DOI: 10.1111/cgf.14745
EUROGRAPHICS 2023 / K. Myszkowski and M. Nießner
(Guest Editors)

COMPUTER GRAPHICS *forum*
Volume 42 (2023), Number 2

Unsupervised Template Warp Consistency for Implicit Surface Correspondences

Mengya Liu[1] Ajad Chhatkuli[1] Janis Postels[1] Luc Van Gool[1] Federico Tombari[2]

[1]ETH Zurich
[2]Google, TU Munich

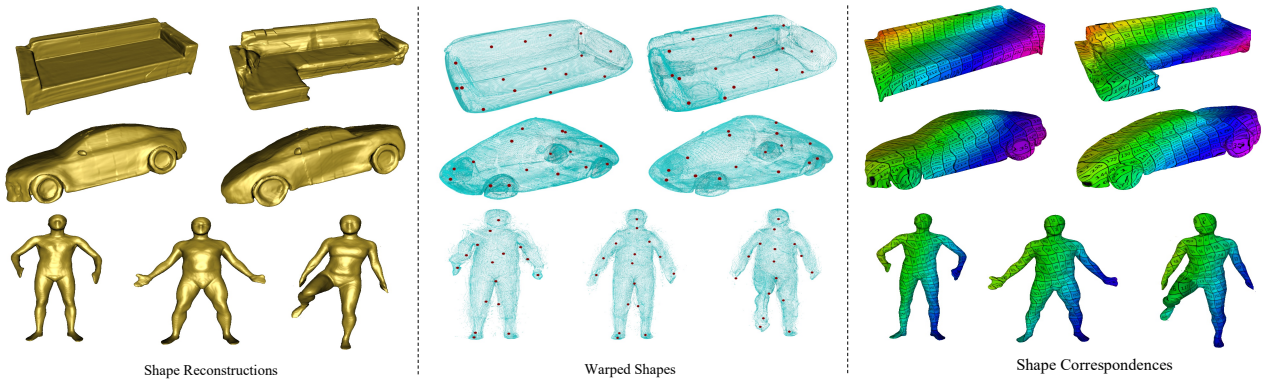

Figure 1: *Our method can implicitly represent 3D shapes with unsupervised dense correspondences for objects with non-rigid transformations or structural variations. For each category, we improve the deformation warp with well-distributed unsupervised sparse keypoints (red points), providing template consistency across the warped shapes and additional feature information. Dense correspondences are presented with the same color and number implying the corresponding mesh. Our method achieves good reconstruction and correspondences with the improvement of template warp consistency.*

Abstract

Unsupervised template discovery via implicit representation in a category of shapes has recently shown strong performance. At the core, such methods deform input shapes to a common template space which allows establishing correspondences as well as implicit representation of the shapes. In this work we investigate the inherent assumption that the implicit neural field optimization naturally leads to consistently warped shapes, thus providing both good shape reconstruction and correspondences. Contrary to this convenient assumption, in practice we observe that such is not the case, consequently resulting in sub-optimal point correspondences. In order to solve the problem, we re-visit the warp design and more importantly introduce explicit constraints using unsupervised sparse point predictions, directly encouraging consistency of the warped shapes. We use the unsupervised sparse keypoints in order to further condition the deformation warp and enforce the consistency of the deformation warp. Experiments in dynamic non-rigid DFaust and ShapeNet categories show that our problem identification and solution provide the new state-of-the-art in unsupervised dense correspondences.

CCS Concepts

• ***Modelling*** → *Shape correspondences;* • ***Modeling*** → *Implicit surface reconstruction;*

1. Introduction

Shape correspondences and representations are keystone problems in computer graphics and vision, essential for applications in shape analysis [LMR*15; SGST20], segmentation [KLF11; KAMC17], animations [WSH*16; ASK*05; WSLG07], and so on. Recent advancements in 3D deep learning, and in particular 3D implicit representations [PFS*19; MON*19; CZ19; SHN*19] and closely related radiance fields [MST*20; PCPM21], have further opened new avenues in shape representation. Similarly, end-to-end training methods have been proposed to obtain unsupervised

sparse [STD*21; JTM*21; SSTN18; FCP*20] as well as dense correspondences [GFK*18b; GFK*18a; ZYDL21; DSO20; JHTG20; UKS*21; YAK*20; RSO19].

An approach that has been particularly useful for learning dense registration of shapes in a category, uses a learned deformation warp on input shape points in a category to a common unknown template space [GFK*18a; DYT21; PSH*21; ZYDL21; LD22; PSB*21]. The consistent common template space is obtained from the deformation warp learned unsupervised, via the point-wise loss [GFK*18a] or implicit function reconstruction loss [DYT21; ZYDL21; LD22; PSB*21]. The latter class simultaneously solves the problem of implicit shape representation and shape correspondences. In particular, the methods of the latter class have shown promising results in both dynamic shapes and rigid shapes of a category. These results are remarkable when we consider the large intra-class variation in shapes, and simple training procedures with no correspondence labels. A key premise of the methods in this class [DYT21; ZYDL21; LD22; PSB*21] is that all input shapes are warped to a common template space discovered during the training, via the minimization of the implicit field loss. The motivation behind this assumption is that in order to predict the correct Signed Distance Field (SDF), the deformation warp must warp all input surface points to a common surface in the template space. We henceforth name this assumption as that of template consistency for convenience.The assumption is directly used but not explored in the original works [ZYDL21; DYT21; LD22] and its derivatives [PSB*21]. Indirectly, this assumption is measured on the quality of the correspondences, as they are obtained by warping the input shape points to the template space and computing nearest neighbors between the points of different warped shapes. Consequently, if the warped shapes are inconsistent, the correspondence estimates will lead to large errors, regardless of the reconstruction performance.

In this paper, we investigate the premise of consistent template discovery [ZYDL21; DYT21] with unsupervised learning for dense correspondences. We observe that, despite reasonable assumptions, the warp optimization often fails in discovering a consistent template space in practice, often despite the good reconstruction accuracy. This is thanks to the implicit function decoder, which can accommodate for inaccuracies of the deformation warp. In order to achieve warped shape consistency, we explore two different aspects. First, we directly address the template inconsistency by using unsupervised sparse keypoints [CLC*20; FCP*20; JTM*21]. The added loss encourages the deformation of the input sparse points in a single batch to be the same on the template space. Second, we modify the network architecture in order to improve the expressive power and the inductive bias. A warp with higher expressive power [ZYDL21; DYT21] can represent larger variations in shapes but may provide less consistent warped shapes due to the weakly constrained optimization loss. On the other hand, a strongly conditioned warp [LD22; PKGF21] may be difficult to optimize for strong deformations such as in a rigid category of shapes with missing parts. Therefore, we opt for a deformation warp that has a suitable balance of both expressive power and strong conditioning inspired from a related work [HTKS19]. We sacrifice the invertibility constraint in [HTKS19] in favor of the expressive power. Finally, we improve the conditional input to the warp by adding point-wise features obtained from the sparse points, which is available for little extra cost. We carefully ablate our network design and explicit consistency constraints using the challenging ShapeNet [CFG*15] and DFaust [BRPB17] datasets. Our comparison on the datasets shows that the method outperforms the baselines and warp-based methods in correspondences as well as reconstruction.

Contributions. We list the contributions of our work below.

- We show the observation that template consistency is not granted from the reconstruction loss in most cases (see Table 4 and 5).
- We propose explicit constraints designed to encourage consistent warped shapes. We choose to use unsupervised sparse points instead of using all the dense points of the shapes, in favor of robustness.
- We choose a deformation network architecture inspired from a recent work [HTKS19] but without the full invertibility constraint, in order to balance the expressive power and a strong inductive bias. We improve the warping function's conditional input by using point-wise feature vectors extracted from the unsupervised sparse keypoints.
- We show with experiments that our method improves correspondences in both dynamic shapes and rigid shapes in a category, along with ablation studies.

2. Related work

We briefly describe the related works on dense 3D shape correspondences for a collection of shapes in a category.

Descriptor-based shape correspondences. Early methods on shape correspondences relied on point-wise descriptors such as [STD14; BK10] in order to establish and refine matches. Rather than solving correspondences for a collection of shapes, methods of this class are optimized for solving correspondences between two shapes. Blended maps [KLF11; GFK*18b] and functional maps [OBS*12; LRR*17; GR20; RSO19; DSO20] are key examples of such methods. While learning has improved their performance [DSO20], methods of this class struggle to generalize to large deformations and across shape categories.

End-to-end learned shape correspondences. A large chunk of recent literature computes dense correspondences over a shape category [GFK*18a; NMOG19; CFB*21; ZYDL21; DYT21; UKS*21; YAK*20; GCV*19]. Learning dense correspondences in such a setting can be accomplished by using a learned deformation warp together with a shape template [GFK*18a; UKS*21]. In contrast to warps and templates selected for specific object types [ZB15; LMR*15], learned warps and templates can generalize to a large variety of deformations and shape categories. A related class of methods uses the learned deformation warps without templates by optimizing shape evolution [NMOG19; CFB*21] in a collection of shapes. However, such a framework requires the input shapes to exhibit limited deformations. In contrast, template-based methods provide correspondences under large deformations while still having an $O(n)$ training complexity with the number of shapes. Recent developments in neural implicit representations [PFS*19; CZ19; SHN*19; MON*19] have further pushed the envelope of 3D shape representation, making simultaneous reconstruction and correspondence estimation possible [ZYDL21; CFB*21; DYT21;

LD22]. Despite these developments, limitations exist due to the assumption that a consistent template is granted through training optimization.

Deformable neural warps for correspondences. Affine warps [GFK*18a] provide the simplest way to train deformable function going from the input shapes to the template. Recent works such as [ZYDL21; DYT21; PSB*21; PKGF21] have favored specific architectures in favor of the expressive power, for example by composing standard LSTM cells [HS97]. Other modifications include [LD22; HTKS19; PKGF21] which use the hard bijective inductive bias via normalizing flow [RM15; KSJ*16] inspired architectures [YHH*19; KBV20; PLS*21; JHTG20]. These methods have been shown to perform very well for continuous deformations such as in human body or animals [LD22]. Although, the flow-based warp provides the necessary inductive-bias of hard bijectivity, it comes at the cost of the warp's expressive power. A recent work [HTKS19] proposed to use a LSTM cell [HS97] on top of each normalizing flow layer in order to encode temporal contexts for modeling deformations. Inspired from the work, we choose a similar network architecture for modeling input to warped shape deformation.

3. Method

3.1. Method Overview

Our work aims to reconstruct the underlying 3D shapes while also solving the correspondence maps between the shapes of a category. Without the ground-truth correspondences, discovering a consistent template via deep INRs is a challenging problem. Previous methods DIT [ZYDL21], CaDEX [LD22] learn the implicit template field by deforming the input shape points to the template 3D space, while DIF-Net [DYT21] achieves it by learning the offset between input shape to the deformed shape; thus solving correspondences in a shape category. We follow the same paradigm and specifically consider the input-to-template deformation [GFK*18a; ZYDL21; LD22] motivated by empirical evidence (see in Section 4.1).

Park et al. [PFS*19] defines an INR for each shape $\mathcal{X}_i$ in a category as a SDF function: $\Phi(\mathsf{x};\mathsf{c}_i) = s_i$, where $\mathsf{x} \in \mathbb{R}^3$ is a point around the surface $\mathcal{X}_i$ and $s_i \in \mathbb{R}$ is the respective scalar field value. Finally $\mathsf{c}_i \in \mathbb{R}^l$ is the latent code representing the shape instance trained together with the weights of Φ. The SDF function Φ is typically parameterized by MLPs [PFS*19; ZYDL21]. We follow the same approach and use an auto-decoder model in order to learn the latent code representing each shape in a category. Further, following [ZYDL21; LD22; DYT21], we decompose Φ as the following function composition:

$$\Phi(\mathsf{x};\mathsf{c}_i) = \mathcal{T} \circ \mathcal{D}_i = s_i, \tag{1}$$

where $\mathcal{D}$ is the warping function that transforms a point x around the shape $\mathcal{X}_i$ to a consistent template space $\Omega \subset \mathbb{R}^3$. $\mathcal{T}$ is the template INR defined on the warped space Ω. An important note regarding the framework is that the template INR is defined on the domain of the deformed space, and directly outputs the SDF of the shape instance $\mathcal{X}_i$. Thus $\mathcal{T}(\mathcal{D}_i) = s_i$ is an SDF field in a strict sense. However, the so-called template space [ZYDL21] defined as $\mathcal{T}(\mathsf{x})$ is strictly speaking not an SDF. Nonetheless, the smooth mapping $\mathcal{D}_i$ or rather its inverse, can result in a meaningful interpretation of $\mathcal{T}(\mathsf{x})$ as an SDF, particularly at the zero-crossings. We consider such interpretations only for visualization and analysis purposes.

We are interested in the study of the deformation warp $\mathcal{D}$ (Section 3.2) so that better correspondences can be established between shapes, possibly also improving the overall INR Φ. We achieve this without using any ground-truth supervision from part or semantic labels. Moreover, we improve the method by introducing explicit constraints, described in Section 3.3.1.

3.2. Deformation Warp

Eq. (1) elegantly advocates that a single template SDF is enough to learn all shapes in the category through the conditional deformation function $\mathcal{D}$. However, we should also note the assumption inherent in the approach. Here we assume that Φ is correct only if $\mathcal{D}, \mathcal{T}$ can be optimized correctly. Specifically, in order to reach a reasonably correct accuracy for Φ, $\mathcal{D}$ must also be correct. However, counter-intuitively, our experiments show otherwise. Thus, we can obtain a good accuracy in the INR Φ even when $\mathcal{D}$ is largely just an identity map. Thus the template INR $\mathcal{T}$ can "cover" for the deficient warp $\mathcal{D}$ during the optimization.

We tackle the deficient $\mathcal{D}$ in three different ways – by adding *warp inputs*, improving *deformation warp architecture* and adding *template consistency loss*.

Local-global shape context. Recent works [ZYDL21; DYT21; LD22] consider $\mathcal{D}$ as the function of the local point position x and the global shape auto-decoder or auto-encoder feature c_i. However, point coordinate x contains little knowledge of the whole shape, while c_i is a global feature learned with reconstruction loss. We argue that local features learned with global shape contexts may significantly help the warp. Intuitively speaking, if the deformation function $\mathcal{D}$ knows that a given point is on a specific leg of the chair, it may learn to warp that point better. With the addition of such features as input, we define the deformation warp as:

$$\mathcal{D}(\mathsf{x},\mathsf{f};\mathsf{c}_i) = \mathsf{p} \in \Omega \subset \mathbb{R}^3. \tag{2}$$

Here, p is the deformed point in the template space Ω. $\mathsf{f} \in \mathbb{R}^t$ is a pointwise feature with local-global shape context for a shape $\mathcal{X}_i$. In our work, we compute f for each point x, with the help of a structural points network:

$$\mathcal{N}(\mathsf{x},\mathsf{X}_i) = \left[\mathsf{f}, \{\mathsf{q}_j\}\right],\ j = 1,\ldots,m, \quad \mathsf{q}_j \in \mathbb{R}^3. \tag{3}$$

We denote the point cloud of the shape $\mathcal{X}_i$ as $\mathsf{X}_i \in \mathbb{R}^{n\times 3}$. More precisely, we train an encoder-decoder network $\mathcal{N}$ in an unsupervised manner to obtain $n \times m$ heatmap representations providing sparse ordered keypoints $\{\mathsf{q}_j\}^m$ for each shape $\mathcal{X}_i$ [FCP*20; CLC*20]. We vectorize the final heatmap representation and use nearest-neighbors between the original points X_i and sparse points $\{\mathsf{q}_j\}$ to extract the $n \times 1$ probability or heatmap features. Note that, in practice $\mathcal{N}$ is evaluated once for the whole shape. However, for mathematical convenience, we also use a single input point x to denote f as its corresponding feature vector in Eq. (3). We use the implementation of [CLC*20] for the network $\mathcal{N}$, due to its simple loss design.

Deformation warp architecture. The network architectures of

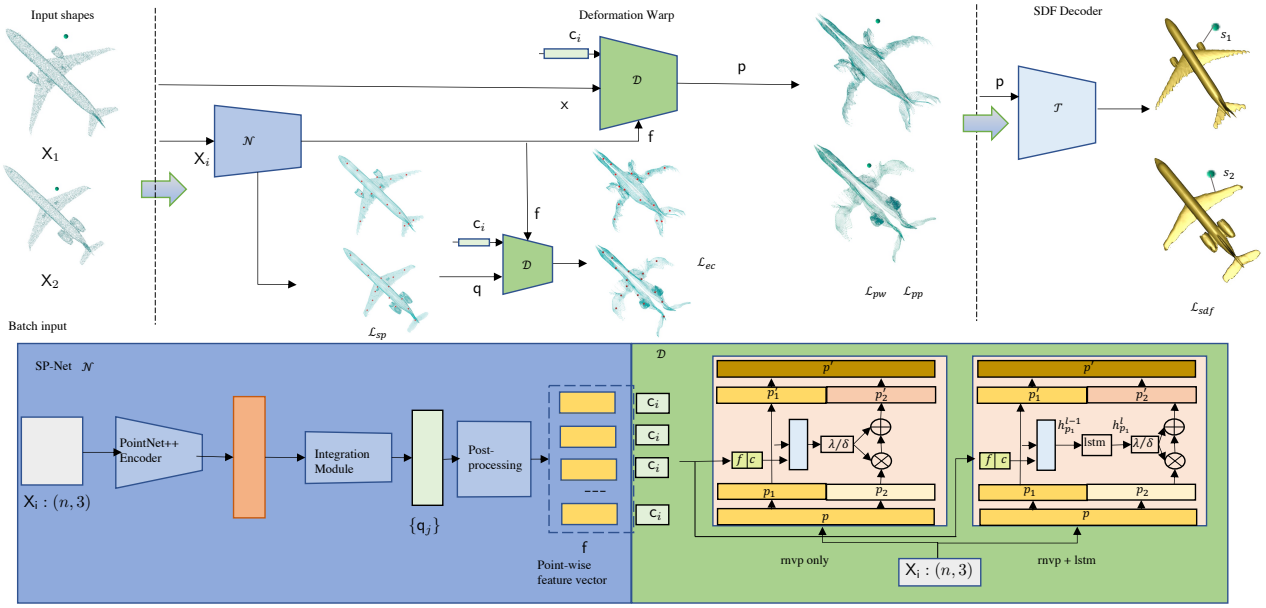

Figure 2: *Network architecture. The network consists of three sub-modules, a Deform-Net $\mathcal{D}$ deforms different shapes into the canonical template space, a Template-Net $\mathcal{T}$ learns to reconstruct the shape by predicting SDF values, and a SP-Net $\mathcal{N}$ predicts sparse keypoints and learn local features for each point. An additional Explicit-Constraint is applied to warped sparse keypoints to force the warped keypoints points to be consistent. The below two pink boxes are the detailed design of one coupling layer in $\mathcal{D}$. We present a basic "rnvp only" framework and a "rnvp + lstm" setup. The latent code* c *is optimized during training, and local feature* $\mathbf{f}$ *is learned from $\mathcal{N}$. They are later concatenated to provide the condition on the warp.*

the deformation warp are designed with specific inductive biases and thus affect the supported space of deformations. We consider bijectivity as well as the flexibility of disregarding the prior when learning a generic deformation warp for correspondences. This is helpful, especially when dealing with shapes in a category consisting of topological changes, missing parts and discontinuities that cannot be explained with hard bijectivity.

We do so by using a warp architecture with a base layer composed of flow-based deformation and simply adding an LSTM cell [HS97] on top of it. Unlike [HTKS19; LD22], we do not constrain our warp with hard bijectivity. See Figure 2 for the illustrated network architecture. The red rectangular box represents the warp $\mathcal{D}$. We first describe the invertible flow-based deformation layer [PKGF21; LD22], henceforth referred to as R-NVP (Real-valued Non-Volume Preserving) [DSB17]. We use generic input and condition variables to formally define the network. Given a point $p \in \mathbb{R}^{d(d=3)}$ and a conditional variable $\mathsf{c} \in \mathbb{R}^l$, a coupling layer first splits p into $p_1 \in \mathbb{R}^{d_1}$ and $p_2 \in \mathbb{R}^{d_2}$ where $d_1 + d_2 = d$, and operate only on p_2 while keeping p_1 unchanged,

$$p_1' = p_1, \quad p_2' = p_2 \odot \mathrm{e}^{\lambda(p_1,c)} + \delta(p_1,c), \tag{4}$$

where λ and δ are independent scaling and translation functions. $\odot$ and $+$ are element-wise product and addition respectively. The architecture allows an exact inverse from the output p' to input p,

$$p_1 = p_1', \quad p_2 = (p_2' - \delta(p_1',c)) \odot \mathrm{e}^{-\lambda(p_1',c)}. \tag{5}$$

We implement the R-NVP blocks in $\mathcal{D}$ with a stack of coupling layers. Scaling and translation functions are composed of simple MLPs, see in Figure 2 for the description of a single layer. R-NVP models strictly bijective deformation warps, which are ideal for deformations such as those of single objects, *e.g.*, human body, animal, clothes, *etc.*. However, it is inadequate in modeling deformations between rigid objects of different 3D structures as in shapes of a category. Shapes of a category such as chairs, and sofas, have missing parts, large scaling, etc. Some of them naturally induce a many-to-one relationship in correspondences. We, therefore, propose to add a recurrent layer [HS97] to learn the relationships which are not bijective and improve warp non-linearity. The recurrent layer $\mathcal{R}$ allows the network to learn and pass a hidden state $h^{(r-1)}$ from the last layer $(r-1)$ to the current layer t before the scaling and translation function.

$$h^r = \mathcal{R}([c, p_1]), \; p_2' = p_2 \odot \exp(\lambda(h^r)) + \delta(h^r), \tag{6}$$

$\mathcal{R}$ denotes the recurrent layer, *e.g.*, an LSTM cell [HS97], and $h \in \mathbb{R}^{d_2}$ denotes the hidden state. Compared to MLPs, a recurrent module helps learn from the previous layer potentially increasing the expressiveness of the warping function. Noticeably, adding a recurrent module in the network directly affects the explicit inverse parameterization since it requires the hidden state from the last layer to be known in order to invert the warp, see in equation (7). Although p_1 is unchanged, the computation of p_2 requires the hidden state h^r which is obtained from layer $(r-1)$.

$$p_1^{r-1} = p_1'^r, \quad p_2^{r-1} = (p_2'^r - \delta(h^r)) \odot \exp(-\lambda(h^r)). \tag{7}$$

Let each shape point cloud X_i is deformed by $\mathcal{D}$ resulting in

$\mathsf{P}_i \in \mathbb{R}^{n\times 3}$ conditioned on its latent representation c_i. Previous methods [ZYDL21; DYT21; PSB*21] assume that each P_i converges to the same template point set P due to the reconstruction loss. Note that Ω as defined in Eq. (2), is an abstraction of P. In Section 4, we show counter-examples of such assumptions and in Section 3.3.1, we show how template consistency can be achieved with explicit constraints.

3.3. Loss Functions

Our model is trained fully end to end. Following Eq. (1), we build the main reconstruction loss using a mean square $\mathcal{L}_{sdf}$ to measure the reconstruction loss as follows.

$$\mathcal{L}_{sdf} = \sum_{\mathcal{X}_i \in \mathcal{C}} \sum_{\mathsf{x}\in \mathsf{X}_i} ||\mathcal{T}(\mathcal{D}(\mathsf{x},\mathsf{f};\mathsf{c}_i)) - \mathbf{SDF}_i||, \tag{8}$$

where $\mathbf{SDF}_i$ represents the corresponding ground-truth SDF values around the shape $\mathcal{X}_i$. $\mathcal{C}$ indicates the set of shapes in the category. By predicting the correct SDF values in the template field, the network gradually learns a template space.

Geometric regularizations. In order to train the deformation warp and limit its space of solutions during optimization, we encourage the warp to preserve local geometric properties. Without such constraints, $\mathcal{D}$ may settle for solutions that do not conform to real shapes, due to the inherent ambiguity of the correspondence problem. Local geometric constraints in shape deformations have been well-studied in the literature. A non-exhaustive list includes isometry [EP09; BGC*15; BPG*20], as-rigid-as-possible [IMH05; SA07] conformality [WWJ*07; SBBG11], Laplacian parametrizations [SCL*04; ZHS*05; AGK*22], point neighborhood and identity priors [YAK*20; ZYDL21], etc. Considering their good performance, we follow those of DIT [ZYDL21]. In particular, we minimize distance and location changes induced by $\mathcal{D}$ using the point-wise regularization $\mathcal{L}_{pw}$ which minimizes the position shift after deformation, and the point-pair regularization $\mathcal{L}_{pp}$ to minimize space distortions.

$$\begin{aligned}\mathcal{L}_{pw} &= \sum_{\mathsf{x}\in\mathsf{X}_i} h(\|\mathcal{D}(\mathsf{x},\mathsf{f};\mathsf{c}_i) - \mathsf{x}\|_2),\\ \mathcal{L}_{pp} &= \sum_{\mathsf{x},\mathsf{y}\in\mathsf{X}_i}\sum_{\mathsf{x}\neq\mathsf{y}} \max\left(\frac{\|\triangle\mathsf{x} - \triangle\mathsf{y}\|_2}{\|\mathsf{x}-\mathsf{y}\|_2} - \varepsilon, 0\right),\end{aligned} \tag{9}$$

where $h(\cdot)$ is the Huber function and $\triangle\mathsf{x} = \mathcal{D}(\mathsf{x},\mathsf{f};\mathsf{c}_i) - \mathsf{x}$ is the position shift of a point $\mathsf{x} \in \mathsf{X}_i$. Additionally, y is a neighbor of x.

3.3.1. Explicit-Constraints on the Deformation Warp

The goal of template space consistency is to ensure that all the deformed or warped shapes in a category are in the same shape space. In order to improve the robustness of the approach, we choose to apply the consistency loss on pre-selected keypoints rather than all points. Unsupervised keypoint selection in a category of shapes is a well-studied problem [CLC*20; FCP*20; JTM*21; JTM*21]. Such sparse keypoints or structural points [CLC*20] describe identifiable and shape-descriptive sparse points in shapes. We train a simple network SP-Net [CLC*20] to extract structural points for the input shapes, see in Figure 2. Given the input point cloud X_i, the sub-network first uses a PointNet++ [QYSG17] encoder to extract sample points along with the probability maps. Following post-processing we obtain the confident sparse points: $\{\mathsf{q}_j\}^m$. We refer the reader to [CLC*20] for more details. Nevertheless, the training loss used for the sparse points is a simple one, as follows:

$$\mathcal{L}_{sp} = \mathrm{CD}(\mathsf{X}_i, \{\mathsf{q}_j\}). \tag{10}$$

The bidirectional Chamfer loss in Eq. (10) enforces that the sparse points are well-distributed and on the original shape.

Once we have discovered the sparse points and they are accurate enough, we proceed with the consistency loss. We use the SP-Net to predict sparse structural points on the input shapes and then warp the predicted structural points to the template space. We then use the distance of warped sparse points for different shapes in the batch as the explicit consistency loss. Note that the sparse points are ordered and therefore already have pre-defined correspondences. Consequently, the Chamfer loss is not needed.

$$\mathcal{L}_{ec} = \sum_{i,k}\sum_{j} \|(\mathcal{D}(\mathsf{q}^i_j, \mathsf{f}^i_j; \mathsf{c}_i) - \mathcal{D}(\mathsf{q}^k_j, \mathsf{f}^k_j; \mathsf{c}_k))\|, \quad i \neq k. \tag{11}$$

By a slight abuse of notation, we use the subscript j to denote the index of the sparse point and the corresponding feature, which have a well-defined order. We use i and k to denote the shape indices in the batch. One important requirement for Eq. (11) is that the batch size of the shapes must be larger than 1. For any batch-size of b, we only keep a fixed b number of comparisons instead of having all possible $C(b,2)$ number of combinations. In Eq. (11), we select all b shapes on one side for the variable i and randomly sample the other b number of shapes for the variable k on the other side, while ensuring $i \neq k$.

Thus, the total loss is defined as below:

$$\mathcal{L} = w_0\mathcal{L}_{sdf} + w_1\mathcal{L}_{\mathbf{c}} + w_2\mathcal{L}_{pp} + w_3\mathcal{L}_{pw} + w_4\mathcal{L}_{sp} + w_5\mathcal{L}_{ec}, \tag{12}$$

where $\mathcal{L}_{\mathbf{c}}$ is the regularization term for latent code in an auto-decoder model, that is: $\mathcal{L}_{\mathbf{c}} = \sum_i ||\mathsf{c}_i||_2^2$. $w_0 \ldots w_5$ are the loss weights. Note that, we don't use $\mathcal{L}_{ec}$ during the training in 500 epochs when the SP-Net module is not stable and the predicted structural points are not accurate. Later, w_5 is set to 0.001 after 500 epochs to apply the explicit constraints. We further provide more detailed parameters in the Appendix.

4. Experiments

In this section, we detail our experiments including the comparisons and analysis against the state-of-the-art methods and the ablation studies. Finally, we provide some applications that are achieved with template-based dense correspondences. Additional results are also provided in the Appendix. Code is available at https://github.com/lmy1001/template_warp_consistency.

Datasets & preprocessing. We train and evaluate our proposed approach as well as baselines on both Shapenet [CFG*15] and DFaust [BRPB17] datasets. Following DeepSDF [PFS*19], we prepare SDF samples and use the same train/test split. For a fair comparison with DIF-Net [DYT21] which uses a different train/test split, we conduct another DIF-Net test set based evaluation. Moreover, inspired by DIF-Net [DYT21], we use semantic labels from ShapeNet-Part [MZC*19] to conduct label transfer experiments. As for DFaust dataset, after preparing SDF values in the same

Method	CD Mean ↓				CD Median ↓				mIoU ↑		
	Airplanes	Sofas	Cars	Chairs	Airplanes	Sofas	Cars	Chairs	Airplanes	Cars	Chairs
NN points	-	-	-	-	-	-	-	-	71.3	65.9	73.1
AtlasNet [GFK*18b]	0.22	0.41	-	0.37	0.065	0.31	-	0.28	-	-	-
SIF [GCV*19]	0.44	0.80	1.08	1.54	-	-	-	-	-	-	-
DeepSDF [PFS*19]	0.14	0.12	0.11	0.24	0.061	0.08	-	0.10	-	-	-
DSIF [HASB20]	0.22	-	-	0.45	0.140	-	-	0.21	-	-	-
C-DeepSDF [DZW*20]	0.07	0.11	0.06	**0.16**	0.033	0.07		**0.06**	-	-	-
DIT [ZYDL21]	0.053	0.102	0.052	0.20	0.027	0.066	0.042	0.07	71.4	65.7	79.6
Ours	**0.050**	**0.098**	**0.050**	0.19	**0.021**	**0.063**	**0.041**	**0.06**	**73.8**	**66.7**	**80.7**

Table 1: *Reconstruction performance on ShapeNet on DeepSDF [PFS*19] test set. CD Mean and CD Median are multiplied by* 10^3. *mIoU is represented with %.*

Method	CD Mean ↓		CD Median ↓		mIoU ↑	
	Airplanes	Chairs	Airplanes	Chairs	Airplanes	Chairs
NN points	-	-	-	-	69.6	75.8
DIF-Net [DYT21]	0.082	0.210	0.032	0.127	60.7	65.5
DIT [ZYDL21]	0.057	0.115	0.021	**0.065**	72.8	81.6
Ours	**0.056**	**0.109**	**0.020**	0.071	**76.2**	**82.5**

Table 2: *Reconstruction performance on ShapeNet on DIF-Net [DYT21] based test set. CD Mean and CD Median are multiplied by* 10^3. *mIoU is represented with %.*

way as ShapeNet, we follow the practice of OFlow [NMOG19] to split the data. In contrast to OFlow, we do not require continuous deformations between shapes, so we prepare the training set by sub-sampling the complete training sequences to 2040 shapes only. DFaust provides high-quality ground-truth correspondences in each sequence, which are not available in ShapeNet. More details about data pre-processing are in the Appendix.

Baselines. We compare against state-of-the-art works on the task of 3D reconstruction on Shapenet: AtlasNet [GFK*18b] (using explicit mesh parameterization), DeepSDF [PFS*19] (deep implicit field), DIT [ZYDL21] (building template space with deep implicit function) and DIF-Net [DYT21] (generating deformed implicit filed with deep implicit function). We use the per-category pre-trained models provided by DIT and DIF-Net for the evaluation on ShapeNet, while other results are from their paper. We also compare to PSGN [FSG17], ONet[MON*19], OFlow [NMOG19] (uses a Neural-ODE [CRBD18] to learn correspondences between frames) and CaDex [LD22] which learns an invertible deformation and achieved good performance on DFaust. We train the model for DIT on DFaust, other results follow OFlow.

Metrics. We use the Chamfer distance ("CD Mean" and "CD Median") for the evaluation of shape reconstructions in ShapeNet, and we evaluate the intersection over union (IoU) according to the ground truth semantic labels and predicted semantic labels for the evaluation of label transfer. The evaluation on DFaust inherits the metrics on OFlow using "CD $\ell 1$" and $\ell 2$ distance error ("Corr") for reconstruction and correspondences respectively.

Moreover, we measure the template consistency "CD Temp $\ell 2/\ell 1$" by computing the Chamfer distance (for Shapenet it is Chamfer $\ell 2$ and for DFaust it is Chamfer $\ell 1$) between the template to all the warped shapes. In order to obtain the template, we use the SDF decoder without the Deform-Net.

Training details. We implement the network $\mathcal{D}$ as an auto-decoder containing 6 layers each consisting of R-NVP with an LSTM cell, together with an SDF decoder, and the SP-Net $\mathcal{N}$. We train the whole network with 4 NVIDIA 2080Ti GPUs for 2000 epochs with batch size 40, with the Adam optimizer. We refer to the Appendix for additional training details.

4.1. Results on Shapenet

Shape Reconstruction. We report results on ShapeNet in Table 1 (tested on DIT test set) and Table 2 (tested on DIF-Net based test set). ShapeNet [CFG*15] contains several object categories, which often have missing parts or often topological changes. Therefore it is natural that normalizing flow-based methods such as [LD22] often fail and we do not report their results. Among the rest of the methods, our method yields the best reconstruction performance in the categories of planes, cars, and sofas. We are slightly worse than C-DeepSDF [DZW*20] on chairs reconstruction but still better than DIT and DIF-Net, given that the chairs category is quite challenging with topological changes thus, discovering a consistent template is difficult. Qualitatively, we observe similar reconstruction quality to the state-of-the-art on chairs. Visualizations are presented in Figure 1 where we show the reconstructed shapes, warped shapes, and shape correspondences for various categories. Equal colors indicate corresponding points. In terms of warped shapes, our method is able to deform different shapes with structural points into the template space, thus our method reconstructs shapes well and yields dense correspondences via the template space.

Dense Correspondences. Due to the lack of ground-truth corre-

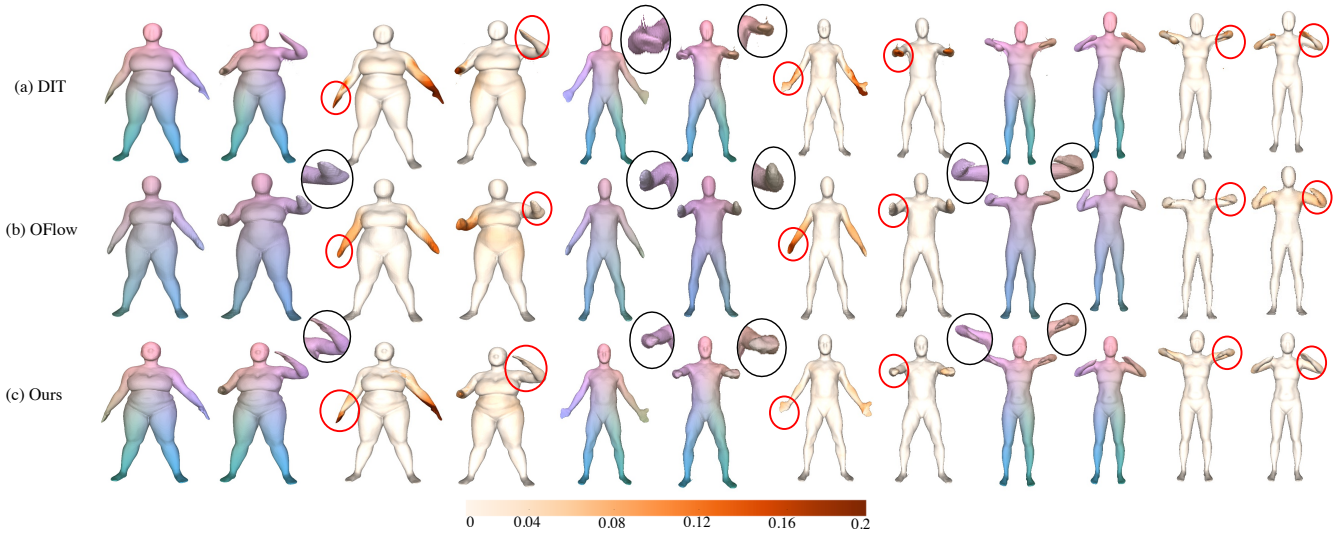

Figure 3: *Qualitative results of reconstruction, correspondence and error color maps on DFaust [BRPB17]. The $1^{st}, 2^{nd}, 5^{th}, 6^{th}, 9^{th}, 10^{th}$ columns display reconstructed shapes with dense correspondences. The same color implies corresponding points, with some parts circled out for clear comparison. The $3^{rd}, 4^{th}, 7^{th}, 8^{th}, 11^{th}, 12^{th}$ columns represent correspondence error in a color map. A darker color denotes a larger error. Parts with large errors are circled out.*

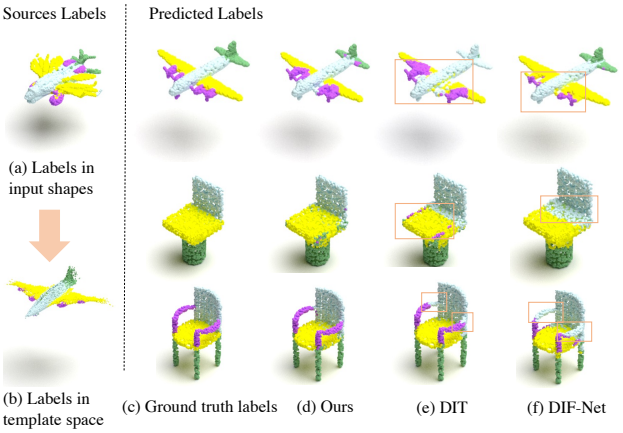

Figure 4: *Label transfer performance in airplanes, chairs and cars categories. The left-most column shows the transfer of source labels from the input shapes to the template space, and the rest columns represent the label transfer results. We show noticeable differences or wrong predictions using the red-boxed area in DIT and DIF-Net.*

spondences in ShapeNet, we resort to semantic label transfer experiment for quantitative evaluation on dense correspondences. Following DIF-Net, we manually select 5 labeled shapes as the source shapes deforming onto the template space, other unlabeled shapes are then deformed onto the template field. We then search for 10 nearest labeled points for each and conduct label voting to predict labels [DYT21]. We compare our method against the nearest neighboring points (NN), DIF-Net and DIT, and report the results in Table 1 and Table 2. Our method yields the best performance in label transfer for all three categories. DIT [ZYDL21] performs similarly to NN on airplanes and cars in Table 1, indicating that the deformation warp is largely identical. Furthermore, as the shapes in these two categories follow similar structures, direct NN also provides decent correspondences. Figure 4 displays the qualitative results. The left-most column shows the generation of source labels from 5 input samples to the template space. The rest columns represent label transfer results. The boxed areas in the figure show noticeable differences with respect to other methods.

4.2. Results on DFaust

Shape Reconstruction and Dense Correspondences. We conduct further experiments to demonstrate the power of modeling non-rigid dynamic shape deformations on DFaust [BRPB17]. The DFaust dataset [BRPB17] contains natural deformation of a human body, different from deformations required to represent a category. Moreover, it consists of high quality ground-truth that is important for our ablation studies. Results are presented in Table 3. "NN points" provides the baseline correspondence between shapes. Surprisingly, even with a smaller training set, we outperform the state-of-the-art methods by a significant margin in reconstruction. The correspondence error follows the "CD Temp $\ell 1$" error, implying the importance of template or warped shape consistency. Although we do not surpass CaDex in correspondence quality, we are able to achieve comparable results with a much smaller training size, while having a more general deformation warp.

We show quantitative results in Figure 3. The figure shows the performance of 3 different sequences (each displays 2 shapes in dense correspondences renders and error color maps) conducted by DIT, OFlow and our method respectively. Equal colors here indicate correspondence. Detailed reconstructions are circled. We ob-

serve that OFlow fails to reconstruct some parts, and DIT sometimes reconstructs with noise. Error color maps are presented to show in which region our method fails. The darker color denotes the larger error. Some parts with large errors in DIT and OFlow are also circled out, where our method predicts good correspondences.

Methods	CD $\ell 1 \downarrow$	Corr $\downarrow$	CD Temp $\ell 1 \downarrow$
NN points	-	0.279	-
PSGN-4D [FSG17]	0.127	3.041	-
ONet-4D [MON*19]	0.140	-	-
O-Flow [NMOG19]	0.095	0.149	-
CaDex [LD22]	0.074	**0.126**	-
DIT [ZYDL21]	0.028	0.237	0.080
Ours	**0.024**	0.137	**0.045**

Table 3: *Results on DFAUST [BRPB17] dataset. CD and CD Temp $\ell 1$ are multiplied by 10, Corr represents correspondence $\ell 2$ error.*

4.3. Ablation Study

We ablate the individual contributions/components of our proposed method. For a clear comparison, we separate the parts of our transformation module into "rnvp + lstm + feature + EC", here "feature" indicates the feature we learned from SP-Net, "EC" denotes the explicit constraint we apply on the deformed shapes. Later we investigate the power of each part by gradually adding different parts including "rnvp only", "rnvp+lstm", "rnvp+lstm+feature", "rnvp+lstm+feature+EC". In order to prove our hypothesis of template or warped shape consistency, we use the consistency measure as described in Metrics. We conduct experiments on airplanes and DFaust one sequence, containing large deformations. Quantitative results are presented in Table 4 and Table 5 respectively.

Results on Shapenet. From the numeric results in Table 4, we see "rnvp only" performs worst on reconstruction and template consistency, but its label transfer result ranks second. The restrictive nature of "rnvp only" network is detrimental to the implicit field defined on the template space. The bad reconstruction implies many outliers in deformed shapes leading to a non-optimal template space and consistency. However, the label transfer experiment is less impacted by outlier transformations. We can see a dramatic improvement on reconstruction when adding the recurrent layer. Moreover, the applied feature is beneficial in both reconstruction and template consistency. Further, the explicit constraints applied on the warp help improve the template consistency resulting in improvement in both reconstruction and label transfer, as well as the template Chamfer distance (last column of the table). "rnvp+lstm+feature+EC" architecture yields the best performance in all metrics. We show the visualization results of deformed shapes regarding the same ground-truth shapes in Figure 5. From left to right are (a) ground-truth shapes, (b) deformed shapes from DIT, (c) from "rnvp + lstm", (d) from "rnvp + lstm +feature" and (e) from "rnvp + lstm + feature + EC". We do not show the warped shapes from "rnvp only" as it always generates large outliers leading to bad visualization. The last column is able to get good template consistency even for shapes with larger structure variations, while (b) DIT label transfer results are similar to "NN points", indicating that the deformation warp is under-performing.

	CD Mean$\downarrow$	mIoU $\uparrow$	CD Temp $\ell 2 \downarrow$
NN points	-	71.3	-
DIT [ZYDL21]	0.053	71.4	2.551
rnvp only	0.303	73.6	23.824
rnvp+lstm	0.057	72.4	4.149
rnvp+lstm+feature	0.052	73.2	2.171
rnvp+lstm+feature+EC	**0.050**	**73.8**	**1.403**

Table 4: *Ablation study on network design, on airplanes category. "CD Mean" and "CD Temp $\ell 2$" are multiplied by 10^3, mIoU is given in %.*

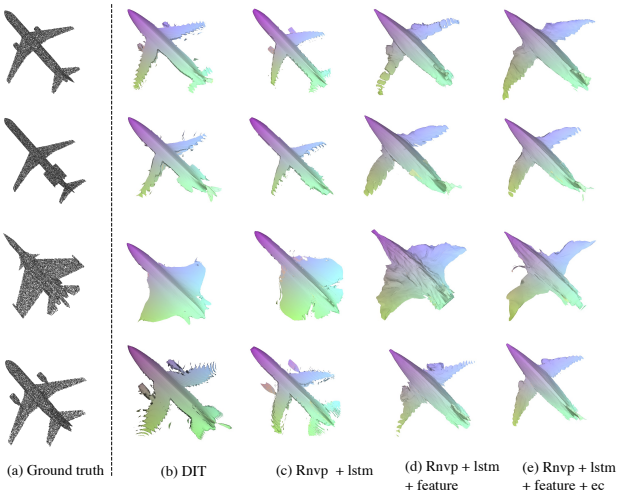

Figure 5: *Qualitative results of deformed shapes on airplanes under different network setups. From left to right are (a)ground truth shapes, and their warped shapes from (b) DIT [ZYDL21], (c) "rnvp + lstm", (d) "rnvp + lstm + feature" and (e) "rnvp + lstm + feature + ec". Equal colors represent correspondences.*

Results on DFaust. In DFaust we use the sequence "50009_chicken_wings" for training/testing due to its higher range of deformation. From Table 5, we see that "rnvp + lstm + feature + EC" yields the best performance in reconstruction and correspondences, worse than "rnvp only" in template consistency. Due to the difficulty of the deformation, both DIT and our base network "rnvp only" fails to reconstruct properly. Adding a recurrent layer is beneficial in reconstruction though it worsens the correspondences. The added feature helps shapes to warp locally and consistently, and the explicit constraint "EC" later alleviates the influence by forcing the template space to be consistent. The template consistency error "CD Temp $\ell 1$" also indicates the correspondence behavior. A surprising result here is that the correspondence error for (b) "rnvp only" is not the lowest despite having the lowest template consistency error. Consequently, while the shape Chamfer distance is low, different parts may be warped to different areas of the shape resulting in sub-optimal correspondences. Furthermore, the high reconstruction error hints that the deformation warp is far from perfect. More visualization results on the deformed shapes on DFaust are presented in Figure 6. From left to right are (a)ground-truth shapes,

deformed shapes from (b) DIT, (c) "rnvp only", (d) "rnvp + lstm", (e) "rnvp + lstm + feature" and (f) "rnvp + lstm + feature + EC". We can observe that the warped shapes in DIT are not consistent, the same happens to the setup in (c) "rnvp + lstm". On the other hand, adding features and "EC" all improve the template consistency.

	CD $\ell 1 \downarrow$	Corr $\downarrow$	CD Temp $\ell 1 \downarrow$
NN points	-	0.207	-
DIT [ZYDL21]	0.039	0.188	0.039
rnvp only	0.035	0.138	**0.022**
rnvp+lstm	0.019	0.165	0.036
rnvp+lstm+feature	0.017	0.113	0.030
rnvp+lstm+feature+EC	**0.016**	**0.099**	0.028

Table 5: *Ablation study on DFAUST [BRPB17] one sequence. CD $\ell 1$ and CD Temp $\ell 1$ are multiplied by 10, Corr represents correspondence $\ell 2$ error.*

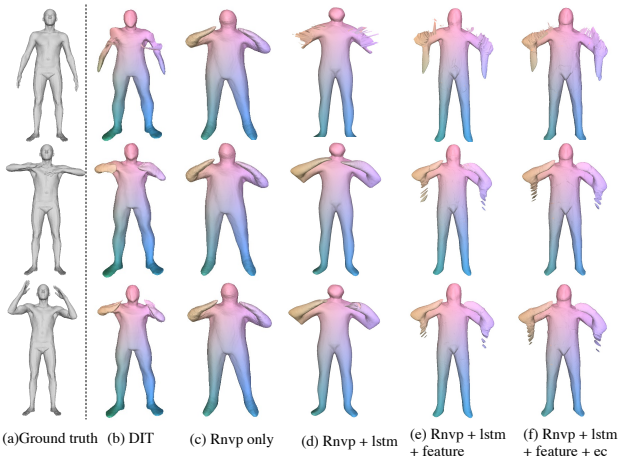

Figure 6: *Qualitative results of warped shapes in DFaust [BRPB17] under different network setups. From left to right are (a) ground truth shapes and their warped shapes from (b) DIT [ZYDL21], (c) "rnvp only", (d) "rnvp + lstm", (e) "rnvp + lstm + feature" and (f) "rnvp + lstm + feature + EC". Equal colors represent correspondences.*

4.4. Applications through Dense Correspondences

To show the flexibility of our method, we explore the application of shape interpolation. The results are presented in Figure 7. By linearly interpolating in the latent space, we obtain the global code of the interpolated shape. We keep the local features, i.e., the sparse point features the same as its closest input shape. With this we are able to interpolate shapes while keeping the dense correspondences. We provide further details and more examples in the Appendix. In Figure 7, the interpolated shapes are visualized with texture. Although the local context can only be approximated, the interpolated latent code can produce a meaningful shape with good correspondences.

Furthermore, with the dense correspondences between input shapes, our method can easily transfer textures between different

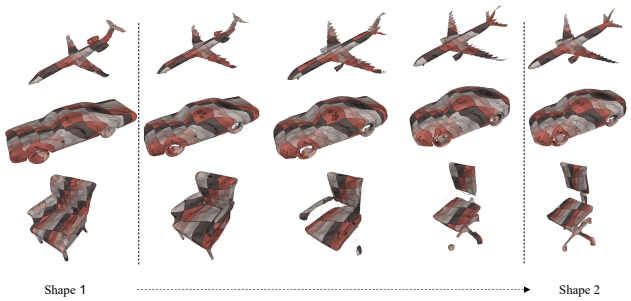

Figure 7: *Shape interpolation between "shape 1" and "shape 2". The same colored checkboard with the same number represents correspondences.*

shapes. We present qualitative results on texture transfer in Figure 8. The texture image is first applied on an input shape using [CWNN20] to generate the textured shape, then the texture is transferred to other shapes through shape correspondences.

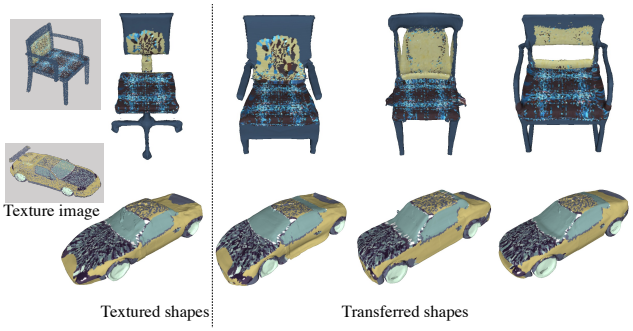

Figure 8: *Texture transfer. The left-most images show the texture image, followed by the textured shapes as input, and the rest are the transferred renders. Our method is able to perform texture transfer with good visual quality using the learned dense correspondences.*

5. Conclusion

We analyzed a key problem in template discovery based unsupervised dense correspondence methods. We discovered that the implicit field reconstruction loss does not always result in consistent warped shapes, thus impacting mainly the correspondences. We tackled the problem in two ways, first by choosing an appropriate deformation warp architecture with additional point-wise feature and second by imposing explicit template consistency constraints. Our experiments showed that the proposed changes significantly improve template consistency and also dense correspondences. Furthermore, the method achieves good performance on both natural quasi-isometric deformations of a human body as well as on ad-hoc deformations between shapes of a category using the exact same warp design.

Acknowledgements. This research is partially funded by the EU Horizon 2020 research and innovation programme under grant

agreement No. 820434 – project ENCORE and VIVO Collaboration Project on Real-time scene reconstruction. Open access funding provided by Eidgenossische Technische Hochschule Zurich.

References

[AGK*22] AIGERMAN, NOAM, GUPTA, KUNAL, KIM, VLADIMIR G, et al. "Neural Jacobian Fields: Learning Intrinsic Mappings of Arbitrary Meshes". *arXiv preprint arXiv:2205.02904* (2022) 5.

[ASK*05] ANGUELOV, DRAGOMIR, SRINIVASAN, PRAVEEN, KOLLER, DAPHNE, et al. "Scape: shape completion and animation of people". *ACM SIGGRAPH 2005 Papers*. 2005, 408–416 1.

[BGC*15] BARTOLI, ADRIEN, GÉRARD, YAN, CHADEBECQ, FRANCOIS, et al. "Shape-from-template". *IEEE transactions on pattern analysis and machine intelligence* 37.10 (2015), 2099–2118 5.

[BK10] BRONSTEIN, MICHAEL M and KOKKINOS, IASONAS. "Scale-invariant heat kernel signatures for non-rigid shape recognition". *CVPR*. 2010 2.

[BPG*20] BEDNARIK, JAN, PARASHAR, SHAIFALI, GUNDOGDU, ERHAN, et al. "Shape reconstruction by learning differentiable surface representations". *CVPR*. 2020 5.

[BRPB17] BOGO, FEDERICA, ROMERO, JAVIER, PONS-MOLL, GERARD, and BLACK, MICHAEL J. "Dynamic FAUST: Registering human bodies in motion". *CVPR*. 2017 2, 5, 7–9.

[CFB*21] CHEN, YUNLU, FERNANDO, BASURA, BILEN, HAKAN, et al. "Neural Feature Matching in Implicit 3D Representations". *ICML*. 2021 2.

[CFG*15] CHANG, ANGEL X, FUNKHOUSER, THOMAS, GUIBAS, LEONIDAS, et al. "Shapenet: An information-rich 3d model repository". *arXiv preprint arXiv:1512.03012* (2015) 2, 5, 6.

[CLC*20] CHEN, NENGLUN, LIU, LINGJIE, CUI, ZHIMING, et al. "Unsupervised learning of intrinsic structural representation points". *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, 9121–9130 2, 3, 5.

[CRBD18] CHEN, RICKY TQ, RUBANOVA, YULIA, BETTENCOURT, JESSE, and DUVENAUD, DAVID K. "Neural ordinary differential equations". *Advances in neural information processing systems* 31 (2018) 6.

[CWNN20] CAO, XU, WANG, WEIMIN, NAGAO, KATASHI, and NAKAMURA, RYOSUKE. "Psnet: A style transfer network for point cloud stylization on geometry and color". *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 2020, 3337–3345 9.

[CZ19] CHEN, ZHIQIN and ZHANG, HAO. "Learning Implicit Fields for Generative Shape Modeling". *CVPR*. 2019 1, 2.

[DSB17] DINH, LAURENT, SOHL-DICKSTEIN, JASCHA, and BENGIO, SAMY. "Density estimation using real nvp". *ICLR* (2017) 4.

[DSO20] DONATI, NICOLAS, SHARMA, ABHISHEK, and OVSJANIKOV, MAKS. "Deep geometric functional maps: Robust feature learning for shape correspondence". *CVPR*. 2020 2.

[DYT21] DENG, YU, YANG, JIAOLONG, and TONG, XIN. "Deformed implicit field: Modeling 3d shapes with learned dense correspondence". *CVPR*. 2021 2, 3, 5–7.

[DZW*20] DUAN, YUEQI, ZHU, HAIDONG, WANG, HE, et al. "Curriculum deepsdf". *ECCV*. 2020 6.

[EP09] EIGENSATZ, MICHAEL and PAULY, MARK. "Positional, metric, and curvature control for constraint-based surface deformation". *Computer Graphics Forum*. Vol. 28. 2. Wiley Online Library. 2009, 551–558 5.

[FCP*20] FERNANDEZ-LABRADOR, CLARA, CHHATKULI, AJAD, PAUDEL, DANDA PANI, et al. "Unsupervised learning of category-specific symmetric 3d keypoints from point sets". *ECCV*. 2020 2, 3, 5.

[FSG17] FAN, HAOQIANG, SU, HAO, and GUIBAS, LEONIDAS J. "A point set generation network for 3d object reconstruction from a single image". *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, 605–613 6, 8.

[GCV*19] GENOVA, KYLE, COLE, FORRESTER, VLASIC, DANIEL, et al. "Learning shape templates with structured implicit functions". *ICCV*. 2019 2, 6.

[GFK*18a] GROUEIX, THIBAULT, FISHER, MATTHEW, KIM, VLADIMIR G, et al. "3d-coded: 3d correspondences by deep deformation". *ECCV*. 2018 2, 3.

[GFK*18b] GROUEIX, THIBAULT, FISHER, MATTHEW, KIM, VLADIMIR G, et al. "A papier-mâché approach to learning 3d surface generation". *CVPR*. 2018 2, 6.

[GR20] GINZBURG, DVIR and RAVIV, DAN. "Cyclic functional mapping: Self-supervised correspondence between non-isometric deformable shapes". *ECCV*. 2020 2.

[HASB20] HAO, ZEKUN, AVERBUCH-ELOR, HADAR, SNAVELY, NOAH, and BELONGIE, SERGE. "Dualsdf: Semantic shape manipulation using a two-level representation". *CVPR*. 2020 6.

[HS97] HOCHREITER, SEPP and SCHMIDHUBER, JÜRGEN. "Long short-term memory". *Neural computation* 9.8 (1997), 1735–1780 3, 4.

[HTKS19] HWANG, SEONG JAE, TAO, ZIRUI, KIM, WON HWA, and SINGH, VIKAS. "Conditional recurrent flow: conditional generation of longitudinal samples with applications to neuroimaging". *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019, 10692–10701 2–4.

[IMH05] IGARASHI, TAKEO, MOSCOVICH, TOMER, and HUGHES, JOHN F. "As-rigid-as-possible shape manipulation". *ACM transactions on Graphics (TOG)* 24.3 (2005), 1134–1141 5.

[JHTG20] JIANG, CHIYU, HUANG, JINGWEI, TAGLIASACCHI, ANDREA, and GUIBAS, LEONIDAS. "ShapeFlow: Learnable Deformations Among 3D Shapes". *NIPS*. 2020 2, 3.

[JTM*21] JAKAB, TOMAS, TUCKER, RICHARD, MAKADIA, AMEESH, et al. "KeypointDeformer: Unsupervised 3D Keypoint Discovery for Shape Control". *CVPR*. 2021 2, 5.

[KAMC17] KALOGERAKIS, EVANGELOS, AVERKIOU, MELINOS, MAJI, SUBHRANSU, and CHAUDHURI, SIDDHARTHA. "3D shape segmentation with projective convolutional networks". *proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, 3779–3788 1.

[KBV20] KLOKOV, ROMAN, BOYER, EDMOND, and VERBEEK, JAKOB. "Discrete point flow networks for efficient point cloud generation". *ECCV*. 2020 3.

[KLF11] KIM, VLADIMIR G, LIPMAN, YARON, and FUNKHOUSER, THOMAS. "Blended intrinsic maps". *ACM transactions on graphics (TOG)* 30.4 (2011), 1–12 1, 2.

[KSJ*16] KINGMA, DURK P, SALIMANS, TIM, JOZEFOWICZ, RAFAL, et al. "Improved variational inference with inverse autoregressive flow". *NIPS* (2016) 3.

[LD22] LEI, JIAHUI and DANIILIDIS, KOSTAS. "CaDeX: Learning Canonical Deformation Coordinate Space for Dynamic Surface Representation via Neural Homeomorphism". *arXiv preprint arXiv:2203.16529* (2022) 2–4, 6, 8.

[LMR*15] LOPER, MATTHEW, MAHMOOD, NAUREEN, ROMERO, JAVIER, et al. "SMPL: a skinned multi-person linear model". *ACM Trans. Graph.* 34.6 (2015), 248:1–248:16 1, 2.

[LRR*17] LITANY, OR, REMEZ, TAL, RODOLA, EMANUELE, et al. "Deep functional maps: Structured prediction for dense shape correspondence". *ICCV*. 2017 2.

[MON*19] MESCHEDER, LARS, OECHSLE, MICHAEL, NIEMEYER, MICHAEL, et al. "Occupancy networks: Learning 3d reconstruction in function space". *CVPR*. 2019 1, 2, 6, 8.

[MST*20] MILDENHALL, BEN, SRINIVASAN, PRATUL P, TANCIK, MATTHEW, et al. "Nerf: Representing scenes as neural radiance fields for view synthesis". *ECCV*. 2020 1.

[MZC*19] MO, KAICHUN, ZHU, SHILIN, CHANG, ANGEL X, et al. "Partnet: A large-scale benchmark for fine-grained and hierarchical part-level 3d object understanding". *CVPR*. 2019 5.

[NMOG19] NIEMEYER, MICHAEL, MESCHEDER, LARS, OECHSLE, MICHAEL, and GEIGER, ANDREAS. "Occupancy flow: 4d reconstruction by learning particle dynamics". *ICCV*. 2019 2, 6, 8.

[OBS*12] OVSJANIKOV, MAKS, BEN-CHEN, MIRELA, SOLOMON, JUSTIN, et al. "Functional maps: a flexible representation of maps between shapes". *ACM Transactions on Graphics (TOG)* 31.4 (2012), 1–11 2.

[PCPM21] PUMAROLA, ALBERT, CORONA, ENRIC, PONS-MOLL, GERARD, and MORENO-NOGUER, FRANCESC. "D-nerf: Neural radiance fields for dynamic scenes". *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, 10318–10327 1.

[PFS*19] PARK, JEONG JOON, FLORENCE, PETER, STRAUB, JULIAN, et al. "Deepsdf: Learning continuous signed distance functions for shape representation". *CVPR*. 2019 1–3, 5, 6.

[PKGF21] PASCHALIDOU, DESPOINA, KATHAROPOULOS, ANGELOS, GEIGER, ANDREAS, and FIDLER, SANJA. "Neural parts: Learning expressive 3d shape abstractions with invertible neural networks". *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, 3204–3215 2–4.

[PLS*21] POSTELS, JANIS, LIU, MENGYA, SPEZIALETTI, RICCARDO, et al. "Go with the Flows: Mixtures of Normalizing Flows for Point Cloud Generation and Reconstruction". *International Conference on 3D Vision* (2021) 3.

[PSB*21] PARK, KEUNHONG, SINHA, UTKARSH, BARRON, JONATHAN T, et al. "Nerfies: Deformable neural radiance fields". *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, 5865–5874 2, 3, 5.

[PSH*21] PARK, KEUNHONG, SINHA, UTKARSH, HEDMAN, PETER, et al. "Hypernerf: A higher-dimensional representation for topologically varying neural radiance fields". *arXiv preprint arXiv:2106.13228* (2021) 2.

[QYSG17] QI, CHARLES R, YI, LI, SU, HAO, and GUIBAS, LEONIDAS J. "PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space". *arXiv preprint arXiv:1706.02413* (2017) 5.

[RM15] REZENDE, DANILO and MOHAMED, SHAKIR. "Variational inference with normalizing flows". *ICML*. 2015 3.

[RSO19] ROUFOSSE, JEAN-MICHEL, SHARMA, ABHISHEK, and OVSJANIKOV, MAKS. "Unsupervised deep learning for structured shape matching". *ICCV*. 2019 2.

[SA07] SORKINE, OLGA and ALEXA, MARC. "As-rigid-as-possible surface modeling". *Symposium on Geometry processing*. Vol. 4. 2007, 109–116 5.

[SBBG11] SOLOMON, JUSTIN, BEN-CHEN, MIRELA, BUTSCHER, ADRIAN, and GUIBAS, LEONIDAS. "As-killing-as-possible vector fields for planar deformation". *Computer Graphics Forum*. Vol. 30. 5. Wiley Online Library. 2011, 1543–1552 5.

[SCL*04] SORKINE, OLGA, COHEN-OR, DANIEL, LIPMAN, YARON, et al. "Laplacian surface editing". *Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing*. 2004, 175–184 5.

[SGST20] SEGU, MATTIA, GRINVALD, MARGARITA, SIEGWART, ROLAND, and TOMBARI, FEDERICO. "3DSNet: Unsupervised Shape-to-Shape 3D Style Transfer". *arXiv preprint arXiv:2011.13388* (2020) 1.

[SHN*19] SAITO, SHUNSUKE, HUANG, ZENG, NATSUME, RYOTA, et al. "PIFu: Pixel-Aligned Implicit Function for High-Resolution Clothed Human Digitization". *ICCV*. 2019 1, 2.

[SSTN18] SUWAJANAKORN, SUPASORN, SNAVELY, NOAH, TOMPSON, JONATHAN J, and NOROUZI, MOHAMMAD. "Discovery of latent 3d keypoints via end-to-end geometric reasoning". *NIPS*. 2018 2.

[STD*21] SUN, WEIWEI, TAGLIASACCHI, ANDREA, DENG, BOYANG, et al. "Canonical Capsules: Self-Supervised Capsules in Canonical Pose". *NIPS*. 2021 2.

[STD14] SALTI, SAMUELE, TOMBARI, FEDERICO, and DI STEFANO, LUIGI. "SHOT: Unique signatures of histograms for surface and texture description". *Computer Vision and Image Understanding* (2014) 2.

[UKS*21] UY, MIKAELA ANGELINA, KIM, VLADIMIR G, SUNG, MINHYUK, et al. "Joint learning of 3D shape retrieval and deformation". *CVPR*. 2021 2.

[WSH*16] WANG, TUANFENG Y, SU, HAO, HUANG, QIXING, et al. "Unsupervised texture transfer from images to model collections." *ACM Trans. Graph.* 35.6 (2016), 177–1 1.

[WSLG07] WEBER, OFIR, SORKINE, OLGA, LIPMAN, YARON, and GOTSMAN, CRAIG. "Context-aware skeletal shape deformation". *Computer Graphics Forum*. Vol. 26. 3. Wiley Online Library. 2007, 265–274 1.

[WWJ*07] WANG, SEN, WANG, YANG, JIN, MIAO, et al. "Conformal geometry and its applications on 3D shape matching, recognition, and stitching". *IEEE Transactions on Pattern Analysis and Machine Intelligence* 29.7 (2007), 1209–1220 5.

[YAK*20] YIFAN, WANG, AIGERMAN, NOAM, KIM, VLADIMIR G, et al. "Neural cages for detail-preserving 3d deformations". *CVPR*. 2020 2, 5.

[YHH*19] YANG, GUANDAO, HUANG, XUN, HAO, ZEKUN, et al. "Pointflow: 3d point cloud generation with continuous normalizing flows". *ICCV*. 2019 3.

[ZB15] ZUFFI, SILVIA and BLACK, MICHAEL J. "The stitched puppet: A graphical model of 3d human shape and pose". *CVPR*. 2015 2.

[ZHS*05] ZHOU, KUN, HUANG, JIN, SNYDER, JOHN, et al. "Large mesh deformation using the volumetric graph laplacian". *ACM SIGGRAPH 2005 Papers*. 2005, 496–503 5.

[ZYDL21] ZHENG, ZERONG, YU, TAO, DAI, QIONGHAI, and LIU, YEBIN. "Deep implicit templates for 3D shape representation". *CVPR*. 2021 2, 3, 5–9.