Canonical Fields: Self-Supervised Learning of Pose-Canonicalized Neural Fields

Rohith Agaram\textsuperscript{1} Shaurya Dewan\textsuperscript{1} Rahul Sajnani\textsuperscript{2} Adrien Poulenard\textsuperscript{3}
Madhava Krishna\textsuperscript{1} Srinath Sridhar\textsuperscript{2}
\textsuperscript{1}RRC, IIIT-Hyderabad \textsuperscript{2}Brown University \textsuperscript{3}Stanford University
ivl.cs.brown.edu/projects/canonicalfields

Figure 1. We present Canonical Field Network (CaFi-Net), a self-supervised method for 3D position and orientation (pose) canonicalization of objects represented as neural fields. We specifically focus on neural radiance fields (NeRFs) fitted to raw RGB images of arbitrarily posed objects (left), and obtain a canonical field (fixed novel view shown on right) where all objects in a category are consistently positioned and oriented.

Abstract

Coordinate-based implicit neural networks, or neural fields, have emerged as useful representations of shape and appearance in 3D computer vision. Despite advances, however, it remains challenging to build neural fields for categories of objects without datasets like ShapeNet that provide “canonicalized” object instances that are consistently aligned for their 3D position and orientation (pose). We present Canonical Field Network (CaFi-Net), a self-supervised method to canonicalize the 3D pose of instances from an object category represented as neural fields, specifically neural radiance fields (NeRFs). CaFi-Net directly learns from continuous and noisy radiance fields using a Siamese network architecture that is designed to extract equivariant field features for category-level canonicalization. During inference, our method takes pre-trained neural radiance fields of novel object instances at arbitrary 3D pose and estimates a canonical field with consistent 3D pose across the entire category. Extensive experiments on a new dataset of 1300 NeRF models across 13 object categories show that our method matches or exceeds the performance of 3D point cloud-based methods.

1. Introduction

Neural fields [59]—coordinate-based neural networks that implicitly parameterize signals—have recently gained significant attention as representations of 3D shape [6, 22, 28], view-dependent appearance [24, 42], and motion [25]. In particular, neural radiance fields (NeRFs) [24], have been successfully used in problems such as novel view synthesis [3, 4, 66], scene geometry extraction [55, 61], capturing dynamic scenes [19, 29, 30, 33, 52], 3D semantic segmentation [53, 68], and robotics [1, 13, 21].

Despite the progress, it remains challenging to build neural fields that represent an entire category of objects. Previous methods sidestep the problem by overfitting on a single instance [24], or learning [22, 28, 63] on datasets like ShapeNet [5] that contain objects that are manually canonicalized — oriented consistently for 3D position and orientation (3D pose) across a category. This strong supervision makes it easier to learn over categories, but limits their application to data that contain these labels. Recent work has proposed methods for self-supervised learning of 3D pose canonicalization [40, 43, 47], however, these operate on 3D point clouds, meshes, or voxels — but not neural fields.

In this paper, we present Canonical Field Network (CaFi-Net), a self-supervised method for category-level canonicalization of the 3D position and orientation of objects represented as neural fields, specifically neural radiance fields. Canonicalizing neural fields is challenging because, unlike 3D point clouds or meshes, neural fields are continuous, noisy, and hard to manipulate since they are parameterized as the weights of a neural network [60]. To address these challenges, we first extend the notion of equiv-
ariance to continuous vector fields and show how networks for processing 3D point clouds \cite{50} can be extended to operate directly on neural radiance fields. We design CaFiNet as a Siamese network that contains layers to extract equivariant features directly on vector fields. These field features are used to learn a canonical frame that is consistent across instances in the category. To handle noise in radiance fields from NeRF, our method incorporates density-based feature weighting and foreground-background clustering.

Our approach learns canonicalization without any supervision labels on a new dataset of 1300 pre-trained NeRF models of 13 common ShapeNet categories in arbitrary 3D pose (see Figure 1). We introduce several self-supervision loss functions that encourage the estimation of a consistent canonical pose. In addition, we present extensive quantitative comparisons with baselines and other methods on standardized canonicalization metrics \cite{37} over 13 object categories. In particular, we show that our approach matches or exceeds the performance of 3D point cloud-based methods. This enables the new capability of directly operating on neural fields rather than converting them to point clouds for canonicalization. To sum up, we contribute:

- Canonical Field Network (CaFi-Net), the first method for self-supervised canonicalization of the 3D position and orientation (pose) of objects represented as neural radiance fields.
- A Siamese neural network architecture with equivariant feature extraction layers that are designed to directly operate on continuous and noisy radiance fields from NeRF.
- A public dataset of 1300 NeRF models from 13 ShapeNet categories including posed images, and weights for evaluating canonicalization performance.

2. Related Work

We focus our review of related work on neural fields, supervised canonicalization, equivariant neural network architectures, and self-supervised canonicalization.

**Neural Fields:** Neural fields are emerging as useful representations for solving problems such as novel view synthesis \cite{3, 4, 15, 24, 66}, shape encoding and reconstruction \cite{22, 28}, dynamic reconstruction \cite{19, 29, 30}, appearance modeling \cite{23, 36, 45}, and human motion modeling \cite{8, 18, 46, 51}. Generalization of neural fields to object categories remains difficult, but some methods have used pre-canonicalized datasets to circumvent this problem \cite{63}. Neural fields are also used outside of computer vision, for instance in reconstructing proteins \cite{69}, physics \cite{35}, or audio \cite{11}. Please see \cite{49, 59} for more details. In this paper, our focus is on canonicalizing for the 3D pose of neural fields, specifically, neural radiance fields (NeRF) \cite{24}.

**Supervised Canonicalization:** Datasets such as ShapeNet \cite{5} and ModelNet40 \cite{58} have 3D shapes that are manually pre-canonicalized. This inductive bias aids category-level generalization in problems such as 3D reconstruction \cite{12, 22, 28, 48}. We can also use these datasets to formulate canonicalization as a supervised learning problem \cite{54} enabling applications such as 6 degree-of-freedom object pose estimation, multi-view reconstruction \cite{17, 44}, and reconstruction of articulating \cite{20, 65} and non-rigid objects \cite{64}. However, our goal is to canonicalize without using manual pose labels.

**Equivariant Neural Networks:** Pose-equivariant networks are equivariant to input pose \cite{9, 50, 56, 57} by design. Some of these methods use Spherical Harmonic functions \cite{10, 50, 56, 57}, or vector neurons \cite{9} to extract equivariant features. Equivariance is closely related to canonicalization since a canonical pose is also equivariant. Thus, previous methods have used pose-equivariance for canonicalization \cite{37, 41, 43, 47}. However, these methods have thus far been limited to 3D point clouds, voxels, or meshes.

**Self-Supervised Canonicalization:** Recent research has shown that self/weak supervision is sufficient for learning pose canonicalization on point clouds \cite{37, 38, 43, 47}, 2D key-points \cite{26}, and images \cite{27}. None of these previous self-supervised methods can operate directly on neural fields – to the best of our knowledge, ours is the first.

3. Background

We first provide background on the essential components and key ideas behind our method.

**Neural Radiance Fields (NeRF):** Given posed RGB images, NeRF and its variants \cite{23, 24} learn a neural network to synthesize novel views of complex scenes. At inference, for any 3D location and viewing direction, it estimates a direction-independent density value $\sigma$, and a direction-dependent color $c$. Although designed for novel view synthesis, NeRF implicitly learns the 3D geometry and viewpoint-dependent appearance of scenes making it a useful representation of shape and appearance. While NeRFs excel at fitting a single object instance or scene, it struggles to generalize to object categories partly because objects can be in arbitrary 3D poses – thus, existing methods use pre-canonicalized datasets \cite{63}. In this paper, we provide a way to canonicalize 3D pose without any supervision for objects represented as neural radiance fields. While our method is aimed at NeRFs, it can be extended to the 3D pose canonicalization of any neural field.

**Equivariance and Canonicalization:** A function $\Gamma$ over $x$ is said to be equivariant to a group operation $O$ if its output changes by a fixed mapping $M$ for any $O$ operating on
Figure 2. Rotation Equivariance in Vector Fields. Rotation equivariance in vector fields $\mathcal{F}$ (here, a simulated 2D magnetic field) require two operations: (1) update the position of the green arrow (the vector), and (2) rotating the green arrow at its new position. A function $\Gamma$ operating on $\mathcal{F}$ is rotationally equivariant over the field if and only if $\Gamma[(\mathcal{R} \cdot \mathcal{F})(x)] = M(\mathcal{R}) \Gamma[\mathcal{F}(\mathcal{R}^{-1} x)]$, where $\mathcal{R}$ is an $SO(2)$ rotation, and $M$ is a rotation-dependent mapping function.

the input $x$, i.e., $\Gamma(Ox) = M(O) \Gamma(x)$. In this paper, we are interested in equivariance to the $SE(3)$ group denoting 3D position and orientation (pose) – in particular we focus on the rotation group $SO(3)$ since 3D translation equivariance is readily achieved through mean centering [26]. Pose canonicalization is closely related to equivariance – our goal is to find an equivariant canonical 3D rigid transformation that maps the neural field of an object in an arbitrary pose to a canonical pose. We call the outputs of our method canonical fields – neural fields that are consistently oriented across a category of shapes.

Rotation Equivariance in Vector Fields: Unlike 3D point clouds, rotation equivariance in fields is more involved. Consider a 2D vector field $\mathcal{F}$ defined $\forall x \in \mathbb{R}^2$ (see Figure 2). Rotating this field requires two operations: (1) updating the positions of vectors in the field to new rotated positions, and (2) rotating the directions of the vectors. A function $\Gamma$ operating on $\mathcal{F}$ is rotationally equivariant over the field if and only if $\Gamma[(\mathcal{R} \cdot \mathcal{F})(x)] = M(\mathcal{R}) \Gamma[\mathcal{F}(\mathcal{R}^{-1} x)]$, where $\mathcal{R}$ is an $SO(2)$ rotation, and $M$ is a rotation-dependent mapping function. In this paper, we operate on NeRF’s density field (a scalar field) but also extract equivariant features that can be vector-valued fields.

Tensor Field Networks (TFNs): Many methods have been proposed for using equivariant features for rotation canonicalization of 3D data, for instance, spherical CNNs [7, 43], vector neurons [9], or capsule networks [47, 67]. However, these and other methods [38] are limited to 3D point cloud or voxel representations. In this paper, we extend Tensor Field Networks [34, 50] pose canonicalization directly on samples from NeRF’s density field. TFNs operate by computing Type-$\ell$ real-valued spherical harmonic functions that are rotation equivariant: $D^\ell \dot{R} Y^\ell(X) = Y^\ell(\dot{R} x)$. Where $D^\ell : SO(3) \rightarrow SO(2\ell + 1)$ is the Wigner matrix of type $\ell$ (please see [34, 50] for details). For a 3D point set $P \in \mathbb{R}^{N \times 3}$, a rotation-equivariant TFN convolution layer (EQConv) at point $p \in P$ is defined as: $\text{EQConv}^J(p) = Q^{(n,\ell)}(\cdot) \left( \sum_{y \in 2\mathcal{N}} k_n^\ell(p - y) \otimes s^\ell(y) \right)$. Where $2\mathcal{N}$ is a set containing neighbors of point $p$ at twice the resolution and $Q^{(n,\ell)}(\cdot)$ is the Clebsh Gordon decomposition to combine type-$n$ kernel $k_n^\ell$ and type-$\ell$ equivariant signal $s^\ell$. The EQConv layer aggregates type-$\ell$ feature $s^\ell$ at multiple receptive fields along with learnable weights to perform equivariant convolution. Our method uses the EQConv layer as the fundamental building block to canonicalize continuous fields.

4. Method

Our method assumes that we have pre-trained NeRF models of scenes containing single object instances at arbitrary poses from common shape categories. Our scenes contain primarily the object, but also some background – perfect foreground-background segmentation is not available. Our goal is to build a method that estimates a canonicalization $SE(3)$ transformation that aligns these object NeRFs to a consistent category-level canonical field.

Rather than operate on 3D point clouds, meshes, or voxels, our method (see Figure 3) operates directly on samples from the continuous neural radiance field. During training, we learn canonical fields from a pre-trained NeRF dataset in a fully self-supervised manner. We use a Siamese network architecture that extracts equivariant field features for canonicalization (see Figure 3). During inference, given a previously unseen NeRF model of a new object instance in an arbitrary pose, our method estimates the transformation that maps it to the canonical field.

4.1. NeRF Sampling

For a pre-trained neural radiance field that maps 3D position $(x, y, z)$ and direction $\theta, \phi$ to color and density $(c, \sigma)$, the input to our method consists of samples on the density field. We do not use color because it is direction-dependent in NeRF and also varies for different instances.

Preprocessing: During both training and inference, we uniformly sample the pre-trained NeRF density field to find the object center and its bounding box (see supplementary document for details). We then sample a uniform grid in the density field of the object’s bounding box. We empirically choose uniform grid sampling over fully random sampling since we found no difference. The uniformly sampled density grid is then normalized to obtain density values $\sigma_D$ in the continuous range $[0, 1]$, i.e.,

$$\sigma_D := \{1 - \exp(-d \cdot \Gamma^\sigma(x)) \mid x \in \mathcal{X}\},$$

(1)

4502
Figure 3. CaFi-Net samples a density field from NeRF and uses its **density**, **position** and **density gradients** as input signals (A) to canonicalize the field. We predict rotation equivariant features and weigh them by density (B) to guide our learning from the occupied regions of the scene. We then compute an invariant embedding by taking a dot product between equivariant features. This invariant embedding is used to canonicalize the field that enables rendering all the objects in the canonical frame (E). Our method also applies an inter-instance consistency loss (D) that aligns different instances of the same category in the canonical frame. We do not assume pre-canonicalized fields, and canonicalize in a self-supervised manner.

where $X$ is the regularly sampled grid within the object bounding box and $d$ is the depth step size used in the coarse NeRF sampling. **Translation** and **scale** canonicalization is simply achieved by moving the bounding box’s center [26] and scaling it, so our next concern is on rotation canonicalization.

### 4.2. CaFi-Net

CaFi-Net (**Canonical Field Network**) is our method for 3D rotation canonicalization using NeRF’s density field (red box in Figure 3). We now describe its components.

**Signal Representation:** At each of the sampled density field locations, we have several choices of signals to use as the input for CaFi-Net. For instance, we can (1) use the NeRF density value directly, (2) use the $xyz$ location of the sampled field, or (3) use the gradient of the density field. Densities alone do not capture the position and directional components of the field, so we use a combination of densities and gradients of densities at query locations of the grid as our input signal for CaFi-Net. Gradients of density capture the object surface and help in canonicalization (see ablation study in Section 5).

**Equivariant Convolution:** To canonicalize for rotation, we leverage equivariance properties of spherical harmonic functions to obtain rotation-equivariant learnable features on the field [34, 50]. Type-$\ell$ spherical harmonic functions ($Y^\ell$) are degree-$\ell$ polynomials of points on the sphere that are equivariant to $SO(2\ell + 1)$. The densities sampled from NeRF reconstructions are often noisy and contain outliers. To promote our method to learn only from occupied regions, we weigh each equivariant feature type by its density. This weighing has no effect on the equivariance of features (see supplementary document for proof). We have the option to weigh equivariant features with the local average of the density to smoothen the signal or to weigh directly by the density. We found weighing by density worked better than taking the local average density (see Section 5). We learn equivariant features of type $\ell$ as:

$$R^\ell_x := f_w(R_x) \left( W^\ell(EQConv^\ell(\nabla(\sigma_D(R_x)))) + \delta \omega b \right),$$

$$f_w(R_x) := \begin{cases} \text{mean}_r(\sigma_D(R_x)), & \text{or} \\ \sigma_D(R_x), & \end{cases}$$

where, $W^\ell$ and $b$ are the weights and biases, $R_x$ is a 3D point queried at resolution $R$ and $\text{mean}_r(\sigma_D(R_x))$ is the average density at location $R_x$ from points sampled at radius $r$. We hierarchically aggregate features at resolutions $\frac{1}{2}$, $\frac{1}{4}$, and $\frac{1}{8}$ to obtain global equivariant features. The $EQConv^\ell$ convolution layer is the same as defined in Section 3. We employ non-linearities from [34] that preserve equivariance and capture better equivariant features. We also compute the max-pool of point-wise features of the last layer to obtain global type-$\ell$ equivariant features $F^\ell$.

**Canonicalization:** After obtaining the global type-$\ell$
equi-variant feature $F^\ell$, we compute its dot product with point-wise spherical harmonics $Z^\ell$ scaled by their norms $Z^\ell(X) = \|X\|_2 Y^\ell(X / \|X\|_2)$ to obtain a pose-invariant embedding $H^\ell$ [37]. As the two vectors $F^\ell$ and $Z^\ell$ are both equivariant to input rotation, their dot product is invariant. We use a linear layer on top of the pose-invariant embedding to estimate grid coordinates in the canonical frame ($P \in \mathbb{R}^{H \times W \times D \times 3}$) for each point in $X$ and $M$ equivariant transformations $E \in \mathbb{R}^{M \times 3 \times 3}$ [37] that orient the canonical coordinates to the input coordinates. We then choose the best canonicalizing transform $E_b$ and penalize it for canonicalization. The invariant embedding $H$, grid coordinates $P$ and equivariant rotation $E$ are given as,

\begin{align}
H^\ell(\sigma_D, X) & := (F^\ell(\sigma_D), Z^\ell(X)), \\
P := \text{MLP}_Y(H(\sigma_D, X)), \\
E := \text{MLP}_E(F(\sigma_D, X)).
\end{align}

Here, we have dropped the type-$\ell$ notation for $H$ and $F$ as we concatenate all equivariant feature types.

**Siamese Network Architecture:** Experimentally, canonicalizing noisy density fields is difficult without guidance on shape similarity within a category. We therefore use a Siamese training strategy [26] where two different object instances are forced to be consistently canonicalized (see the lower branch in Figure 3).

**Clustering:** CaFi-Net predicts a canonical coordinate for every grid point, but unlike point clouds, fields have many unoccupied regions that do not provide any information. We penalize the network on foreground regions only by performing K-means clustering on the densities within the object bounding box with $K = 2$ and choose the cluster $C_f$ with a higher mean. Note that we still operate on continuous fields, but only cluster samples to guide our training.

### 4.3. Training

CaFi-Net is trained on a large dataset of pre-trained NeRF models over 13 object categories (see Section 5). We use the following loss functions to train our model.

**Canonicalization Loss:** To self-supervise our learning, we transform the predicted canonical grid coordinates to the input grid using the canonical rotation $E_b$ and compute a point-wise L2 distance between them. This loss forces the predicted invariant grid coordinates $P$ to reconstruct the shape and forces the predicted canonicalizing transform to be equivariant to the input transformation. The final loss is computed over the canonicalizing transform that minimizes this loss:

\begin{align}
E_b := \min_j \left( \text{mean}_{i \in C_f} \|X_i - E_j P_i\|_2^2 \right), \\
\mathcal{L}_{\text{canon}} := \text{mean}_{i \in C_f} \|X_i - E_b P_i\|_2^2,
\end{align}

where $C_f$ is a set of points belonging to the foreground.

**Orthonormality Loss:** The predicted equivariant transformations $E$ should have orthonormal vectors and we use the following to force orthonormality using:

\begin{align}
U_j, D_j, V_j := \text{SVD}(E_j), \\
\mathcal{L}_{\text{ortho}} := \frac{1}{M} \sum_j \|E_j - U_j V_j^T\|_2^2,
\end{align}

where $\text{SVD}(E_j)$ computes the singular value decomposition of the $j$th equivariant transformation.

**Siamese Shape Loss:** We use a Siamese loss that penalizes for consistency between different shapes belonging to the same category. Given two un-canonicalized fields in different frames of reference $\sigma_1^j$ and $\sigma_2^j$, we canonicalize both the fields to the canonical frame and compute chamfer distance between their predicted canonical coordinates for points in the foreground cluster $C_f$, i.e.,

\begin{align}
\mathcal{L}_{\text{siamese}} := \text{CD}(P(\sigma_1^j), P(\sigma_2^j)).
\end{align}

This loss regularizes the training by ensuring that instances of the same category should be aligned in the canonical frame.

**Architecture and Hyper-parameters:** CaFi-Net predicts a 128 dimensional invariant embedding for each point in space and performs convolution at resolutions 1/2, 1/4, and 1/8. Each equivariant convolution layer aggregates features from 512 neighboring points at twice the resolution. We then use equivariant non-linearities introduced in [34] by applying Batch Normalization [14] and ReLU activation [2] in the inverse spherical harmonic transform domain. We use a three-layer MLP with intermediate Batch Normalization and ReLU layers followed by a final linear layer to predict canonical coordinates. To train our network, we weigh the canonicalization loss ($\mathcal{L}_{\text{canon}}$) the highest with weight 2.0 and 1.0 for all the other loss functions. All our models are written in PyTorch [31] and are trained for 300 epochs with a batch size of 2 on an Nvidia 1080-Ti GPU. We use the Adam Optimizer [16] with an initial learning rate of 6e − 4 and L2 weight regularization of 1e − 5 for all our experiments. Please see the supplementary document for more details.

## 5. Experiments

We now provide details on our extensive experiments to evaluate the quality of our pose canonicalization. To our knowledge, we are the first method for pose canonicalization of neural radiance fields which makes direct comparisons with other work challenging. Nonetheless, we have designed experiments to compare our method with three 3D point cloud-based canonicalization methods. Overall, we present three classes of experiments: (1) evaluation of the performance of our method on 13 different categories,
Table 1. This table compares the canonicalization performance of our method (F - operating on fields) with other 3D point cloud-based methods (P) on three standard metrics (IC, CC and GEC) on our dataset of 13 categories. We compare with PCA, Canonical Capsules (CaCa) [47] and ConDor [37]. All metrics are multiplied by 100 for ease of reading. The top two performing methods are highlighted in magenta (best) and blue (second best). We are better than SOTA [37] on the Ground Truth Equivariance Consistency (GEC) and Category-Level Consistency (CC) with lower mean and median canonicalization error. However, we perform on par with ConDor [37] on the Instance-Level Consistency.

<table>
<thead>
<tr>
<th>Instance-Level Consistency (IC)</th>
<th>bench</th>
<th>cabinet</th>
<th>car</th>
<th>cellph.</th>
<th>chair</th>
<th>couch</th>
<th>firearm</th>
<th>lamp</th>
<th>monitor</th>
<th>plane</th>
<th>speaker</th>
<th>table</th>
<th>water</th>
<th>avg.</th>
<th>med.</th>
</tr>
</thead>
<tbody>
<tr>
<td>PCA (P)</td>
<td>14.65</td>
<td>5.94</td>
<td>6.13</td>
<td>0.67</td>
<td>6.13</td>
<td>7.63</td>
<td>15.07</td>
<td>12.84</td>
<td>6.78</td>
<td>7.90</td>
<td>5.40</td>
<td>10.31</td>
<td>10.72</td>
<td>8.47</td>
<td>7.63</td>
</tr>
<tr>
<td>CaCa (P) [47]</td>
<td>2.89</td>
<td>2.09</td>
<td>2.01</td>
<td>2.81</td>
<td>1.08</td>
<td>1.90</td>
<td>0.23</td>
<td>2.97</td>
<td>2.35</td>
<td>1.51</td>
<td>2.21</td>
<td>2.42</td>
<td>2.60</td>
<td>2.08</td>
<td>2.21</td>
</tr>
<tr>
<td>ConDor (P) [37]</td>
<td>0.56</td>
<td>1.27</td>
<td>0.36</td>
<td>0.53</td>
<td>0.96</td>
<td>0.32</td>
<td>0.60</td>
<td>3.78</td>
<td>0.44</td>
<td>0.64</td>
<td>1.97</td>
<td>1.32</td>
<td>2.15</td>
<td>1.15</td>
<td>0.64</td>
</tr>
<tr>
<td>Ours (F)</td>
<td>2.62</td>
<td>1.15</td>
<td>0.39</td>
<td>1.28</td>
<td>0.81</td>
<td>1.43</td>
<td>0.74</td>
<td>3.87</td>
<td>0.41</td>
<td>0.36</td>
<td>0.60</td>
<td>1.09</td>
<td>2.74</td>
<td>1.34</td>
<td>1.09</td>
</tr>
</tbody>
</table>

| Category-Level Consistency (CC) | | | | | | | | | | | | | | |
|--------------------------------|-------|---------|-----|---------|-------|-------|---------|------|---------|-------|---------|-------|-------|------|------|
| CaCa (P) [47]                  | 2.79  | 0.38    | 1.05| 0.83    | 1.82  | 1.59  | 1.31    | 4.34 | 0.39    | 1.83  | 0.65   | 1.91  | 2.06  | 1.61 | 1.59 |
| ConDor (P) [37]                | 1.98  | 1.68    | 0.47| 1.04    | 1.28  | 0.88  | 0.96    | 4.44 | 1.58    | 1.26  | 2.09   | 1.20  | 2.56  | 1.72 | 1.58 |
| Ours (F)                       | 2.77  | 1.36    | 0.48| 1.26    | 0.75  | 1.59  | 0.92    | 3.92 | 0.58    | 0.63  | 0.77   | 1.48  | 2.35  | 1.45 | 1.26 |

| Ground Truth Equivariance Consistency (GEC) | | | | | | | | | | | | | | |
|--------------------------------|-------|---------|-----|---------|-------|-------|---------|------|---------|-------|---------|-------|-------|------|------|
| PCA (P)                        | 14.40 | 5.86    | 6.32| 1.64    | 6.32  | 8.12  | 15.01   | 12.49| 6.82    | 8.16  | 5.32   | 10.17 | 11.27 | 8.61 | 8.12 |
| CaCa (P) [47]                  | 3.65  | 2.28    | 2.40| 2.74    | 1.90  | 2.22  | 1.28    | 4.69 | 2.48    | 2.09  | 2.22   | 3.01  | 2.79  | 2.59 | 2.40 |
| ConDor (P) [37]                | 2.12  | 1.79    | 0.50| 1.17    | 1.34  | 0.93  | 1.04    | 4.58 | 1.59    | 1.31  | 2.22   | 2.24  | 2.67  | 1.81 | 1.59 |
| Ours (F)                       | 3.26  | 1.51    | 0.54| 1.43    | 0.94  | 1.90  | 1.01    | 4.37 | 0.63    | 0.66  | 0.81   | 1.63  | 3.02  | 1.67 | 1.43 |

(2) comparison with other work, and (3) ablations to justify method design choices.

Dataset: All our experiments use a new large synthetic NeRF dataset of shapes from 13 categories (that overlap with [12, 37, 47]) from the ShapeNet [5] dataset. For each category, we pick 100 instances from ShapeNet, rotate the shape randomly, and render 54 omnidirectional views sampled in a cube around the shape using Blender. Differ- ent from these datasets, we also simulate cluttered back- grounds. These posed views are used to train 100 NeRF models per category using the public PyTorch implementation of NeRF [62] for a total of 1300 NeRF models. To our knowledge, this is one of the largest (compared to [11, 53]) 360° synthetic NeRF datasets – we will publicly release the simulator, raw posed images, and trained NeRF weights. Each NeRF model is trained for 400 epochs with 1024 randomly selected rays per iteration, a coarse sampling resolution of 64 points, and a fine sampling resolution of 128 points along each ray. Rather than train NeRF to maximize PSNR on a novel view, we fix the number of epochs for all models to enable a fair comparison between instances. As a consequence, some NeRF models may have a lower PSNR, but our goal is not to increase NeRF quality but rather improve canonicalization quality – hence we fix the number of epochs. For each category, we set aside 20% of the models for testing. Metrics: Since there are no known metrics for evaluating canonical fields, we resort to 3 metrics used for 3D point cloud canonicalization [37]: (1) Instance-Level Consistency (IC), a metric that measures how consistently we can canonicalize the same instance in different poses, (2) Category-Level Consistency (CC) to measure how consistently we can canonicalize across different instances in the same category, and (3) Ground Truth Equivariance Consistency (GEC), a variant of the Ground Truth Consistency (GC) metric proposed in [37] to measure canonicalization performance compared to manual labels. We use the GEC because we observed that the original GC metric is prone to degeneracy when the canonicalizing transform is identity. We fix this issue by taking three point clouds \( P_i, P_j, \) and \( P_k \) from a canonicalized dataset \( P \) and rotating \( P_i \) by \( R_1, P_j \) by \( R_2 \) where both \( R_1 \) and \( R_2 \) are random rotations. Let \( C(P) \) predict a canonicalizing rotation for a point cloud \( P \), we then compute the Ground Truth Equivariance consistency as:

\[
GEC := \frac{1}{|P|^3} \sum_{P_i, P_j, P_k \in P} CD(C(R_1P_i)R_1P_k, C(R_2P_j)R_2P_k),
\]

where \( CD \) refers to the Chamfer distance. Here, we apply canonicalizing transforms of shape \( P_i \) and \( P_j \) to the shape \( P_k \). This modified metric will not be degenerate for identity canonicalizing transforms and is more rigorous in evaluating ground truth consistency. To compute our metrics, we used point clouds sampled from the Shapenet meshes [5] using poisson disk sampling.

5.1. Evaluation & Comparisons

We first analyze the performance of our method on the three metrics across 13 different categories. In Table 1, we highlight our method that operates on a field as

\[
Ours (F)
\]

Our method performs well across all categories. We observe similar trends across metrics for a set of categories (e.g., firearms,lamps) indicating that category shape plays a
role in canonicalization performance.

Next, we compare our method against three benchmarks: Principal Component Analysis (PCA) [32], Canonical Capsules (CaCa) [47], and ConDor [37]. All prior methods canonicalize clean point clouds with all points on the surface. To have a fair comparison, we train the above methods on point clouds (P) sampled from NeRF scenes for points on and within the object. Most scenes have points in the range of 120–1200 when sampled uniformly on the 32^3 grid. We resample the scene when points are less than 1024 and train both Canonical Capsules and ConDor on the resulting point clouds. CaFi-Net estimates a canonicalizing transform directly from the density field that we use to transform its volume. Our average ground truth canonicalization error over 4 categories is lower when using gradient as compared to using density. We hypothesize that gradient of density weighs equivariant features closer to the surface of the object higher. This improves the canonicalization performance.

5.2. Ablations

We conduct additional experiments to justify key design choices in our method. For all the ablation studies we use a smaller subset of our data consisting of 4 categories: bench, cellphone, chair, and airplane.

**Choice of Signal Representation:** To select the appropriate signal representation to use (see Section 4), we conducted an ablation study using just the density field $xyz$ locations or its gradients. As seen in Table 2 (a), the average canonicalization error over 4 categories is lower when using gradient as compared to using density $xyz$ locations on all three metrics. We hypothesize that gradient of density weighs equivariant features closer to the surface of the object higher. This improves the canonicalization performance.

**Weighting Features by Local Average Density:** Next, we justify why weighting the equivariant features using the density is helpful. Table 2 (b) shows results for weighting features by local average density compared to weighting features by density. Weighing features by local average smoothens out details that are necessary to canonicalize the volume. Our average ground truth canonicalization error
Figure 4. This figure visually compares the canonicalization performance of our method with 3D point cloud-based methods. Input NeRFs shows one of the RGB views used to learn our NeRF input (notice how orientations of instances are random). CaFi-Net (Ours) shows the results of our canonical field rendered from a novel view unseen during NeRF training. PCA (blue), CaCa [47] (red), and ConDor [37]'s (green) results are also presented for the same instances. Our method matches or exceeds ConDor. We also show some failure cases for several instances (red box) which occur either due to thin structures or symmetry.

### Conclusion

We presented Canonical Field Network (CaFi-Net), a method for self-supervised category-level canonicalization of the 3D pose of objects represented as neural radiance fields. CaFi-Net consists of a Siamese network architecture with rotation-equivariant convolution layers to extract features for pose canonicalization. Our approach operates directly on NeRF’s noisy but continuous density field. We train our method on a large dataset of 1300 NeRF models obtained for 13 common ShapeNet categories. During inference, our method is able to canonicalize arbitrarily oriented NeRFs. Experiments show that our method matches or outperforms 3D point cloud-based methods.

**Limitations & Future Work:** Our approach has several limitations that future work should investigate. First, we require a 360° NeRF model of objects and cannot handle partial views from front-facing NeRFs. Second, canonicalizing densities with uncertainty is more difficult as the uncertainty increases for very thin/small structures that can be missed (see chair in Figure 4). Furthermore, we also inherit the issues that TFNs have with symmetry (see monitor case in Figure 4). In future work, we plan to extend our approach to create a large real dataset of common object categories without requiring manual pose canonicalization and combat the symmetry issue in CaFi-Net as in [39].

**Acknowledgments:** This work was supported by NSF grant CNS-2038897, an AWS Cloud Credits award, NSF CloudBank, and a gift from Meta Reality Labs. We thank Chandradeep Pokhariya and Ishaan Shah.
References


[23] Xueyu Liu, Yan Xu, Qi Tian, Yandong Li, Xi Sui, and Antonio Torralba. 3d neural scene representations for visuomotor control. In *Proceedings of Robotics: Science and Systems*, 2021. 1


[26] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nef: Representing scenes as neural radiance fields for view syn-


