Learning Implicit Functions for Dense 3D Shape Correspondence of Generic Objects

Feng Liu, *Member, IEEE,* and Xiaoming Liu, *Fellow, IEEE*

Abstract—The objective of this paper is to learn dense 3D shape correspondence for topology-varying generic objects in an unsupervised manner. Conventional implicit functions estimate the occupancy of a 3D point given a shape latent code. Instead, our novel implicit function produces a probabilistic embedding to represent each 3D point in a part embedding space. Assuming the corresponding points are similar in the embedding space, we implement dense correspondence through an inverse function mapping from the part embedding vector to a corresponded 3D point. Both functions are jointly learned with several effective and uncertainty-aware loss functions to realize our assumption, together with the encoder generating the shape latent code. During inference, if a user selects an arbitrary point on the source shape, our algorithm can automatically generate a confidence score indicating whether there is a correspondence on the target shape, as well as the corresponding semantic point if there is one. Such a mechanism inherently benefits man-made objects with different part constitutions. The effectiveness of our approach is demonstrated through unsupervised 3D semantic correspondence and shape segmentation.

Index Terms—Dense 3D shape correspondence, uncertainty-aware, unsupervised learning, implicit functions, inverse implicit functions, topology-varying, and generic objects.

1 Introduction

Finding dense correspondence between 3D shapes is a key algorithmic component in problems such as statistical modeling [1]–[3], cross-shape texture mapping [4], and space-time 4D reconstruction [5]. Dense 3D shape correspondence can be defined as: *given two 3D shapes belonging to the same object category, one can match an arbitrary point on one shape to its semantically equivalent point on another shape if such a correspondence exists*. For instance, given two chairs, the dense correspondence of the middle point on one chair's arm should be the similar middle point on another chair's arm, despite different shapes of arms; or alternatively, declare the non-existence of correspondence if another chair has no arm.

The dense 3D correspondence problem is difficult because it involves understanding the shapes at both the local and global levels. Prior dense correspondence methods [6]–[13] have proven to be effective on organic shapes, *e.g.*, human bodies and mammals. However, those methods become less suitable for generic topology-varying or man-made objects, *e.g.*, chairs or vehicles [14].

It remains a challenge to build dense 3D correspondence for a generic object category with large variations in geometry, structure, and even topology. First of all, the lack of annotations on dense correspondence often leaves *unsupervised learning* the only option. Second, most prior works make an inadequate assumption [15] that there is a similar topological variability between matched shapes. Man-made objects such as chairs shown in Fig. 1 are particularly challenging to tackle, since they often differ not only by geometric deformations but also by *part constitutions*. In these cases, existing correspondence methods for man-made objects either perform fuzzy [16], [17] or part-level [18], [19] correspondences or predict a constant number of semantic points [20], [21]. As a result, they cannot determine whether the established correspondence is a "missing match" or not. As shown in Fig. 1(c), for instance, we may find non-convincing correspondences in legs between an office chair and a 4-legged chair, or even no correspondences in arms for some pairs. Ideally, given a query point on the source shape, a dense correspondence method should be able to determine whether there exists a correspondence on the target shape, and identify the corresponding point if there is. This objective lies at the core of this work.

Shape representation is highly relevant to, and can impact, the approach of dense correspondence. Recently, compared to point cloud [22]–[24] or mesh [25]–[27], deep implicit functions have shown to be highly effective as 3D shape representations [28]–[34], since it can handle generic shapes of arbitrary topology, which is favorable as a representation for dense correspondence. Often learned as a multilayer perceptron (MLP), conventional implicit functions input the 3D shape represented by a latent code $\mathbf{z}$ and a query location $\mathbf{x}$ in the 3D space, and estimate its occupancy $O = f(\mathbf{x}, \mathbf{z})$.

In this work, we propose to plant the dense correspondence capability into the implicit function by learning a semantic part embedding. Specifically, we first adopt a branched implicit function [32] to learn a part embedding vector (PEV), $\mathbf{o} = f(\mathbf{x}, \mathbf{z})$, where the max-pooling of PEV $\mathbf{o}$ gives the occupancy O. In this way, each branch is tasked to learn a representation for one universal part of the input shape, and PEV represents the occupancy of the point w.r.t. all the branches/semantic parts. By assuming that PEVs between a pair of corresponding points are similar, we then establish dense correspondence via an inverse function $\hat{\mathbf{x}} = g(\mathbf{o}, \mathbf{z})$ mapping the PEV back to the 3D space. To

• *F. Liu and X. Liu are with the Department of Computer Science and Engineering, Michigan State University, East Lansing, MI 48824, U.S.A. E-mail: {liufeng6,liuxm}@msu.edu*

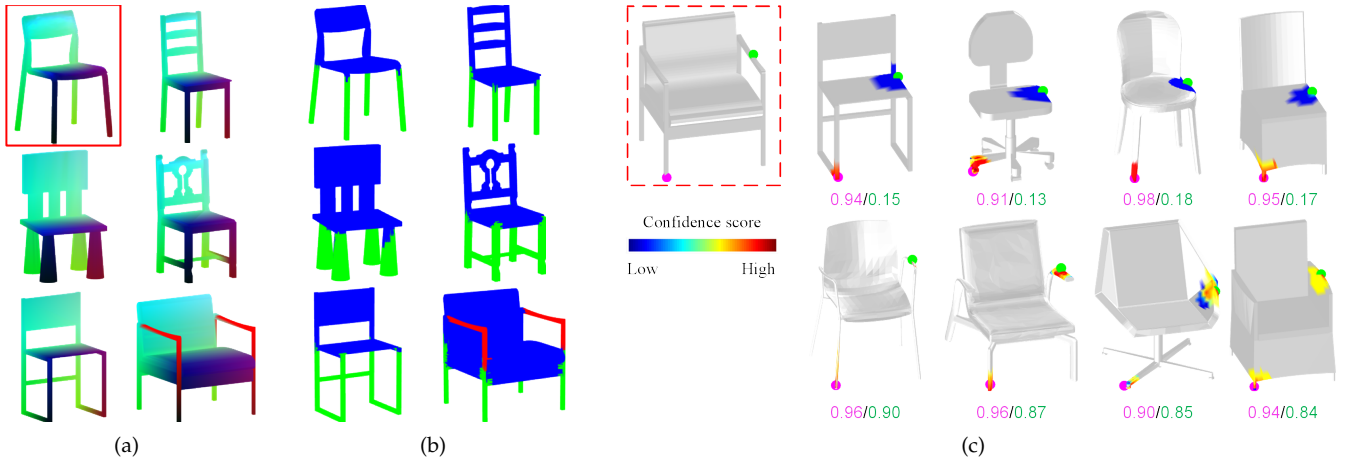


Fig. 1: Given a set of 3D shapes, our category-specific unsupervised method learns pair-wise dense correspondence (a) between any source and target shape (red box), and shape segmentation (b). Give an *arbitrary* point on the source shape (red box), our method predicts its corresponding point on any target shape, and a score measures the correspondence confidence (c). For each target, we show the confidence scores of red/green points and score maps around corresponded points. A score less than a threshold (*e.g.*, 0.2) deems the correspondence as "non-existing"– a desirable property for topology-varying shapes with missing parts, *e.g.*, chair's arm.

further satisfy the assumption, we devise an unsupervised learning framework with a joint loss measuring both the occupancy error and shape reconstruction error between $\mathbf{x}$ and $\hat{\mathbf{x}}$. In addition, a cross-reconstruction loss is proposed to enforce part embedding consistency by mapping between a pair of shapes in the collection. Besides, we adopt a probabilistic solution to the semantic part embedding learning, where each 3D point is represented as a Gaussian distribution in the semantic latent space. The mean of the distribution encodes the most likely PEVs while the variance shows the uncertainty along each feature dimension of PEVs. During inference, the "likelihood" between two Gaussian distributions can then be naturally derived to produce a confidence score for measuring the accuracy of the established point-to-point correspondence. And the learned uncertainty can be interpreted as the model's confidence along each feature dimension of PEVs, which can visualize the distribution of the "hard" points of a shape for dense correspondence.

A preliminary version of this work was published in the 34th Annual Conference on Neural Information Processing Systems (NeurIPS) 2020 [35]. We further extend the work from three aspects: (i) we design a novel ***deep*** branched implicit function, which gives a more powerful shape representation and improves shape correspondences. (ii) Instead of representing each point as a deterministic point in the semantic embedding space, we propose to use probabilistic embeddings for semantic part feature learning, which enables our framework to inherently capture the uncertainty of each point correspondence by its probabilistic PEV. and (iii) we further carry out dense correspondence evaluation and comparison on real scans (3D body shapes from Faust dataset [3]);

In summary, this paper makes these contributions.

- We propose a novel paradigm leveraging the implicit function representation for category-specific *unsupervised and uncertainty-aware* dense 3D shape correspondence, applicable to generic objects with diverse variations.
- We devise several effective loss functions to learn a semantic part embedding, which enables both shape segmentation and dense correspondence. Based on the learned probabilistic part embedding, our method further produces a confidence score indicating whether the predicted correspondence is valid.
- Through extensive experiments, we demonstrate the superiority of our method in 3D shape segmentation, 3D semantic correspondence, and dense 3D shape correspondence.

The rest of this paper is organized as follows. Section 2 briefly reviews related work in the literature. Section 3 introduces in detail the proposed unsupervised dense 3D shape correspondence algorithm based on implicit shape representation and its implementations. Section 4 reports the experimental results. Section 5 concludes the paper.

2 Related Work

2.1 Dense Shape Correspondence

While there are many dense correspondence works for organic shapes [6]–[12], [42], [43], here we focus on methods designed for man-made objects, including optimization and learning-based methods. For the former, most prior works build correspondences only at a *part* level [18], [19], [44]–[46]. Kim *et al.* [16] propose a diffusion map to compute point-based "fuzzy correspondence" for every shape pair. This is only effective for a small collection of shapes with limited shape variations. [38] and [47] present a template-based deformation method, which can find point-level correspondences after rigid alignment between the template and target shapes. However, these methods only predict coarse and discrete correspondence, leaving the structural

TABLE 1: A comparison of shape correspondence methods designed for generic objects is presented. 'Template' refers to methods that are template-based or require templates. Note that the methods requiring templates and template-based methods might not be suitable for man-made objects, since they have significant differences in the number and arrangement of their parts.

Method	Type	Supervision	Template	Corr. Level	Shape Representation	Non-Existence Detection	Content, Uncertainty-aware
Slavcheva *et al.* [36]	Optimization	unsupervised	✗	dense	implicit function	✗	bodies, ✗
3D-CODED [8]	Learning	self-supervised	✓	dense	mesh	✗	bodies, ✗
LoopReg [37]	Learning	self-supervised	✗	dense	implicit function	✗	bodies, ✗
Kim12 [16]	Optimization	unsupervised	✗	dense	point	✗	man-made objects, ✗
Kim13 [38]	Optimization	unsupervised	✓	dense	point	✗	man-made objects, ✗
LMVCNN [20]	Learning	supervised	✗	dense	point	✗	man-made objects, ✗
ShapeUnicode [39]	Learning	supervised	✗	sparse	point	✗	man-made objects, ✗
Chen *et al.* [21]	Learning	unsupervised	✗	sparse	point	✗	man-made objects, ✗
DIF [40]	Learning	unsupervised	✓	dense	implicit function	✗	man-made objects, ✗
Zheng *et al.* [41]	Learning	unsupervised	✓	dense	implicit function	✗	man-made objects, bodies, ✗
Proposed	Learning	unsupervised	✗	dense	implicit function	✓	man-made objects, bodies, ✓

or topological discrepancies between matched parts or part ensembles unresolved.

A series of learning-based methods [20], [39], [48]–[50] are proposed to learn local descriptors, and treat correspondence as 3D semantic landmark estimation. For example, ShapeUnicode [39] learns a unified embedding for 3D shapes and demonstrates its ability in correspondence among 3D shapes. However, these methods require *ground-truth* pairwise correspondences for training. Recently, Chen *et al.* [21] present an unsupervised method to estimate 3D structure points. Unfortunately, it estimates a *constant* number of *sparse* structured points. As shapes may have diverse part constitutions, it may not be meaningful to establish the correspondence between all of their points. Groueix *et al.* [51] also learn a parametric transformation between two surfaces by leveraging cycle consistency, and apply it to the segmentation problem. However, the deformation-based method always deforms all points on one shape to another, even for the points from a non-matching part. In contrast, our *unsupervised and uncertainty-aware* learning model can perform pairwise *dense* correspondence for any two shapes of a man-made object. We summarize the comparison in Tab. 1.

2.2 Implicit Shape Representation

Due to the advantages of being a continuous representation and handling complicated topologies, implicit functions have been adopted for learning representations for 3D shape generation [28]–[30], [33], encoding texture [31], [52], [53], 3D reconstruction [54]–[56], and 4D reconstruction [5]. Meanwhile, some extensions have been proposed to learn deep structured [57], [58] or segmented implicit functions [32], or separate implicit functions for shape parts [59]. Further, some works [36], [37], [40], [41], [60]–[62] leverage the implicit representation together with a deformation model for shape registration. However, these methods rely on the deformation model, which might prevent their usage for topology-varying objects. Slavcheva *et al.* [36] implicitly obtain correspondence for organic shapes by predicting the evolution of the signed distance field. However, as they require a Laplacian operator to be invariant, it is limited to small shape variations. Recently, Zheng *et al.* [41] present a deep implicit template, a new 3D shape representation that factors out the implicit template from deep implicit functions. Additionally, a spatial warping module deforms the template's implicit function to form specific object instances, which reasons dense correspondences between different shapes. Similarly, DIF [40] introduces a deep implicit template field together with a deformation module to represent 3D models with correspondences. However, these methods assume that the object instances within a category are mostly composed of a few common semantic structures, which inevitably limits their effectiveness for topology-varying objects.

On the other hand, in order to preserve fine-grained shape details in implicit function learning, Wang *et al.* [63] propose to combine 3D query point features with local image features to predict the SDF values of the 3D points, which is able to generate shape details. Meanwhile, instead of encoding the shape in a single latent code $\mathbf{z}$, Chibane *et al.* [64] and Peng *et al.* [65] propose to extract a learnable multi-scale tensor of deep features. Then, instead of classifying point coordinates $\mathbf{x}$ directly, they classify deep features extracted at continuous query points, preserving local details. In this paper, we propose a deep branched implicit function, which can also improve the fidelity of shape representations.

2.3 Uncertainty in Deep Learning

Recent years have witnessed a trend to estimate uncertainty in deep neural networks (DNNs) [66]–[69]. Specific to deep learning models for the computer vision field, the uncertainties can be classified into two main types: model (or epistemic) uncertainty and data (or aleatoric) uncertainty. Model uncertainty accounts for uncertainty in the model parameters and can be remedied with sufficient training data [70]–[72]. Data uncertainty captures the noise inherent in the training data, which cannot be reduced even with enough data [67]. Recently, uncertainty learning has been widely applied to various tasks, such as semantic segmentation [72], [73], depth estimation [74], [75], depth completion [76], [77], multi-view stereo [78], visual correspondence [79], face alignment [80], 3D reconstruction [56], and face recognition [81]–[83]. In this work, we introduce

an uncertainty solution in our dense correspondence model by representing 3D points as distributions instead of deterministic points in our semantic part embedding space. Consequently, the learned variance of PEVs can be used as the measurement of the point-wise correspondence, which is suitable for generic objects with rich geometric and topological variations.

2.4 Unsupervised Shape Co-Segmentation

Co-segmentation is one of the fundamental tasks in geometry processing. Prior works [84]–[86] develop clustering strategies for meshes, given a handcrafted similarity metric induced by an embedding or graph [18], [45], [87]. The segmentation for each cluster is computed independently without accounting for statistics of shape variations, and the overall complexity of these methods is quadratic in the number of shapes in the collection. Recently, BAE-NET [32] presents an unsupervised branched autoencoder with 3 fully-connected layers that discovers coarse segmentation of shapes by predicting implicit fields for each part. According to the evaluation in the BAE-NET [32], the 3-layer network is the best choice for independent shape extraction, making it a suitable candidate for shape segmentation. However, the shallow network structure results in a limited shape representation power. In contrast, we extend the branched implicit function with a deep architecture, making it suitable for shape reconstruction as well.

3 Proposed Method

3.1 Preliminaries

Let us first formulate the dense 3D correspondence problem. Given a collection of 3D shapes of the same object category, one may encode each shape $\mathbf{S} \in \mathbb{R}^{n\times 3}$ in a latent space $\mathbf{z} \in \mathbb{R}^d$. As show in Fig. 2(a), for any point $p \in \mathbf{S}_A$ in the source shape $\mathbf{S}_A$, dense 3D correspondence will find its semantic corresponding point $q \in \mathbf{S}_B$ in the target shape $\mathbf{S}_B$. if a semantic embedding function (SEF) $f : \mathbb{R}^3 \times \mathbb{R}^d \rightarrow \mathbb{R}^k$ is able to satisfy

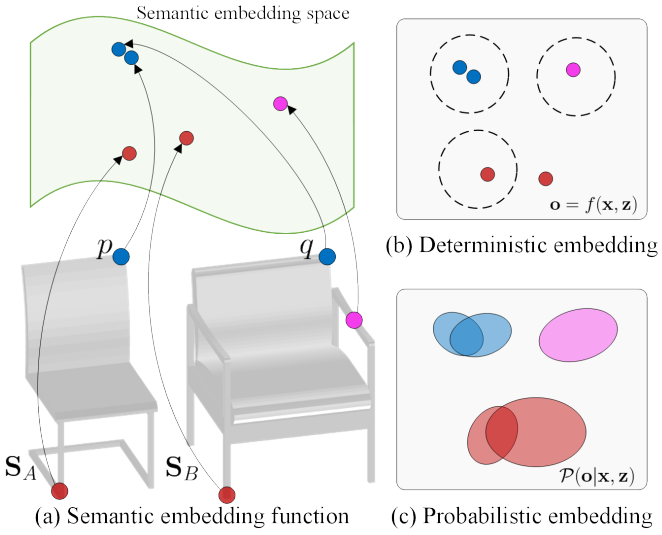
$$\left(\min_{q\in\mathbf{S}_B} ||f(p,\mathbf{z}_A) - f(q,\mathbf{z}_B)||_2\right) < \tau, \quad \forall p \in \mathbf{S}_A. \tag{1}$$

Here the SEF is responsible for mapping a point from its 3D Euclidean space to the semantic embedding space. When p and q have sufficiently similar locations in the semantic embedding space, they have similar semantic meaning, or functionality, in their respective shapes. Hence q is the corresponding point of p. On the other hand, if their distance in the embedding space is too large ($\geq \tau$), there is no corresponding point in $\mathbf{S}_B$ for p. If SEF could be learned for a small τ, the corresponded point q of p can be solved via $q = f^{-1}(f(p,\mathbf{z}_A),\mathbf{z}_B)$, where $f^{-1}(:,:)$ is the inverse function of f that maps a point from the semantic embedding space back to the 3D space. Therefore, the dense 3D correspondence problem amounts to learning the SEF and its inverse function.

Probabilistic Semantic Embedding Learning. Our preliminary work [35] adopts a deterministic point representation for each 3D point in the semantic embedding space. However, it is difficult to estimate an accurate point embedding for shape parts with semantic ambiguity, which usually has larger uncertainty in the embedding space (Fig. 2(b)). Also, these ambiguous features will negatively affect the mapping of the inverse function, leading a poor correspondence accuracy. To address this issue, we propose to utilize probabilistic embeddings to predict a distributional estimation $\mathcal{P}(\mathbf{o}|\mathbf{x},\mathbf{z})$ instead of a point estimation $f(\mathbf{x},\mathbf{z})$, for each 3D point of the shapes in the semantic embedding space (Fig. 2(c)). Specifically, we define the PEV in the latent space as a Gaussian distribution:

$$\mathcal{P}(\mathbf{o}|\mathbf{x},\mathbf{z}) = \mathcal{N}(\mathbf{o};\mathbf{o}_\mu,\mathbf{o}_\sigma^2\mathbf{I}), \tag{2}$$

where the mean and variance of the Gaussian distribution are predicted by the function f: $(\mathbf{o}_\mu,\mathbf{o}_\sigma) = f(\mathbf{x},\mathbf{z})$. Here, The mean $\mathbf{o}_\mu$ can be regarded as the semantic feature of the point. The variance $\mathbf{o}_\sigma$ encodes the model's uncertainty along each feature dimension.



(a) Semantic embedding function (b) Deterministic embedding (c) Probabilistic embedding

Fig. 2: (a) We seek to learn a semantic embedding function, which maps a point from its 3D Euclidean space to the semantic embedding space. Consequently, when p and q locate in similar locations in the semantic embedding space, they have similar semantic meanings in their respective shapes. (b) In our preliminary work [35], the learned semantic embedding is a deterministic model, which represents each 3D point as a **deterministic point** in the latent space without considering its feature ambiguity (*i.e.*, the leg point of $\mathbf{S}_A$). (c) In this work, we propose to use probabilistic embeddings to give a **distributional** estimation of PEVs in the semantic space, which is able to capture a point-wise uncertainty in the dense correspondence model.

Implementation Solution. As shown in Fig. 3, we propose to leverage the topology-free implicit function, a conventional shape representation, to jointly serve as the SEF. By assuming that corresponding points are similar in the embedding space, we explicitly implement an inverse function mapping from the embedding space to the 3D space, so that the learning objectives can be more conveniently defined in the 3D space rather than the embedding space. Both functions are jointly learned with an occupancy loss for accurate shape representation, and an uncertainty-aware

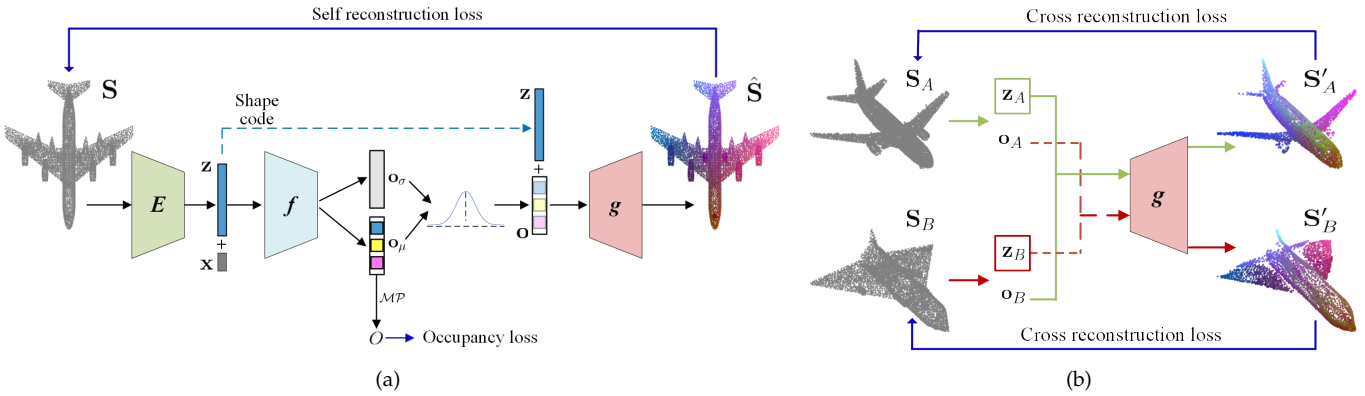


Fig. 3: Model Overview. (a) Given a shape $\mathbf{S}$, PointNet E is used to extract the shape feature code $\mathbf{z}$. The parameters ($\mathbf{o}_\mu$, $\mathbf{o}_\sigma$) of the Gaussian distribution are predicted via a deep implicit function f. Then a stochastic part embedding vector $\mathbf{o}$ is sampled from $\mathcal{N}(\mathbf{o}; \mathbf{o}_\mu, \mathbf{o}_\sigma^2\mathbf{I})$ in the semantic embedding space. We implement dense correspondence through an inverse function mapping from $\mathbf{o}$ to recover the 3D shape $\hat{\mathbf{S}}$. (b) To further make the learned part embedding consistent across all the shapes, we randomly select two shapes $\mathbf{S}_A$ and $\mathbf{S}_B$. By swapping the part embedding vectors, a cross-reconstruction loss is used to enforce the inverse function to recover to each other. $\mathcal{MP}$ denotes the max-pooling operator.

self-reconstruction loss for the inverse function to recover itself. In addition, we propose an uncertainty-aware cross-reconstruction loss enforcing two objectives. One is that the two functions can deform source shape points to be sufficiently close to the target shape. The other is that the offset vectors between corresponding points, $\overrightarrow{pq}$, are *locally* smooth within the neighborhood of p.

3.2 PointNet Encoder

To perform dense correspondence for a 3D shape, we need to first obtain a latent representation describing its overall shape. In this work, given a shape $\mathbf{S} \in \mathbb{R}^{n\times 3}$, we utilize a PointNet-based network to encode the shape into a latent code space. We adopt the original PointNet [23] without the STN module to extract a global shape code $\mathbf{z} \in \mathbb{R}^d$:

$$E : \mathbb{R}^{n\times 3} \rightarrow \mathbb{R}^d. \tag{3}$$

3.3 Uncertainty-aware Implicit Function

Based on the shape code $\mathbf{z}$ of an object, as in [29], [33], the 3D shape of the object can be reconstructed by an implicit function. That is, given the 3D coordinate of a query point $\mathbf{x} \in \mathbb{R}^3$, the implicit function assigns an occupancy probability O between 0 and 1: $\mathbb{R}^3 \times \mathbb{R}^d \rightarrow [0, 1]$, where 1 indicates $\mathbf{x}$ is inside the shape, and 0 outside.

This conventional function can not serve as our SEF, given its simple 1D output. Motivated by the unsupervised part segmentation [32], we adopt its branched layer as the final layer of our implicit function, whose outputs are denoted by $\mathbf{o}_\mu \in \mathbb{R}^k$ and $\mathbf{o}_\sigma \in \mathbb{R}^k$:

$$f : \mathbb{R}^3 \times \mathbb{R}^d \rightarrow (\mathbb{R}^k, \mathbb{R}^k). \tag{4}$$

A max-pooling operator ($\mathcal{MP}$) leads to the final occupancy $O = \mathcal{MP}(\mathbf{o}_\mu)$ by selecting one branch from $\mathbf{o}_\mu$, whose index indicates the unsupervisedly estimated part where $\mathbf{x}$ belongs to. Conceptually, each element of $\mathbf{o}_\mu$ shall indicate the occupancy value of $\mathbf{x}$ w.r.t. the respective part. Since $\mathbf{o}$ appears to represent the occupancy of $\mathbf{x}$ w.r.t. all semantic parts of the object, the latent space of $(\mathbf{o}_\mu, \mathbf{o}_\sigma)$ can be the desirable probabilistic semantic embedding. In our implementation, f is a 3-layer multilayer perceptron (MLP). The final layer consists of two separate fully connected (FC) layers designed to produce the mean $\mathbf{o}_\mu$ and variance $\mathbf{o}_\sigma$ of the Gaussian distribution.

Deep Branched Implicit Function. As detailed in the work [32], the network structure of 3-layer implicit function can be sensitive to the initial parameters and it cannot infer semantic information when the number of layers is greater than 3. As a result, the limited depth of the network makes it difficult to represent shape details, as well as obtain fine-grained semantic correspondence. To address these issues, we introduce a ***deep*** branched implicit function network. Specifically, as shown in Fig. 4, we first generate deep point-specific features via four parallel MLPs. We then concatenate those features to produce a point-specific latent code $\hat{\mathbf{z}}$. The final 3 FC layers produce the occupancy value of the query point. Querying deep features extracted at continuous 3D locations used in implicit function learning allows us to reconstruct the local geometric structure of generic objects, enhancing the point-wise semantic representation in the semantic embedding.

3.4 Inverse Implicit Function

Given the objective function in Eqn. 1, one may consider that learning SEF, f, would be sufficient for dense correspondence. However, there are two problems with this. First of all, to find correspondence of p, we need to compute $f^{-1}(f(p, \mathbf{z}_A), \mathbf{z}_B)$, *i.e.*, assuming the output of $f(q, \mathbf{z}_B)$ equals $f(p, \mathbf{z}_A)$ and solve for q via iterative back-propagation. This could be computationally inefficient during inference. Secondly, it is easier to define shape-related constraints or loss functions between $f^{-1}(f(p, \mathbf{z}_A), \mathbf{z}_B)$ and q in the 3D space, rather than those between $f(q, \mathbf{z}_B)$ and $f(p, \mathbf{z}_A)$ in the embedding space.

To this end, we define the inverse implicit function to take the probabilistic part embedding vector (PEV) $\mathbf{o}$ and

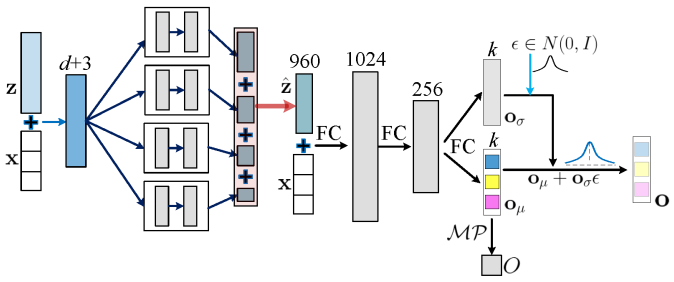


Fig. 4: Uncertainty-aware deep branched implicit function network. Four branched MLPs are first utilized to produce point features in different scales. Then the features are aggregated into a single point-wise latent vector $\hat{\mathbf{z}}$. The final 4 fully connected layers predict the mean $\mathbf{o}_\mu \in \mathbb{R}^k$ and variance $\mathbf{o}_\sigma \in \mathbb{R}^k$ of the Gaussian distribution. The part embedding vector $\mathbf{o}$ of each point $\mathbf{x}$ is not a deterministic point embedding any point, but a stochastic embedding sampled from $\mathcal{N}(\mathbf{o}; \mathbf{o}_\mu, \mathbf{o}_\sigma^2\mathbf{I})$. A max-pooling operator leads to the final occupancy $O{=}\mathcal{MP}(\mathbf{o}_\mu)$.

the shape code $\mathbf{z}$ as inputs, and recover the corresponding 3D location:

$$g : \mathbb{R}^k \times \mathbb{R}^d \rightarrow \mathbb{R}^3. \tag{5}$$

The probabilistic PEV can be reformulated as $\mathbf{o} = \mathbf{o}_\mu + \epsilon\mathbf{o}_\sigma$, where $\epsilon \sim \mathcal{N}(0, \mathbf{I})$. We also use an MLP network to implement g. With g, we can efficiently compute $g(f(p, \mathbf{z}_A), \mathbf{z}_B)$ via forward passing, without iterative back-propagation.

3.5 Training Loss Functions

We jointly train our implicit function and inverse function by minimizing three losses: the occupancy loss $\mathcal{L}^{occ}$, the uncertainty-aware self-reconstruction loss $\mathcal{L}^{SR}$, and the uncertainty-aware cross-reconstruction loss $\mathcal{L}^{CR}$, *i.e.*,

$$\mathcal{L}^{all} = \mathcal{L}^{occ} + \mathcal{L}^{SR} + \mathcal{L}^{CR}, \tag{6}$$

where $\mathcal{L}^{occ}$ measures how accurately f predicts the occupancy of the shapes, $\mathcal{L}^{SR}$ enforces that g is an inverse function of f, and $\mathcal{L}^{CR}$ strives for part embedding consistency across all shapes in the collection. We first explain how we prepare the training data, and then provide the details of the loss functions.

3.5.1 Training Samples

Given a collection of N raw 3D surfaces $\{\mathbf{S}_i^{raw}\}_{i=1}^N$ with a consistent upright orientation, we first normalize the raw surfaces by uniformly scaling the object such that the diagonal of its tight bounding box has a constant length and make the surfaces watertight by converting them to voxels. In order to train the implicit function model, following the sample scheme of [33], we randomly sample and obtain K spatial points $\{\mathbf{x}_j\}_{j=1}^K$ and their occupancy labels $\{\tilde{O}_j\}_{j=1}^K \in \{0, 1\}$ near the surface, which are 1 for the inside points and 0 otherwise. In addition, to learn discriminative shape codes, we further uniformly sample n *surface points* to represent 3D shapes, resulting in $\{\mathbf{S}_i\}_{i=1}^n$.

3.5.2 Occupancy Loss

This is a L_2 error between the label and estimated occupancy of all shapes:

$$\mathcal{L}^{occ} = \sum_{i=1}^{N}\sum_{j=1}^{K} \|\mathcal{MP}(f_{\mathbf{o}_\mu}(\mathbf{x}_j, \mathbf{z}_i)) - \tilde{O}_j\|_2^2. \tag{7}$$

3.5.3 Uncertainty-aware Self-Reconstruction Loss

the inverse function aims to map from the embedding space to the 3D space. The variance could actually be regarded as the uncertainty measuring the confidence of the inverse mapping. Following the minimisation objective suggested by [67], we supervise the inverse function by recovering input surface $\mathbf{S}_i$:

$$\begin{aligned}\mathcal{L}^{SR} = &\sum_{i=1}^{N}\sum_{j=1}^{n} \frac{1}{2}(\mathbf{o}_\sigma^{(j)})^{-2}\|g(f(\mathbf{S}_i^{(j)}, \mathbf{z}_i), \mathbf{z}_i) - \mathbf{S}_i^{(j)}\|_2^2 \\ &+ \frac{1}{2}\log(\mathbf{o}_\sigma^{(j)})^2,\end{aligned} \tag{8}$$

where $\mathbf{S}_i^{(j)}$ is the j-th point of shape $\mathbf{S}_i$ and $\mathbf{o}_\sigma^{(j)}$ denotes the mean value across all dimensions of its variance. The first term $\frac{1}{2}(\mathbf{o}_\sigma^{(j)})^{-2}$ serves as a weighted distance which assigns larger weights to less uncertainty vectors. The second term $\frac{1}{2}\log(\mathbf{o}_\sigma^{(j)})^2$ penalizes points with high uncertainties. In practice, we train the function f to predict the log variance $\log \mathbf{o}_\sigma^2$ for stable optimization.

3.5.4 Uncertainty-aware Cross-Reconstruction Loss

The cross-reconstruction loss is designed to encourage the resultant PEVs to be similar for densely corresponded points from any two shapes. As shown in Fig. 3, from a shape collection we first randomly select two shapes $\mathbf{S}_A$ and $\mathbf{S}_B$. The implicit function f generates PEV sets $\{\mathbf{o}_A\}$ ($\{\mathbf{o}_B\}$), given $\mathbf{S}_A$ ($\mathbf{S}_B$) and their respective shape codes $\mathbf{z}_A$ ($\mathbf{z}_B$) as inputs. Then we swap their PEVs and feed the concatenated vectors to the inverse function g: $\mathbf{S}_A^{'(j)} = g(\mathbf{o}_B^{(j)}, \mathbf{z}_A), \mathbf{S}_B^{'(j)} = g(\mathbf{o}_A^{(j)}, \mathbf{z}_B)$. If the part embedding is point-to-point consistent across all shapes, the inverse function should recover each other, *i.e.*, $\mathbf{S}'_A \approx \mathbf{S}_A$, $\mathbf{S}'_B \approx \mathbf{S}_B$. Towards this goal, we exploit several loss functions to minimize the pairwise difference for each of these two shape pairs:

$$\mathcal{L}^{CR} = \lambda_1\mathcal{L}^{CD} + \lambda_2\mathcal{L}^{EMD} + \lambda_3\mathcal{L}^{nor} + \lambda_4\mathcal{L}^{smo}, \tag{9}$$

where $\mathcal{L}^{CD}$ is the Chamfer Distance (CD) loss, $\mathcal{L}^{EMD}$ the Earth Mover distance (EMD) loss, $\mathcal{L}^{nor}$ the surface normal loss, $\mathcal{L}^{smo}$ the smooth correspondence loss, and λ_i are the weights. The first three terms focus on shape similarity, while the last one encourages the correspondence offsets to be locally smooth. We empirically apply uncertainty learning in $\mathcal{L}^{CD}$ only since $\mathcal{L}^{CD}$ essentially reflects the main results of the cross-reconstruction process.

Uncertainty-aware Chamfer Distance Loss. is defined as:

$$\mathcal{L}^{CD} = d_{CD}(\mathbf{S}_A, \mathbf{S}'_A) + d_{CD}(\mathbf{S}_B, \mathbf{S}'_B), \tag{10}$$

Algorithm 1 Dense correspondence inference.

Input: Two surface point sets: $\mathbf{S}_A$ (Source) and $\mathbf{S}_B$ (Target).
Output: The corresponding point sets and their confidence scores $\mathbf{S}_{A\to B} = \{q, C\}$ on $\mathbf{S}_B$.

Initialisation :

$\mathbf{z}_A \leftarrow E(\mathbf{S}_A)$, $\mathbf{z}_B \leftarrow E(\mathbf{S}_B)$;
$\{\mathbf{o}_{\mu B}\} \leftarrow, f(\mathbf{z}_B, \mathbf{S}_B)$, $\{\mathbf{o}_{\mu A}\} \leftarrow f(\mathbf{z}_A, \mathbf{S}_A)$;
$\mathbf{S}'_A \leftarrow g(\mathbf{z}_A, \{\mathbf{o}_{\mu B}\})$;
$\mathbf{S}_{A\to B} \leftarrow \emptyset$

LOOP Search Function:

for each point p in $\mathbf{S}_A$ **do**
  Find a preliminary correspondence q' in $\mathbf{S}'_A$ via $q' = \arg\min_{q'\in\mathbf{S}'_A} \|p - q'\|_2$;
  Knowing the index of q' in $\mathbf{S}'_A$, the same index in $\mathbf{S}_B$ refers to the final correspondence $q \in \mathbf{S}_B$;
  Compute the confidence score via Eqn. 17.
  if $\mathcal{C} > \tau'$ **then**
    $\mathbf{S}_{A\to B} \leftarrow \mathbf{S}_{A\to B} \,||\, (q, C)$;
  else
    $\mathbf{S}_{A\to B} \leftarrow \mathbf{S}_{A\to B} \,||\, (\emptyset, C)$;
  end if
end for
return $\mathbf{S}_{A\to B}$

where CD is calculated as [23]:

$$d_{CD}(\mathbf{S}, \mathbf{S}') = \frac{1}{2}(\mathbf{o}_\sigma^{(p)})^{-2} \sum_{p\in\mathbf{S}} \min_{q\in\mathbf{S}'} \|p - q\|_2^2 + \frac{1}{2} log(\mathbf{o}_\sigma^{(p)})^2 + \frac{1}{2}(\mathbf{o}_\sigma^{(p)})^{-2} \sum_{q\in\mathbf{S}'} \min_{p\in\mathbf{S}} \|p - q\|_2^2 + \frac{1}{2} log(\mathbf{o}_\sigma^{(p)})^2, \tag{11}$$

where $\mathbf{o}_\sigma^{(p)}$ is the mean of all values in variance of point p. Similarly, the variance in the semantic embedding learns the correspondence uncertainty through such cross-reconstruction.

Earth Mover Distance Loss. is defined as:

$$\mathcal{L}^{EMD} = d_{EMD}(\mathbf{S}_A, \mathbf{S}'_A) + d_{EMD}(\mathbf{S}_B, \mathbf{S}'_B), \tag{12}$$

where EMD is the minimum of the sum of distances between a point in one set and a point in another set over all possible permutations of correspondences [23]:

$$d_{EMD}(\mathbf{S}, \mathbf{S}') = \min_{\Phi:\mathbf{S}\to\mathbf{S}'} \sum_{p\in\mathbf{S}} \|p - \Phi(p)\|_2, \tag{13}$$

where Φ is a bijective mapping.

Surface Normal Loss. An appealing property of implicit representation is that the surface normal can be analytically computed using the spatial derivative $\frac{\partial \mathcal{MP}(f(\mathbf{x},\mathbf{z}))}{\partial \mathbf{x}}$ via back-propagation through the network. Hence, we are able to define the surface normal distance on the point sets.

$$\mathcal{L}^{nor} = d_{nor}(\mathbf{n}_A, \mathbf{n}'_A) + d_{nor}(\mathbf{n}_B, \mathbf{n}'_B), \tag{14}$$

where $\mathbf{n}_*$ is the surface normal of $\mathbf{S}_*$. We measure d_{nor} by the Cosine similarity distance:

$$d_{nor}(\mathbf{n}, \mathbf{n}') = \frac{1}{n}\sum_i (1 - \mathbf{n}_i \cdot \mathbf{n}'_i), \tag{15}$$

where $\cdot$ denotes the dot-product.

Smooth Correspondence Loss. encourages that the correspondence offset vectors $\Delta\mathbf{S}_{AB} = \mathbf{S}'_B - \mathbf{S}_A$, $\Delta\mathbf{S}_{BA} = \mathbf{S}'_A - \mathbf{S}_B$ of neighboring points are as similar as possible to ensure a smooth deformation:

$$\mathcal{L}^{smo} = \sum_{a\in\mathbf{S}_A, a'\in\mathbb{N}(a)} \|\Delta\mathbf{S}_{AB}^{(a)} - \Delta\mathbf{S}_{AB}^{(a')}\|_2 + \sum_{b\in\mathbf{S}_B, b'\in\mathbb{N}(b)} \|\Delta\mathbf{S}_{BA}^{(b)} - \Delta\mathbf{S}_{BA}^{(b')}\|_2, \tag{16}$$

where $\mathbb{N}(a)$ and $\mathbb{N}(b)$ are neighborhoods for a and b respectively. Here, for the local neighborhood selection, we utilize the radius-based ball query strategy [24] with the radius being 0.1.

3.6 Inference

During inference, our method can offer both shape segmentation and dense correspondence for 3D shapes. As each element of PEV learns a compact representation for one common part of the shape collection, the shape segmentation of p is the index of the element being max-pooled from its PEV. As both the implicit function f and its inverse g are point-based, the number of input points to f can be arbitrary during inference. As depicted in Algorithm 1, given two point sets $\mathbf{S}_A$, $\mathbf{S}_B$ with shape codes $\mathbf{z}_A$ and $\mathbf{z}_B$, f generates the mean of PEVs $\mathbf{o}_{\mu A}$ and $\mathbf{o}_{\mu B}$, and g outputs cross-reconstructed shape $\mathbf{S}'_A$. For any query point $p \in \mathbf{S}_A$, a preliminary correspondence may be found by the nearest neighbor search in $\mathbf{S}'_A$: $q' = \arg\min_{q'\in\mathbf{S}'_A} \|p - q'\|_2$. Knowing the index of q' in $\mathbf{S}'_A$, the same index in $\mathbf{S}_B$ refers to the final correspondence $q \in \mathbf{S}_B$.

Finally, given the probabilistic semantic embedding $(\mathbf{o}_A^{i_p}, \mathbf{o}_B^{i_q})$ of the corresponding points (p,q), the correspondence confidence can be computed by measuring the "likelihood" of them:$\mathcal{P}(\mathbf{o}_A^{i_p} = \mathbf{o}_B^{i_q})$, where $\mathbf{o}_A^{i_p} \sim \mathcal{P}(\mathbf{o}|p, \mathbf{z}_A)$ and $\mathbf{o}_B^{i_q} \sim \mathcal{P}(\mathbf{o}|q, \mathbf{z}_B)$. In practice, we adopt the mutual likelihood score as the confidence score [81]:

$$\mathcal{C} = -\sum_l^k \left(\frac{({\mathbf{o}_{\mu A}^{i_p}}^{(l)} - {\mathbf{o}_{\mu B}^{i_q}}^{(l)})}{(\mathbf{o}_{\sigma A}^{i_p})^{2(l)} + (\mathbf{o}_{\sigma B}^{i_q})^{2(l)}} + \log\left((\mathbf{o}_{\sigma A}^{i_p})^{2(l)} + (\mathbf{o}_{\sigma B}^{i_q})^{2(l)}\right) \right) \tag{17}$$

where i_p is the index of p in $\mathbf{S}_A$. ${\mathbf{o}_\mu}^{(l)}$ refers to the l^{th} dimension of $\mathbf{o}_\mu$ and similarly for ${\mathbf{o}_\sigma}^{(l)}$. Here, $\mathcal{C}$ is normalized to the range of $[0, 1]$ with min-max normalization for all the testing samples. Since the learned part embedding is discriminative among different parts of a shape, the distance of PEVs is suitable to define confidence. When $\mathcal{C}$ is larger than a pre-defined threshold τ', this $p \to q$ correspondence is valid; otherwise p has no correspondence in $\mathbf{S}_B$.

3.7 Implementation Detail

3.7.1 Sampling Point-Value Pairs

The training of implicit function network needs point-value pairs. Following the sampling strategy of [33], we obtain the

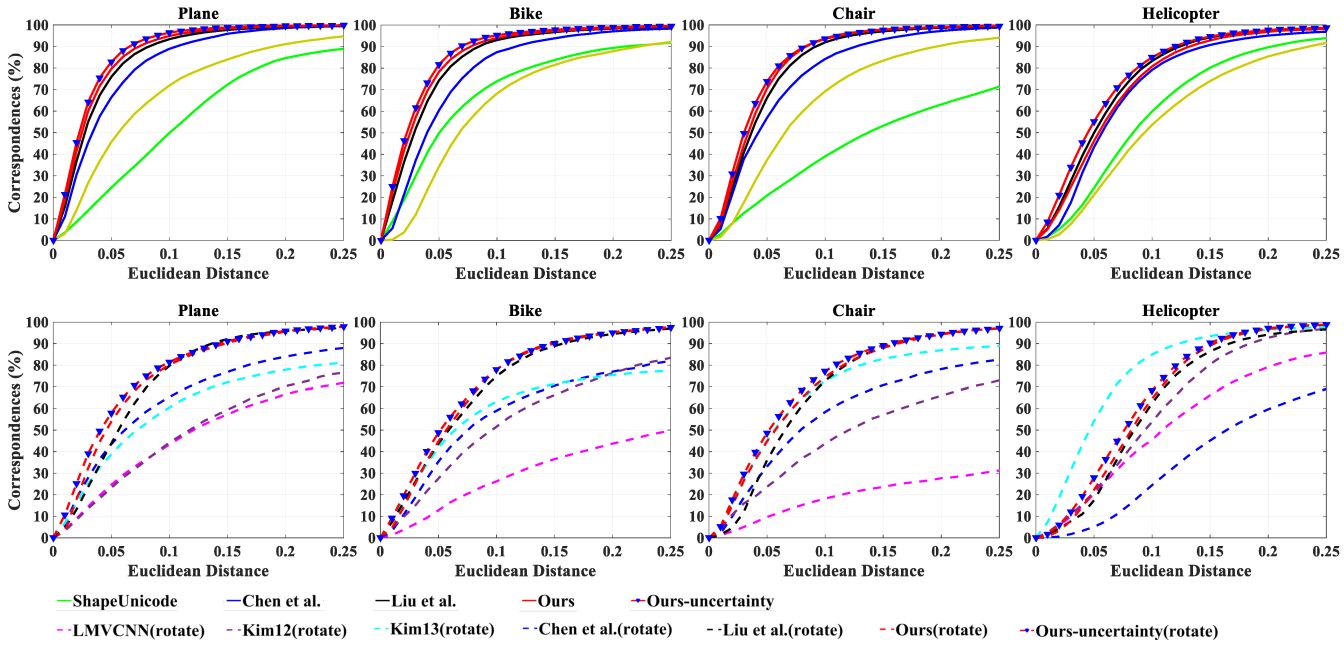


Fig. 5: Correspondence accuracy for 4 categories in the BHCP benchmark. The solid and dashed lines are for the aligned (top) and unaligned (bottom) setting respectively. All baseline results are quoted from [21], [38].

TABLE 2: Three stages of the training process.

Stage	Updated networks	Loss functions
1	E, f	$\mathcal{L}^{occ}$
2	E, f, g	$\mathcal{L}^{occ}$ and $\mathcal{L}^{SR}$
3	E, f, g	$\mathcal{L}^{all}$

paired data $\{\mathbf{x}_j, \tilde{O}_j\}_{j=1}^{K}$ offline. $\mathbf{x}_j, \tilde{O}_j$ are the spatial point and its corresponding occupancy label. For each 3D shape, we utilize the technique of Hierarchical Surface Prediction (HSP) [88] to generate the voxel models at different resolutions (16^3, 32^3, 64^3). We then respectively sample points ($K = 4,096$, $K = 8,192$, $K = 32,768$) on three resolutions in order to train the implicit function *progressively*.

3.7.2 Training Process

We summarize the training process in Tab. 2. In order to speed up the training process, our method is trained in three stages: 1) To encode the shape codes of the input shapes, PointNet E and implicit function f are first trained on sampled point-value pairs via Eqn. 7; 2) To enforce inverse implicit function g to recover 3D points from PEVs, E, f, and inverse function g are jointly trained via Eqn. 7 and 8; 3) To further facilitate the learned PEVs to be consistent for densely corresponded points from any two shapes, we jointly train E, f, and g with $\mathcal{L}^{all}$. In Stage 1, we adopt a progressive training technique [33] to train our implicit function on data with gradually increasing resolutions ($16^3 \rightarrow 32^3 \rightarrow 64^3$), which stabilizes and significantly speeds up the training process.

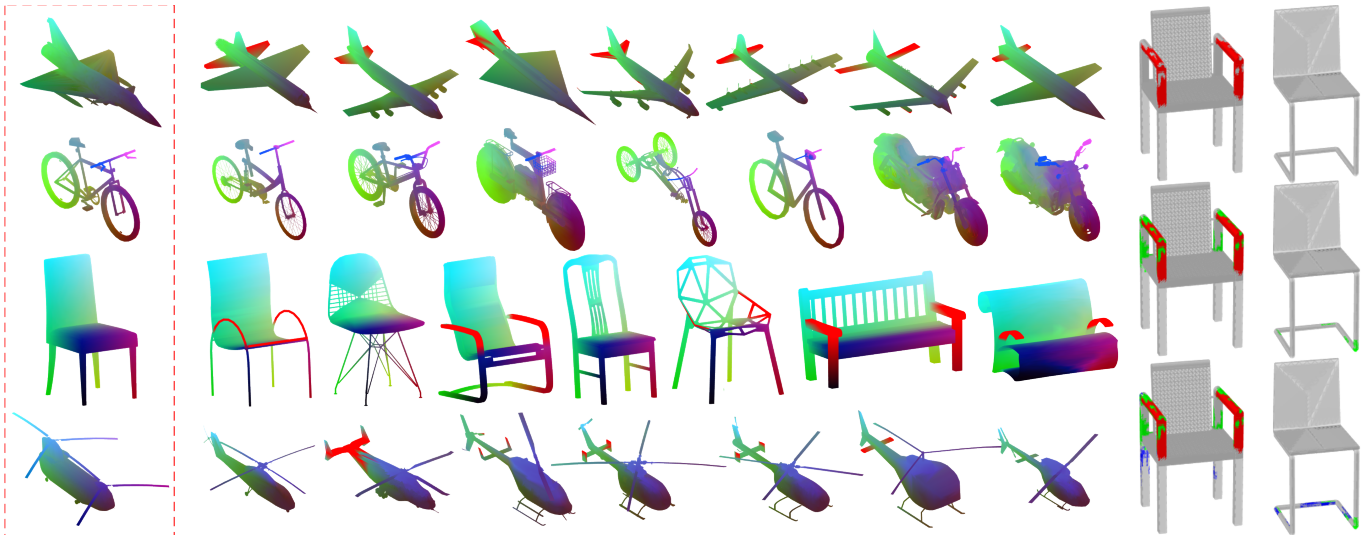
In experiments, we set $n = 8{,}192$, $d = 256$, $k = 12$, $\tau' = 0.2$, $\lambda_1 = 10$, $\lambda_2 = 1$, $\lambda_3 = 0.01$, $\lambda_4 = 0.1$. We implement our model in Pytorch and use Adam optimizer at a learning rate of 0.0001 in all three stages.

4 Experiments

4.1 3D Semantic Correspondence

Data. We evaluate our proposed algorithm on the task of 3D semantic point correspondence, a special case of dense correspondence, with two motivations: 1) no database of man-made objects has ground-truth dense correspondence; and 2) there is far less prior work in dense correspondence for man-made objects than the semantic correspondence task, which has strong baselines for comparison. Thus, to evaluate semantic correspondence, we train on ShapeNet [89] and test on BHCP [38] following the experimental protocol of [20], [21]. For training, we use a subset of ShapeNet including plane (500), bike (202), and chair (500) categories to train 3 individual models. For testing, BHCP provides ground-truth semantic points (7-13 per shape) of 404 shapes including plane (104), bike (100), chair (100), and helicopter (100). We generate all pairs of shapes for testing, *i.e.*, $9,900$ pairs for bikes. The helicopter category is tested with the plane model as [20], [21] did. As BHCP shapes are with rotations, prior works test on either one or both settings: aligned and unaligned, *i.e.*, 0° vs. arbitrary relative pose of two shapes. We evaluate both settings.

Baseline. We compare our work with multiple state-of-the-art (SoTA) baselines. Kim12 [16] and Kim13 [38] are traditional optimization methods that require part labels for templates and employ collection-wise co-analysis. LMVCNN [20], ShapeUnicode [39], AtlasNet2 [90], Chen *et al.* [21] and Liu *et al.* [35] are all learning based, where [20] require ground-truth correspondence labels for training. Despite [21] only estimates a fixed number of sparse points,



(a) (b)

Fig. 6: (a) Dense correspondences in 4 categories. Each row shows one target shape $\mathbf{S}_B$ (red box) and its pair-wise corresponded 6 source shapes $\mathbf{S}_A$. Given a spatially colored $\mathbf{S}_B$, the $p \rightarrow q$ correspondence enables to assign $p \in \mathbf{S}_A$ with the color of $q \in \mathbf{S}_B$, or with red if q is non-existing. (b) For one pair of shapes, the non-existence correspondences are impacted by the confidence threshold τ'. The colored regions progressively show the non-existence correspondences between the two shapes where the confidence score $\mathcal{C}$ is in the range of $[0, 0.3]$ (red), $(0.3, 0.5]$ (green), and $(0.5, 0.7]$ (blue).

[21] and ours are trained **without** labels. As optimization-based methods and [20] are designed for the unaligned setting, we also train a rotation-invariant version of ours by supervising E to predict additional rotation parameters, which is applied to rotate the input query point before feeding the point to f. In addition, we report results from two models: without uncertainty learning (**Ours**) and with uncertainty learning (**Ours-uncertainty**).

Results. The correspondence accuracy is measured by the fraction of correspondences whose error is below a given threshold of Euclidean distances. As in Fig. 5, the solid lines show the results on the aligned data and dotted lines on the unaligned data. We can clearly observe that our method outperforms baselines in the plane, bike, and chair categories on aligned data. Note that Kim13 [38] has slightly higher accuracy than ours on the helicopter category, likely due to the fact that [38] tests with the *helicopter-specific model*, while we test on the *unseen* helicopter category with a plane-specific model. At the distance threshold of 0.05, compared to our preliminary work [35], the **Ours** setting improves on average 5.3% accuracy relatively in 3 (Plane, Bike, and Chair) categories. While the **Ours-uncertainty** setting improves on average 9.2% accuracy, and achieves 9.6% improvement in the unseen Helicopter category. Moreover, compared to the best-performing SoTA baseline [21], our average relative improvement is 29.3% in 4 categories. We can clearly observe that both proposed deep branched implicit function and uncertainty learning can significantly improve the performance of semantic correspondence. *In the rest of experiments, we will use the model trained with **Ours-uncertainty** setting (unless specified otherwise).*

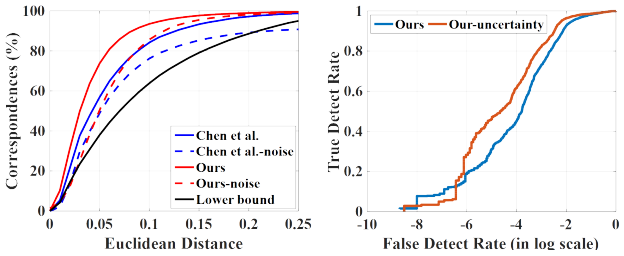
For unaligned data, both two settings achieve competitive performance as baselines. While it has the best AUC overall, it is worse at the threshold between $[0, 0.05]$. The main reason is the implicit network itself is sensitive to rotation. Note that this comparison shall be viewed in the context that most baselines use extra cues during training or inference, as well as the high inference speed of our learning-based approach. For example, Kim13 requires a part-based template during inference.

Some visual results of dense correspondences are shown in Fig. 6(a). Note the amount of non-existent correspondence is impacted by the threshold τ' as in Fig. 6(b). A larger τ' discovers more subtle non-existence correspondences. This is expected as the division of semantically corresponded or not can be blurred for some shape parts.

In the aligned setting, one naive approach to semantic correspondence is to find the closest point q on another 3D shape given a point p in one shape. We report the accuracy of this approach as the black curve in Fig. 7(a). Clearly, our accuracy is much higher than this "lower bound", indicating our method doesn't rely much on the canonical orientation. To further validate noisy real data, we evaluate the Chair category with additive noise $\mathcal{N}(0, 0.02^2)$ and compare with Chen *et al.* [21]. As shown in Fig. 7(a), the accuracy is slightly worse than testing on clean data. However, our method still outperforms the baseline on noisy data.

Visualization of Correspondence Confidence Score. To further visualize the correspondence confidence score, we provide confidence score maps for some examples. As shown in Fig. 8, the confidence score can show the probability around corresponded points between the target shape (red box) and its pair-wise source shapes. For example, for the source shapes with arms, we can clearly see the confidence scores of the arm part is significantly lower than other parts.

Detecting Non-Existence of Correspondences. Our method can build dense correspondences for 3D shapes



(a) (b)

Fig. 7: (a) Additional semantic correspondence results for the chair category in BHCP. (b) ROC curve of the non-existence of correspondence detection.

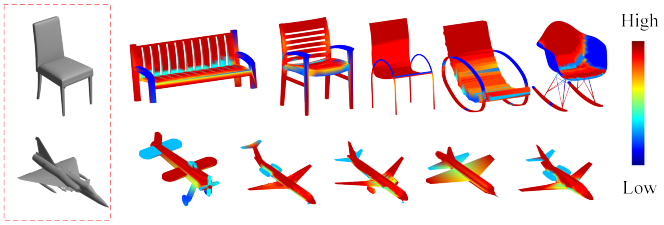


Fig. 8: Visualization of the confidence score. The confidence score maps show the probability around corresponded points between the target shape (red box) and its pair-wise source shapes.

with different topologies, and automatically declare the non-existence of correspondence. The experiment in Fig. 5 cannot fully depict this capability of our algorithm because no semantic point was annotated on a non-matching part. Also, there is no benchmark providing the non-existence label between a shape pair. We thus build a dataset with $1,000$ paired shapes from the chair category of ShapeNet part dataset [91]. Within a pair, one has the arm part while the other does not. For the former, we respectively annotate 5 arm points and 5 non-arm points based on provided part labels. We utilize this data to measure our detection of the non-existence of correspondence. Based on our confidence scores, we report the ROCs of both **Ours** and **Ours-uncertainty** (AUC: 96.58% vs 97.24%) in Fig. 7(b). The results show our strong capability in detecting unreliable correspondence.

4.2 Understand Uncertainty Learning

To better understand the impact of uncertainty on dense 3D shape correspondence, we provide the distribution of estimated uncertainty on BHCP 4 categories in Fig. 9. As can be seen, the uncertainty increases in the following order: planes $<$ bikes $<$ chairs $<$ helicopters. The estimated uncertainty is proportional to the complexity of the object's shape topology, which is intuitive and consistent with semantic correspondence results in Fig. 5. In addition, we visualize the point-wise uncertainty in shapes (Fig. 9). The estimated uncertainty clearly discovers the "hard" shape regions in the dense correspondence task, such as the chairs' legs and arms, and bikes' handlebars, which often suffer from large variations in geometric topology. Therefore, dense

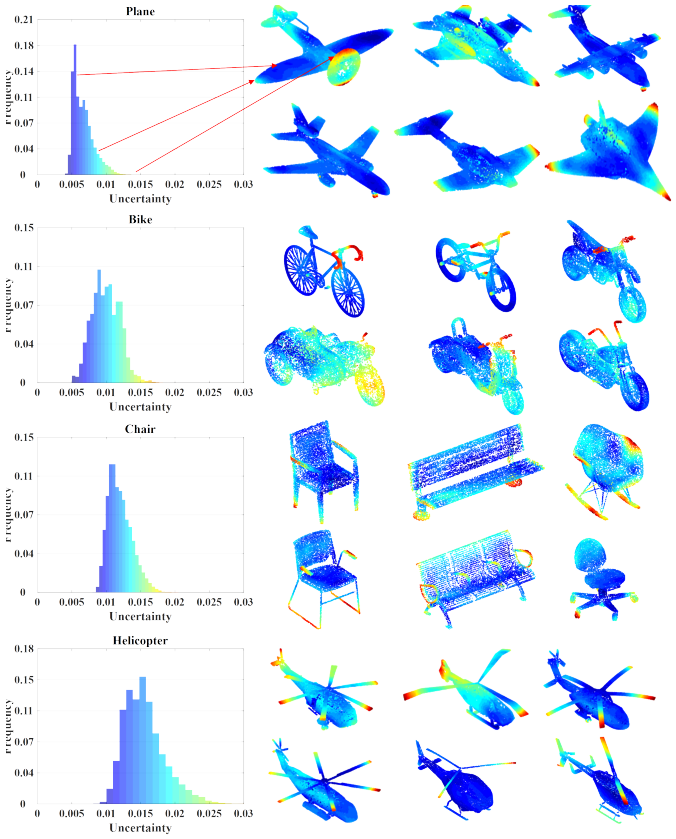


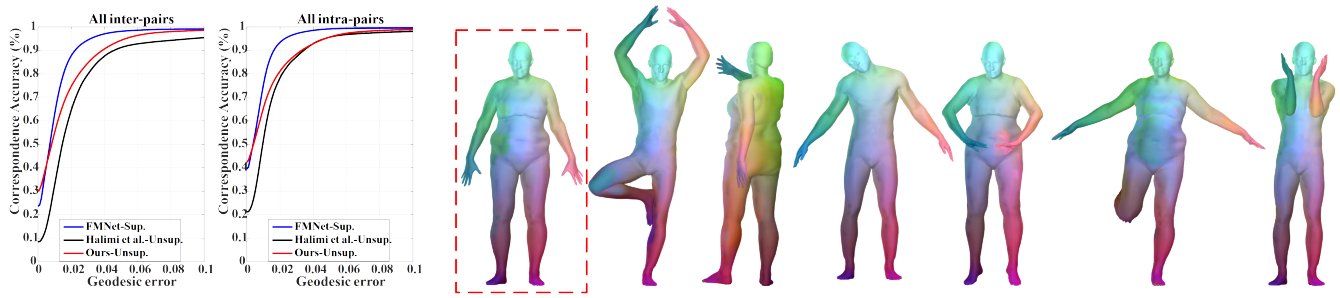
Fig. 9: Distribution of point-wise uncertainty in the semantic embedding on the BHCP 4 categories. Here, "uncertainty" refers to the mean value of $\mathbf{o}_\sigma^2$ across all feature dimensions. On the right-hand side, we show the uncertainty distribution by shapes. It can be observed that the learned uncertainty increases along with the shape regions with semantic ambiguity, *e.g.*, the arms and legs of chairs, which often differ among instances.

correspondence models with uncertainty learning have two benefits. First, the learned uncertainty can be utilized as a measurement of the complexity of objects' geometric topology in a shape collection. Second, the learned uncertainty can also be regarded as a "confidence indicator" to identify reliable established point-to-point correspondences.

4.3 Dense Correspondence on Human Body

Although our method is designed to handle challenging man-made or topology-varying objects, we choose to conduct additional experiments for organic shapes for two reasons. One is that datasets of organic shapes such as human bodies do provide annotations on *dense* correspondence. Thus evaluation of dense correspondence will complement well with our sparse semantic correspondence test in Sec. 4.1. The other is to evaluate the generalization capability of our method to diverse generic object types.

To this end, we evaluate the FAUST humans dataset [3], and compare with two representative SOTA baselines: supervised (FMNet [7]) and unsupervised (Halimi *et al.* [9]) methods. We follow the same dataset split as in [9] and [7] where the first 80 shapes of 8 subjects are used for training, and a validation set of 20 shapes of 2 other subjects



(a) (b)

Fig. 10: (a) Geodesic error comparison of intra-subject pairs and inter-subject pairs on the FAUST dataset [3], for three methods: ours (unsup.), FMNet [7] (sup.) and Halimi *et al.* [9] (unsup.) (b) One target shape and 6 pairwise corresponded source shapes.

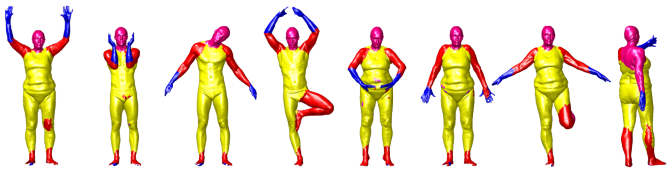


Fig. 11: Unsupervised segmentation results of the proposed method on 3D human shapes.

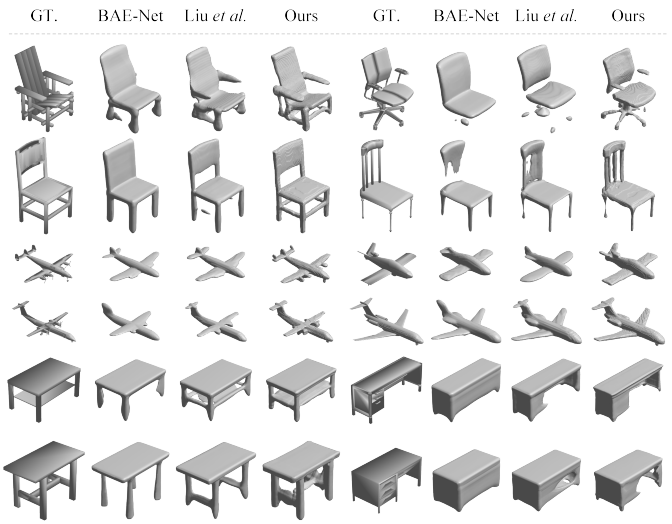


Fig. 12: Shape representation power comparison. Our reconstructions closely match the ground-truth (GT.) shapes than BAE-Net [32] and Liu *et al.* [35].

is utilized for testing. The ground-truth densely aligned shapes are used for evaluation. As shown in Fig. 10(a), our method, as an unsupervised method, outperforms the unsupervised method [9]. Fig. 10(b) visualizes the estimated correspondences. We also show the unsupervised segmentation results of some testing shapes in Fig. 11. As can be seen, meaningful and consistent segmentation appears across 3D human shapes.

4.4 Unsupervised Shape Segmentation

In order to produce 3D shape segmentation, prior template-based [38] or feature point estimation [21] correspondence methods usually need an additional part template to transfer pre-defined segmentation labels to the estimated corresponded points. However, in contrast to these methods, our framework is able to generate co-segmentation results in an unsupervised manner. For a fair comparison of shape segmentation, we only compare with the SoTA unsupervised method, BAE-Net [32], which is a *solely* optimized method for shape segmentation.

Following the same protocol [32], we train category-specific models and test on the same 8 categories of ShapeNet part dataset [91]: plane $(2,690)$, bag (76), cap (76), chair $(3,758)$, mug (184), skateboard (152), table $(5,271)$, and chair* (a joint chair+table set with $9,029$ shapes). Intersection over Union (IoU) between prediction and the ground truth is a common metric for segmentation. Since unsupervised segmentation is not guaranteed to produce the same part counts exactly as the ground truth, *e.g.*, combining the seat and back of a chair as one part, we report a modified IoU [32] measuring against both parts and part combinations in the ground-truth. As shown in Tab. 3, our model achieves a higher average segmentation accuracy than BAE-Net and on-par results with our preliminary work [35]. As BAE-Net is similar to the model of [35] trained in Stage 1, these results show that our dense correspondence task helps the PEV to better segment the shapes into parts, thus producing a more semantically meaningful embedding. Some visual results of segmentation are shown in Fig. 13.

4.5 Shape Representation Power of Implicit Function

We hope our novel implicit function f still serves as an effective shape representation while achieving dense correspondence. Hence its shape representation power shall be evaluated. Following the setting of unsupervised shape segmentation in Sec. 4.4, we first pass a ground-truth point set from the test set to E and extract the shape code $\mathbf{z}$. By feeding $\mathbf{z}$ and a grid of points to f, we can reconstruct the 3D shape by Marching Cubes [92]. We evaluate how well the reconstruction matches with the ground-truth point set. As shown in Tab. 3, the average Chamfer distance (CD-L_1) among branched implicit function (BAE-Net) [32], our preliminary work [35], and the proposed method on the 7 categories is 0.032, 0.024, and 0.017, respectively. Note that our relative improvement over our preliminary work [35]

Fig. 13: Qualitative results of our unsupervised segmentation in Tab. 3: 8 shapes in each of the 8 categories.

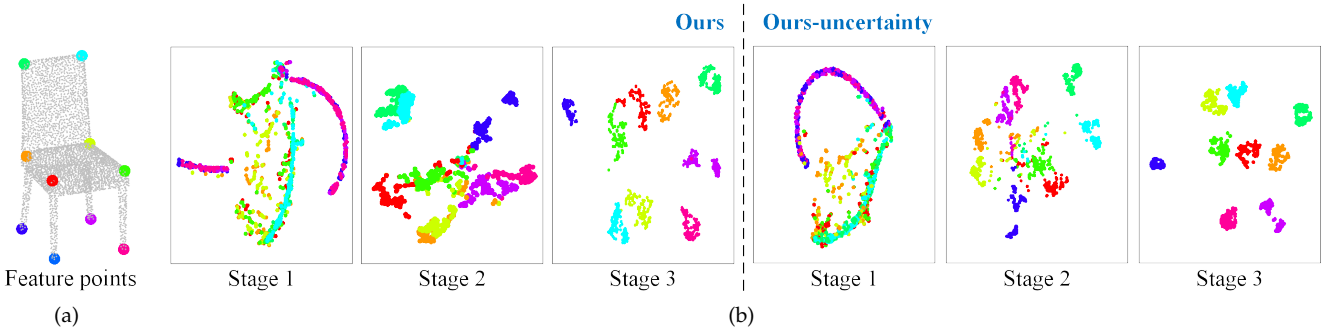


Fig. 14: (a) 10 semantic points overlaid with the shape. (b) The t-SNE comparison of the estimated PEVs over 3 training stages of models trained with the **Ours** and **Ours-uncertainty** settings. Points of the same color are the PEVs of ground-truth corresponding points in 100 chairs. 10 colors refer to the 10 points in (a).

TABLE 3: Unsupervised segmentation, shape representation comparisons (IoU/CD-L_1) on ShapeNet part. We use #parts in the evaluation and k=12 for all models.

Category (#parts)	plane (3)	bag (2)	cap (2)	chair (3)	chair* (4)	mug (2)	skateboard (2)	table (2)	Average
Segmented parts	body,tail, wing+engine	body, handle	panel, peak	back+seat, leg, arm	back, seat, leg, arm	body, handle	deck, wheel+bar	top, leg+support	
BAE-Net [32]	80.4/0.020	82.5/0.059	87.3/0.047	86.6/0.031	83.7/–	93.4/0.028	88.1/0.017	87.0/0.025	86.1/0.032
Liu *et al.* [35]	81.0/0.015	85.4/0.044	**87.9**/0.033	88.2/0.016	86.2/–	**94.7**/0.023	**91.6**/0.015	88.3/0.021	**88.0**/0.024
Ours	**82.7**/**0.009**	**85.7**/**0.035**	87.2/**0.021**	**88.6**/**0.013**	**86.9**/–	91.9/**0.014**	88.2/**0.011**	**88.7**/**0.016**	87.5/**0.017**

is 29%. This substantially lower CD shows that our novel design of semantic embedding and deep branched implicit function actually improves the shape representation power. It is understandable that the higher shape representation power is a prerequisite to more precise 3D correspondence, as shown in Fig. 5. Additionally, Fig. 12 shows the visual quality comparisons of the three categories' reconstructions.

4.6 Ablations and Analysis

4.6.1 Ablations Study

Loss Terms on Correspondence. Since the point occupancy loss and self-reconstruction loss are essential, we only ablate each term in the cross-reconstruction loss for the Chair category. Correspondence results in Fig. 15(a) demonstrate that, while all loss terms contribute to the final performance, $\mathcal{L}^{CD}$ and $\mathcal{L}^{smo}$ are the most crucial ones. $\mathcal{L}^{CD}$ forces $\mathbf{S}'_B$ to resemble $\mathbf{S}_B$. Without $\mathcal{L}^{smo}$, it is possible that $\mathbf{S}'_B$ may resemble $\mathbf{S}_B$ well, but with erroneous correspondences locally.

Part Embedding over Training Stages. The assumption of learned PEVs being similar for corresponding points motivates our algorithm design. To validate this assumption, we visualize the PEVs of 10 semantic points, defined in Fig. 14(a), with their ground-truth corresponding points across 100 chairs. The t-SNE visualizes the 100×10 k-dim PEVs in a 2D plot with one color per semantic point, after each training stage. The model after Stage 1 training resembles BAE-Net. As shown in Fig. 14(b), the 100 points corresponding to the same semantic point, *i.e.*, 2D points of the same color, scatter and overlap with other semantic (colored) points. With the inverse function and self-reconstruction loss in Stage 2, the part embedding shows a more promising grouping of colored points. Finally, the part embedding after Stage 3 has well-clustered and more discriminative grouping, which means points correspond-

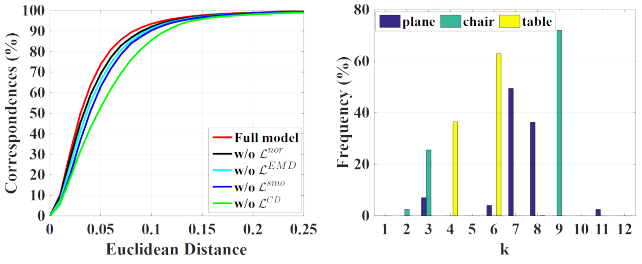


Fig. 15: (a) 3D semantic correspondence reflecting the contribution of our loss terms and (b) Active branch distribution of PEVs on three categories (plane, chair, and table). Each branch either represents a shape part or outputs nothing, *i.e.*, for the table category, No. 3 and 10 branches are active.

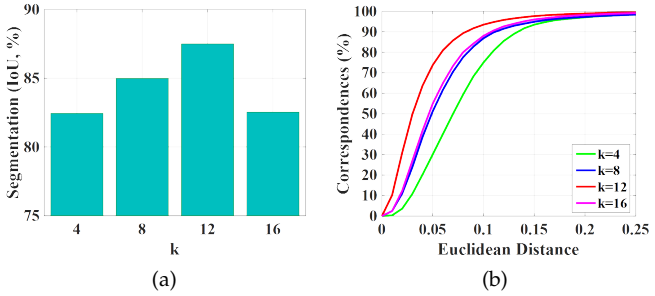


Fig. 16: (a) Shape segmentation and (b) 3D semantic correspondence performances on the Chair category over different dimensionalities of PEV (k).

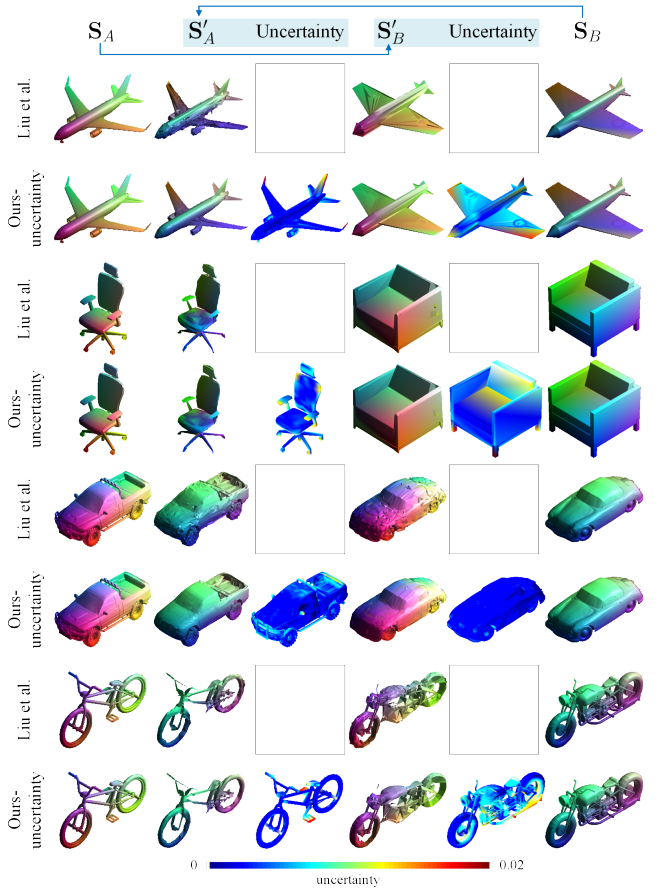


Fig. 17: Comparison of cross 3D reconstruction ($\mathbf{S}_A \rightarrow \mathbf{S}'_B$, $\mathbf{S}_B \rightarrow \mathbf{S}'_A$) between Liu *et al.* [35] and our method. Corresponding points are assigned the same color in ($\mathbf{S}_A, \mathbf{S}'_B$) and ($\mathbf{S}_B, \mathbf{S}'_A$). For our method, we also show the estimated uncertainty, which visualizes the learned variances of cross-reconstructed shapes. Best viewed in zoom-in.

ing to the same semantic location do have similar PEVs. The improvement trend of part embedding across 3 stages shows the effectiveness of our loss design and training scheme, as well as validates the key assumption that motivated our algorithm. In addition, we compare the t-SNE plot between the **Ours** and **Ours-uncertainty** in Fig. 14(b). Here, for the **Ours-uncertainty**, we apply t-SNE with the mean PEVs ($\mathbf{o}_\mu$), As can be seen, the part embedding of the **Ours-uncertainty** has highly well-clustered than the **Ours** (Stage 3). It demonstrates that uncertainty learning can further improve the intra-class compactness and inter-class separability in semantic embedding.

Dimensionality of PEV. As mentioned in Sec. 3.3, in the final output layer, we utilize a max-pooling operator ($\mathcal{MP}$) to select one branch output and form the final occupancy value. Here, we denote the selected branch as an "active" branch and Fig. 15(b) shows its distribution of PEVs with $k = 12$. As can be observed, the active branch distribution across different categories is random and only a small part of branches is active, which implies that shape segmentation might not require a high-dimensional PEV. To verify this, we conduct experiments on the dimensionality of PEV. Fig. 16(a) and 16(b) show the shape segmentation and semantic correspondence results over the dimensionality of PEV. Our algorithm performs the best in both when $k = 12$.

One-hot vs. Continuous Embedding. Ideally, our implicit function, adopted from BAE-Net [32], should output a one-hot vector before $\mathcal{MP}$, which would benefit unsupervised segmentation the most. In contrast, our PEVs prefer continuous embedding rather than one-hot. To better understand PEV, we compute the statistics of Cosine Similarity (CS) between the PEVs and their corresponding one-hot vectors: 0.972 ± 0.020 (BAE-Net) vs. 0.966 ± 0.040 (ours). This shows our learned PEVs are *approximately* one-hot vectors. Compared to BAE-Net, our smaller CS and larger variance are likely due to the limited network capability, as well as our desire to learn a *continuous* embedding benefiting correspondence.

4.6.2 *Expressiveness of Inverse Implicit Function*

Given our inverse implicit function, we are able to cross-reconstruct each other between two paired shapes by swapping their part embedding vectors. Further, we can interpolate shapes both in learned semantic embedding space and maintain the point-level correspondence consistently.

Cross-Reconstruction Performance. We first show the cross-reconstruction performances in Fig. 17. From a shape collection, we can randomly select two shapes $\mathbf{S}_A$ and $\mathbf{S}_B$. Their shape codes $\mathbf{z}_A$ and $\mathbf{z}_B$ can be predicted by the PointNet encoder. With their respectively generated the

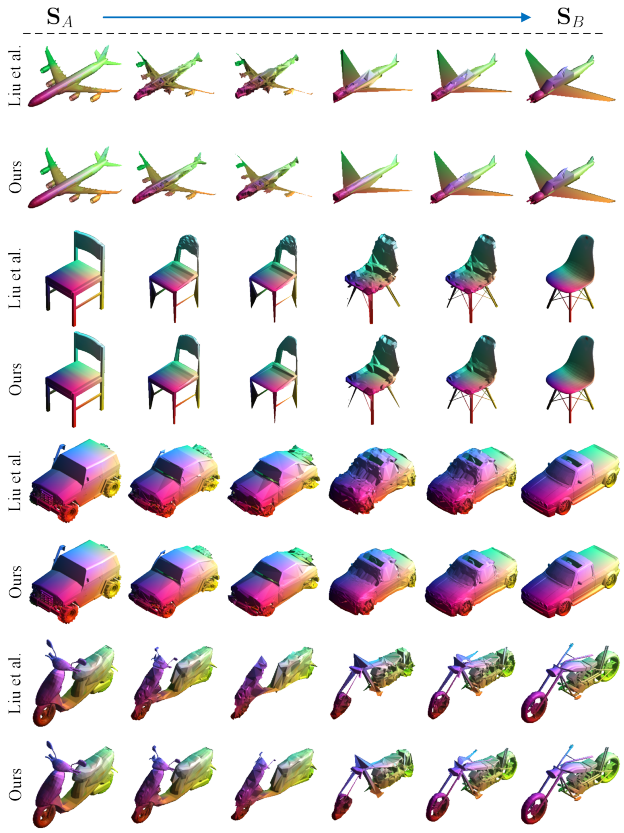


Fig. 18: Comparison of interpolation in the semantic embedding space between Liu *et al.* [35] and our method. Our interpolations are more smooth and point-to-point consistent than Liu *et al.* [35]. Best viewed in zoom-in.

mean of PEVs $\mathbf{o}_{\mu A}$ and $\mathbf{o}_{\mu B}$, we swap their PEVs, send the concatenated vectors to the inverse function, and obtain $\mathbf{S}'_A = g(\mathbf{o}_{\mu B}, \mathbf{z}_A), \mathbf{S}'_B = g(\mathbf{o}_{\mu A}, \mathbf{z}_B)$. As compared to our preliminary work [35] in Fig. 17, our cross reconstructions are more closely resemble each other in detail. Additionally, this work can produce uncertainty, which can reveal the reliability of the cross reconstructions. Here, we provide the cross-reconstruction performance of the car category, where the car-specific model is trained on 659 shapes of the ShapeNet Part database.

Interpolation in the Semantic Embedding Space. An alternative way to explore the correspondence ability is to evaluate the interpolation capability of the inverse implicit function. In this experiment, we interpolate shapes in the latent space. Given two shapes $\mathbf{S}_A$ and $\mathbf{S}_B$, we first obtain their $\mathbf{z}_A$, $\mathbf{z}_B$, $\mathbf{o}_{\mu A}$, and $\mathbf{o}_{\mu B}$ by the trained encoder, implicit and inverse implicit functions. The intermediate shape code can be calculated as $\tilde{\mathbf{z}} = \alpha \mathbf{z}_A + (1-\alpha)\mathbf{z}_B$ ($\alpha \in [0,1]$), and then we send the concatenated vectors ($\tilde{\mathbf{z}}$ and $\mathbf{o}_{\mu A}$) to the inverse function to generate an intermediate cross-reconstructed shape $\tilde{\mathbf{S}}$. Since $\mathbf{S}_A$ and $\tilde{\mathbf{S}}$ are point-to-point corresponded, we can easily show the correspondences in the same color. As observed in Fig. 18, our inverse implicit function generalizes well the different shape deformations. Moreover, our interpolations are more smooth and point-to-point consistent than our preliminary work [35]. It also demonstrates that the proposed deep branched implicit function and uncertainty learning enhance the discriminative ability of the learned semantic part embedding among different parts of the shape.

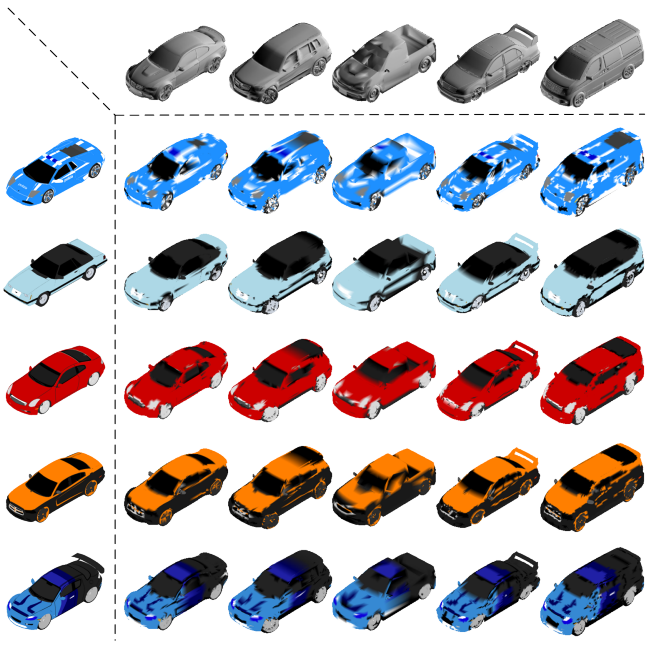


Fig. 19: Texture transfer from the source shapes (1^{st} column) to the target shapes (1^{st} row) based on the dense correspondences estimated by the proposed method.

Texture Transfer. As shown in Fig. 19, based on the correspondences generated by our method, we are able to transfer textures from one shape to another. As can be observed, the texture can be semantically transferred to the correct places in new shapes.

4.6.3 *Computation Time*

Our training on one category (500 samples) takes ~ 8 hours to converge with a GTX1080Ti GPU, where 1.5, 2, and 8 hours are spent at Stage 1, 2, 3 respectively. In inference, the average runtime to pair two shapes (n=8,192) is 0.21 second including the runtimes of E, f, g networks (on GPU, $\sim 2.1ms$), and neighbor search, confidence calculation (on CPU, $\sim$ $208ms$). The inference time is similar to our preliminary work [35] since the deep branched implicit function network only brings an additional $0.6ms$ time cost.

5 Conclusion

In this work, we propose a novel framework including an implicit function and its inverse for dense 3D shape correspondences of topology-varying generic objects. Based on the learned probabilistic semantic part embedding via our implicit function, dense correspondence is established via the inverse function mapping from the part embedding to the corresponding 3D point. In addition, our algorithm can automatically calculate a confidence score measuring the probability of correspondence, which is desirable for generic objects with large topological variations. The comprehensive experimental results show the superiority of the proposed method in unsupervised shape correspondence and segmentation.

REFERENCES

[1] V. Blanz and T. Vetter, "Face recognition based on fitting a 3D morphable model," *TPAMI*, 2003.

[2] S. Zuffi, A. Kanazawa, D. W. Jacobs, and M. J. Black, "3D menagerie: Modeling the 3D shape and pose of animals," in *CVPR*, 2017.

[3] F. Bogo, J. Romero, M. Loper, and M. J. Black, "FAUST: Dataset and evaluation for 3D mesh registration," in *CVPR*, 2014.

[4] V. Kraevoy, A. Sheffer, and C. Gotsman, "Matchmaker: constructing constrained texture maps," *TOG*, 2003.

[5] M. Niemeyer, L. Mescheder, M. Oechsle, and A. Geiger, "Occupancy flow: 4D reconstruction by learning particle dynamics," in *ICCV*, 2019.

[6] M. Ovsjanikov, M. Ben-Chen, J. Solomon, A. Butscher, and L. Guibas, "Functional maps: a flexible representation of maps between shapes," *TOG*, 2012.

[7] O. Litany, T. Remez, E. Rodolà, A. Bronstein, and M. Bronstein, "Deep functional maps: Structured prediction for dense shape correspondence," in *ICCV*, 2017.

[8] T. Groueix, M. Fisher, V. G. Kim, B. C. Russell, and M. Aubry, "3D-CODED: 3D correspondences by deep deformation," in *ECCV*, 2018.

[9] O. Halimi, O. Litany, E. Rodola, A. M. Bronstein, and R. Kimmel, "Unsupervised learning of dense shape correspondence," in *CVPR*, 2019.

[10] J.-M. Roufosse, A. Sharma, and M. Ovsjanikov, "Unsupervised deep learning for structured shape matching," in *ICCV*, 2019.

[11] S. C. Lee and M. Kazhdan, "Dense point-to-point correspondences between genus-zero shapes," in *Computer Graphics Forum*, 2019.

[12] F. Steinke, V. Blanz, and B. Schölkopf, "Learning dense 3D correspondence," in *NeurIPS*, 2007.

[13] F. Liu, L. Tran, and X. Liu, "3D face modeling from diverse raw scan data," in *ICCV*, 2019.

[14] Q. Huang, F. Wang, and L. Guibas, "Functional map networks for analyzing and exploring large shape collections," *TOG*, 2014.

[15] O. Van Kaick, H. Zhang, G. Hamarneh, and D. Cohen-Or, "A survey on shape correspondence," in *Computer Graphics Forum*, 2011.

[16] V. G. Kim, W. Li, N. J. Mitra, S. DiVerdi, and T. Funkhouser, "Exploring collections of 3D models using fuzzy correspondences," *TOG*, 2012.

[17] J. Solomon, A. Nguyen, A. Butscher, M. Ben-Chen, and L. Guibas, "Soft maps between surfaces," in *Computer Graphics Forum*, 2012.

[18] O. Sidi, O. van Kaick, Y. Kleiman, H. Zhang, and D. Cohen-Or, "Unsupervised co-segmentation of a set of shapes via descriptor-space spectral clustering," in *SIGGRAPH Asia*, 2011.

[19] I. Alhashim, K. Xu, Y. Zhuang, J. Cao, P. Simari, and H. Zhang, "Deformation-driven topology-varying 3D shape correspondence," *TOG*, 2015.

[20] H. Huang, E. Kalogerakis, S. Chaudhuri, D. Ceylan, V. G. Kim, and E. Yumer, "Learning local shape descriptors from part correspondences with multiview convolutional networks," *TOG*, 2017.

[21] N. Chen, L. Liu, Z. Cui, R. Chen, D. Ceylan, C. Tu, and W. Wang, "Unsupervised learning of intrinsic structural representation points," in *CVPR*, 2020.

[22] P. Achlioptas, O. Diamanti, I. Mitliagkas, and L. Guibas, "Learning representations and generative models for 3D point clouds," in *ICML*, 2018.

[23] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3D classification and segmentation," in *CVPR*, 2017.

[24] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," in *NeurIPS*, 2017.

[25] T. Groueix, M. Fisher, V. G. Kim, B. C. Russell, and M. Aubry, "Atlasnet: A papier-mâché approach to learning 3D surface generation," in *CVPR*, 2018.

[26] J. J. Georgia Gkioxari, Jitendra Malik, "Mesh R-CNN," in *ICCV*, 2019.

[27] N. Wang, Y. Zhang, Z. Li, Y. Fu, W. Liu, and Y.-G. Jiang, "Pixel2mesh: Generating 3D mesh models from single RGB images," in *ECCV*, 2018.

[28] J. J. Park, P. Florence, J. Straub, R. Newcombe, and S. Lovegrove, "DeepSDF: Learning continuous signed distance functions for shape representation," in *CVPR*, 2019.

[29] L. Mescheder, M. Oechsle, M. Niemeyer, S. Nowozin, and A. Geiger, "Occupancy networks: Learning 3D reconstruction in function space," in *CVPR*, 2019.

[30] S. Liu, S. Saito, W. Chen, and H. Li, "Learning to infer implicit surfaces without 3D supervision," in *NeurIPS*, 2019.

[31] S. Saito, Z. Huang, R. Natsume, S. Morishima, A. Kanazawa, and H. Li, "PIFu: Pixel-aligned implicit function for high-resolution clothed human digitization," in *ICCV*, 2019.

[32] Z. Chen, K. Yin, M. Fisher, S. Chaudhuri, and H. Zhang, "BAE-NET: branched autoencoder for shape co-segmentation," in *ICCV*, 2019.

[33] Z. Chen and H. Zhang, "Learning implicit fields for generative shape modeling," in *CVPR*, 2019.

[34] M. Atzmon, N. Haim, L. Yariv, O. Israelov, H. Maron, and Y. Lipman, "Controlling neural level sets," in *NeurIPS*, 2019.

[35] F. Liu and X. Liu, "Learning implicit functions for topology-varying dense 3D shape correspondence," in *NeurIPS*, 2020.

[36] M. Slavcheva, M. Baust, and S. Ilic, "Towards implicit correspondence in signed distance field evolution," in *ICCV*, 2017.

[37] B. L. Bhatnagar, C. Sminchisescu, C. Theobalt, and G. Pons-Moll, "Loopreg: Self-supervised learning of implicit surface correspondences, pose and shape for 3D human mesh registration," in *NeurIPS*, 2020.

[38] V. G. Kim, W. Li, N. J. Mitra, S. Chaudhuri, S. DiVerdi, and T. Funkhouser, "Learning part-based templates from large collections of 3D shapes," *TOG*, 2013.

[39] S. Muralikrishnan, V. G. Kim, M. Fisher, and S. Chaudhuri, "Shape unicode: A unified shape representation," in *CVPR*, 2019.

[40] Y. Deng, J. Yang, and X. Tong, "Deformed implicit field: Modeling 3D shapes with learned dense correspondence," *arXiv preprint arXiv:2011.13650*, 2020.

[41] Z. Zheng, T. Yu, Q. Dai, and Y. Liu, "Deep implicit templates for 3D shape representation," *arXiv preprint arXiv:2011.14565*, 2020.

[42] D. Boscaini, J. Masci, E. Rodolà, and M. Bronstein, "Learning shape correspondence with anisotropic convolutional neural networks," in *NeurIPS*, 2016.

[43] M. Bahri, E. O. Sullivan, S. Gong, F. Liu, X. Liu, M. Bronstein, and S. Zafeiriou, "Shape my face: Registering 3d face scans by surface-to-surface translation," *IJCV*, vol. 129, pp. 2680–2713, June 2021.

[44] E. Kalogerakis, A. Hertzmann, and K. Singh, "Learning 3D mesh segmentation and labeling," in *SIGGRAPH*, 2010.

[45] Q. Huang, V. Koltun, and L. Guibas, "Joint shape segmentation with linear programming," in *SIGGRAPH Asia*, 2011.

[46] C. Zhu, R. Yi, W. Lira, I. Alhashim, K. Xu, and H. Zhang, "Deformation-driven shape correspondence via shape recognition," *TOG*, 2017.

[47] H. Huang, E. Kalogerakis, and B. Marlin, "Analysis and synthesis of 3D shape families via deep-learned generative models of surfaces," in *Computer Graphics Forum*, 2015.

[48] L. Yi, H. Su, X. Guo, and L. J. Guibas, "SyncspecCNN: Synchronized spectral CNN for 3D shape segmentation," in *CVPR*, 2017.

[49] M. Sung, H. Su, R. Yu, and L. J. Guibas, "Deep functional dictionaries: Learning consistent semantic structures on 3D models from functions," in *NeurIPS*, 2018.

[50] Y. You, Y. Lou, C. Li, Z. Cheng, L. Li, L. Ma, C. Lu, and W. Wang, "Keypointnet: A large-scale 3D keypoint dataset aggregated from numerous human annotations," in *CVPR*, 2020.

[51] T. Groueix, M. Fisher, V. G. Kim, B. C. Russell, and M. Aubry, "Unsupervised cycle-consistent deformation for shape matching," in *Computer Graphics Forum*, 2019.

[52] M. Oechsle, L. Mescheder, M. Niemeyer, T. Strauss, and A. Geiger, "Texture fields: Learning texture representations in function space," in *ICCV*, 2019.

[53] V. Sitzmann, M. Zollhöfer, and G. Wetzstein, "Scene representation networks: Continuous 3D-structure-aware neural scene representations," in *NeurIPS*, 2019.

[54] F. Liu, L. Tran, and X. Liu, "Fully understanding generic objects: Modeling, segmentation, and reconstruction," in *CVPR*, 2021.

[55] F. Liu and X. Liu, "Voxel-based 3d detection and reconstruction of multiple objects from a single image," in *NeurIPS*, 2021.

[56] ——, "2d GANs meet unsupervised single-view 3d reconstruction," in *ECCV*, 2022.

[57] K. Genova, F. Cole, D. Vlasic, A. Sarna, W. T. Freeman, and T. Funkhouser, "Learning shape templates with structured implicit functions," in *ICCV*, 2019.

[58] K. Genova, F. Cole, A. Sud, A. Sarna, and T. Funkhouser, "Local deep implicit functions for 3D shape," in *CVPR*, 2020.

[59] D. Paschalidou, L. van Gool, and A. Geiger, "Learning unsupervised hierarchical part decomposition of 3D objects from a single rgb image," in *CVPR*, 2020.
[60] X. Huang, S. Zhang, Y. Wang, D. Metaxas, and D. Samaras, "A hierarchical framework for high resolution facial expression tracking," in *CVPRW*, 2004.
[61] X. Huang, N. Paragios, and D. N. Metaxas, "Shape registration in implicit spaces using information theory and free form deformations," *TPAMI*, 2006.
[62] T. Yenamandra, A. Tewari, F. Bernard, H.-P. Seidel, M. Elgharib, D. Cremers, and C. Theobalt, "i3DMM: Deep implicit 3D morphable model of human heads," *arXiv preprint arXiv:2011.14143*, 2020.
[63] Q. Xu, W. Wang, D. Ceylan, R. Mech, and U. Neumann, "DISN: Deep implicit surface network for high-quality single-view 3D reconstruction," in *NeurIPS*, 2019.
[64] J. Chibane, T. Alldieck, and G. Pons-Moll, "Implicit functions in feature space for 3D shape reconstruction and completion," in *CVPR*, 2020.
[65] S. Peng, M. Niemeyer, L. Mescheder, M. Pollefeys, and A. Geiger, "Convolutional occupancy networks," in *ECCV*, 2020.
[66] C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra, "Weight uncertainty in neural network," in *ICML*, 2015.
[67] A. Kendall and Y. Gal, "What uncertainties do we need in bayesian deep learning for computer vision?" in *NeurIPS*, 2017.
[68] A. Malinin and M. Gales, "Predictive uncertainty estimation via prior networks," in *NeurIPS*, 2018.
[69] J. Van Amersfoort, L. Smith, Y. W. Teh, and Y. Gal, "Uncertainty estimation using a single deep deterministic neural network," in *ICML*, 2020.
[70] R. M. Neal, *Bayesian learning for neural networks*. Springer Science & Business Media, 2012, vol. 118.
[71] Y. Gal and Z. Ghahramani, "Dropout as a bayesian approximation: Representing model uncertainty in deep learning," in *ICML*, 2016.
[72] A. Kendall, V. Badrinarayanan, and R. Cipolla, "Bayesian segnet: Model uncertainty in deep convolutional encoder-decoder architectures for scene understanding," in *BMVC*, 2015.
[73] P.-Y. Huang, W.-T. Hsu, C.-Y. Chiu, T.-F. Wu, and M. Sun, "Efficient uncertainty estimation for semantic segmentation in videos," in *ECCV*, 2018.
[74] M. Poggi, F. Aleotti, F. Tosi, and S. Mattoccia, "On the uncertainty of self-supervised monocular depth estimation," in *CVPR*, 2020.
[75] T. Ke, T. Do, K. Vuong, K. Sartipi, and S. I. Roumeliotis, "Deep multi-view depth estimation with predicted uncertainty," in *ICRA*, 2021.
[76] S. Imran, X. Liu, and D. Morris, "Depth completion with twin-surface extrapolation at occlusion boundaries," in *CVPR*, 2021.
[77] Y. Long, D. Morris, X. Liu, M. P. G. Castro, P. Chakravarty, and P. Narayanan, "Radar-camera pixel depth association for depth completion," in *In Proceeding of IEEE Computer Vision and Pattern Recognition*, Nashville, TN, June 2021.
[78] H. Xu, Z. Zhou, Y. Wang, W. Kang, B. Sun, H. Li, and Y. Qiao, "Digging into uncertainty in self-supervised multi-view stereo," in *ICCV*, 2021.
[79] P. Truong, M. Danelljan, L. Van Gool, and R. Timofte, "Learning accurate dense correspondences and when to trust them," in *CVPR*, 2021.
[80] A. Kumar*, T. K. Marks*, W. Mou*, Y. Wang, M. Jones, A. Cherian, T. Koike-Akino, X. Liu, and C. Feng, "Luvli face alignment: Estimating landmarks' location, uncertainty, and visibility likelihood," in *CVPR*, 2020.
[81] Y. Shi and A. K. Jain, "Probabilistic face embeddings," in *ICCV*, 2019.
[82] J. Chang, Z. Lan, C. Cheng, and Y. Wei, "Data uncertainty learning in face recognition," in *CVPR*, 2020.
[83] M. Kim, A. K. Jain, and X. Liu, "Adaface: Quality adaptive margin for face recognition," in *CVPR*, 2022.
[84] L. Yi, H. Huang, D. Liu, E. Kalogerakis, H. Su, and L. Guibas, "Deep part induction from articulated object pairs," in *SIGGRAPH Asia 2018 Technical Papers*, 2018.
[85] S. Tulsiani, H. Su, L. J. Guibas, A. A. Efros, and J. Malik, "Learning shape abstractions by assembling volumetric primitives," in *CVPR*, 2017.
[86] Z. Shu, C. Qi, S. Xin, C. Hu, L. Wang, Y. Zhang, and L. Liu, "Unsupervised 3D shape segmentation and co-segmentation via deep learning," *CAGD*, 2016.
[87] K. Xu, H. Li, H. Zhang, D. Cohen-Or, Y. Xiong, and Z.-Q. Cheng, "Style-content separation by anisotropic part scales," *TOG*, 2010.
[88] C. Häne, S. Tulsiani, and J. Malik, "Hierarchical surface prediction for 3D object reconstruction," in *3DV*, 2017.
[89] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, J. Xiao, L. Yi, and F. Yu, "ShapeNet: An information-rich 3D model repository," *arXiv preprint arXiv:1512.03012*, 2015.
[90] T. Deprelle, T. Groueix, M. Fisher, V. Kim, B. Russell, and M. Aubry, "Learning elementary structures for 3D shape generation and matching," in *NeurIPS*, 2019.
[91] L. Yi, V. G. Kim, D. Ceylan, I.-C. Shen, M. Yan, H. Su, C. Lu, Q. Huang, A. Sheffer, and L. Guibas, "A scalable active framework for region annotation in 3D shape collections," *TOG*, 2016.
[92] W. E. Lorensen and H. E. Cline, "Marching cubes: A high resolution 3D surface construction algorithm," *ACM siggraph computer graphics*, vol. 21, no. 4, pp. 163–169, 1987.

Feng Liu is currently a post-doc researcher in the Computer Vision Lab at Michigan State University. He received the Ph.D. degree in Computer Science from Sichuan University, China in 2018. His main research interests span the areas of joint analysis of 2D images and 3D shapes, including 3D modeling, semantic correspondence, and coherent 3D scene reconstruction. He is a member of the IEEE.

Xiaoming Liu is a Anil K. and Nandita Jain Endowed Professor and MSU Foundation Professor at the Department of Computer Science and Engineering of Michigan State University. He received the Ph.D. degree in Electrical and Computer Engineering from Carnegie Mellon University in 2004. Before joining MSU in Fall 2012, he was a research scientist at General Electric (GE) Global Research. His research interests include computer vision, machine learning, and biometrics. As a co-author, he is a recipient of Best Industry Related Paper Award runner-up at ICPR 2014, Best Student Paper Award at WACV 2012 and 2014, Best Poster Award at BMVC 2015, and Michigan State University College of Engineering Withrow Endowed Distinguished Scholar Award. He has been the Area Chair for numerous conferences, including CVPR, ICCV, ECCV, ICLR, NeurIPS, the Program CO-Chair of WACV'18, BTAS'18, IJCB'22, AVSS'22 conferences, and General Co-Chair of FG'23 conference. He is an Associate Editor of Pattern Recognition and IEEE Transactions on Image Processing. He has authored more than 150 scientific publications, and has filed 29 U.S. patents. He is a fellow of IEEE and IAPR.