LEARNING DENSE NERF CORRESPONDENCE THROUGH GENERATIVE STRUCTURAL PRIORS

Anonymous authors
Paper under double-blind review

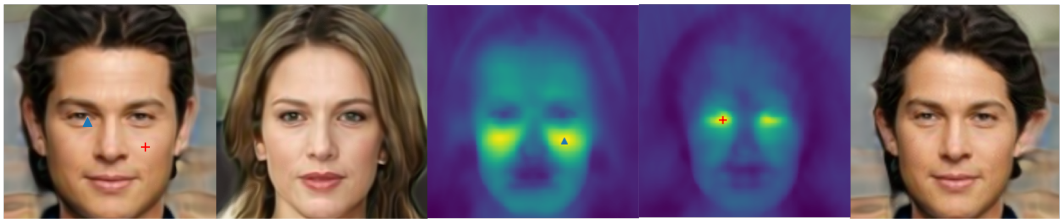


Figure 1: Given a trained generator $G(\cdot)$, we have two synthesis on the first two columns. On the first identity's face we randomly sample two keypoints. We extract each point's feature from G and calculate the feature similarity between the keypoints' features and the feature map of the second identity. The third and fourth columns show the similarity heatmap where the highlights position have the same semantic meaning with the keypoints, which means they are dense correspondences to each other. In this work we leverage this prior from pretrained NeRF-based GANs to learn 3D NeRF dense correspondence.

ABSTRACT

Neural radiance field (NeRF), a kind of 3D shape representation, has shown promising results over building geometry and textures from images. However, unlike mesh or signed distance function (SDF) based representation, it remains an open problem to build correspondences across radiance fields, limiting its application in many downstream tasks. Assumptions of prior arts on the availability of either correspondence annotations or 3D shapes as supervision signals do not apply to NeRF. This paper shows that by leveraging rich structural priors encapsulated in a pretrained NeRF generative adversarial network (GAN), we can learn correspondence in a self-supervised manner without using any correspondence or 3D supervision. To exploit the priors, we devise a novel Bijective Deformation Field (BDF), a way to establish a bijective shape deformation field for 3D radiance fields. Our experiments demonstrate that the GAN-derived priors are discriminative enough to guide the learning of accurate, smooth and robust 3D dense correspondence. We also show that BDF can produce high-quality dense correspondences across different shapes belonging to the same object category. We further demonstrate how the accurate correspondences facilitate downstream applications such as texture transfer, segmentation transfer, and deformation transfer. Code and models will be released.

1 INTRODUCTION

The success of neural radiance fields (NeRF) (Mildenhall et al., 2020) has led to remarkable progress in learning 3D representations. Unlike voxel- and mesh-based methods, NeRF-based approaches represent each 3D object as a distribution of per-point colored densities in the 3D space. And by approximating this distribution with a continuous parametric function, they show great potential to capture geometric scene details and render realistic novel views.

In this work, we wish to establish dense correspondence between NeRFs, a meaningful prerequisite for many downstream applications such as non-rigid tracking, appearance transfer and shape manipulation. The problem is non-trivial as existing methods are not directly applicable to our problem.

Specifically, while point-wise correspondences are needed for NeRF-based object representations, existing methods mainly focus on mesh-based object representations, providing only vertex-wise and part-wise correspondences. Importantly, the training of existing methods require ground-truth annotations as supervision signals, which are hard to obtain for NeRF-based object representations. There are studies on finding correspondences over object representations in the form of signed distance functions (SDFs) (Zheng et al., 2021; Deng et al., 2021), which are not applicable as well due to the reliance on SDF-specific priors, i.e., exact object surface can be defined.

We overcome the limitations above by exploiting NeRF-based generative adversarial networks (GANs) (Chan et al., 2021; Niemeyer & Geiger; Schwarz et al., 2020). Specifically, NeRF-based GANs treat novel views rendering like image synthesis in conventional image-based GANs but employ NeRFs as the representations. In this study, we show that such NeRF-based GANs implicitly capture rich structural priors after learning from a massive set of 2D images. Such priors are indicative to correspondences across two objects' NeRF – corresponding points of both objects would share similar features derived from the NeRF GAN. We demonstrate that one can naturally exploit such cross-instance feature similarity as the geometric preservation descriptor to establish shape correspondences.

To build dense correspondences between object NeRFs using generator features of a NeRF-based GAN, an intuitive solution is by learning a conditional mapping that estimates per-point 3D offsets from a source NeRF to a target NeRF, conditioned on the latent codes of these two NeRFs. However, such a straightforward approach is less effective in practice as the objective is too challenging for a single mapping to capture. In light of this, we propose to replace the conditional mapping with a bijective deformation field referred to as ***BDF***. It consists of two separate mapping functions, namely a mapping function $W_F(\cdot)$ that estimates per-point 3D offsets from the source NeRF to a fixed template NeRF, and a mapping function $W_B(\cdot)$ that estimates per-point 3D offsets from the template NeRF to the target NeRF. Fig 2 illustrates the notion of BDF. The proposed BDF is appealing as two separate NeRFs can be seamlessly and accurately bridged by the fixed template NeRF in a self-supervised manner without any explicit supervision.

We contribute a careful study on the losses and regularizations to achieve self-supervised learning in the proposed BDF framework. Specifically, we minimize the cosine distance between generator features of estimated corresponding points as the primary objective. To make the learning more effective, we introduce some useful regularizations, e.g., encouraging the cycle consistency of two mappings, emphasizing points with large densities via importance sampling, and encouraging the smoothness of pair-wise deformation. In addition, we present a curriculum training strategy to improve the stability of BDF learning – the learning starts from using pairs of object NeRFs with small differences and gradually includes more challenging pairs.

To our knowledge, this study is the first attempt that builds 3D dense correspondences between two shapes represented as NeRF. Our technical contributions include the notion of exploiting prior from NeRF-based GAN and the self-supervised objectives for learning the bijective deformation field. BDF produces high-quality 3D dense correspondences, surpassing previous self-supervised methods by a large margin while being on par with supervised counterparts. By conducting GAN-Inversion and applying BDF subsequently, one can achieve several interesting downstream tasks like texture transfer.

2 Related Work

3D Shape correspondences The problem of establishing dense correspondences between 3D shapes is of key importance to a series of downstream tasks (Loper et al., 2015; Egger et al., 2020), and has been studied extensively in recent survey (Kaick et al., 2011; Sahillioglu, 2019). Traditional approaches build correspondence between shapes represented by mesh or point clouds. They can be roughly divided into registration based and similarity based methods, where the former adopts Laplacian coordinate δ_i for vertex v_i as geometric preservation descriptor after registration. Similarity-based solutions do not change the geometry of given shapes and calculates the similarity between vertices with learnable feature descriptors. To mitigate the complexity of point-to-point matching, functional maps (Ovsjanikov et al., 2012) reduce shape matching from vertex-based to the spectral space of 3D shapes. With recent advances in geometric machine learning (Wang et al., 2019; Qi et al., 2017), researchers extend traditional framework by replacing hand-crafted descriptor such as

SHOT (Tombari et al., 2010), with learnable feature descriptors (Litany et al., 2017). Ground truth correspondences (Fan et al., 2019) or 3D models (Zhou et al., 2016) are usually needed during training. To mitigate these requirements, (Roufosse et al., 2019; Halimi et al., 2019; Eisenberger et al., 2021; Groueix et al., 2018) extend 3D correspondence to unsupervised settings. However, explicit geometry representations intrinsically suffer from deformation artifacts and require dense vertices to recover surface details.

As a parallel class of shape representation, implicit functions represent shapes as the iso-surface (Carr et al., 2001) of a continuous volumetric field. Recent advances in implicit functions (Mescheder et al., 2019; Park et al.; Chen & Zhang, 2019) have demonstrated their excellence when representing complicated geometry. However, building dense correspondence across shapes represented by implicit functions are intrinsically challenging since ground truth correspondence are impossible to acquire. Recent attempts to build correspondence over implicit representations (Zheng et al., 2021; Deng et al., 2021) tried to bypass this requirement by defining F as the SDF values of the deformed points and d as the marginal L_1 loss as in (Park et al.). Similarly, (Liu & Liu, 2020) (**?**) followed similar principles as functional maps and adopts occupancy loss as supervision, while the basis functions are learned from data. Though dense correspondence over implicit functions could be derived, these methods are unable to establish consistent bijective correspondence and still require 3D supervision during training. Moreover, these methods are all constrained on synthetic dataset (Chang et al., 2015) which limit the applications on real scenes. We emphasize that our methods is essentially different from them in three ways. First, our method builds on NeRF which are more practical in representing realistic scenes. Second, our method is fully free of 3D annotations like sparse correspondence labeling or 3D models. This facilitated more downstream applications where only 2D images are available. Lastly, our method is able to build bijection correspondences between two shapes, which provides more flexibility and scalability to deform between two shapes.

Generative models and 3D-aware image synthesis Deep generative models, especially GANs (Goodfellow et al.; Karras et al.; Brock et al., 2019), have shown promising results in generating photorealistic images. To further extend GANs to synthesize images in a 3D-consistent manner, many recent approaches investigated how to incorporate 3D inductive bias into generative training. Motivated by the success of NeRF (Mildenhall et al., 2020), recently researchers resort to the continuous power of radiance fields as the incorporated 3D inductive bias in GANs (Chan et al., 2021; Schwarz et al., 2020; Niemeyer & Geiger). Impressive results have been achieved on both 3D-aware image synthesis and multi-view consistency. Our work employs radiance fields-based GANs, specifically π-GAN (Chan et al., 2021), as both a robust correspondence similarity metric and an infinite 3D shapes dataset. Beyond the study of improving the synthesis quality, few work probes how to apply the representations learned by GANs for downstream tasks. (Bau et al., 2020; Shen et al., 2020; Jahanian et al., 2020) interpret the semantics encoded by GANs and apply them for image editing. (Zhang et al., 2021a; Tritrong et al., 2021; Zhang et al., 2021b) leverage the rich semantics in GAN's features for fine-grained annotation synthesis, few-shot segmentation as well as multi-view data generation respectively. Concurrently (Pan et al.; Eslami et al.; Jahanian et al., 2019; Zhang et al., 2020) show that GAN trained on 2D images can learn implicit notion of 3D environment. But it remains less explored whether the learned GAN representations are transferable to more challenging 3D tasks, like dense correspondence estimation.

3 METHODOLOGY

In this paper, we propose a novel framework that exploits generator features of a NeRF-based GAN trained with unposed 2D images to build dense 3D correspondences between NeRF representations of 3D objects. At the core of our framework is a bijection deformation field that estimates the correspondence of a NeRF point based on its generator feature. While our framework does not require any ground-truth 3D annotations, a key insight of our framework is that a pre-trained NeRF-based GAN will embed the geometric details of NeRFs in the semantics of its generator features, so that corresponding points across different NeRFs have the most similar features, as shown in Fig. **??**. In the following, we at first introduce the details of NeRF-based GANs as background knowledge, and then briefly describe formulate the problem of 3D correspondence learning. Finally, we will introduce our framework in detail.

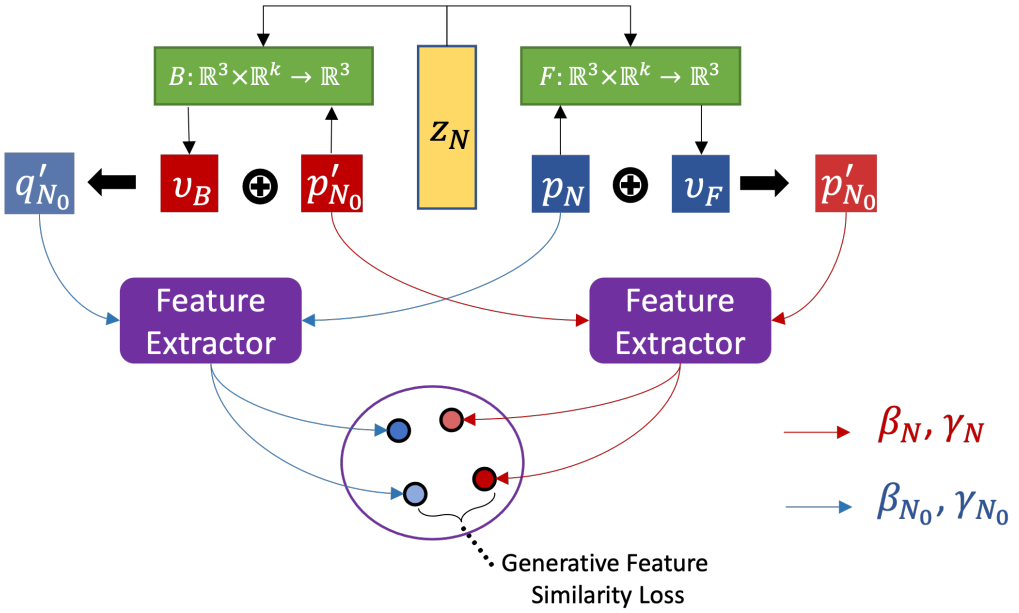


Figure 2: Demonstration of our deformation field through texture transfer. For each samples we transfer the texture from target NeRF according to the their correspondences. This is achieved by volume rending over deformed geometry and texture from the target NeRF. The first two rows show the forward deformation using $F(\cdot)$ which map a NeRF to the template, and the last two rows show the inverse mapping from template to the instance via $B(\cdot)$.

3.1 Background on NeRF-based GANs

Inspired by the success of NeRF as an efficient 3D representation, NeRF-based generative adversarial networks (GANs) employ NeRF as their internal representation for 3D-aware image synthesis. We adopt π-GAN (Chan et al., 2021) in this paper. Specifically, the generator of the π-GAN contains a mapping network and a multi-layer perceptron (MLP) network. Starting from a latent code $\boldsymbol{z} \sim p_Z$ that follows the Gaussian prior distribution, the mapping network m first maps $\boldsymbol{z}$ to a set of modulation signals $(\boldsymbol{\beta} = \{\beta_i\}, \boldsymbol{\gamma} = \{\gamma_i\})$. In π-GAN, a NeRF is obtained by the MLP network, which estimates the view-dependent density $\sigma \in \mathbb{R}^+$ and the color vector $\boldsymbol{c} \in \mathbb{R}^3$ for each 3D point, taking its coordinate $\boldsymbol{x} \in \mathbb{R}^3$ and a viewing direction $\boldsymbol{d} \in \mathbb{S}^2$ as input. To associate a latent code to its corresponding NeRF, the modulation signals will be injected into the MLP network, serving as FiLM conditions (Perez et al., 2018; Dumoulin et al., 2018) to modulate its features at different layers as $\boldsymbol{f}_{i+1} = \sin(\gamma_i \cdot (\boldsymbol{W}_i \boldsymbol{f}_i + \boldsymbol{b}_i) + \beta_i)$.

Image synthesis in π-GAN is achieved by sampling a latent code and a viewing direction, and subsequently rendering an image from the corresponding NeRF. Following the volume rendering of NeRF (Mildenhall et al., 2020), each pixel color C of the image is obtained via sampling a set of points along the ray $r(t) = \boldsymbol{o} + t\boldsymbol{d}$ and accumulating their color vectors weighted by their densities:

$$\hat{C}(\boldsymbol{r}) = \sum_{i=1}^{N} T(t_i)(1 - \exp(-\sigma_i \delta))\boldsymbol{c}_i, \text{ where } T(t) = \exp\left(-\sum_{j-1}^{i=1} \sigma_j \delta_j\right), \tag{1}$$

where $\delta i = t_{i+1} - t_i$ is the distance between adjacent samples. Using a set of unposed 2D images, π-GAN is trained progressively with the non-saturating GAN loss and the R1 regularization (Mescheder et al., 2018).

3.2 PROBLEM FORMULATION

In general, the dense correspondences between two objects $\mathcal{X}$ and $\mathcal{Y}$ refer to pairs of surface points $\{(x, y); x \in \mathcal{X}, y \in \mathcal{Y}\}$ that have analogous geometric semantics. To conduct dense correspondence learning between two given objects $\mathcal{X}$ and $\mathcal{Y}$, previous attempts usually learns a deformation field T that estimates an offset Δ in 3D coordinate for a given surface point in $\mathcal{X}$, so that $corr_{\mathcal{X},\mathcal{Y}}(x) = x + \Delta = y$. The learning objective of this deformation field T can be summarized as

$$\mathcal{L} = \arg\min_{T} \sum_{x \in \mathcal{X}} d(x, corr_{\mathcal{X},\mathcal{Y}}(x)) + \lambda L_{reg}(T), \tag{2}$$

where d is a measurement that evaluates the distance between two points, each of which belongs to a different shape, and L_{reg} regularizes the deformation of T to satisfy certain properties, such as local smoothness.

In our case, since each object $\mathcal{X}$ is represented as a neural radiance field (NeRF) $\mathcal{N}_X$, we will find correspondences for all points in $\mathcal{N}_X$ that have non-zero densities. While previous attempts (Liu & Liu, 2020; Deng et al., 2021; Zheng et al., 2021) for dense correspondence learning rely on ground-truth annotations to learn their deformation fields, collecting such annotations for NeRF-based representations is infeasible, where there are infinite points with non-zero densities, and the object surface is not explicitly modeled.

3.3 BIJECTIVE DEFORMATION FIELD

We propose a novel method to learn dense correspondences between two object NeRFs without relying on any ground-truth annotations. The key idea of our method is to employ a pre-trained π-GAN $G(\cdot)$ that plays the *dual role*: 1) a source of infinite object NeRFs $\mathcal{N}_{i=1}^{\inf}$ and 2) a robust semantic embedding function that maps corresponding points across different NeRFs into semantically similar features, as shown in Fig. **??**. Specifically, points in the 3D Euclidean space are first mapped to the semantic embedding space via the generator G. For an object NeRF $\mathcal{N}_X$, G will output a feature vector $\boldsymbol{f}_x$ for each point x that has non-zero density in $\mathcal{N}_X$, taking the latent code $\boldsymbol{z}_X$ of $\mathcal{N}_X$ and the coordinate of x as input. Based on such per-point generator features, the original learning objective in equation 2 can thus be reformulated in the feature space of G, where d is implemented as the cosine distance between two features. And there is no need to use any 3D ground-truth annotations for learning the deformation field.

A straightforward solution to model the correspondences between NeRFs is leveraging a conditional neural deformation field $T : \mathbb{R}^3 \times \mathbb{R}^{z_X} \times \mathbb{R}^{z_Y} \mapsto \mathbb{R}^3$ which estimates the offset for each point x of the source NeRF $\mathcal{N}_X$, taking its coordinate and the latent codes z_X and z_Y of source and target NeRFs as input. However, since the source and target NeRFs vary in each iteration, such a solution requires a large model capacity and fails to converge in practice.

To alleviate the computational complexity, we sample a fixed NeRF with a latent code $\boldsymbol{z}_0$ from G as the intermediate template $\mathcal{N}_0$, and reformulate the deformation field T as the composition of two separate conditional neural deformation fields, namely a forward deformation field F that estimates the deformation from a source NeRF $\mathcal{N}_X$ to the template $\mathcal{N}_0$, and a backward deformation field B that estimates the deformation from the template $\mathcal{N}_0$ to a target NeRF $\mathcal{N}_Y$. In this way, for each point x in $\mathcal{N}_X$, its corresponding point y in $\mathcal{N}_Y$ can be retrieved by:

$$x' = x + F(x, \boldsymbol{z}_X), \tag{3}$$

$$y = x' + B(x', \boldsymbol{z}_Y), \tag{4}$$

where $\boldsymbol{z}_X$ and $\boldsymbol{z}_Y$ are corresponding latent codes, and both F and B are implemented as a MLP consisting of 3 fully connected layers. By decomposing the original deformation field T between arbitrary two NeRFs into two deformation fields F and B bridged by a fixed template NeRF, the overall learning complexity is significantly reduced. In practice, the template NeRF $\mathcal{N}_0$ is chosen as $\left(\gamma_{\mathcal{N}_0} = \overline{\gamma}_i, \beta_{\mathcal{N}_0} = \overline{\beta}_i\right)$ which can be intuitively seen as the average shape of the trained dataset.

In the following we at first introduce our novel training objective, which, as described above, are based on generator features of a pre-trained π-GAN. Subsequently, a curriculum training strategy is further introduced to enhance the learning efficiency.

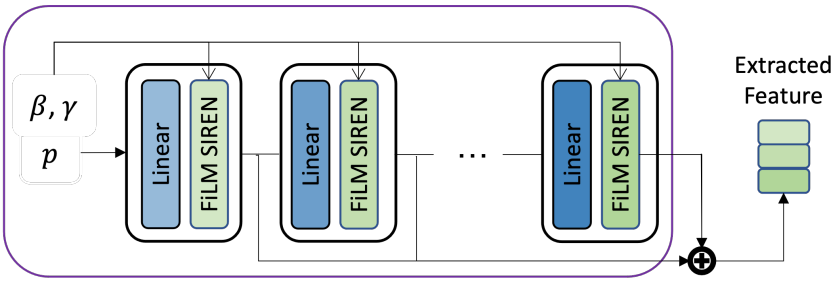


Figure 3: Overview of the generator feature extractor $\mathbf{f}$. We use multiple layers from different channel as the final extracted feature. The generator $G(\cdot)$ takes in a point $p \in \mathbb{R}^3$ and a modulation signal (β, γ) as input to to control the generated content.

3.4 Training Objective

Following equation 2, our overall training objective contains a generator feature similarity loss for estimated correspondences and two additional regularizations for our deformable fields F and B, namely a cycle consistency regularization and a deformation smoothness regularization.

Generator Feature Similarity Loss. Given a collection of n NeRFs $\{\mathcal{N}_i\}_{i=1}^n$ that are sampled from G with corresponding latent codes $\{\boldsymbol{z}_i\}_{i=1}^n$, each of these NeRFs will serve as a source NeRF for F to compute its deformation to the template. For each pair of estimated corresponding points (x, y) where x belongs to one of these sampled NeRFs and y belongs to the template, the cosine similarity between their generator features from G will be used to measure the distance between them. Note that we also include NeRF features after concatenating view directions in the feature extraction. Consequently, the loss for F can be written as:

$$\mathcal{L}_F = \sum_{i=1}^{n} \sum_{x \in \mathcal{P}_{\mathcal{N}_i}} w_x * \text{Similarity}(G(x, \boldsymbol{z}_i), G(x + F(x, \boldsymbol{z}_i), \boldsymbol{z}_0)), \tag{5}$$

where the loss of each point x is weighted by $w_x = T(t_x)$, so that F is encouraged to focus more on points with large densities, as they are close to the object surface with rich semantics. It is worth noting that to reduce the computational redundancy and complexity, we will sample only a subset $\mathcal{P}_{\mathcal{N}_i}$ of points from each NeRF $\mathcal{N}_i$ by the sampling strategy introduced in the next section. Similarly, each of these NeRFs will also serve as a target NeRF for B to compute the deformation of the template to it. The loss for B is thus:

$$\mathcal{L}_B = \sum_{i=1}^{n} \sum_{y \in \mathcal{P}_{\mathcal{N}_0}} w_y * \text{Similarity}(G(y, \boldsymbol{z}_0), G(y + B(y, \boldsymbol{z}_i), \boldsymbol{z}_i)). \tag{6}$$

In practice, instead of using a single generator feature, we adopt features of G at multiple layers and concatenate them to better reflect the semantics of a point.

Cycle Consistency Regularization. Since the conditional deformation fields F and B are supposed to restore the original deformation field T, when the same NeRF $\mathcal{N}_i$ is used as both the source and target NeRF, they should satisfy $B(x+F(x, \boldsymbol{z}_i), \boldsymbol{z}_i) = x$ for all valid points x. Therefore we further apply a cycle consistency regularization for F and B:

$$L_{cycle} = \sum_{i=1}^{n} \sum_{x \in \mathcal{P}_{\mathcal{N}_i}} \|B(x+F(x, \boldsymbol{z}_i), \boldsymbol{z}_i) - x\|_2^2 + \sum_{i=1}^{n} \sum_{y \in \mathcal{P}_{\mathcal{N}_0}} \|F(B(y, \boldsymbol{z}_i), \boldsymbol{z}_i) - y\|_2^2, See figure\ 4 for a clear overview. \tag{7}$$

Deformation Smoothness Regularization. To encourage the smoothness of deformation and reduce spatial distortion, a deformation smoothness regularization is also included. For each point pair (x_1, x_2) in the same NeRF, it requires the distance between this pair to be the same as that of

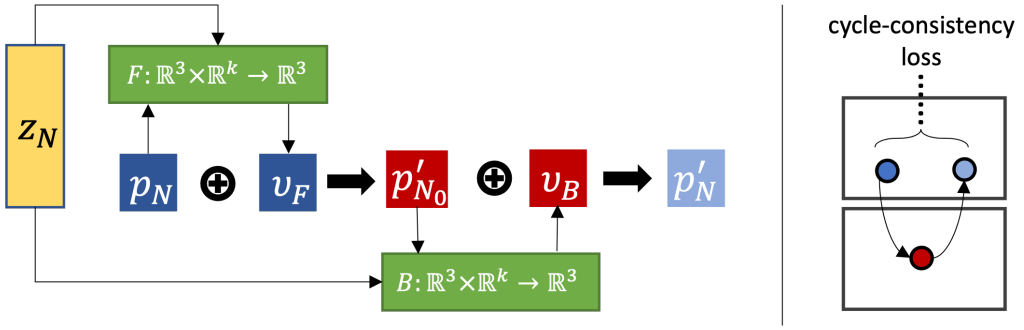


Figure 4: Our model contains two mapping functions: the forward mapping $F(\cdot)$ to map point from a NeRF instance $p_{\mathcal{N}}$ to its corresponding point on template $p_{\mathcal{N}_0}$, and vice cersa for $B(\cdot)$. To regularize the mapping, we introduce cycle consistency regularisation that captures the principle that the bijective mapping of a point should remain itself.

their corresponding points in another NeRF. Therefore it can be written as:

$$\begin{aligned} L_{pair} = &\sum_{i=1}^{n} \sum_{x_1, x_2 \in \mathcal{P}_{\mathcal{N}_i}} \|e^{\mathcal{N}_i}(x_1, x_2) - e^{\mathcal{N}_0}(x_1 + F(x_1, \boldsymbol{z}_i), x_2 + F(x_2, \boldsymbol{z}_i))\|_2 \\ &+ \sum_{i=1}^{n} \sum_{y_1, y_2 \in \mathcal{P}_{\mathcal{N}_0}} \|e^{\mathcal{N}_0}(y_1, y_1) - e^{\mathcal{N}_i}(y_1 + B(y_1, \boldsymbol{z}_i), y_2 + B(y_2, \boldsymbol{z}_i))\|_2^2, \end{aligned} \tag{8}$$

where $e^{\mathcal{N}_i} = \|x_1, x_2\|_2$.

The total objective is thus:

$$\mathcal{L}_{total} = \mathcal{L}_F + \mathcal{L}_B + \lambda_{cycle}\mathcal{L}_{cycle} + \lambda_{pair}\mathcal{L}_{pair}, \tag{9}$$

where λ_{cycle} and λ_{pair} are balancing coefficients, which are respectively set to 0.5 and $1e-4$ in practice.

3.5 Training Strategy

While the generator G of a pre-trained π-GAN serves as the source of infinite training object NeRFs, in each iteration of the training process we will sample a batch of NeRFs $\{\mathcal{N}_i\}_{i=1}^n$ with corresponding latent codes $\{\boldsymbol{z}_i\}_{i=1}^n$. To further sample a point set $\mathcal{P}_{\mathcal{N}_i}$ for each sampled NeRF $\mathcal{N}_i$, for each pixel within the resolution $H \times W$ we shoot a ray $r(t) = o + td$ where d identifies the direction from the camera to the pixel. Subsequently, for each ray we follow (Mildenhall et al., 2020) and conduct a hierarchical sampling to obtain a *fine* set of points. Finally, we take the union of all these fine sets as $\mathcal{P}_{\mathcal{N}_i}$.

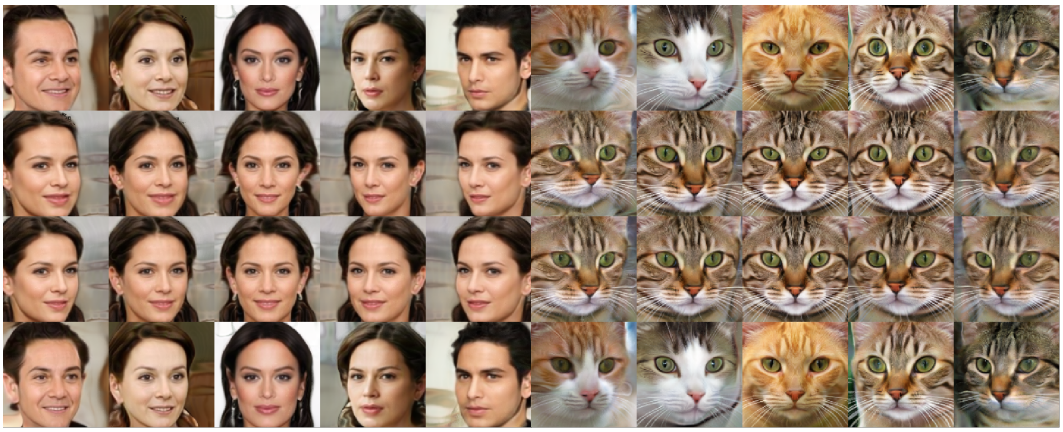
Curriculum Sampling of NeRFs. In practice, we find the variation between a sampled NeRF and the template NeRF significantly affects the training process, which may even collapose at the beginning stage if it gets a sampled NeRF that varies substantially from the template.

To improve training stability and efficiency, we thus adopt a curriculum sampling strategy when obtaining NeRFs from G, which morphs the template NeRF gradually in the latent space to sample NeRFs with growing complexity. Specifically, since in π-GAN the semantics of a sampled NeRF is determined by the modulation signals $(\boldsymbol{\beta}, \boldsymbol{\gamma})$, we can linearly interpolate between two sets of modulation signals to gradually morph one NeRF into another. Inspired by this property of π-GAN, when we sample a set of n NeRFs $\{\mathcal{N}_i\}_{i=1}^n$, we will compute their corresponding modulation signals $\{(\boldsymbol{\beta}_i, \boldsymbol{\gamma}_i)\}_{i=1}^n$ from their latent codes. Subsequently, we will adjust the learning complexity by blending them with the template NeRF as

$$\boldsymbol{\beta}_i(t) = \boldsymbol{\beta}_0 + t \cdot (\boldsymbol{\beta}_i - \boldsymbol{\beta}_0), \tag{10}$$

$$\boldsymbol{\gamma}_i(t) = \boldsymbol{\gamma}_0 + t \cdot (\boldsymbol{\gamma}_i - \boldsymbol{\gamma}_0), \tag{11}$$

where $(\boldsymbol{\beta}_0, \boldsymbol{\gamma}_0)$ are the modulation signals of the template NeRF, and t will gradually increase from 0 to 0.6 as the training process proceeds.



(a) Texture transfer results on CelebA dataset (b) Texture transfer results on Cats dataset

Figure 5: Demonstration of our deformation field through texture transfer. For each samples we transfer the texture from target NeRF according to the their correspondences. This is achieved by volume rending over deformed geometry and texture from the target NeRF. The first two rows show the forward deformation using $F(\cdot)$ which map a NeRF to the template, and the last two rows show the inverse mapping from template to the instance via $B(\cdot)$. **Best viewed with zoom**

4 EXPERIMENTS

4.1 EXPERIMENT SETUP

Here we adopt π-GAN (Chan et al., 2021) as the NeRF-based Generator for dataset generation and point feature extraction 3 and report visual results over Human Face and Cats dataset. Please see the supplementary material for more implementation details.

4.2 RESULTS

We evaluate our methods on three pretrained NeRF-based Generator models for correspondence learning. In Fig. 4.2, we present the dense correspondences generated by our methods between the template NeRF and randomly sampled NeRF through texture transfer, where we transfer textures from sampled NeRF to template through $F(\cdot)$ and inversely, through $B(\cdot)$, to help check the correspondence quality. Visually inspected, our method could establish plausible bijective dense correspondences that indicates the semantic relationship across various NeRF despite their structure variations. This validates that our bijective deformation network learns the underlying structural semantics of different NeRF though no explicit an are provided.

4.3 ABLATION STUDY.

In this section we conduct ablation study to validate the efficacy of our regularisation loss terms. To validate the effectiveness of cycle consistency loss, we conduct self-reconstruction where given a randomly sampled instance, namely a point on a NeRF is first deformed to the template and then deformed back to itself. Fig. 6 showed the rendered reconstructed 3D object as well as the projected 2D loss heatmap between the deformed and the original NeRF. Through the visualization we can easily seen that bijective deformation with consistency loss leads to smaller reconstruction error, as well as fewer distortions during deformation. Regarding point pair regularisation, we qualitatively showed in 7 that the learned correspondences are inferior as well as distorted.

Figure 6: Rendering from self-reconstructed point through cycle deformation. From the left is the input image, rendered reconstructed NeRF with generative feature similarity loss only, and on the right we add cycle-consistency loss. Note that deformation model trained with cycle-consistency loss can perfectly reconstruct itself, which means the learned correspondence are consistent across shapes.

Figure 7: Output from deformation network trained without and with point-pair loss. Without point-pair regularisations, the deformation network tends output distorted visual results. We show deformation results from B on the top row and F on the second row.

5 APPLICATIONS.

Texture Transfer. Using dense correspondence generated by BDF, we are able to transfer textures from one identity to another. Texture transfer has a wide range of applications and intrinsically requires high quality dense correspondence. Here in Fig. 8 we show the texture transfer results across randomly sampled identities. Visual results show that the texture patterns can be preserved and transferred to corresponding semantic areas. This is achieved by querying the density and texture of correspondence point on target NeRF bridged by template. Please see supplementary for the qualitative results.

6 CONCLUSIONS

In this paper, we have presented a network architectures and training strategies to establish robust 3D dense correspondence across NeRF without annotations. Lying in the core of our method is to leverage rich structural priors encapsulated in a pretrained NeRF generative adversarial network (GAN), in which way dense correspondences can be learned in a self-supervised manner. Our experiments further demonstrate that 3D dense correspondences learned from the GAN-derived priors are accurate, smooth and robust to support promising downstream applications. To the best of our knowledge, this is the first method that tries to establish dense correspondence across NeRF representation. We believe this is an inspiring direction and introduces a new solution for NeRF-based human expression editing as well as video reenactment.

REFERENCES

David Bau, Jun-Yan Zhu, Hendrik Strobelt, Agata Lapedriza, Bolei Zhou, and Antonio Torralba. Understanding the role of individual units in a deep neural network. *Proceedings of the National Academy of Sciences*, 2020. ISSN 0027-8424. doi: 10.1073/pnas.1907375117. URL https://www.pnas.org/content/early/2020/08/31/1907375117.

Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale GAN training for high fidelity natural image synthesis. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL https://openreview.net/forum?id=B1xsqj09Fm.

J. Carr, R. Beatson, J. Cherrie, T. J. Mitchell, W. R. Fright, B. C. McCallum, and Tim R. Evans. Reconstruction and representation of 3d objects with radial basis functions. *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, 2001.

Eric Chan, Marco Monteiro, Petr Kellnhofer, Jiajun Wu, and G. Wetzstein. pi-gan: Periodic implicit generative adversarial networks for 3d-aware image synthesis. In *CVPR*, 2021.

Angel X. Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, Li Yi, and Fisher Yu. ShapeNet: An Information-Rich 3D Model Repository. Technical Report arXiv:1512.03012 [cs.GR], Stanford University — Princeton University — Toyota Technological Institute at Chicago, 2015.

Zhiqin Chen and Hao Zhang. Learning implicit fields for generative shape modeling. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5932–5941, 2019.

Yu Deng, Jiaolong Yang, and Xin Tong. Deformed implicit field: Modeling 3d shapes with learned dense correspondence. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10286–10296, 2021.

Vincent Dumoulin, Ethan Perez, Nathan Schucher, Florian Strub, Harm de Vries, Aaron C. Courville, and Yoshua Bengio. Feature-wise transformations. 2018.

B. Egger, W. Smith, Ayush Tewari, Stefanie Wuhrer, M. Zollhöfer, T. Beeler, Florian Bernard, Timo Bolkart, Adam Kortylewski, S. Romdhani, C. Theobalt, V. Blanz, and T. Vetter. 3d morphable face models—past, present, and future. *ACM Transactions on Graphics (TOG)*, 39:1 – 38, 2020.

M. Eisenberger, David Novotný, Gael Kerchenbaum, Patrick Labatut, N. Neverova, D. Cremers, and A. Vedaldi. Neuromorph: Unsupervised shape interpolation and correspondence in one go. In *CVPR*, 2021.

S. M. Ali Eslami, Danilo Jimenez Rezende, Frederic Besse, Fabio Viola, Ari S. Morcos, Marta Garnelo, Avraham Ruderman, Andrei A. Rusu, Ivo Danihelka, Karol Gregor, David P. Reichert, Lars Buesing, Theophane Weber, Oriol Vinyals, Dan Rosenbaum, Neil Rabinowitz, Helen King, Chloe Hillier, Matt Botvinick, Daan Wierstra, Koray Kavukcuoglu, and Demis Hassabis. Neural scene representation and rendering. 360(6394):1204–1210. ISSN 0036-8075, 1095-9203. doi: 10.1126/science.aar6170. URL https://www.sciencemag.org/lookup/doi/10.1126/science.aar6170.

Zhenfeng Fan, Xiyuan Hu, C. Chen, and S. Peng. Boosting local shape matching for dense 3d face correspondence. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10936–10946, 2019.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. pp. 9.

Thibault Groueix, Matthew Fisher, Vladimir G. Kim, Bryan C. Russell, and Mathieu Aubry. 3d-coded: 3d correspondences by deep deformation. In *ECCV*, 2018.

Oshri Halimi, Or Litany, Emanuele Rodola, Alex M Bronstein, and Ron Kimmel. Unsupervised learning of dense shape correspondence. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4370–4379, 2019.

Ali Jahanian, Lucy Chai, and Phillip Isola. On the" steerability" of generative adversarial networks. *arXiv preprint arXiv:1907.07171*, 2019.

Ali Jahanian, Lucy Chai, and Phillip Isola. On the "steerability" of generative adversarial networks. In *International Conference on Learning Representations*, 2020.

O. V. Kaick, Hao Zhang, G. Hamarneh, and D. Cohen-Or. A survey on shape correspondence. *Computer Graphics Forum*, 30, 2011.

Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. URL http://arxiv.org/abs/1812.04948.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *ICLR*, abs/1412.6980, 2015.

O. Litany, Tal Remez, E. Rodolà, A. Bronstein, and M. Bronstein. Deep functional maps: Structured prediction for dense shape correspondence. *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 5660–5668, 2017.

Feng Liu and Xiaoming Liu. Learning implicit functions for topology-varying dense 3d shape correspondence. In *In Proceeding of 2020 Conference on Neural Information Processing Systems*, Virtual, December 2020.

Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J Black. Smpl: A skinned multi-person linear model. *ACM transactions on graphics (TOG)*, 34(6):1–16, 2015.

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*, pp. 4460–4470, 2019.

Lars M. Mescheder, Andreas Geiger, and S. Nowozin. Which training methods for gans do actually converge? In *ICML*, 2018.

Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *European Conference on Computer Vision*, pp. 405–421. Springer, 2020.

Michael Niemeyer and Andreas Geiger. GIRAFFE: Representing scenes as compositional generative neural feature fields. URL http://arxiv.org/abs/2011.12100.

M. Ovsjanikov, M. Ben-Chen, J. Solomon, A. Butscher, and L. Guibas. Functional maps. *ACM Transactions on Graphics (TOG)*, 31:1 – 11, 2012.

Xingang Pan, Bo Dai, Ziwei Liu, Chen Change Loy, and Ping Luo. Do 2d GANs know 3d shape? unsupervised 3d shape reconstruction from 2d image GANs. URL http://arxiv.org/abs/2011.00844.

Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. DeepSDF: Learning continuous signed distance functions for shape representation. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 165–174. IEEE. ISBN 978-1-72813-293-8. doi: 10.1109/CVPR.2019.00025. URL https://ieeexplore.ieee.org/document/8954065/.

Ethan Perez, Florian Strub, Harm de Vries, Vincent Dumoulin, and Aaron C. Courville. Film: Visual reasoning with a general conditioning layer. In *AAAI*, 2018.

C. Qi, Hao Su, Kaichun Mo, and L. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 77–85, 2017.

Jean-Michel Roufosse, Abhishek Sharma, and M. Ovsjanikov. Unsupervised deep learning for structured shape matching. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 1617–1627, 2019.

Y. Sahillioglu. Recent advances in shape correspondence. *The Visual Computer*, 36:1705 – 1721, 2019.

Katja Schwarz, Yiyi Liao, Michael Niemeyer, and Andreas Geiger. Graf: Generative radiance fields for 3d-aware image synthesis. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Yujun Shen, Ceyuan Yang, Xiaoou Tang, and Bolei Zhou. Interfacegan: Interpreting the disentangled face representation learned by gans. *IEEE transactions on pattern analysis and machine intelligence*, PP, 2020.

Federico Tombari, Samuele Salti, and Luigi Di Stefano. Unique signatures of histograms for local surface description. In *European conference on computer vision*, pp. 356–369. Springer, 2010.

Nontawat Tritrong, Pitchaporn Rewatbowornwong, and Supasorn Suwajanakorn. Repurposing gans for one-shot semantic part segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E. Sarma, M. Bronstein, and J. Solomon. Dynamic graph cnn for learning on point clouds. *ACM Transactions on Graphics (TOG)*, 38:1 – 12, 2019.

Yuxuan Zhang, Wenzheng Chen, Huan Ling, Jun Gao, Yinan Zhang, Antonio Torralba, and Sanja Fidler. Image gans meet differentiable rendering for inverse graphics and interpretable 3d neural rendering. *arXiv preprint arXiv:2010.09125*, 2020.

Yuxuan Zhang, Wenzheng Chen, Huan Ling, Jun Gao, Yinan Zhang, Antonio Torralba, and Sanja Fidler. Image gans meet differentiable rendering for inverse graphics and interpretable 3d neural rendering. In *International Conference on Learning Representations*, 2021a.

Yuxuan Zhang, Huan Ling, Jun Gao, K. Yin, Jean-Francois Lafleche, Adela Barriuso, A. Torralba, and S. Fidler. Datasetgan: Efficient labeled data factory with minimal human effort. In *CVPR*, 2021b.

Zerong Zheng, Tao Yu, Qionghai Dai, and Yebin Liu. Deep implicit templates for 3d shape representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1429–1439, 2021.

Tinghui Zhou, Philipp Krähenbühl, Mathieu Aubry, Qixing Huang, and Alexei A. Efros. Learning dense correspondence via 3d-guided cycle consistency. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 117–126, 2016.

Figure 8: Given two randomly sampled identities, we transfer the texture from the second to the first via dense correspondence query bridged by template NeRF. We show six cases here.

A TRAINING DETAILS.

In the experiments, we set the learning rate to $1e-5$ and exponentially decay in $5000, 10000, 15000 and 20000$ iterations. In each iteration, we randomly sample 10 latent codes $z_{i=1}^{10}$ for training. For each synthesized NeRF from latent code z_i, we randomly sample 20% rays per image which generates about $2e^{17}$ points in a single batch. To improve sampling efficiency, we use the integrated depth mask as sampling hint and ignore those background regions for sampling.

To train F and B simultaneously, we sample the same number of points on template NeRF. For each sampled point we conduct positional encoding on their 3D coordinates, which we find facilitates training. All parameters are trained end-to-end using the Adam (Kingma & Ba, 2015) optimizer. Training takes about 8 hours on a NVIDIA V100 GPU with a batchsize of 10 NeRF and 2^{17} samples per NeRF.

B MODEL ARCHITECTURE.

Deformation network F and B has identical architecture. It has 3 layers ReLU-MLP with dim 256. The model concatenates the positional-encoding points $PE(p)$ and the corresponding conditions z_i as input and output the deformation offsets.

C 3D GENERATOR.

We adopt the officially released π-GAN pretrained checkpoint for dense correspondence learning. To extract network features, we use the features starting from layer 4. We find the middle layer features have more correlation with the underlying semantics of given region, while the last few layers are more sensitive to low-level details such as the color variations, which could not provide meaningful clues for dense correspondence learning.

D ADDITIONAL EXPERIMENT RESULTS

See 8.