthesize novel-view feature maps for different viewpoints and performs keypoints transfer on top of it. Fig. 6 (a) shows qualitative results of novel-view keypoints transfer. Since 2D foundation models can not synthesize novel-view feature maps, we do not apply them to this novel-view setting.

**2D Part Segmentation Label Transfer.** We report mIoU results of 2D cross-instance part segmentation label transfer in the left part of Tab. 1. FeatureNeRF distilled with DINO features significantly outperforms other approaches




(a) 2D results (b) 3D results

Figure 5: **Qualitative results for cross-instance part segmentation label transfer**. Each row contains a source image and its 2D/3D transfer results. After distilling, FeatureNeRF learns richer semantic information, produces better boundaries and preserves details like small parts. Note that the segmentation label for the source instance is omitted.




(a) Keypoints transfer (b) Part co-segmentation

Figure 6: **Qualitative results of novel-view keypoints transfer and part co-segmentation.** FeatureNeRF learns a 3D representation from 2D observations, which makes it possible to synthesize feature maps from other viewpoints and transfer keypoints and segmentation labels to them.

| | 2D part co-segmentation | | | | | | 3D part co-segmentation | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Chair | Car | Plane | Table | Bottle | Motorbike | Chair | Car | Plane | Table | Bottle | Motorbike |
| SemanticNeRF* | 50.32 | 34.61 | 56.19 | 54.87 | 51.58 | 27.62 | | | | - | | |
| Diffusion Baseline | 41.59 | 42.46 | 44.60 | 57.77 | 41.72 | 25.50 | 33.60 | 24.74 | 30.45 | 44.34 | 53.81 | 20.01 |
| DINO Baseline | 62.43 | 54.59 | 57.81 | 64.01 | 59.73 | 37.05 | 52.67 | 29.72 | 40.16 | 49.98 | 70.79 | 29.25 |
| NeRF Feature | 72.02 | 58.57 | 72.57 | 70.41 | 55.87 | 44.00 | 65.23 | 59.19 | 71.23 | 61.64 | 66.63 | 44.88 |
| Ours (Diffusion) | 65.39 | 63.02 | 72.95 | 70.59 | 53.91 | 44.84 | 63.65 | 61.27 | 73.41 | 63.93 | 66.91 | 49.05 |
| Ours (DINO) | **76.55** | **66.85** | **74.60** | **74.06** | **61.13** | **49.56** | **73.85** | **64.99** | **74.20** | **66.52** | **72.33** | **52.56** |

Table 1: **Cross-instance part segmentation label transfer results**. We report mIoU of part co-segmentation for each category. The left part is for 2D and the right is for 3D. By distilling features to the 3D space, our proposed representation contains richer semantic information which is apt for this co-segmentation task.

for all 6 categories. However, the performance of FeatureNeRF distilled with diffusion features is not as good as the DINO one. This is possible since the keypoints transfer task may focus on predicting accurate locations of sparse pixels that contain the richest semantic information, but part co-segmentation requires denser correspondences. Qualitative results are shown Fig. 5 (a). We can find that NeRF feature often produces unexpected artifacts. We attribute this phenomenon to NeRF feature's training only relying on RGB information. In contrast, by distilling pre-trained features, our method produces better boundaries and preserves details like small parts. Note that the segmentation label of the source object is also required during the transfer process, which is omitted in the figure.

We also perform novel-view part co-segmentation and report both quantitative and qualitative results in Tab. 2 and Fig. 6 (b) respectively. The rendered feature maps from other viewpoints still exhibit promising performance for co-

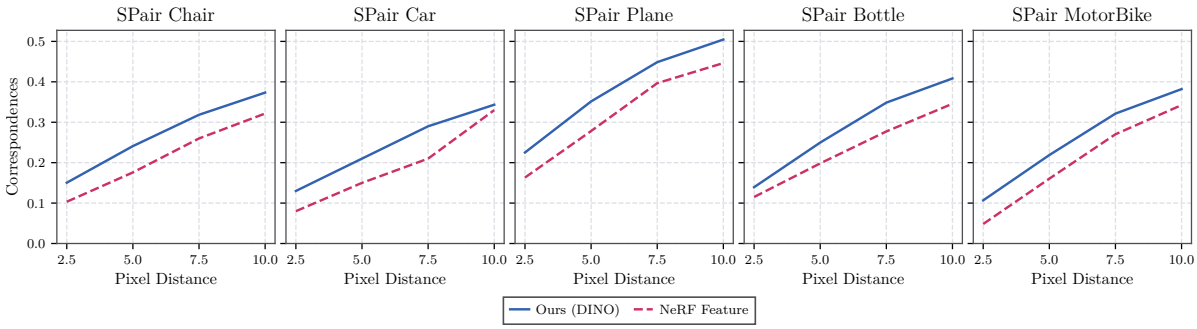



Figure 7: **Keypoints transfer on SPair**. Our method outperforms NeRF feature on 5 categories on the real-world dataset.


Figure 8: **Keypoints and segmentation label transfer on CO3D**. After training on CO3D, our method can accurately transfer keypoints and segmentation labels with in-the-wild image inputs

segmentation, which proves that FeatureNeRF learns a 3D-consistent feature representation.

**Real-world Experiments.** To show the generalizability to real-world images, we fine-tune our model on CO3D datasets and evaluate its keypoints transfer performance on two categories from the SPair dataset. The results in Fig. 7 confirm the effectiveness of our approach over NeRF feature on real images. The qualitative results on CO3D are shown in Fig. 8. Even with in-the-wild images, our method can still transfer keypoints and segmentation labels accurately. Note that pre-process steps (e.g. segment and normalize the foreground object) are required.

### 4.3. 3D Semantic Understanding Tasks

We further validate our proposed method on 3D semantic understanding tasks, which aims to find semantic correspondences between two sets of 3D points based on image observations. We only evaluate under the cross-instance setting for 3D tasks.

**Metrics.** The metrics for 3D tasks are 3D versions of their counterparts in 2D tasks. For the keypoints transfer, we utilize Euclidean distance instead of 2D pixel distance with

| | Chair | Car | Plane | Table | Bottle | Motorbike |
|---|---|---|---|---|---|---|
| NeRF Feat. | 80.73 | 75.10 | 69.45 | 84.65 | 87.14 | 67.49 |
| Ours (Diff.) | 73.45 | 71.16 | 59.16 | 83.91 | **88.01** | 64.28 |
| Ours (DINO) | **81.93** | **76.50** | **71.57** | **87.97** | 87.89 | **68.22** |

Table 2: **Novel-view 2D part segmentation label transfer results**. The rendered feature maps from other viewpoints still exhibit promising performance for co-segmentation, which proves that FeatureNeRF learns a 3D-consistent feature representation.

the threshold of $(0.025, 0.05, 0.075, 0.1)$ in the normalized space. For the part co-segmentation, we calculate mIoU based on point clouds instead of pixels. We randomly generate 1000 combinations per category for the evaluation.

**3D Keypoints Transfer.** For each 3D keypoint from the source shape, we try to find its corresponding point from a set of sampled candidate points from the target shape. For DINO and diffusion baselines, we simply project the 3D points onto 2D feature maps and utilize the interpolated feature vectors for matching. As those features are not 3D-aware, it is expected that their performance will be suboptimal. The quantitative results are shown in the second row of Fig 3. We can find that our method has a larger advantage in 3D tasks, which can also be observed from the qualitative results in Fig 4. Our learned 3D representation contains accurate semantic information and thus are able to transfer 3D keypoints in a more accurate way.

**3D Part Segmentation Label Transfer.** Similar to keypoints transfer, we try to propagate segmentation labels from the source point cloud to the target point cloud based on image observations. Note that the point cloud is only used as a query for segmentation, our method constructs a semantic feature volume for the whole 3D space and allows predicting feature vectors for any given 3D points. The right part of Tab 1 reports mIoUs for all methods. FeatureNeRF consistently outperforms all baselines in all categories, suggesting that distilling feature to 3D-aware representation leads to improved 3D co-segmentation performance. Fig. 5 presents qualitative results. Note that even for occluded areas in 2D images, FeatureNeRF can transfer its labels cor-

| Method | Chair | Plane |
|---|---|---|
| w/o $\mathcal{L}_{\text{coord}}$ | 75.39 | 72.83 |
| w/o internal features | 74.62 | 72.29 |
| full model | **76.55** | **74.60** |

Table 3: **Ablation study**. The coordinate loss and the use of internal NeRF features both boost performance.

| | | Chair | Car | Plane | Table | Bottle | Motorbike |
|---|---|---|---|---|---|---|---|
| PSNR | PixelNeRF [66] | 23.29 | 22.86 | 24.46 | 25.61 | **26.04** | 20.36 |
| | Ours (Diff.) | **23.36** | **23.09** | 24.39 | 25.42 | 25.75 | **20.64** |
| | Ours (DINO) | 23.20 | 22.92 | **24.49** | **25.69** | 26.01 | 20.59 |
| SSIM | PixelNeRF [66] | 0.92 | 0.91 | **0.93** | 0.89 | 0.89 | 0.80 |
| | Ours (Diff.) | **0.92** | 0.91 | 0.91 | 0.87 | 0.89 | **0.81** |
| | Ours (DINO) | 0.91 | **0.91** | 0.91 | **0.89** | **0.90** | 0.80 |

Table 4: **Novel-view synthesis results**. The proposed distillation process does not hurt the performance of novel-view synthesis, our method achieves comparable performance with PixelNeRF on novel-view synthesis.

rectly in 3D space.

### 4.4. Ablation Study

We mainly ablate the coordinate loss $\mathcal{L}_{\text{coord}}$ and the use of internal NeRF features $\mathbf{v}_{\text{NeRF}}$ (instead of final output feature vector $\mathbf{v}$) for semantic understanding. We conduct experiments of 2D parts co-segmentation on the Chair and Plane classes. The mIoU results are reported in Tab. 3. We can see that both design choices boost performance.

In addition, we compare the performance of novel-view synthesis with PixelNeRF [66] to see if the proposed feature distillation technique hurts the synthesis ability. Both FeatureNeRF and PixelNeRF are trained on our rendered dataset with the same parameters. From Tab. 4, we see that our method achieves comparable performance with PixelNeRF on novel-view synthesis.

### 4.5. Editing Applications

The learned FeatureNeRF model can also be leveraged to editing applications. Here, we take the 3D part texture swapping as an example. Given a source image and its part segmentation label, we can construct a 3D feature volume for the source image. Then, for a target image, we also construct its 3D feature volume and transfer the segmentation label from the source image to it. When rendering the target image, for a 3D point $\mathbf{x}_{\text{tgt}}$ that belongs to the part of interests (e.g. chair back) in the target feature volume, we find its closet point in the source feature volume:

$$\mathbf{x}_{\text{closet}} = \arg\min_{\mathbf{x}_{\text{src}}} \| v_{\text{NeRF}}(\mathbf{x}_{\text{tgt}}) - v_{\text{NeRF}}(\mathbf{x}_{\text{src}}) \|_2,$$

where $\mathbf{x}_{\text{src}} \in \mathbf{X}_{\text{src}}$ and $\mathbf{X}_{\text{src}}$ is the set of sampled points in the source feature volume. Finally, we use the color of the closet point $\mathbf{x}_{\text{closet}}$ to replace the original color of $\mathbf{x}_{\text{tgt}}$ for the rendering.

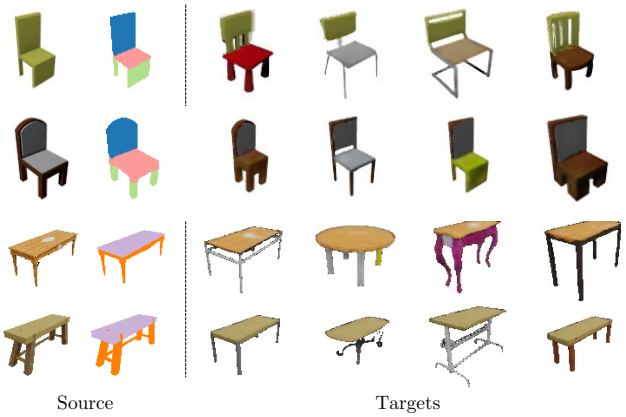



Figure 9: **Editing application: 3D part texture swapping**. The part textures from source images are successfully transferred to the target images. Note that the edited target objects can also be rendered from novel views, please see our video for 3D visualizations.

The results are shown in Fig. 9. We can find that the part textures from source images are successfully transferred to the target images. Besides, the details are also well preserved (the brown boundary from the source image at the second row still exists in the target images). Note that the edited target instances can also be rendered from novel views, please see our video for 3D visualizations.

## 5. Conclusion

In contrast to the success of 2D foundation models, it is still unclear how to build foundation models in 3D given the lack of web-scale datasets and effective architecture. In this paper, we present FeatureNeRF, a unified framework for learning generalizable NeRFs from distilling 2D vision foundation models. FeatureNeRF explores the use of internal NeRF features as 3D visual descriptors and distills knowledge from 2D foundation models into 3D space via neural rendering. Given a single image, FeatureNeRF allows to predict a 3D semantic feature representation, which can be leveraged for downstream tasks. Specifically, we demonstrate the effectiveness of FeatureNeRF on the tasks of 2D/3D keypoints transfer and part co-segmentation from a single image. In addition, FeatureNeRF can also serve as a general-purpose feature extractor for downstream 3D tasks such as open-world 3D classification/segmentation and robotics tasks.

**Acknowledgements.** Prof. Wang's lab was supported, in part, by Amazon Research Award, Sony Research Award, Adobe Data Science Research Award, and gifts from Qualcomm.

## References


[1] Shir Amir, Yossi Gandelsman, Shai Bagon, and Tali Dekel. Deep vit features as dense visual descriptors. *arXiv preprint arXiv:2112.05814*, 2(3):4, 2021. 1, 2

[2] Dmitry Baranchuk, Ivan Rubachev, Andrey Voynov, Valentin Khrulkov, and Artem Babenko. Label-efficient semantic segmentation with diffusion models. *ArXiv*, abs/2112.03126, 2022. 2

[3] Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021. 2

[4] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020. 2

[5] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9650–9660, 2021. 1, 2, 4, 5

[6] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015. 5

[7] 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. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14124–14133, 2021. 1

[8] Yinbo Chen, Sifei Liu, and Xiaolong Wang. Learning continuous image representation with local implicit image function. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8628–8638, 2021. 1, 2

[9] Yinbo Chen and Xiaolong Wang. Transformers as meta-learners for implicit neural representations. In *European Conference on Computer Vision*, pages 170–187. Springer, 2022. 2

[10] Zhiqin Chen and Hao Zhang. Learning implicit fields for generative shape modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5939–5948, 2019. 1, 2

[11] An-Chieh Cheng, Xueting Li, Min Sun, Ming-Hsuan Yang, and Sifei Liu. Learning 3d dense correspondence via canonical point autoencoder. In *Advances in Neural Information Processing Systems*, 2021. 2

[12] Subhabrata Choudhury, Iro Laina, C. Rupprecht, and Andrea Vedaldi. Unsupervised part discovery from contrastive reconstruction. In *NeurIPS*, 2021. 2

[13] Christopher Bongsoo Choy, Jaesik Park, and Vladlen Koltun. Fully convolutional geometric features. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 8957–8965, 2019. 2

[14] Blender Online Community. *Blender - a 3D modelling and rendering package*. Blender Foundation, Stichting Blender Foundation, Amsterdam, 2018. 5

[15] 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. 2

[16] Prafulla Dhariwal and Alex Nichol. Diffusion models beat gans on image synthesis. *ArXiv*, abs/2105.05233, 2021. 2

[17] Xiao Fu, Shangzhan Zhang, Tianrun Chen, Yichong Lu, Lanyun Zhu, Xiaowei Zhou, Andreas Geiger, and Yiyi Liao. Panoptic nerf: 3d-to-2d label transfer for panoptic urban scene segmentation. *arXiv preprint arXiv:2203.15224*, 2022. 4

[18] Huy Ha and Shuran Song. Semantic abstraction: Open-world 3d scene understanding from 2d vision-language models. *ArXiv*, abs/2207.11514, 2022. 2

[19] Byeongho Heo, Jeesoo Kim, Sangdoo Yun, Hyojin Park, Nojun Kwak, and Jin Young Choi. A comprehensive overhaul of feature distillation. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1921–1930, 2019. 2

[20] Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. Distilling the knowledge in a neural network. *ArXiv*, abs/1503.02531, 2015. 2

[21] Jonathan Ho, Ajay Jain, and P. Abbeel. Denoising diffusion probabilistic models. *ArXiv*, abs/2006.11239, 2020. 2

[22] Zehao Huang and Naiyan Wang. Like what you like: Knowledge distill via neuron selectivity transfer. *ArXiv*, abs/1707.01219, 2017. 2

[23] Wonbong Jang and Lourdes Agapito. Codenerf: Disentangled neural radiance fields for object categories. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12949–12958, 2021. 2

[24] Wei Jiang, Eduard Trulls, Jan Hendrik Hosang, Andrea Tagliasacchi, and Kwang Moo Yi. Cotr: Correspondence transformer for matching across images. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6187–6197, 2021. 2

[25] Tero Karras, Miika Aittala, Samuli Laine, Erik Härkönen, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Alias-free generative adversarial networks. *Advances in Neural Information Processing Systems*, 34:852–863, 2021. 2

[26] Sosuke Kobayashi, Eiichi Matsumoto, and Vincent Sitzmann. Decomposing nerf for editing via feature field distillation. *arXiv preprint arXiv:2205.15585*, 2022. 2

[27] Kai-En Lin, Lin Yen-Chen, Wei-Sheng Lai, Tsung-Yi Lin, Yi-Chang Shih, and Ravi Ramamoorthi. Vision transformer for nerf-based view synthesis from a single input image. In *WACV*, 2023. 1, 2

[28] Feng Liu and Xiaoming Liu. Learning implicit functions for topology-varying dense 3d shape correspondence. *ArXiv*, abs/2010.12320, 2020. 2

[29] Nelson Max. Optical models for direct volume rendering. *IEEE Transactions on Visualization and Computer Graphics*, 1(2):99–108, 1995. 3

[30] Luke Melas-Kyriazi, Christian Rupprecht, Iro Laina, and Andrea Vedaldi. Deep spectral methods: A surprisingly

strong baseline for unsupervised semantic segmentation and localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8364–8375, 2022. 1, 2

[31] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4460–4470, 2019. 1, 2

[32] 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. *Communications of the ACM*, 65(1):99–106, 2021. 1, 2, 3

[33] Juhong Min, Jongmin Lee, Jean Ponce, and Minsu Cho. Spair-71k: A large-scale benchmark for semantic correspondence. *ArXiv*, abs/1908.10543, 2019. 5

[34] Kaichun Mo, Shilin Zhu, Angel X. Chang, Li Yi, Subarna Tripathi, Leonidas J. Guibas, and Hao Su. PartNet: A large-scale benchmark for fine-grained and hierarchical part-level 3D object understanding. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 5

[35] Alex Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. *ArXiv*, abs/2102.09672, 2021. 2

[36] Michael Niemeyer, Lars Mescheder, Michael Oechsle, and Andreas Geiger. Occupancy flow: 4d reconstruction by learning particle dynamics. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 5379–5389, 2019. 2

[37] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 165–174, 2019. 1, 2

[38] Wonpyo Park, Dongju Kim, Yan Lu, and Minsu Cho. Relational knowledge distillation. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3962–3971, 2019. 2

[39] Nikolaos Passalis and Anastasios Tefas. Learning deep representations with probabilistic knowledge transfer. In *ECCV*, 2018. 2

[40] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021. 1, 2

[41] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, Peter J Liu, et al. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(140):1–67, 2020. 2

[42] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022. 2

[43] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *International Conference on Machine Learning*, pages 8821–8831. PMLR, 2021. 1, 2

[44] Nikhila Ravi, Jeremy Reizenstein, David Novotny, Taylor Gordon, Wan-Yen Lo, Justin Johnson, and Georgia Gkioxari. Accelerating 3d deep learning with pytorch3d. *arXiv:2007.08501*, 2020. 5

[45] Jeremy Reizenstein, Roman Shapovalov, Philipp Henzler, Luca Sbordone, Patrick Labatut, and David Novotny. Common objects in 3d: Large-scale learning and evaluation of real-life 3d category reconstruction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10901–10911, 2021. 1, 2, 5

[46] Konstantinos Rematas, Ricardo Martin-Brualla, and Vittorio Ferrari. Sharf: Shape-conditioned radiance fields from a single view. *arXiv preprint arXiv:2102.08860*, 2021. 2

[47] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10684–10695, 2022. 1, 2, 4, 5

[48] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015. 4

[49] Dávid Rozenberszki, Or Litany, and Angela Dai. Language-grounded indoor 3d semantic segmentation in the wild. In *ECCV*, 2022. 2

[50] Vincent Sitzmann, Michael Zollhöfer, and Gordon Wetzstein. Scene representation networks: Continuous 3d-structure-aware neural scene representations. *Advances in Neural Information Processing Systems*, 32, 2019. 2

[51] Jiaming Sun, Zehong Shen, Yuang Wang, Hujun Bao, and Xiaowei Zhou. Loftr: Detector-free local feature matching with transformers. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8918–8927, 2021. 2

[52] Ayush Tewari, Justus Thies, Ben Mildenhall, Pratul Srinivasan, Edgar Tretschk, W Yifan, Christoph Lassner, Vincent Sitzmann, Ricardo Martin-Brualla, Stephen Lombardi, et al. Advances in neural rendering. In *Computer Graphics Forum*, volume 41, pages 703–735. Wiley Online Library, 2022. 2

[53] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive representation distillation. *ArXiv*, abs/1910.10699, 2020. 2

[54] Alex Trevithick and Bo Yang. Grf: Learning a general radiance field for 3d representation and rendering. *arXiv preprint arXiv:2010.04595*, 2020. 1, 2

[55] Vadim Tschernezki, Iro Laina, Diane Larlus, and Andrea Vedaldi. Neural feature fusion fields: 3d distillation of self-supervised 2d image representations. *arXiv preprint arXiv:2209.03494*, 2022. 2

[56] Qianqian Wang, Zhicheng Wang, Kyle Genova, Pratul P Srinivasan, Howard Zhou, Jonathan T Barron, Ricardo

Martin-Brualla, Noah Snavely, and Thomas Funkhouser. Ibrnet: Learning multi-view image-based rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4690–4699, 2021. 2

[57] Yangtao Wang, Xiaoke Shen, Shell Xu Hu, Yuan Yuan, James L. Crowley, and Dominique Vaufreydaz. Self-supervised transformers for unsupervised object discovery using normalized cut. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14523–14533, 2022. 2

[58] Chen Wei, Haoqi Fan, Saining Xie, Chaoxia Wu, Alan Loddon Yuille, and Christoph Feichtenhofer. Masked feature prediction for self-supervised visual pre-training. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14648–14658, 2022. 2

[59] Julia Wolleb, Robin Sandkühler, Florentin Bieder, Philippe Valmaggia, and Philippe C. Cattin. Diffusion models for implicit image segmentation ensembles. *ArXiv*, abs/2112.03145, 2021. 2

[60] Xiaoshi Wu, Hadar Averbuch-Elor, J. Sun, and Noah Snavely. Towers of babel: Combining images, language, and 3d geometry for learning multimodal vision. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 418–427, 2021. 2

[61] Yiheng Xie, Towaki Takikawa, Shunsuke Saito, Or Litany, Shiqin Yan, Numair Khan, Federico Tombari, James Tompkin, Vincent Sitzmann, and Srinath Sridhar. Neural fields in visual computing and beyond. In *Computer Graphics Forum*, volume 41, pages 641–676. Wiley Online Library, 2022. 2

[62] Zhenda Xie, Yutong Lin, Zhuliang Yao, Zheng Zhang, Qi Dai, Yue Cao, and Han Hu. Self-supervised learning with swin transformers. *arXiv preprint arXiv:2105.04553*, 2021. 2

[63] Li Yi, Vladimir G. Kim, Duygu Ceylan, I-Chao Shen, Mengyan Yan, Hao Su, Cewu Lu, Qixing Huang, Alla Sheffer, and Leonidas Guibas. A scalable active framework for region annotation in 3d shape collections. *SIGGRAPH Asia*, 2016. 5

[64] Junho Yim, Donggyu Joo, Ji-Hoon Bae, and Junmo Kim. A gift from knowledge distillation: Fast optimization, network minimization and transfer learning. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7130–7138, 2017. 2

[65] Yang You, Yujing Lou, Chengkun Li, Zhoujun Cheng, Liangwei Li, Lizhuang Ma, Cewu Lu, and Weiming Wang. Keypointnet: A large-scale 3d keypoint dataset aggregated from numerous human annotations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13647–13656, 2020. 5

[66] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelnerf: Neural radiance fields from one or few images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4578–4587, 2021. 1, 2, 5, 9

[67] Lu Yuan, Dongdong Chen, Yi-Ling Chen, Noel Codella, Xiyang Dai, Jianfeng Gao, Houdong Hu, Xuedong Huang, Boxin Li, Chunyuan Li, et al. Florence: A new foundation model for computer vision. *arXiv preprint arXiv:2111.11432*, 2021. 1, 2

[68] Shuaifeng Zhi, Tristan Laidlow, Stefan Leutenegger, and Andrew J Davison. In-place scene labelling and understanding with implicit scene representation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15838–15847, 2021. 4, 5