Unsupervised cycle-consistent deformation for shape matching

Thibault Groueix, Matthew Fisher, Vladimir G. Kim, Bryan C. Russell, Mathieu Aubry

Eurographics Symposium on Geometry Processing 2019
D. Bommes and H. Huang
(Guest Editors)

Volume 38 (2019), Number 5

Unsupervised cycle-consistent deformation for shape matching

Thibault Groueix[1], Matthew Fisher[2], Vladimir G. Kim[2], Bryan C. Russell[2], Mathieu Aubry[1]

[1]LIGM (UMR 8049), École des Ponts, UPE, [2]Adobe Research
http://imagine.enpc.fr/~groueixt/cycleconsistentdeformations/

Abstract

We propose a self-supervised approach to deep surface deformation. Given a pair of shapes, our algorithm directly predicts a parametric transformation from one shape to the other respecting correspondences. Our insight is to use cycle-consistency to define a notion of good correspondences in groups of objects and use it as a supervisory signal to train our network. Our method does not rely on a template, assume near isometric deformations or rely on point-correspondence supervision. We demonstrate the efficacy of our approach by using it to transfer segmentation across shapes. We show, on Shapenet, that our approach is competitive with comparable state-of-the-art methods when annotated training data is readily available, but outperforms them by a large margin in the few-shot segmentation scenario.

1. Introduction

Large collections of 3D models enable data-driven techniques for interactive geometry modeling, shape synthesis, image-based reconstruction, and shape completion [MWZ*14]. Many of these techniques require the collection to have additional surface annotations such as segmentation into functional [YKC*16] or geometric [LSD*18] parts. The notion of parts and their granularity can vary significantly across different tasks, so many novel applications require new types of annotations [MZC*19, YLZ*19, WZS*19]. Deep learning algorithms have recently achieved state-of-the-art in automatically predicting such surface annotations [QSMG16, QYSG17, WSL*18]. However, they typically require a significant number of training examples for every shape category, which limits their applicability, and bears significant start-up cost in introducing a new type of annotation. In this work, we propose a new deep learning approach which leverages large non-annotated object collections to perform few-shot segmentation.

We rely on the idea to use shape matching to transfer labels from similar examples. This approach has been shown to be robust in extreme "few-shot" learning scenarios [YKC*16] and can work robustly even in heterogeneous datasets as long as labeled models roughly span all the shape variations. The few-shots segmentation problem then amount to the fundamental problem of identifying correspondences between shapes. There is a vast amount of work on shape matching, which can be roughly separated in two trends: (i) classical optimization based approaches; (ii) recent approaches where correspondences are directly predicted by a neural network.

Traditional, optimization-based methods such as iterative closest point (ICP) algorithm, are fast and effective with good initial guesses and few degrees of freedom (e.g., a rigid motion) [RL01].

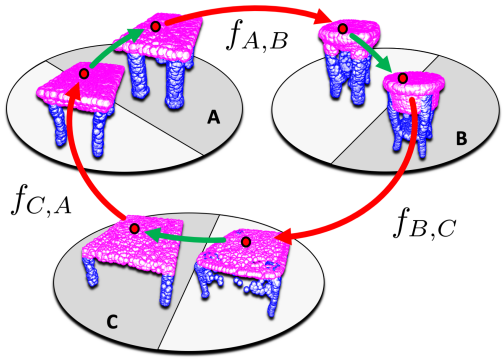


Figure 1: ***Shape deformation with cycle-consistency.*** *Our approach takes a pair (A,B) of pointclouds as input and predicts a deformation of A into B. During training, a cycle-consistent loss on a shape triplet (A, B, C) allows the method to learn semantically consistent deformations $f_{A,B}$, $f_{B,C}$, $f_{C,A}$ without any priors. Red arrows represent the learned shape deformation function and green arrows indicate the projection of the deformed shape onto the nearest point on the surface of the target shape.*

More flexible correspondence algorithms for dissimilar models usually require significantly more compute time to optimize for larger number of degrees of freedom [BR07, KLF11, CK15]. Since directly matching dissimilar shapes poses significant challenges, these methods often rely on joint analysis of the entire collection [KLM*12], leveraging cycle consistency priors during optimization [HG13, NBCW*11]. These joint correspondence estima-

tion methods tend to be very compute heavy and as new models are added to the collection, the entire optimization needs to be repeated. We thus turned to deep learning-based approaches.

Indeed, with the recent advances in neural networks for geometry analysis, learning-based methods have been proposed to address the matching problem. Of particular interest to us is the method of Groueix et al. [GFK*18a], which demonstrate that one can learn how to deform a human body template to the target point cloud, even without correspondence supervision. In their approach, the target point cloud is encoded into a latent descriptor space (via PointNet encoder [QSMG16]), and then the deformation network takes the target descriptor and a point on the template, and maps the point to new position so that it aligns to the target. This approach is efficient, since it only requires a forward pass through a network. It also has the benefit of holistic understanding of shape deformations, since the same neural network is trained for all models in the input collection. However, it has to be trained specifically for each template, limiting this method to analysis of geometrically and topologically similar shape collections, such as human bodies. If such a template is not available, one can pick a very generic shape (e.g., a sphere) and still obtain some correspondences via the intermediate domain [GFK*18b]. However, as we will show, the quality of the correspondences will degrade significantly as shapes deviate from that domain.

In this work we propose a novel neural network architecture that learns to match shapes directly, without relying on a pre-defined template, by learning to predict deformations that aligns points on the source shape to points on the target. Note that the transformation can be much more complex than a rigid transformation, and that the space of meaningful transformation is defined implicitly by the (unlabelled) training data. We encode both source and target shapes and then predict the deformed position for every point on the source conditioned on these two codes, unlike prior work that use a fixed template common to all the shapes. We show that the results obtained can be greatly improved if the network is trained not only with a reconstruction loss, which encourages it to deform the source shape into the target shape, but also using a cycle consistency loss. Indeed a deformation which respects correspondences should be consistent between pairs of shapes *i.e.*, the deformation from A to B should be the inverse of the deformation from B to A . More generally, in larger cycles of shapes $[A_1, ..., A_i, .., A_N]$, global consistency is achieved if the composition of the N successive mappings from A_i to A_{i+1} is identity. This new consistency loss used during training can be seen as playing a role similar to the global consistency objective used in optimization-based approaches. Finally, our network is trained in an self-supervised manner using only shape reconstruction and cycle consistency losses.

We demonstrate the effectiveness of our approach for shape matching by propagating segmentations in a few-shot learning setting on the ShapeNet part dataset [YKC*16]. We first show that in this extreme case with very few training examples, PointNet [QSMG16], a strongly supervised method, fails to generalize. Then, we propose several strategies for picking source shapes and propagate the signal from them, using our predicted correspondences. We demonstrate that even with a simple strategy, such as picking the source with smallest Chamfer distance, our method is better at transferring segmentations than other fast correspondence techniques such as ICP with rigid transformation and a prior learning-based method that aligns sphere and plane templates [GFK*18b].

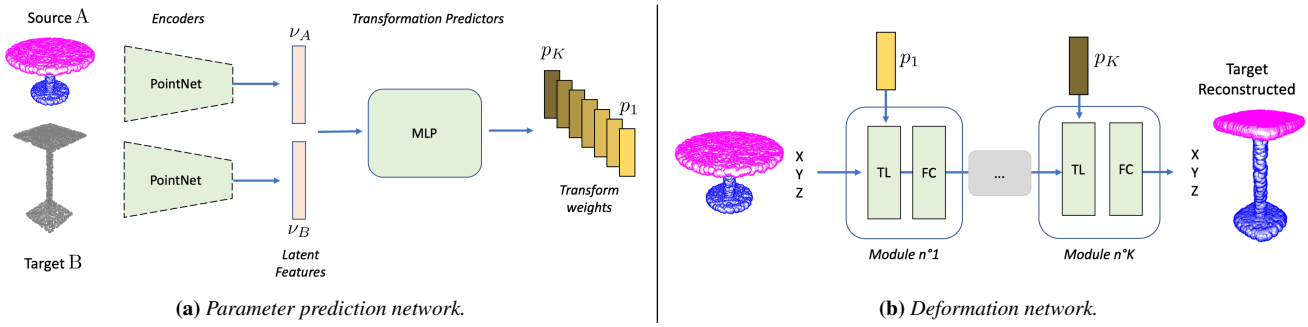
2. Related Work

Shape matching is a long-standing problem in shape analysis [vKZHCO11]. It is often done explicitly, by deforming a source shape to a target [RL01, BR07, LSP08, HAWG08, ZSCO*08], or implicitly, by mapping points [KLF11, CK15, OMMG10, BBK06] or functions [OBCS*12, RPWO18, EBC17] on one shape to another. The deformation-based methods typically aim to minimize the amount of distortion introduced by the deformation, and the mapping-based approaches often assume that shapes to be near-isometric. Both assumptions do not hold for very dissimilar shapes.

To address this challenge, some prior methods leverage additional context of the entire shape collection in a joint optimization [KLM*12, NBCW*11]. These techniques often use cycle-consistency as additional cue [HZG*12, HG13, ROA*13]. This, enables estimating correspondences even between dissimilar objects by mapping via intermediate shapes. While these traditional optimization techniques are very powerful, non-rigid matching involves optimizing for many degrees of freedom with complex non-convex objective functions, and takes minutes or hours. To make matters worse, joint analysis usually scales in a super-linear manner with number of models, and if a new shape is added to a collection, the entire optimization needs to be repeated.

Recently, learning-based correspondence techniques were used to address these limitations. They are fast, typically only requiring a forward pass through a neural network, and they enable joint analysis of a collection of shapes, since multiple shapes are typically used during training. Descriptor-based methods embed each shape point into some high-dimensional space, where corresponding points are embedded nearby [HKC*18, BMRB16, WHC*16]. In most cases, however, a more holistic mapping for the entire shape is often preferred, since it is more capable of preserving the intrinsic shape structure. Litany et al. [LRR*17] use a deep neural network to predict a soft inter-surface mapping a common representation used in functional map framework. Groueix et al. [GFK*18a] propose to train a network that predicts a deformation for each point on a template. A similar method that uses planes or spheres can be used in case such a template is not available [GFK*18b]. These techniques struggle with diverse shape collections when matched shapes have very different topology and geometry. Instead, we propose a method that takes both source and target shape as input and infers the mapping. We also propose a novel regularization term favoring cycle-consistency when mapping across multiple shapes in the collection. A similar cycle-consistency loss for training deep networks to predict correspondences between images of different instances of objects from the same category has recently been used in [ZKA*16]. In this work, views rendered from different viewpoints from a 3D model were used to avoid the trivial identity flow solution, but no correspondence between 3D shapes was predicted.

We demonstrate the value of our method for few-shot segmentation transfer. While many techniques have been developed



(a) *Parameter prediction network.* **(b)** *Deformation network.*

Figure 2: ***Shape Deformation approach.*** *Our methods take as input a pair (source A, target B) of shapes and aims at predicting the deformation of A in B. In* ***(a)****, A and B are encoded with Pointnets [QSMG16] into a latent feature vector, from which an MLP predicts transformation parameters, used in* ***(b)*** *to deform A into B, by stacking Transformation Layers* ***(TL)*** *and Fully-Connected Layers* ***(FC)****.*

for strongly supervised mesh segmentation [QSMG16, QYSG17, WSL*18, LSD*18, KAMC17, KHS10], they typically rely on many training examples and fail in a few-shot scenarios (see Table 1). In these cases, some framework propose to rely on propagating annotations from most similar annotated shapes via global or local shape matching [YKC*16]. In fact, it is common for correspondence techniques to be evaluated and used for transferring various signals between shapes [OBCS*12, KLF11, ACBCO17, CFG*15].

3. Learning asymmetric cycle-consistent shape matching

We address the surface matching problem by training a model that takes as inputs a source shape, a target shape, and a point on the source shape and generates the corresponding point on the target shape. As pointed out in Groueix et al. [GFK*18a], a learnable model allows for efficient surface matching, which is in contrast to approaches requiring optimization over a collection of pairwise shape matches [NBCW*11].

We assume that shapes are represented as point sets sampled from the shapes' surface. Given point sets A and B, our goal is to learn a mapping function $f_{A,B}$ that takes a 3D point $\mathbf{p} \in A$ to its corresponding point $\mathbf{q} \in B$. If f is a function on points and A a set of points, we denote by $f(A)$ the set $\{f(\mathbf{p}), \forall \mathbf{p} \in A\}$.

First, building on work on unsupervised template-based shape correspondence [GFK*18a] we use a Chamfer loss to minimize the distance between deformed source $f_{A,B}(A)$ and the target B. Unlike prior work, however, we do not assume that all of our shapes are derived from the same template and directly predict template-free correspondences between pairs of shapes.

Second, we seek to leverage the success of cycle consistency, which has been used in shape collection optimization [NBCW*11] and more recently in self-supervised learning [ZPIE17], during training of our learnable mapping function. Formally, for N shapes $X_1, \ldots, X_N$ that are assumed to be put into correspondence, we enforce that the learnable mapping function $f_{A,B}$ satisfies,

$$\forall \mathbf{p} \in X_1,\ f_{X_1,X_2} \circ \cdots \circ f_{X_{N-1},X_N} \circ f_{X_N,X_1}(\mathbf{p}) = \mathbf{p}. \qquad (1)$$

We use cycle-consistency training losses for cycles of lengths two and three as it implies consistency for cycles of any length [NBCW*11]. We visualize our cycle-consistency loss in Figure 1.

4. Approach

We describe our learnable mapping function $f_{A,B}$, implemented as a two-stage neural network, in Section 4.1, our training losses in Section 4.2, and application to segmentation in Section 4.3.

4.1. Architecture

The architecture of our shape transformation model from a source shape A to a target shape B is visualized in Figure 2 and can be separated into two parts: (a) a parameter prediction network which outputs transformation parameters given the two shapes (Figure 2a); (b) a deformation network that transforms the first shape into the second one using the predicted parameters (Figure 2b). We now describe these two components.

To predict transformation parameters, A and B are first passed into two independent PointNet networks [QSMG16] leading to feature encodings v_A and v_B of size 512. The resulting concatenated descriptor $v_{AB} = [v_A, v_B]$ contains information about the pair (A, B). A multilayer perceptron (MLP) then predicts transformation parameters vectors $p_1, \cdots, p_K$ from this concatenated feature.

The deformation network (Figure 2b) takes a surface point in $\mathbb{R}^3$ and outputs the associated deformed point. The network is composed of K modules each with the same architecture. Let's call x_{k-1} the input of module k and x_k its output. The operation computed by this module is:

$$x_k = A_k\left(W_k\left(s_k \cdot x_{k-1} + b_k\right)\right), \qquad (2)$$

where W_k is the matrix of parameters of a fully-connected layer in $\mathbb{R}^{64x64}$, "$\cdot$" refers to the Hadamard (term to term) product, A_k is the activation function for module k and $[s_k, b_k] = p_k$ are the transformation parameters, both in in $\mathbb{R}^{64}$, corresponding to a scale and a bias in each dimension. Note that this is similar to the architecture of the T-net modules in [QSMG16, JSZ*15], but using fewer predicted parameters. Also note that equation 2 is differentiable, which enables the two sub-networks to be trained jointly in an end-to-end fashion. In all of our experiments we used $K = 7$ modules, 64 dimensions for each intermediary feature and ReLU activations for all but the last layer, for which we used a hyperbolic tangent. We train for 500 epochs with Adam [KB14] starting with a learning rate of 0.01 divided by 10 after 400 epochs.

4.2. Training Losses

We train our deformation by minimizing the ~~weighted~~ sum over several components: a loss enforcing cycle consistency $L_{\mathbf{Cy}}$, Chamfer distance loss $L_{\mathbf{Ch}}$, and a self reconstruction loss $L_{\mathbf{SR}}$:

$$L_{\mathbf{total}} = L_{\mathbf{SR}} + L_{\mathbf{Ch}} + L_{\mathbf{Cy}}$$

We only use the self-reconstruction loss to stabilize the beginning of the training and disable it after 30 epochs to focus on cycle consistency and reconstruction losses. We train all parameters in our network by sampling triplets (A,B,C) of shapes which are needed by our 3-cycle consistency and enforcing all other losses on all the associated deformations. We first explain how we sampled these triplets, then detail the different terms of our loss.

4.2.1. Training shape sampling

For our cycle-consistency loss, we require a valid mapping across shape triplet (A, B, C). As different shape categories may have different topologies, we train category-specific networks. Furthermore, as there may be topological changes within a single category, for shape A, we randomly sample shapes B and C from the K nearest neighbors of A under chamfer distance. We take $K = 20$ and demonstrate in the ablation study the superiority of this approach over random sampling of shape triplets.

We apply data augmentation ψ on each sampled shape in this order : a random rotation around the Z axis of a random angle between -40° and 40°, an anisotropic scaling of random scale between 0.75 and 1.25, a bounding box normalization, and a small random translation below 0.03.

4.2.2. Cycle-consistency loss

The cycle consistency loss is based on the intuition that a point deformed through any cycle of deformations should be mapped back to itself. One way to enforce consistency would be to compute composite functions, for two shapes X and Y minimizing $\|\mathbf{p} - f_{Y,X} \circ f_{X,Y}(\mathbf{p})\|$ for all $\mathbf{p}$ in X. However $f_{X,Y}(\mathbf{p})$ is typically not an element of Y, and computing $f_{Y,X} \circ f_{X,Y}(\mathbf{p})$ would thus require computing the deformations $f_{Y,X}$ of other points than the points of Y. To avoid this, we consider instead projections of the deformed shapes to the target shapes. More precisely, we define the shape projection operator π

$$\pi_X(\mathbf{p}) = \text{argmin}_{\mathbf{q}\in X} \|\mathbf{p} - \mathbf{q}\| \tag{3}$$

and enforce 2-cycle consistency between X and Y by minimizing

$$Cy_2(X,Y) = \frac{1}{|X|} \sum_{\mathbf{p}\in X} \left|\mathbf{p} - f_{Y,X} \circ \pi_Y \circ f_{X,Y}(\mathbf{p})\right|_2 \tag{4}$$

and cycle consistency for the (X,Y,Z) cycle by minimizing

$$Cy_3(X,Y,Z) = \frac{1}{|X|} \sum_{\mathbf{p}\in X} \left|\mathbf{p} - f_{Z,X} \circ \pi_Z \circ f_{Y,Z} \circ \pi_Y \circ f_{X,Y}(\mathbf{p})\right|_2 \tag{5}$$

Our full cycle-consistency loss $L_{\mathbf{Cy}}$ is simply defined by summing over possible all possible two and three cycles using a sampled triplet (A, B, C).

$$L_{\mathbf{Cy}} = \sum_{X,Y,Z\in\{A,B,C\} s.t. \{X,Y,Z\}=\{A,B,C\}} Cy_2(X,Y) + Cy_3(X,Y,Z) \tag{6}$$

Enforcing 2- and 3-cycle consistency implies consistency for any cycle [NBCW*11].

4.2.3. Reconstruction loss

As discussed in section 3, we want to enforce that every point in the target shape is well reconstructed, but not necessarily that any point in the source shape is mapped to the target shape, in case some part appear in the source and not the target. We thus used asymmetric Chamfer distance to quantify how well the network has generated the target shape. More precisely, given a pair of shapes (X,Y), the asymmetric chamfer $Ch(X,Y)$ computes the average distance between a point $\mathbf{q} \in Y$ and its nearest neighbor in X.

$$Ch(X,Y) = \frac{1}{|X|} \sum_{\mathbf{q}\in Y} \min_{\mathbf{p}\in X} \|p - q\|_2 . \tag{7}$$

Given a training triplet (A,B,C), we define the reconstruction loss by summing the asymmetric chamfer loss on all 6 possible (source, target) couples.

$$L_{\mathbf{Ch}} = \sum_{X,Y\in\{(A,B),(A,C),(B,C)\}} Ch(f_{X,Y}(X),Y) + Ch(f_{Y,X}(Y),X) \tag{8}$$

If segmentation is available for the training shapes, we can compute the distance in equation 7 on each segment independently, which would add supervision on the correspondences. We of course do not use such labels for our few-shot learning experiments, but show in Table 2 it can be used if available to slightly boost our results.

4.2.4. Self-reconstruction loss

We can fully supervise the deformation by manually deforming a shape with a known transformation. We found such a supervision was helpful to stabilize and speed up the beginning of our training. Concretely, we sampled deformations ψ similar to what we did for data augmentation (described above in 4.2.1) by composing (1) a rotation, (2) an anisotropic scaling, and (3) a rescaling to a centered bounding box. Given a transformation ψ, we compute the average distance between the two images of a point $\mathbf{p} \in A$ under ψ and the predicted mapping function $f_{A,\psi(A)}$.

$$SR(A,\psi) = \frac{1}{|A|} \sum_{\mathbf{p}\in A} \left\| f_{A,\psi(A)}(p) - \psi(p) \right\|_2 \tag{9}$$

Our corresponding self-reconstruction loss $L_{\mathbf{SR}}$ is the sum of this loss for each of the three point clouds in the triplet (A, B, C) with different random transformations.

$$L_{\mathbf{SR}} = SR(A,\psi) + SR(B,\psi') + SR(C,\psi'') \tag{10}$$

4.3. Application to segmentation

Learning a deformation between two shapes provides an intuitive method to transfer label information, such as a part segmentation, from a labeled shape to an unlabeled one. In this formulation, we assume we are given a (small) number of labeled shapes, and seek to label each point on an unlabeled test shape. This requires us to decide which of the labeled shapes we should use as the source to propagate labels to the target shapes.

Selection Criteria. Given a target T, We manually define 4 possible source selection criteria:

- **Nearest Neighbor**: The source shape S that minimizes the Chamfer distance between S and T is selected.
- **Deformation Distance**: The source shape S that minimizes the Chamfer distance between $f_{S,T}(S)$ and T is selected.
- **Cosine Distance**: The source shape S that minimizes the cosine distance distance between the PointNet encodings v_S and v_T is selected.
- **Cycle Consistency**: The source shape S that minimizes 2-cycle loss for the pair (S,T) is selected.

Having selected a pair (S,T), labels can be transferred directly with our approach.

Voting strategy. Instead of selecting a single source shape to get labels from, combining several voting shapes allows for better segmentation. We select the K-best sources, and make each source shape vote with equal weight for the label of each target point. We evaluate the benefits of this voting approach in Section 5.2.2.

5. Results

In this section, we show qualitative and quantitative results on the tasks of few-shot and supervised semantic segmentation and compare against several baselines.

Data and evaluation criteria. We evaluated our approach on the standard ShapeNet part dataset [YKC*16]. We restricted ourselves to the 5 most populated categories, namely Airplane, Car, Chair, Lamp, and Table. Point clouds sampled on mesh objects are densely labeled for segmentation with one to five parts. We follow Qi et al. [QSMG16] and report the mean intersection over union (mIoU) between the predicted and ground truth segmentation across instances in a category.

Baselines. We compare our unsupervised approach against supervised and unsupervised approaches. We used PointNet as a supervised baseline. Our unsupervised baselines include a learned approach derived from Atlasnet [GFK*18b] and variants of iterative closest points (ICP) [Zha94, BM92]. AtlasNet is a template-based reconstruction method that predicts a transformation of the template matching the target shape. The learned deformations have been previously observed to be semantically consistent [GFK*18a]. To transfer segmentation labels from a source to a target, we project the source labels on the source reconstruction through nearest neighbors, then on the template through dense correspondence between the template and the source reconstruction. Similarly, we transfer labels on the template to the target by dense correspondence and nearest neighbors. AtlasNet is trained on the same train/test splits as our approach. We consider two settings of AtlasNet – with 10 patches or 1 sphere as the template. Additionally, we use two standard shape alignment baselines. First, labels can be transferred from source to target through nearest neighbor matching, which we call the *Identity* baseline. An immediate refinement over this baseline is to apply ICP to align the source to the target, and then use nearest neighbors. We call the latter the *ICP* baseline.

5.1. Qualitative Results

Correspondences. In figure 5 we visualize in more detail the correspondences obtained with our approach. We visualize how each point on the source shape is deformed and transferred to the target shape using a colored checkerboard. For each example, we show a successful deformation (top) and a failure case (bottom). Note how the checkerboard appears nicely deformed in the case of successful deformation, and still appears consistent on some parts in the failure cases.

Cycle-consistency. In figure 6 we compare the mappings learned by our approach with and without cycle-consistency loss. The Chamfer Distance is a point based loss with no control over the amount of distorsion. Notice in this case that the deformed source has large triangles. It indicates that the mapping learned by a Chamfer loss alone is not smooth, and can't be used in label tranfer. On the other hand, the cycle-consistency loss leads to a smooth and high quality mapping.

Segmentation transfer. When looking at the results, a first surprising observation is the high quality of the identity baseline (this is quantitatively confirmed in Table 2). Indeed, the different criteria tend to select shapes that are really close to the target. To focus on interesting examples, we selected in Figure 3 the pairs that maximize the performance improvement provided by our method compared to the identity baseline using the cycle-consistency-selection criterion. The richness of the learned deformations allows our method to find meaningful correspondences in cases where the training example is far from the target shape and the identity baseline does not work. Note that the deformations are often far from isometric. Thus, methods that rely on regularization toward identity, a popular approach to regularize learned deformations [GFK*18a, KTEM18, WZL*18], would likely fail.

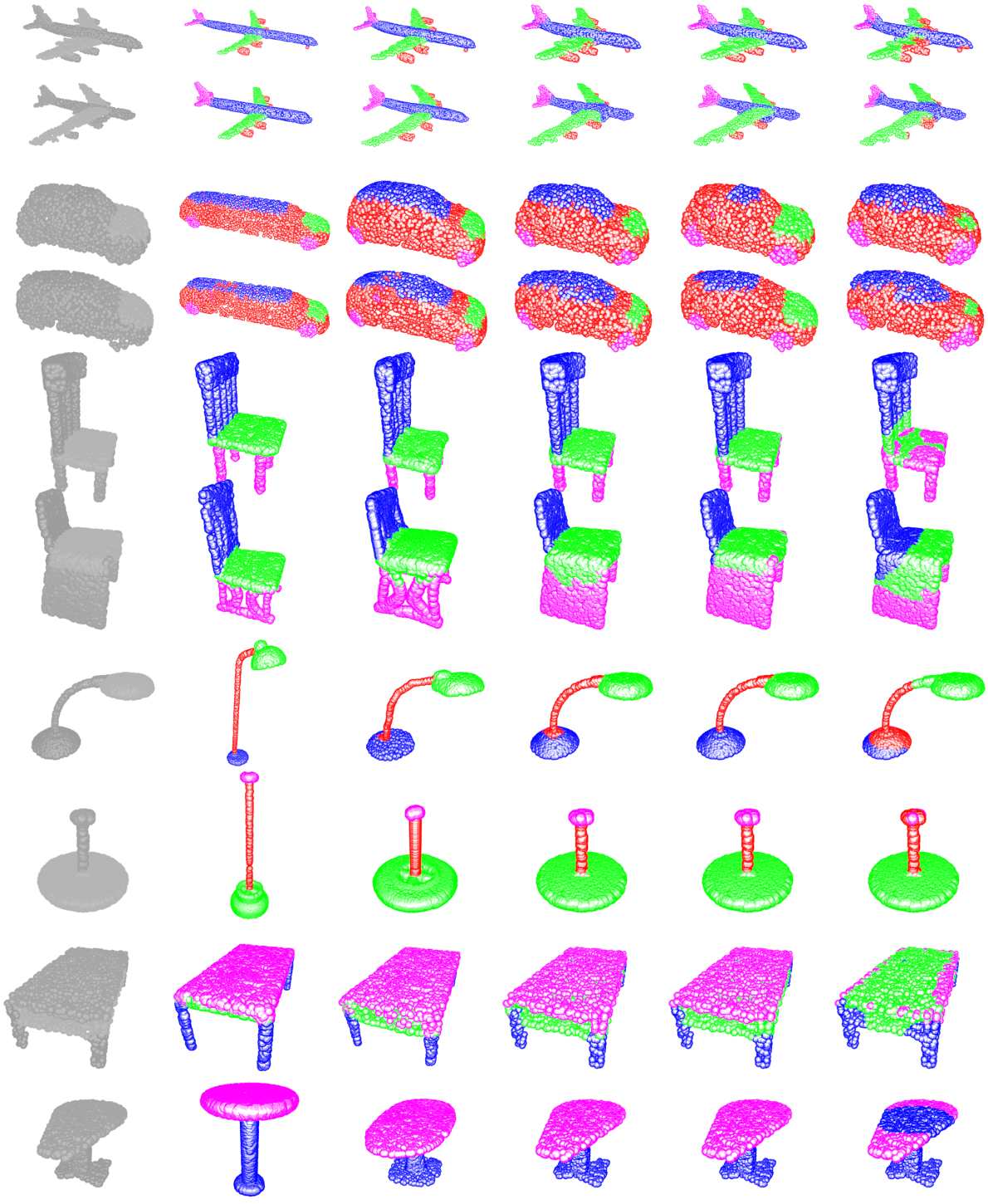
Failure cases. Figure 4 shows failures of our method. We show for each category the pair (S,T) which minimizes our segmentation transfer performance. It is clear that the corresponding shapes are rare and specific object instances. We observe two main sources of errors. First, in some cases where we correctly deform S in T, the ground truth labeling was inconsistent, leading to large errors. For example, notice how the source airplane has a single label. Second, S and T are sometimes too distant topologically so that a high-fidelity reconstruction of T is impossible by deforming S. For example, notice how the pole of the lamp has been erroneously inflated to match the target shape.

5.2. Quantitative Results

5.2.1. Few-shot Segmentation

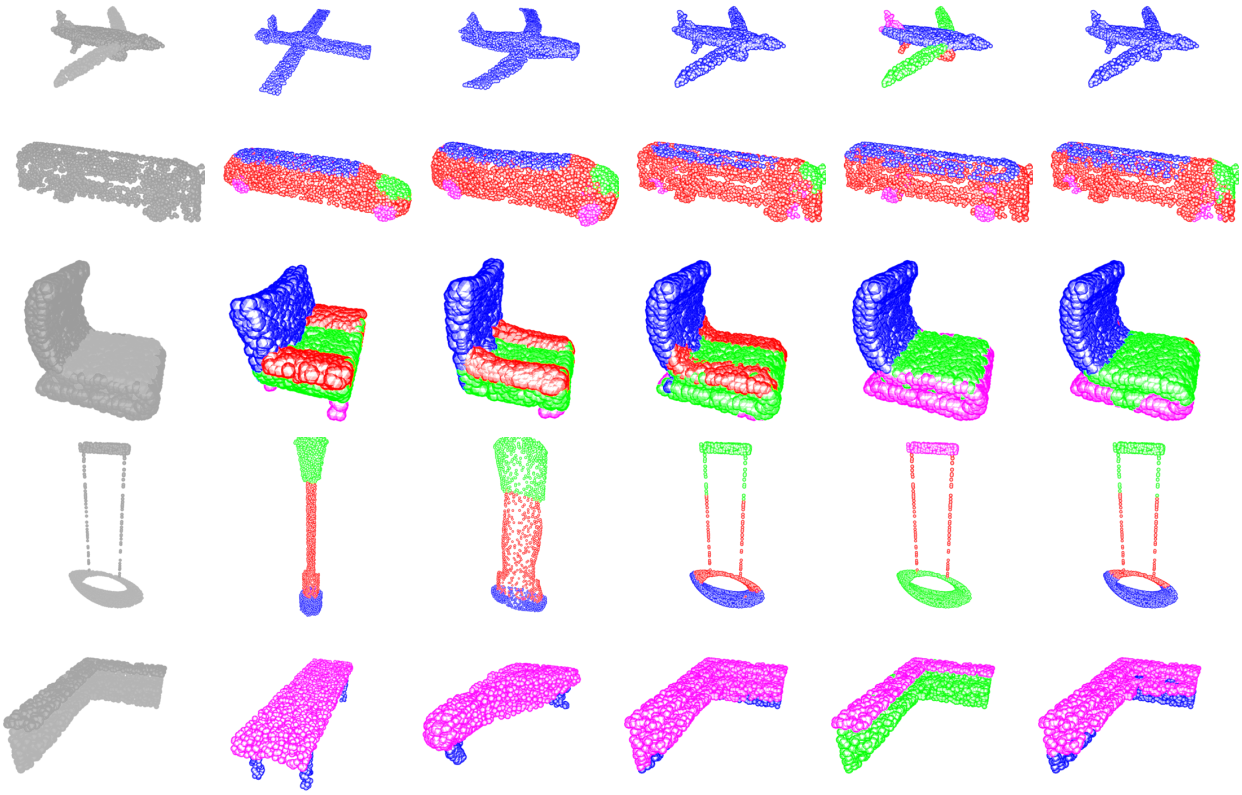
In this section, we evaluate our approach on the task of transferring semantic labels from a small set of segmented shapes to unlabeled data.

We report quantitative results for few-shot semantic segmentation on point clouds in Table 1. Note that the learning-based methods are all trained separately for each category. Since the results depend on the sampled shapes used in the training set, we report the



(a) *Input shape (target)* **(b)** *Retrieved shape (source)* **(c)** *Deformed retrieved shape* **(d)** *Transferred labels to input* **(e)** *Ground truth labels* **(f)** *Identity baseline segmentation*

Figure 3: ***Qualitative results.*** *For each input shape (a), we select the top nearest neighbor from 400 training examples with part segmentations using the cycle-consistency criterion (b). We apply our approach to deform the retrieved shape to align with the input shape (c). Given the deformed shape, we transfer the labels onto the input shape (d). For each category, we show the top results that maximize IoU with the ground truth (e). For comparison, we show the Identity baseline in (f). Notice how our method successfully transfers labels and improves over the baseline.*



(a) *Input shape (target)* **(b)** *Retrieved shape (source)* **(c)** *Deformed retrieved shape* **(d)** *Transferred labels to input* **(e)** *Ground truth labels* **(f)** *Identity baseline segmentation*

Figure 4: ***Failures.*** *Example failures include when a retrieved shape has inconsistent annotation (rows 1,2,5) and poor deformation due to different topology (rows 3,4).*

10 shots	Selection Criterion	Airplane	Car	Chair	Lamp	Table
(a) Pointnet	-	14.0 ± 8.0	11.7 ± 10.4	21.1 ± 13.1	26.0 ± 13.2	43.5 ± 15.5
(b) Atlasnet Patch	Nearest Neighbors	62.6 ± 2.4	52.3 ± 9.1	72.1 ± 1.2	62.8 ± 2.2	61.6 ± 3.7
(c) Atlasnet Sphere	Nearest Neighbors	62.2 ± 2.2	52.9 ± 9.1	70.2 ± 1.2	59.3 ± 1.8	60.0 ± 5.1
(d) ICP	Nearest Neighbors	65.5 ± 3.1	61.3 ± 1.1	75.8 ± 1.2	64.8 ± 5.0	64.9 ± 3.9
(e) Ours	Nearest Neighbors	$\mathbf{67.1 \pm 2.9}$	$\mathbf{61.4 \pm 1.1}$	$\mathbf{78.9 \pm 1.1}$	$\mathbf{65.8 \pm 5.2}$	$\mathbf{66.1 \pm 4.5}$
(f) Ours	Cycle Consistency	67.9 ± 3.0	60.2 ± 3.4	81.8 ± 0.7	69.1 ± 5.4	68.8 ± 4.0
(g) Ours	Oracle	74.9 ± 3.0	68.6 ± 2.4	86.4 ± 0.6	80.3 ± 3.8	77.8 ± 2.1

Table 1: ***Few-shot segmentation:****. We compare* ***(e, f)*** *our approach with* ***(a)*** *Pointnet [QSMG16], a supervised method, trained per category,* ***(b, c)*** *two unsupervised baselines based on Atlasnet [GFK*18b] and* ***(e)*** *ICP. We pre-train all* ***(b, c , e, f)*** *unsupervised approaches on the train splits (without labels). Given a target shape T and 10 segmented train samples, we select T's nearest neighbors S. In Atlasnet* ***(b, c)****, labels are propagated through the template. In our approach* ***(e, f, g)****, labels are propagated from* T_S *to T. We report in* ***(g)*** *the best performance of our method over the 10 shots. The mean IoU is reported. Results are averaged over 10 runs.*

average and standard deviation over ten randomly sampled training sets. We use the Nearest Neighbors criterion to pair sources and targets and compare our approach against all baselines **(b, c, d, e)**. Notice that our approach out-performs all baselines on all categories. Interestingly, the AtlasNet baseline is not on par with ICP, hinting at the difficulty of predicting two consistent deformations of the template.

We find that the Cycle Consistency criterion **(f)** is a stronger selection criterion than Nearest Neighbors and boosts the results simply by selecting a better (Source, Target) pair. We also report an

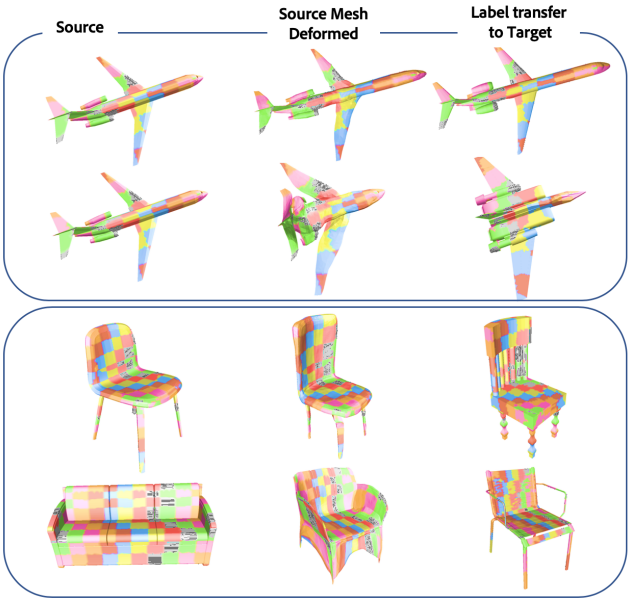


Figure 5: ***Mapping function quality.*** *We apply a checkerboard colorization scheme on the source (**left**), and use our approach to deform (**middle**) the source shape to the target shape (**right**). The labels are transferred from the deformed shape to the target shape through nearest neighbors. For each category, we show a example of good reconstruction (**top**) and poor reconstruction (**bottom**). Notice the high quality of the mapping in both cases.*

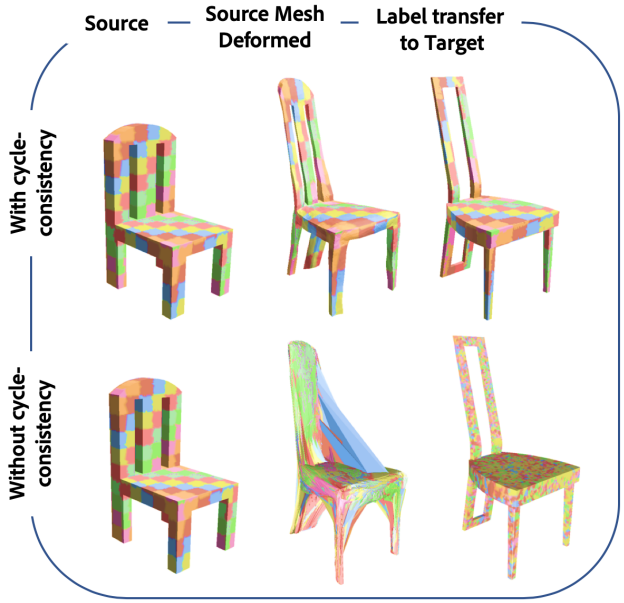


Figure 6: ***Cycle-consistency performance.*** *We apply a checkerboard colorization scheme on the source (**left**), and use our approach with cycle-constistency (**top**) and without (**bottom**) to deform (**middle**) the source shape to the target shape (**right**). The labels are transferred from the deformed shape to the target shape through nearest neighbors.*

oracle source-shape selection with our approach where the source shape maximising IoU with the target is selected, which corresponds to the scenario where an optimal source shape is selected. Notice the large improvement of the oracle, showing the quality of our deformations and the potential of our method.

5.2.2. Supervised segmentation

Our method is not designed to be competitive when many training samples are available. Indeed, it solves for the deformation against each of the provided segmented shapes, which for large numbers of examples can be computationally expensive compared to feed-forward segmentation predictions like PointNet [QSMG16]. One forward pass through our network deforms a source shape in a target shape in 7 milliseconds (ms), with a 7ms standard deviation (std). ICP takes 28 ms with a 17 std[†]. Here, however, we study the performance of our method in this case, using the segmentation of the many training shapes as supervision during training and making the ten best shapes vote during testing. We report results of our unsupervised method. In addition, we consider adding supervision to our approach by computing Chamfer distances over points with the same segmentation label. The corresponding results are reported in Table 2

[†] We use Open3D [ZPK18] to compute ICP ran on Intel i7-6900K - 3.2 GHz and run our method on an NVIDIA TITAN X.

Table 2 shows that, when using all the annotations, nearest neighbors is again a surprisingly good baseline, only slightly below performance of PointNet. Despite the good performance of the identity baseline, our method outperforms it in all categories and performs on par with PointNet. Note that the encoders of our approach incorporate two PointNet architectures, which makes this result intuitive.

Table 2 also highlights the importance of the criterion selection. Notice the significant boost in each category gained by carefully choosing the selection criterion over the Nearest Neighbors criterion. The exciting performance of the oracle, way over the PointNet baseline, is another incentive at carefully designing selection criteria.

Finally, notice that our unsupervised trained model is on par with our supervised one. The boost gained by supervised training is marginal except in the car category. It confirms that our cycle-consistent loss is efficient to enforce meaningful part correspondence.

5.2.3. Selection criteria and voting strategy

Figure 7 shows a quantitative comparison on all criteria, on all category for the identity baseline and our approach using a voting strategy with different number of shapes. The oracle, and PointNet performances are also reported. The Deformation Distance criterion outperforms all other criteria but remains far from the oracle. The oracle performs better than the PointNet baseline across all categories. As a sanity check, we observe that our method outperforms

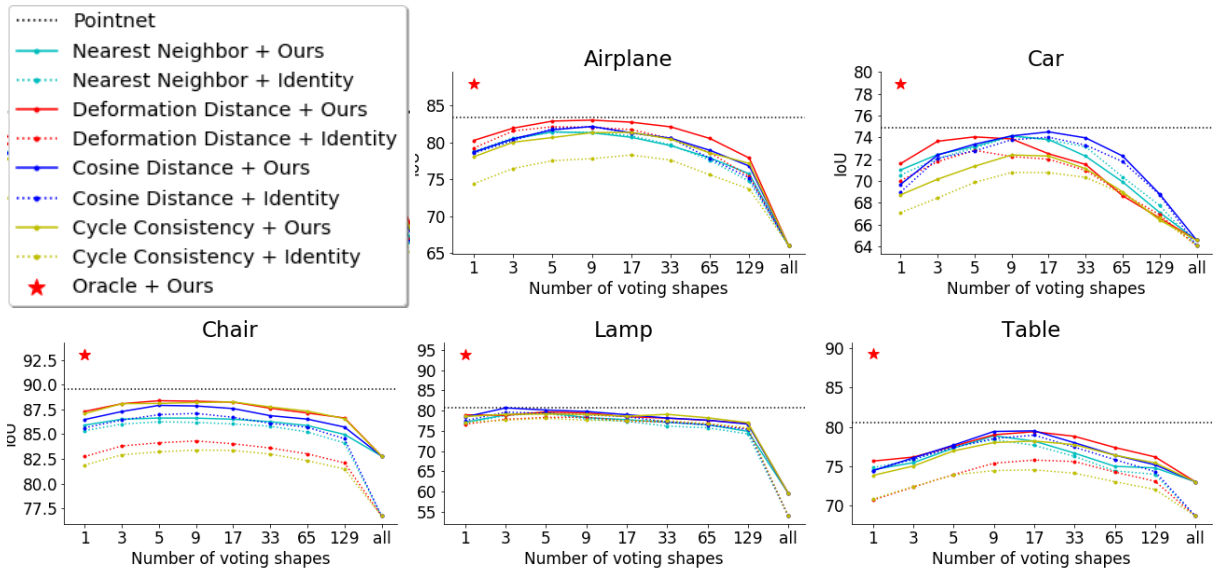


Figure 7: ***Criteria and voting strategies.*** *Study of the number of voting shapes for the transfer of segmentation label, across 4 criteria (see 4.3) - Nearest Neighbors, Deformation Distance, Cosine Distance and Cycle Consistency -, and across 5 Shapenet categories. Our transformation method (solid lines) almost always enhance the identity baseline (dashed lines). We report a supervised baseline, Pointnet [QSMG16] and the oracle source which maximizes IoU for our method. Notice how the oracle significantly outperforms the Pointnet baseline, making the search of a strong selection criterion a good direction. Our models are category specific and trained without segmentation supervision. All of the train set is searched to maximize each criterion.*

	Selection	Airplane	Car	Chair	Lamp	Table
(a) Pointnet	-	83.4	74.9	**89.6**	**80.8**	**80.6**
(b) Identity	NN	81.3	74.0	86.1	78.4	78.9
(c) Ours unsup	NN	81.5	73.9	86.6	78.8	79.2
(d) Ours unsup	Best criterion	83.4	74.6	88.4	79.8	79.7
(e) Ours unsup	Oracle	87.9	78.9	93.0	93.9	89.3
(f) Ours sup	NN	81.2	75.9	86.9	78.4	79.0
(g) Ours sup	Best criterion	**83.5**	**76.4**	88.8	79.3	79.9
(h) Ours sup	Oracle	88.0	80.2	93.1	93.4	89.4

Table 2: ***Supervised segmentation:****. We compare our approach with* ***(a)*** *Pointnet [QSMG16] and* ***(b)*** *Identity baseline. Our approach can be trained with part supervision* ***(f, g, h)*** *or without* ***(c, d, e)****. Given a target shape T and the segmented train set, we compare 3 types of source shapes :* ***(b, c, f)*** *T's Nearest Neighbors;* ***(d, g)*** *the best shape among all criteria see 4.3; and* ***(e, h)*** *the a posteriori best shape over all train sample. A voting strategy is used on the top 10 shapes in* ***(b, c, d, f, g)****. The mean IoU is reported.*

the identity baseline in all settings, showing that it helps to apply our method to transfer labels from S to T.

Figure 7 also confirms that using several source shapes is beneficial when many annotated examples are available. In the limit, when all source shapes vote and selection criterion does not matter anymore, an average labelling is predicted with poor performances, which again outlines the importance of source selection. Using nine source shapes performs the best across most criteria and categories when all the training annotations can be used.

5.3. Ablation Study

In this section we conduct an ablation study to empirically validate our approach. Table 3 shows performances without the cycle loss, without Chamfer loss, and without any specific triplet sampling strategy during training, simply selecting random shapes.

Table 3 shows that the cycle consistency loss is critical to the success of our method (relative drop of 23% in IoU). Training without Chamfer distance as a reconstruction loss performs slightly better than the identity baseline and 3% below our approach. This highlight the fact that the cycle consistency loss also acts as a reconstruction loss. Finally, our triplet sampling strategy during training provides a small boost.

Car/100 shots	Nearest Neighbor	Oracle
(a) Identity	67.60	73.59
(b) Ours	**68.19**	**75.87**
(c) Ours w/o cycle loss	52.78	59.63
(d) Ours w/o chamfer	66.21	74.31
(e) Ours w/o knn restriction	67.70	75.23

Table 3: ***Ablation Study:****. Given a target shape T and 100 segmented train samples, we select T's nearest neighbors S (1st column), and the oracle source shape which maximizes performances for our approach . (2nd column). We compare* ***(a)*** *the identity baseline, with* ***(b)*** *our approach, trained without label supervision, and* ***(c, d, e)*** *its ablations. The mean IoU is reported. Results are computed on the Car category.*

5.4. Hyperparameter Study

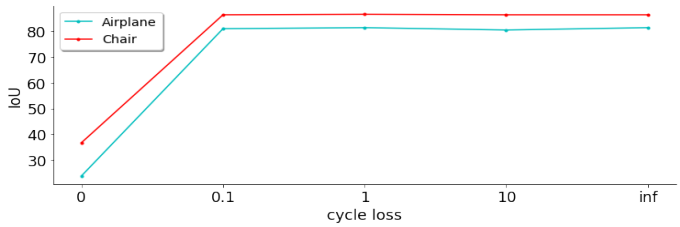


Figure 8: ***Hyperparameter study.*** *Study of the influence of the cycle consistency loss from not having it (abscissa point "0") to having only the cycle loss (abscissa point "inf"). For each target shape, we use the Nearest Neighbors (see 4.3) criterion to select sources from the full training set. A voting strategy is used on the top 10 source shapes. The mean IoU is reported*

Figure 8 demonstrates once more that the cycle-consistency loss is the pivotal insight of our method. It also outlines the stability of the results for different weightings of our losses. Note how performances are maintained even in the extreme case with only the cycle-consistency loss. Indeed, the identity function is not a trivial minimum of the cycle consistency loss because of the projection step.

6. Conclusion

We have presented a method for learning a parametric transformation between two surfaces that leverages cycle-consistency as a supervisory signal to predict meaningful correspondences. Our method does not require an object template, can operate without any inter-shape correspondences supervision, and does not assume the deformation is nearly isometric. We demonstrate that our method is able to transfer segmentation labels from a very small number of labeled examples significantly better than state-of-the-art methods, and match the segmentation performance when a larger training dataset is provided.

We believe that the large gap between our performance and the "oracle shape" which provides maximal accuracy shows that using learned deformations to transfer labels, investigating ways to better understand what source models should be selected and new ways to aggregate information across multiple sources is a very promising research direction.

References

[ACBCO17] AZENCOT O., CORMAN E., BEN-CHEN M., OVSJANIKOV M.: Consistent functional cross field design for mesh quadrangulation. *ACM Trans. Graph. 36*, 4 (July 2017), 92:1–92:13. 3

[BBK06] BRONSTEIN A. M., BRONSTEIN M. M., KIMMEL R.: Generalized multidimensional scaling: A framework for isometry-invariant partial surface matching. *Proceedings of the National Academy of Sciences 103*, 5 (2006), 1168–1172. 2

[BM92] BESL P. J., MCKAY N. D.: A method for registration of 3-d shapes. *IEEE Trans. Pattern Anal. Mach. Intell. 14*, 2 (Feb. 1992), 239–256. 5

[BMRB16] BOSCAINI D., MASCI J., RODOLÀ E., BRONSTEIN M. M.: Learning shape correspondence with anisotropic convolutional neural networks. *CoRR abs/1605.06437* (2016). 2

[BR07] BROWN B., RUSINKIEWICZ S.: Global non-rigid alignment of 3-D scans. *ACM Transactions on Graphics (Proc. SIGGRAPH) 26*, 3 (Aug. 2007). 1, 2

[CFG*15] CHANG A. X., FUNKHOUSER T. A., GUIBAS L. J., HANRAHAN P., HUANG Q., LI Z., SAVARESE S., SAVVA M., SONG S., SU H., XIAO J., YI L., YU F.: Shapenet: An information-rich 3d model repository. *CoRR abs/1512.03012* (2015). 3

[CK15] CHEN Q., KOLTUN V.: Robust nonrigid registration by convex optimization. *ICCV* (2015). 1, 2

[EBC17] EZUZ D., BEN-CHEN M.: Deblurring and denoising of maps between shapes. *Comput. Graph. Forum 36*, 5 (Aug. 2017), 165–174. 2

[GFK*18a] GROUEIX T., FISHER M., KIM V. G., RUSSELL B., AUBRY M.: 3d-coded : 3d correspondences by deep deformation. In *ECCV* (2018). 2, 3, 5

[GFK*18b] GROUEIX T., FISHER M., KIM V. G., RUSSELL B., AUBRY M.: AtlasNet: A Papier-Mâché Approach to Learning 3D Surface Generation. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)* (2018). 2, 5, 7

[HAWG08] HUANG Q., ADAMS B., WICKE M., GUIBAS L. J.: Non-rigid registration under isometric deformations. In *Computer Graphics Forum* (2008), vol. 27, pp. 1449–1457. 2

[HG13] HUANG Q.-X., GUIBAS L.: Consistent shape maps via semidefinite programming. In *Proceedings of the Eleventh Eurographics/ACMSIGGRAPH Symposium on Geometry Processing* (Aire-la-Ville, Switzerland, Switzerland, 2013), SGP '13, Eurographics Association, pp. 177–186. 1, 2

[HKC*18] HUANG H., KALOGERAKIS E., CHAUDHURI S., CEYLAN D., KIM V. G., YUMER E.: Learning local shape descriptors from part correspondences with multi-view convolutional networks. *Transactions on Graphics* (2018). 2

[HZG*12] HUANG Q.-X., ZHANG G.-X., GAO L., HU S.-M., BUTSCHER A., GUIBAS L.: An optimization approach for extracting and encoding consistent maps in a shape collection. *ACM Trans. Graph. 31*, 6 (Nov. 2012), 167:1–167:11. 2

[JSZ*15] JADERBERG M., SIMONYAN K., ZISSERMAN A., ET AL.: Spatial transformer networks. In *Advances in neural information processing systems* (2015), pp. 2017–2025. 3

[KAMC17] KALOGERAKIS E., AVERKIOU M., MAJI S., CHAUDHURI S.: 3D shape segmentation with projective convolutional networks. In *Proc. IEEE Computer Vision and Pattern Recognition (CVPR)* (2017). 3

[KB14] KINGMA D. P., BA J.: Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014). 3

[KHS10] KALOGERAKIS E., HERTZMANN A., SINGH K.: Learning 3D Mesh Segmentation and Labeling. *ACM Transactions on Graphics 29*, 3 (2010). 3

[KLF11] KIM V. G., LIPMAN Y., FUNKHOUSER T.: Blended intrinsic maps. *Transactions on Graphics (Proc. of SIGGRAPH)*, 4 (2011). 1, 2, 3

[KLM*12] KIM V. G., LI W., MITRA N. J., DIVERDI S., FUNKHOUSER T.: Exploring Collections of 3D Models using Fuzzy Correspondences. *Transactions on Graphics (Proc. of SIGGRAPH)*, 4 (2012). 1, 2

[KTEM18] KANAZAWA A., TULSIANI S., EFROS A. A., MALIK J.: Learning category-specific mesh reconstruction from image collections. 5

[LRR*17] LITANY O., REMEZ T., RODOLÀ E., BRONSTEIN A. M., BRONSTEIN M. M.: Deep functional maps: Structured prediction for dense shape correspondence. *CoRR abs/1704.08686* (2017). 2

[LSD*18] LI L., SUNG M., DUBROVINA A., YI L., GUIBAS L. J.: Supervised fitting of geometric primitives to 3d point clouds. *CVPR* (2018). 1, 3

[LSP08] LI H., SUMNER R. W., PAULY M.: Global correspondence optimization for non-rigid registration of depth scans. *Computer Graphics Forum (Proc. SGP'08) 27*, 5 (July 2008). 2

[MWZ*14] MITRA N. J., WAND M., ZHANG H., COHEN-OR D., KIM V. G., HUANG Q.-X.: Structure-Aware Shape Processing. *SIGGRAPH Course notes* (2014). 1

[MZC*19] MO K., ZHU S., CHANG A., YI L., TRIPATHI S., GUIBAS L., SU H.: PartNet: A large-scale benchmark for fine-grained and hierarchical part-level 3D object understanding. 1

[NBCW*11] NGUYEN A., BEN-CHEN M., WELNICKA K., YE Y., GUIBAS L.: An optimization approach to improving collections of shape maps. In *Computer Graphics Forum* (2011), vol. 30, Wiley Online Library, pp. 1481–1491. 1, 2, 3, 4

[OBCS*12] OVSJANIKOV M., BEN-CHEN M., SOLOMON J., BUTSCHER A., GUIBAS L.: Functional maps: A flexible representation of maps between shapes. *ACM Trans. Graph. 31*, 4 (July 2012), 30:1–30:11. 2, 3

[OMMG10] OVSJANIKOV M., MÃL'RIGOT Q., MÃL'MOLI F., GUIBAS L. J.: One point isometric matching with the heat kernel. *Comput. Graph. Forum 29*, 5 (2010), 1555–1564. 2

[QSMG16] QI C. R., SU H., MO K., GUIBAS L. J.: Pointnet: Deep learning on point sets for 3d classification and segmentation. *arXiv preprint arXiv:1612.00593* (2016). 1, 2, 3, 5, 7, 8, 9

[QYSG17] QI C. R., YI L., SU H., GUIBAS L. J.: Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *arXiv preprint arXiv:1706.02413* (2017). 1, 3

[RL01] RUSINKIEWICZ S., LEVOY M.: Efficient variants of the icp algorithm. In *Proceedings Third International Conference on 3-D Digital Imaging and Modeling* (2001). 1, 2

[ROA*13] RUSTAMOV R. M., OVSJANIKOV M., AZENCOT O., BEN-CHEN M., CHAZAL F., GUIBAS L.: Map-based exploration of intrinsic shape differences and variability. *ACM Trans. Graph. 32*, 4 (July 2013), 72:1–72:12. 2

[RPWO18] REN J., POULENARD A., WONKA P., OVSJANIKOV M.: Continuous and orientation-preserving correspondences via functional maps. *ACM Trans. Graph. 37*, 6 (Dec. 2018), 248:1–248:16. 2

[vKZHCO11] VAN KAICK O., ZHANG H., HAMARNEH G., COHEN-OR D.: A survey on shape correspondence. *Computer Graphics Forum 30*, 6 (2011), 1681–1707. 2

[WHC*16] WEI L., HUANG Q., CEYLAN D., VOUGA E., LI H.: Dense human body correspondences using convolutional networks. In *Computer Vision and Pattern Recognition (CVPR)* (2016). 2

[WSL*18] WANG Y., SUN Y., LIU Z., SARMA S. E., BRONSTEIN M. M., SOLOMON J. M.: Dynamic graph CNN for learning on point clouds. *CoRR abs/1801.07829* (2018). 1, 3

[WZL*18] WANG N., ZHANG Y., LI Z., FU Y., LIU W., JIANG Y.-G.: Pixel2mesh: Generating 3d mesh models from single rgb images. In *ECCV* (2018). 5

[WZS*19] WANG X., ZHOU B., SHI Y., CHEN X., ZHAO Q., XU K.: Shape2motion: Joint analysis of motion parts and attributes from 3d shapes. In *CVPR* (2019), p. to appear. 1

[YKC*16] YI L., KIM V. G., CEYLAN D., SHEN I.-C., YAN M., SU H., LU C., HUANG Q., SHEFFER A., GUIBAS L.: A scalable active framework for region annotation in 3d shape collections. *SIGGRAPH Asia* (2016). 1, 2, 3, 5

[YLZ*19] YU F., LIU K., ZHANG Y., ZHU C., XU K.: Partnet: A recursive part decomposition network for fine-grained and hierarchical shape segmentation. In *CVPR* (2019), p. to appear. 1

[Zha94] ZHANG Z.: Iterative point matching for registration of free-form curves and surfaces, 1994. 5

[ZKA*16] ZHOU T., KRÄHENBÜHL P., AUBRY M., HUANG Q., EFROS A. A.: Learning dense correspondence via 3d-guided cycle consistency. In *Computer Vision and Pattern Recognition (CVPR)* (2016). 2

[ZPIE17] ZHU J.-Y., PARK T., ISOLA P., EFROS A. A.: Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Computer Vision (ICCV), 2017 IEEE International Conference on* (2017). 3

[ZPK18] ZHOU Q.-Y., PARK J., KOLTUN V.: Open3D: A modern library for 3D data processing. *arXiv:1801.09847* (2018). 8

[ZSCO*08] ZHANG H., SHEFFER A., COHEN-OR D., ZHOU Q., VAN KAICK O., TAGLIASACCHI A.: Deformation-drive shape correspondence. *Computer Graphics Forum (Special Issue of Symposium on Geometry Processing) 27*, 5 (2008), 1431–1439. 2