Fine-grained Object Semantic Understanding from Correspondences

Yang You[1*] Chengkun Li[1*] Yujing Lou[1*] Zhoujun Cheng[1] Liangwei Li[1] Lizhuang Ma[1]
Weiming Wang[1] Cewu Lu[1†]

[1]Shanghai Jiao Tong University, China

Abstract

*Fine-grained semantic understanding of 3D objects is crucial in many applications such as object manipulation. However, it is hard to give a universal definition of point-level semantics that everyone would agree on. We observe that people are pretty sure about semantic correspondences between two areas from different objects, but less certain about what each area means in semantics. Therefore, we argue that by providing human labeled correspondences between different objects from the same category, one can recover rich semantic information of an object. In this paper, we propose a method that outputs dense semantic embeddings based on a novel geodesic consistency loss. Accordingly, a new dataset named **CorresPondenceNet** and its corresponding benchmark are designed. Several state-of-the-art networks are evaluated based on our proposed method. We show that our method could boost the fine-grained understanding of heterogeneous objects and the inference of dense semantic information is possible.*

1. Introduction

Object understanding [23, 32, 48] is one of the holy grails in computer vision. Being able to fully understand object semantics is crucial for various applications such as self-driving [7, 34] and attribute transfer [26]. Recently, significant advances have been made in both category-level and instance-level understanding of objects [9, 22]. However, having category-level or instance-level knowledge of objects is far from enough for fine-grained tasks such as object manipulation [25, 30]. Fine-grained semantic understanding of objects is of great importance and still remains challenging.

One of the key problems with fine-grained semantic understanding lies in the ambiguous definitions of semantics. In the past decades, researchers have proposed key-points [24, 27, 38, 41] and skeletons [4] to explicitly define object semantics. These methods have made success in tasks like human body parsing [19], however, it is hard or even impossible to give consistent definitions of key-points or skeletons for a general object. Recently, part based representations of objects are also adopted by researchers [9, 47, 32], where an object is decomposed into semantic parts by experts, with a predefined semantic label on each part. The above methods all impose an explicit definition of object semantics, which is inevitably biased or flawed since different people may hold different opinions of what the semantics of an object are.

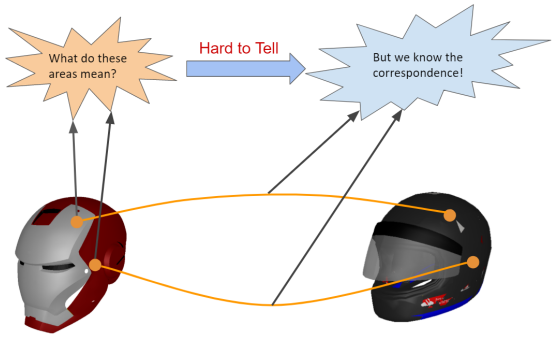

Figure 1. We observe that it is hard to tell the exact meanings of some areas on an object, while correspondences between different objects are clear.

In this paper, we explore a brand new way to deal with this vagueness in fine-grained object understanding. Instead of explicitly giving semantic components and labels, we leverage the semantic correspondence between objects to implicitly infer their semantic meanings. This is based on the observation that while it is hard to tell the exact meanings of some sub-object areas, almost everyone would agree on their semantic correspondence across different objects, as shown in Figure 1. Consequently, comprehensive object understanding can be achieved by collecting multiple unambiguous semantic correspondences from a large population.

To that end, we introduce **C**orres**P**ondence**Net** (CPNet):

*Contributed equally.

†Cewu Lu is the corresponding author: lucewu@sjtu.edu.cn

a *diverse* and *high-quality* dataset on top of ShapeNet [9] with *cross-object*, *point-level* fine-grained 3D semantic correspondence annotations. In this dataset, every annotator gives multiple sets of semantic-consistent points across different intra-class objects, which we call "correspondence sets", as shown in Figure 2.

Using these correspondence sets, we aim to obtain the pointwise embeddings of an object to represent its fine-grained semantic information. We propose a novel method to learn these embeddings. While a simple push-pull loss fails to generate meaningful embeddings, we leverage a geodesic consistency loss. On one hand, points in the same correspondence set get pulled in the embedding space. On the other hand, points across different correspondence sets get pushed according to their average geodesic distances. By considering geodesic relationships between different correspondence sets, points with similar semantics are more likely to be grouped together in the embedding space.

In summary, our key contributions are as follows:

- We explore a new way towards fine-grained semantic understanding of objects, where explicit definitions are avoided but point-level semantic correspondences across heterogeneous objects are leveraged.
- We introduce *CPNet*, the first correspondence based dataset for 3D object understanding, which contains 100K+ high-quality semantic-consistent points.
- We design a novel geodesic consistency loss to learn dense embeddings. To evaluate these embeddings, a brand-new semantic understanding benchmark — semantic correspondence estimation, is proposed. A variety of state-of-the-art neural networks are evaluated.

2. Related Work

Datasets on Semantic Analysis Big data and deep learning have witnessed several large 2D/3D datasets these years aiming to parse semantic information from objects. In the world of 2D images, SPAIR-71k [31] proposes a large-scale dataset with rich annotations on viewpoints, keypoints and segmentations, which is mainly used for semantic matching between different images. Recently, Ham et al. [17] and Taniai et al. [42] have introduced datasets with groundtruth correspondences. Since then, PF-WILLOW and PF-PASCAL [17] have been used for evaluation in many works. In addition, plenty of datasets on human pose analysis [3, 2] have been proposed recently. These 2D image datasets have their advantages in that they are relatively large and pertain diversity across different scenes and objects.

On the other hand, there exists a rich set of 3D model datasets that try to directly process meshes or point clouds. There are generally two types of them: ones that focus on rigid models and some others that focus on non-rigid models. For rigid model analysis, ShapeNet Core 55 [9] is proposed to help object-level classification while ShapeNet part dataset [47] pushes it one step forward with intra-object part classification. As a followup, PartNet [32] comes up with a much more complete and manually defined hierarchical structures of parts. Alternatively, dataset proposed by Dutagaci et al. [13] focuses on sparse semantic keypoints on objects. For non-rigid (deformable) models, FAUST [6] and TOSCA [8] provide dense correspondence labels for humans and animals, respectively. These methods leverage the clear anatomy structure underlying humans and animals and can be applied to pose transfer, pose synthesis, etc.

Methods on Fine-grained Semantic Understanding In the last decade, plenty of methods have been proposed to find semantic correspondences between paired images. Earlier methods like Okutomi et al. [33], Horn et al. [18] and Matas et al. [29] propose to find semantic correspondences within the same scene. Semantic flows like SIFT flow [28] and ProposalFlow [17] further explore to find dense correspondence across different scenes. Kulkarni et al. [21] and Zhou et al. [49] utilize a synthesis 3D model as a medium to enforce semantic cycle-consistency. Florence et al. [14] and Schmidt et al. [39] leverage an unsupervised method to learn consistent dense embeddings across different objects.

When it comes to the domain of 3D shapes, Blanz et al. [5] and Allen et al. [1] are the pioneers on finding 3D correspondence between human faces and bodies. Recently, 3D dense semantic correspondence has been boosted by a variety of deep learning methods. Halimi et al. [16], Groueix et al. [15] and Roufosse et al. [37] propose unsupervised methods on learning dense correspondences between humans and animals. Deep functional dictionaries [40] gives a small dictionary of basis functions for each shape, a dictionary whose span includes the semantic functions provided for that shape.

3. Understanding Semantic Information from Humans

Understanding semantics from arbitrary objects is of great importance. However, explicitly expressing semantics in a well defined format is extremely hard as the definition of semantics is vague and diverse.

We observe that people are pretty sure about the correspondence between two areas but less sure about what each area means in semantics. As shown in Figure 1, almost everyone would agree on the lined correspondences between two helmets. However, it is hard to tell the exact semantic meanings of the colored areas.

Therefore, unlike all previous methods where an explicit

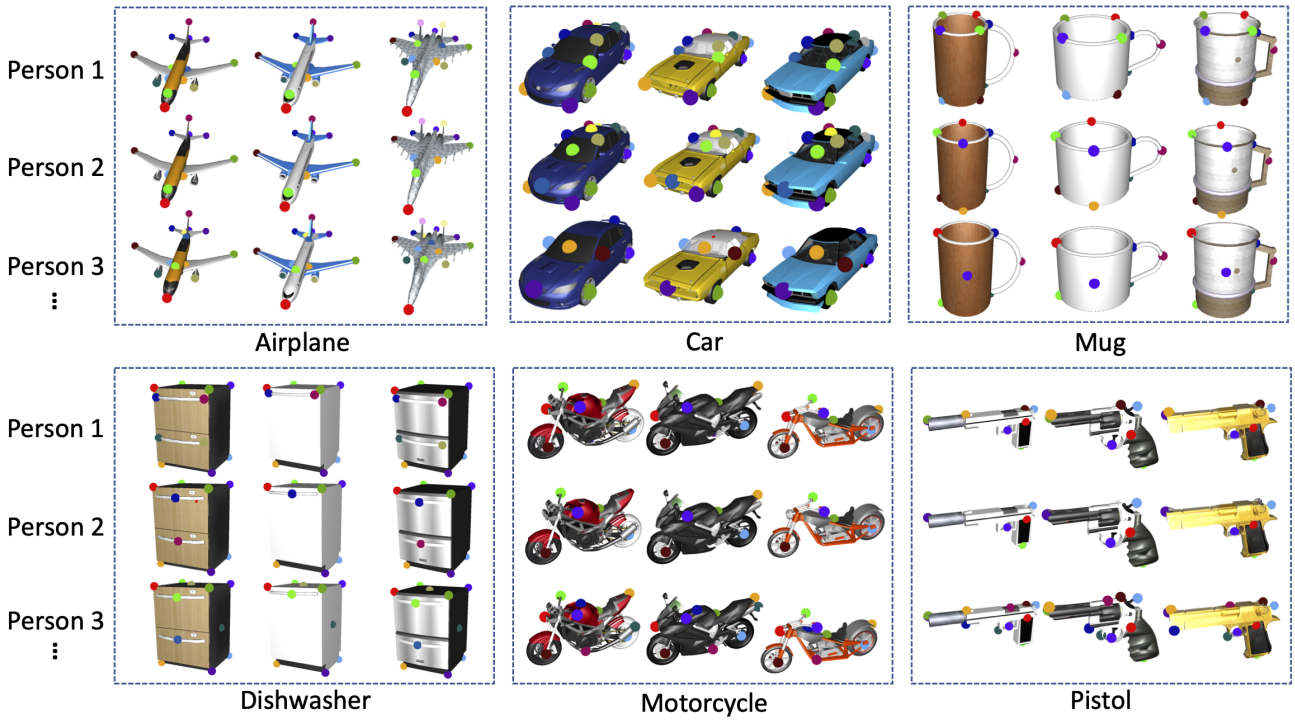

Figure 2. **CPNet dataset**. Each person annotates multiple sets of corresponding points. Points in the same correspondence set are in the same color. It can be seen that people could have his/her own understanding of semantic points as long as they are consistent across different models within the same category.

definition of keypoints or parts is given, we instead focus on sparse correspondences annotated by humans, based on the assumption that all the corresponding points labeled by the same person share the same semantic meaning.

4. CorresPondenceNet

Corres**P**ondenceNet (CPNet) has a collection of 25 categories, 2000+ models based on ShapeNetCore. Each model is annotated with a number of semantic points from multiple annotators, as shown in Figure 2. Unlike other 2D or 3D keypoint datasets which manually set a keypoint template and let annotators to follow, semantic points in our dataset are not deliberately defined by anyone. The key is that every annotator can have his/her own understanding of semantic points, as long as they are consistent across different models within the same category. In the following subsections, we discuss how we collect models, how we annotate models and annotation types in details.

4.1. Dataset Collections

Our dataset is based on ShapeNetCore [9]. ShapeNetCore is a subset of the full ShapeNet dataset with single clean 3D models and manually verified category and alignment annotations. There are 51,300 unique 3D models from 55 common object categories in ShapeNetCore. We select 25 categories that are mostly seen in daily life to build our dataset. To keep a balanced dataset, for each category we keep at most 100 models. For those categories with less than 100 models, all the models are selected.

4.2. Annotation Process

We hire 80 professional annotators in total. Each model is annotated by at least 10 persons to enrich the dataset.

For each category, every annotator is allowed to create 1 to 6 templates with his/her own understanding of semantic points. Templates are then listed to guide the annotations of rest models, so that he/she is able to keep the consistency. Consider an airplane as an example, if one annotator marks the nose as No.1 semantic point, then he/she is supposed to mark all the noses on other airplanes as No.1. It does not matter if another annotator marks the nose as No.2 semantic point, or even neglecting it, as long as one annotator obeys his own rules across all the models. For those points that may not exist on all the models such as propeller, one can just skip this point on the models without it. The annotator is free to choose any points from his/her perspective.

Each annotator is asked to mark at most 16 semantic points per model. All points are annotated on raw meshes, which is more accurate than those annotated on point clouds. Moreover, it is straightforward to extend these annotations to point clouds by sampling from the mesh

	Airplane	Bathtub	Bed	Bench	Bottle	Bus	Cap	Car	Chair	Dishwasher	Display	Earphone	Faucet
N_K	5527	6033	6464	5421	4489	6404	949	7938	6140	5343	4509	904	1612
N_A	10	10	10	10	10	10	10	10	10	10	10	10	10
N_M	100	100	100	100	100	100	38	100	100	77	100	58	100
C_{min}	35	40	40	30	41	50	20	64	50	60	20	14	10
C_{med}	54	60	60	50	45	64	25	80	70	70	50	15	15
C_{max}	72	96	80	70	46	81	30	82	78	84	51	21	22

	Guitar	Helmet	Knife	Lamp	Laptop	Motorcycle	Mug	Pistol	Rocket	Skateboard	Table	Vessel	All
N_K	2832	1500	2109	1683	2987	3878	7668	3358	2315	3822	4008	5214	104861
N_A	10	10	10	10	10	10	10	10	10	10	10	10	-
N_M	100	95	100	100	100	100	100	100	66	100	100	100	2334
C_{min}	19	27	10	13	20	30	66	17	21	20	39	40	-
C_{med}	30	35	12	15	30	40	77	35	32	40	40	54	-
C_{max}	32	37	15	21	36	40	78	41	49	43	44	56	-

Table 1. **CPNet statistics.** N_K gives the number of annotated points of each category; N_A gives the number of annotators for each category; N_M is the number of models in each category; C_{min}, C_{med}, C_{max} give minimum, median and maximum number of correspondence sets per instance in each category.

while fixing the locations of semantic points.

4.3. Annotation Type

Denote all the models as $\mathbf{M} = \{\mathcal{M}_i\}$, where $\mathcal{M}_i$ represents a single model. Each mdoel $\mathcal{M}_i$ is associated with a set of semantic points $\mathcal{P}_i = \{p_{i,j}^{(n)}\}$ where i, j, n denote the j-th semantic point of the n-th annotator on the i-th model.

In addition, we ask each annotator to give consistent points across different models, so that $p_{i_1,j}^{(n)}$ and $p_{i_2,j}^{(n)}$ have the same semantic meaning. Therefore, we define a set of correspondence sets $\Omega = \{\mathcal{C}_j | j = 1, \cdots, N_\Omega\}$, where each correspondence set $\mathcal{C}_j = \{p_{i,j} | i = 1, \cdots, N_\mathbf{M}\}$ contains all the points with the same semantic label. Note that we dropped the index of the annotator since distinct point correspondence from the same person can be treated the same as those from different persons.

Each annotated point contains attributes about (1) xyz coordinate, (2) color, (3) face index and (4) uv coordinate. By providing these attributes, methods based on either point clouds or meshes can be applied easily.

4.4. Statistics

CPNet provides instance-level keypoint annotation for 2,334 models with 104,861 keypoints from 25 object categories. Table 1 gives the detailed statistics of our dataset.

5. Proposed Method

We now propose a method on learning dense semantic embeddings from human labeled correspondences across various intra-class models.

5.1. Problem Statement

Given a set of 3D models $\mathbf{M} = \{\mathcal{M}_i | i = 1, \cdots, N_\mathbf{M}\}$ and a set of correspondence sets $\Omega = \{\mathcal{C}_j | j = 1, \cdots, N_\Omega\}$ defined in Section 4.3, our goal is to produce a set of pointwise embeddings for each model $\mathcal{M}_i$. The embeddings encode semantic information across different models and points with similar semantics are close in embedding space. We define f as an embedding function, such that $f(p)$ gives the embedding for point p on the model. In practice, we approximate f with a deep neural network and explain how to optimize f as follows.

5.2. Method Details

Pull Loss It is natural to come up with a pull loss since we would like to ensure the semantic consistency within every correspondence set. As illustrated in Figure 3, the points with the same color belong to the same correspondence set and reveal similar semantic information. For one specific correspondence $\mathcal{C}_k$ like the green line shown in Figure 3, we aim to pull the embedding vectors of the points within it. Any two of points in the same correspondence set form a positive pair. The pairwise embedding distances are then summed over all positive pairs to form our pull loss:

$$L_{pull} = \frac{1}{N_{pos}} \sum_k \sum_{p,q \in \mathcal{C}_k, p \neq q} \|f(p) - f(q)\|_2, \quad (1)$$

where N_{pos} is the number of all possible positive point pairs.

Geodesic Consistency Loss The pull loss in Equation 1 enforces the points in the same correspondence set to have similar embeddings. However, there is a trivial solution

Figure 3. **Correspondence sets across different airplanes.** $\mathcal{C}_i, \mathcal{C}_j$ and $\mathcal{C}_k$ denote three semantic correspondence sets respectively.

where f outputs a constant embedding (e.g. $\mathbf{0}$) for all points, which is a global optimum when minimizing L_{pull} only. Such a trivial solution is due to the ignorance of an important principle: we ought to ensure that those points with distinct semantics to have a large embedding distance. Therefore, a push loss guided by geodesic consistency is proposed to fulfill this goal. We leverage a prior to determine whether two different correspondence sets have distinct semantics: if all pairs of points from these two sets have large geodesic distances on models, they are more likely to reveal different semantic information.

Based on this insight, we design a distance measure $\mathbf{d}$ for a pair of correspondence sets $\mathcal{C}_i$ and $\mathcal{C}_j$:

$$\mathbf{d}(\mathcal{C}_i, \mathcal{C}_j) = \frac{1}{N_{\mathbf{M}}} \sum_k \sum_{p,q \in \mathcal{M}_k} \mathbf{d}_{geo}(p, q), \quad s.t.\ p \in \mathcal{C}_i, q \in \mathcal{C}_j, \tag{2}$$

where $\mathbf{d}_{geo}(p, q)$ is the geodesic distance between point p and q. This distance measure $\mathbf{d}$ represents the average geodesic distance between point pairs from two correspondence sets.

Then, the push loss can be written as,

$$L_{push} = \frac{1}{N_{neg}} \sum_{i \neq j} \sum_{p \in \mathcal{C}_i} \sum_{q \in \mathcal{C}_j} \max\{0, \mathbf{d}(\mathcal{C}_i, \mathcal{C}_j) - \|f(p) - f(q)\|_2\}, \tag{3}$$

where N_{neg} is the number of all possible negative pairs formed by points from different correspondence sets.

In Equation 3, the push loss is only activated when $\|f(p) - f(q)\|_2$ is smaller than $\mathbf{d}(\mathcal{C}_i, \mathcal{C}_j)$. In other words, the larger $\mathbf{d}(\mathcal{C}_i, \mathcal{C}_j)$ is, the further $f(p)$ and $f(q)$ are separated in the embedding space. This is based on the observation that some points in two correspondence sets may have similar semantic information (like the red and orange lines in Figure 3) while some have totally different meanings (like the orange and green lines in Figure 3). Therefore, only for those correspondence sets with a large average geodesic distance, a large distance between their embeddings is expected.

Our final loss is,

$$L = L_{pull} + \lambda L_{push}, \tag{4}$$

where λ is a weight factor.

Hard Negative Mining In practice, negative pairs to be pushed are combinatorially more than positive pairs to be pulled, since negative pairs are sampled from different correspondence sets. In such case, we borrow the idea from [11] to utilize hard negative mining. Within each batch, only those negative pairs with smallest embedding distances are taken into consideration, matching the number of positive pairs.

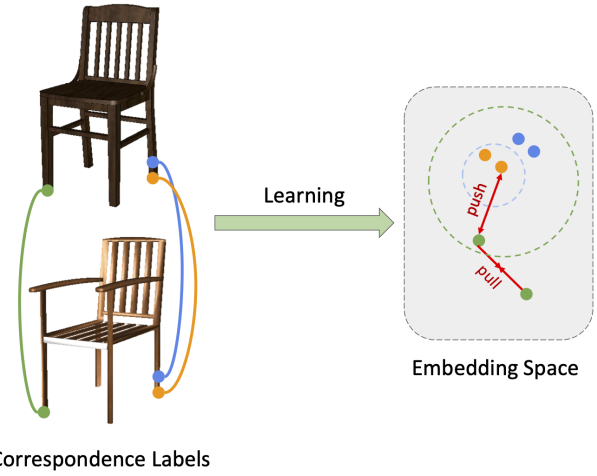

Figure 4. Given correspondence sets, we pull the points in the same correspondence set and push points from different correspondence sets adaptively, according to their average geodesic distances. The blue and orange correspondence sets are close so that they can stay close in embedding space, while the orange and green ones are far away in average geodesic distance so their embeddings are pushed further from each other.

Our method is summarized in Figure 4.

6. Experiments

Algorithm 1 mean Geodesic Error calculation

Input: model set Ω, an embedding function f to be evaluated
Output: mean Geodesic Error (mGE) ε of f
$\varepsilon = 0$
for $\mathcal{C}_i$ **in** Ω **do**
  for p **in** $\mathcal{C}_i$ **do**
    for q **in** $\mathcal{C}_i$ **do**
      $x = \arg\min_{x \in \mathcal{M}_q} \|f(x) - f(p)\|_2$, where $\mathcal{M}_q$ denotes the model that point q lies on.
      $\varepsilon = \varepsilon + \mathbf{d}_{geo}(q, x)$
    end for
  end for
end for
$\varepsilon = \frac{\varepsilon}{N_\Omega N_{\mathbf{M}}^2}$

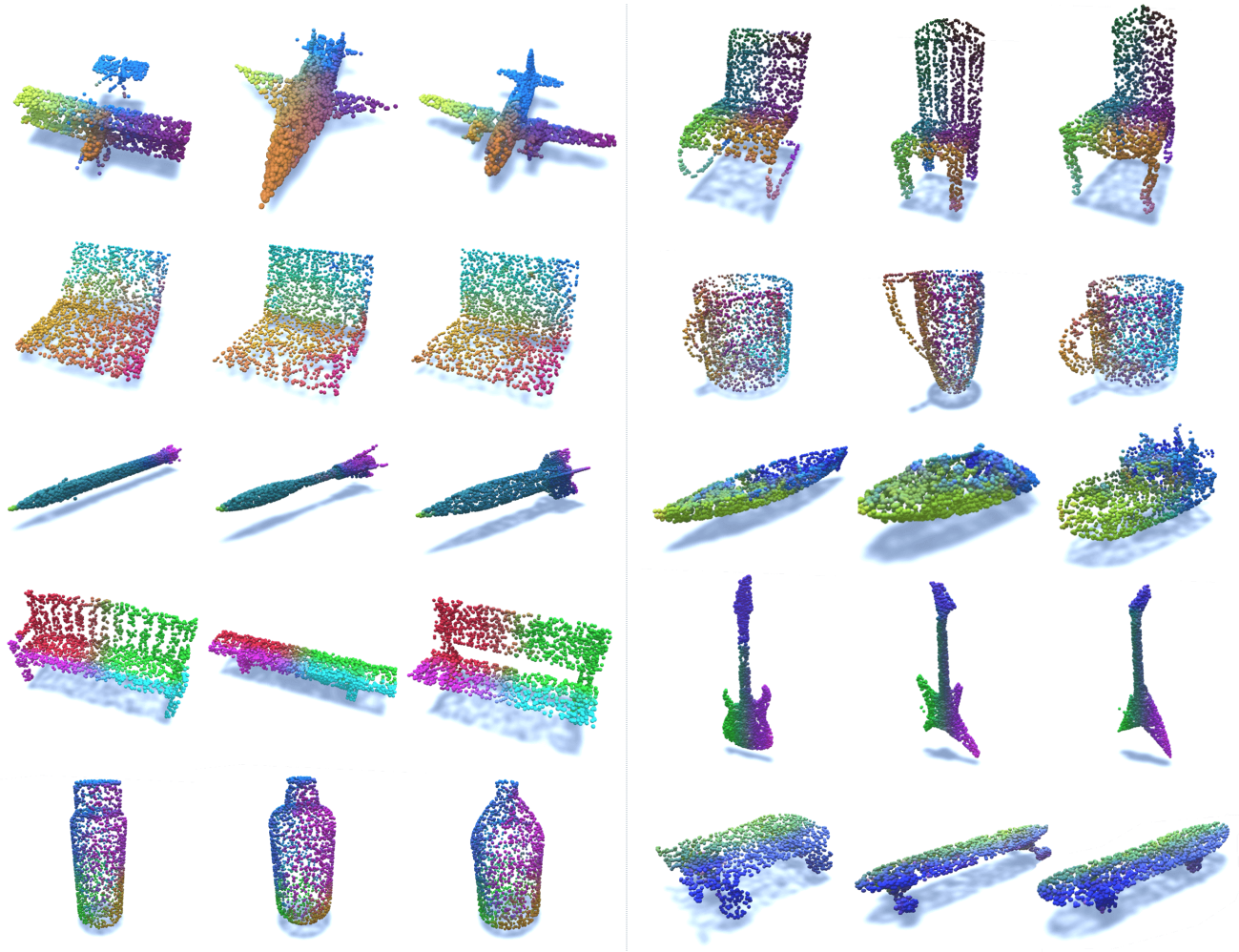

Figure 5. **Predicted semantic embeddings for PontConv.** Same colors indicate similar embeddings.

In this section, we demonstrate that our proposed method on learning pointwise embeddings can effectively help fine-grained object semantic understanding. We first introduce a new metric to evaluate predicted embeddings. Then seven state-of-the art neural network architectures are chosen as our method's backbones and benchmarked. We additionally compare our approach, which is based on human labeled correspondences, with that based on part-level supervision.

Evaluation Metric We introduce mean Geodesic Error (mGE) to evaluate predicted semantic embeddings. mGE is calculated individually for each category and measures how well the generated embedding vectors fit with annotated correspondence sets. Algorithm 1 presents the calculation procedure of mGE for a given embedding function f. Intuitively, for each annotated points on a model, we find their corresponding points that minimize the embedding distance on other models. After that, the geodesic distances between these points and human labeled corresponding points are accumulated. It is easy to verify that if all the embeddings are identical within the same correspondence set but are distinct across different correspondence sets, mGE $= 0$, which means that the predicted semantic embeddings are consistent with human labels.

Benchmark Neural Networks We benchmark three kinds of backbones: point cloud, graph and voxel based neural networks. Point cloud based architectures PointNet [35], PointNet++ [36] and PointConv [46] take unordered point sets as the input and generate embeddings directly from these point sets. Graph based architectures DGCNN [45] and GraphCNN [12] use graph based convolutional neural networks to extract embeddings. Voxel based architecture MinkowskiNet [10] takes voxels as the

	Airplane	Bathtub	Bed	Bench	Bottle	Bus	Cap	Car	Chair	Dishwasher	Display	Earphone	Faucet
PointNet	0.088	**0.245**	0.231	**0.198**	0.106	**0.082**	0.123	0.074	0.198	**0.124**	**0.180**	0.101	0.170
PointNet++	0.083	0.307	0.254	0.210	0.218	0.142	0.123	0.077	0.199	0.168	0.207	0.130	0.189
RS-Net	0.095	0.280	**0.212**	0.280	**0.105**	0.084	**0.086**	**0.065**	0.187	0.149	0.183	**0.092**	0.150
PointConv	0.078	0.284	0.237	0.220	0.107	0.107	0.099	0.083	**0.185**	0.142	0.186	0.096	0.150
DGCNN	**0.075**	0.273	0.223	0.216	0.144	0.115	0.110	0.087	0.215	0.159	0.239	**0.092**	**0.141**
GraphCNN	0.091	0.291	0.256	0.217	0.139	0.123	0.121	0.113	0.212	0.166	0.220	0.138	0.164
Minkowski	0.108	0.286	0.270	0.243	0.152	0.136	0.144	0.099	0.238	0.189	0.253	0.117	0.161
SHOT	0.229	0.488	0.539	0.530	0.382	0.405	0.345	0.386	0.474	0.515	0.455	0.495	0.274
Random	0.290	0.489	0.526	0.507	0.427	0.396	0.484	0.401	0.478	0.507	0.459	0.599	0.337

	Guitar	Helmet	Knife	Lamp	Laptop	Motorcycle	Mug	Pistol	Rocket	Skateboard	Table	Vessel	Average
PointNet	**0.095**	0.177	0.061	0.265	0.171	0.123	**0.070**	0.168	0.186	0.155	0.075	**0.119**	**0.143**
PointNet++	0.116	0.186	0.079	0.263	0.183	0.128	0.106	0.185	0.163	0.179	0.093	0.159	0.166
RS-Net	0.110	**0.167**	**0.054**	0.273	0.138	**0.122**	0.110	**0.161**	**0.152**	0.166	0.089	0.135	0.146
PointConv	0.109	0.176	0.076	0.270	**0.137**	0.128	0.085	0.173	0.168	0.156	0.097	0.144	0.148
DGCNN	0.124	0.173	0.068	**0.261**	0.181	0.148	0.139	0.174	0.172	**0.150**	**0.069**	0.162	0.156
GraphCNN	0.135	0.184	0.116	0.279	0.168	0.152	0.132	0.185	0.181	0.169	0.099	0.199	0.170
Minkowski	0.148	0.213	0.105	0.290	0.206	0.170	0.149	0.194	0.195	0.173	0.109	0.172	0.181
SHOT	0.305	0.387	0.194	0.425	0.543	0.340	0.414	0.334	0.271	0.381	0.607	0.377	0.404
Random	0.326	0.406	0.426	0.451	0.543	0.358	0.488	0.375	0.298	0.378	0.544	0.378	0.435

Table 2. **Mean Geodesic Error (mGE) results**.

input and utilize sparse 3D convolutions. In addition, we report the performance of a local geometry based descriptor SHOT [43] and random embeddings.

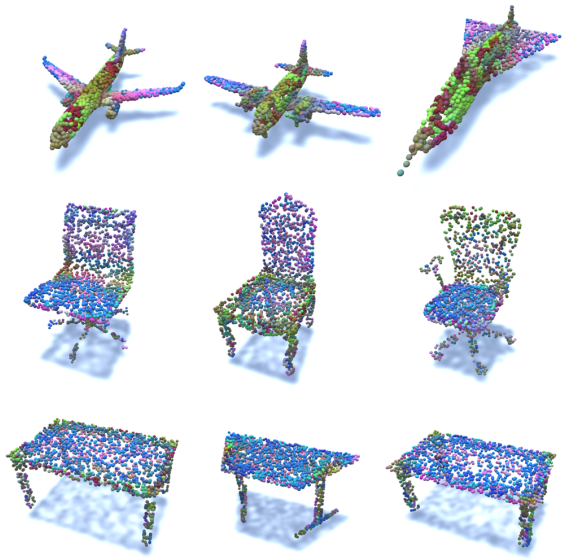

Figure 6. **Predicted embeddings for SHOT.** Same colors indicate similar embeddings.

Evaluation and Results We split our dataset into train (70%), validation (15%) and test (15%) set. Train and validation sets are used during training and all the results are reported on the test set. We use ADAM optimizer [20] with initial learning rate $\alpha = 0.001$, $\beta_1 = 0.9$, $\beta_2 = 0.999$ and batch size 4. The learning rate is multiplied by 0.9 every 10 epochs and the hyperparameter λ in Equation 4 is set to 1. The output point embedding vector is 128-dimensional for all neural networks.

Table 2 gives mGE of all the compared architectures. SHOT fails to predict correct semantic correspondences across objects, whose performance is just slightly better than random point embeddings. The reason is that SHOT only considers local geometric properties, without aggregation of the global structure and semantic information. The visualization of embeddings computed by SHOT are shown in Figure 6. In contrast, all deep learning based methods using our geodesic consistency loss achieve much smaller mGE. Among them, PointNet, RS-Net and PointConv are relatively superior to the other nets on extracting semantic correspondence information. The visualization of learned embeddings by PointConv is shown in Figure 5. From Figure 5, we can see that consistent pointwise embeddings are generated across heterogeneous objects. We get reasonable dense embeddings of all points on objects though only sparse correspondence annotations are used. A possible explanation is that the annotated correspondences impose a sparse set of pairwise constraints on the embedding function approximated by a deep neural network. Deep neural networks are usually Lipschitz-continuous and therefore, by fitting these imposed correspondence constraints, dense continuous embeddings could be inferred.

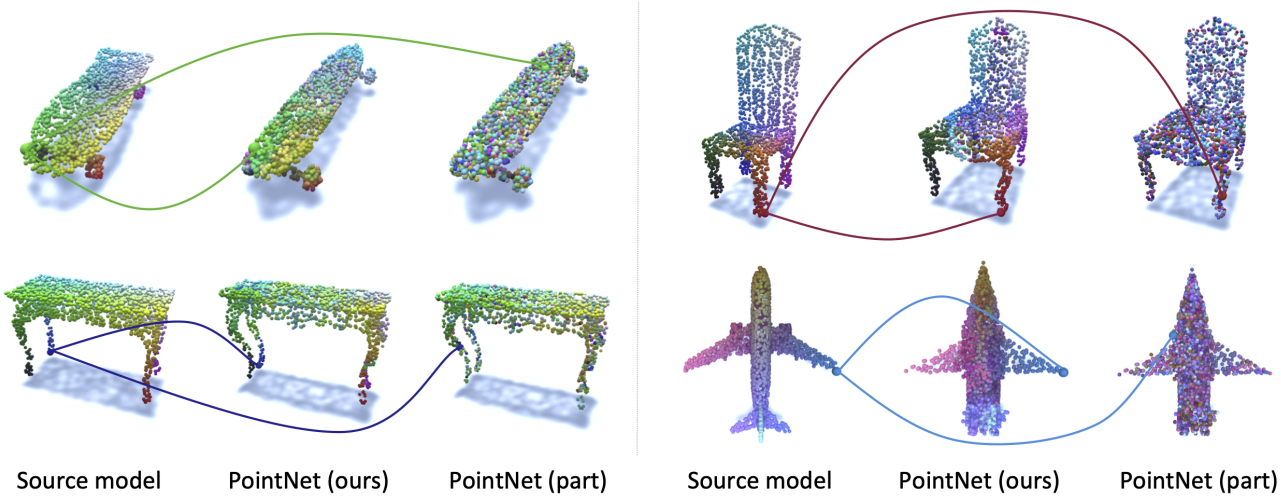

Figure 7. **Comparison between our method and part-level supervision.** Given a point on the source model, we find its closest point in embedding space on the target model and post-process the founded correspondences with PMF [44] to ensure bijectiveness. The corresponding points are in the same color.

Comparison to Part-level Supervision To further illustrate the advantage of our proposed semantic correspondence sets, we compare our method with that supervised by part-level annotations.

We train a PointNet using correspondence labels and part labels respectively. For PointNet trained on part labels, we use the same experiment settings for part segmentation as the original paper [35] and extract features from the last but one layer as point embeddings. Then given a point on a source model, we use embeddings to find its corresponding point on the target model and results are shown in Figure 7. Qualitatively, we can see that when trained on our correspondence labels, points of the same semantic have similar embeddings while part-level supervision fails to give consistent semantic embeddings across objects. In addition, we compare them quantitatively using mGE, as shown in Table 3. Clearly, PointNet trained on our correspondence labels achieves better performance. On the contrary, with only part-level supervision, points in the same part are hard to be distinguished from each other, resulting in inferior performance. Note that the number of training data for part-level supervision (10240) is seven times more than that for correspondence based supervision (1362).

	PointNet	PointNet(Part)
Airplane	**0.088**	0.182
Cap	**0.123**	0.271
Car	**0.074**	0.246
Chair	**0.198**	0.278
Earphone	**0.101**	0.140
Guitar	**0.095**	0.114
Knife	**0.061**	0.065
Lamp	**0.265**	0.313
Laptop	0.171	**0.114**
Motorcycle	**0.123**	0.237
Mug	**0.070**	0.182
Pistol	**0.168**	0.204
Rocket	**0.186**	0.218
Skateboard	**0.155**	0.330
Table	**0.075**	0.282
Average	**0.130**	0.212

Table 3. **Comparison of the results trained on human labeled correspondences and part annotations using PointNet.** We can see that part-level supervision are far from enough for inferring finer object semantics while our method on human labeled correspondences could help improve the semantic understanding of objects.

7. Conclusion

In this paper, we explored a new way to obtain fine-grained semantic understanding of 3D objects. Instead of explicitly defining semantic labels on an object, we leveraged an observation that though semantic meanings on a single object can be ambiguous and hard to depict, the correspondences of certain points across objects are clear. We thus built a dataset named **C**orres**P**ondence**Net** (CPNet) based on human labeled correspondences, and proposed a novel geodesic guided push-pull loss to recover dense and rich semantic information of objects. Mean Geodesic Error (mGE) metric is introduced to evaluate our method with various backbones. As shown in the experiments, our method can effectively learn pointwise semantic embeddings, which are implicitly inferred from correspondences.

References

[1] Brett Allen, Brian Curless, Brian Curless, and Zoran Popović. The space of human body shapes: reconstruction and parameterization from range scans. In *ACM transactions on graphics (TOG)*, volume 22, pages 587–594. ACM, 2003. 2

[2] M. Andriluka, U. Iqbal, E. Ensafutdinov, L. Pishchulin, A. Milan, J. Gall, and Schiele B. PoseTrack: A benchmark for human pose estimation and tracking. In *CVPR*, 2018. 2

[3] Mykhaylo Andriluka, Leonid Pishchulin, Peter Gehler, and Bernt Schiele. 2d human pose estimation: New benchmark and state of the art analysis. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014. 2

[4] Oscar Kin-Chung Au, Chiew-Lan Tai, Hung-Kuo Chu, Daniel Cohen-Or, and Tong-Yee Lee. Skeleton extraction by mesh contraction. In *ACM transactions on graphics (TOG)*, volume 27, page 44. ACM, 2008. 1

[5] Volker Blanz, Thomas Vetter, et al. A morphable model for the synthesis of 3d faces. In *Siggraph*, volume 99, pages 187–194, 1999. 2

[6] Federica Bogo, Javier Romero, Matthew Loper, and Michael J Black. Faust: Dataset and evaluation for 3d mesh registration. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3794–3801, 2014. 2

[7] Mariusz Bojarski, Davide Del Testa, Daniel Dworakowski, Bernhard Firner, Beat Flepp, Prasoon Goyal, Lawrence D Jackel, Mathew Monfort, Urs Muller, Jiakai Zhang, et al. End to end learning for self-driving cars. *arXiv preprint arXiv:1604.07316*, 2016. 1

[8] Alexander M Bronstein, Michael M Bronstein, and Ron Kimmel. *Numerical geometry of non-rigid shapes*. Springer Science & Business Media, 2008. 2

[9] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015. 1, 2, 3

[10] Christopher Choy, JunYoung Gwak, and Silvio Savarese. 4d spatio-temporal convnets: Minkowski convolutional neural networks. *arXiv preprint arXiv:1904.08755*, 2019. 6

[11] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. 2005. 5

[12] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In *Advances in neural information processing systems*, pages 3844–3852, 2016. 6

[13] Helin Dutagaci, Chun Pan Cheung, and Afzal Godil. Evaluation of 3d interest point detection techniques via human-generated ground truth. *The Visual Computer*, 28(9):901–917, 2012. 2

[14] Peter R Florence, Lucas Manuelli, and Russ Tedrake. Dense object nets: Learning dense visual object descriptors by and for robotic manipulation. *arXiv preprint arXiv:1806.08756*, 2018. 2

[15] Thibault Groueix, Matthew Fisher, Vladimir G Kim, Bryan C Russell, and Mathieu Aubry. 3d-coded: 3d correspondences by deep deformation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 230–246, 2018. 2

[16] Oshri Halimi, Or Litany, Emanuele Rodolà, Alex Bronstein, and Ron Kimmel. Self-supervised learning of dense shape correspondence. *arXiv preprint arXiv:1812.02415*, 2018. 2

[17] Bumsub Ham, Minsu Cho, Cordelia Schmid, and Jean Ponce. Proposal flow: Semantic correspondences from object proposals. *IEEE transactions on pattern analysis and machine intelligence*, 40(7):1711–1725, 2017. 2

[18] Berthold KP Horn and Brian G. Schunck. " determining optical flow": A retrospective. 1993. 2

[19] Mahdi M Kalayeh, Emrah Basaran, Muhittin Gökmen, Mustafa E Kamasak, and Mubarak Shah. Human semantic parsing for person re-identification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1062–1071, 2018. 1

[20] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 7

[21] Nilesh Kulkarni, Abhinav Gupta, and Shubham Tulsiani. Canonical surface mapping via geometric cycle consistency. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2202–2211, 2019. 2

[22] Abhijit Kundu, Yin Li, and James M Rehg. 3d-rcnn: Instance-level 3d object reconstruction via render-and-compare. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3559–3568, 2018. 1

[23] Biao Leng, Yu Liu, Kai Yu, Xiangyang Zhang, and Zhang Xiong. 3d object understanding with 3d convolutional neural networks. *Information sciences*, 366:188–201, 2016. 1

[24] Stefan Leutenegger, Margarita Chli, and Roland Siegwart. Brisk: Binary robust invariant scalable keypoints. In *2011 IEEE international conference on computer vision (ICCV)*, pages 2548–2555. Ieee, 2011. 1

[25] Sergey Levine, Peter Pastor, Alex Krizhevsky, Julian Ibarz, and Deirdre Quillen. Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection. *The International Journal of Robotics Research*, 37(4-5):421–436, 2018. 1

[26] Jing Liao, Yuan Yao, Lu Yuan, Gang Hua, and Sing Bing Kang. Visual attribute transfer through deep image analogy. *arXiv preprint arXiv:1705.01088*, 2017. 1

[27] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 1

[28] Ce Liu, Jenny Yuen, and Antonio Torralba. Sift flow: Dense correspondence across scenes and its applications. *IEEE transactions on pattern analysis and machine intelligence*, 33(5):978–994, 2010. 2

[29] Jiri Matas, Ondrej Chum, Martin Urban, and Tomás Pajdla. Robust wide-baseline stereo from maximally stable extremal regions. *Image and vision computing*, 22(10):761–767, 2004. 2

[30] Jan Matas, Stephen James, and Andrew J Davison. Sim-to-real reinforcement learning for deformable object manipulation. *arXiv preprint arXiv:1806.07851*, 2018. 1

[31] Juhong Min, Jongmin Lee, Jean Ponce, and Minsu Cho. Spair-71k: A large-scale benchmark for semantic correspondence. *arXiv preprint arXiv:1908.10543*, 2019. 2

[32] Kaichun Mo, Shilin Zhu, Angel X Chang, Li Yi, Subarna Tripathi, Leonidas J Guibas, and Hao Su. Partnet: A large-scale benchmark for fine-grained and hierarchical part-level 3d object understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 909–918, 2019. 1, 2

[33] Masatoshi Okutomi and Takeo Kanade. A multiple-baseline stereo. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (4):353–363, 1993. 2

[34] Brian Paden, Michal Čáp, Sze Zheng Yong, Dmitry Yershov, and Emilio Frazzoli. A survey of motion planning and control techniques for self-driving urban vehicles. *IEEE Transactions on intelligent vehicles*, 1(1):33–55, 2016. 1

[35] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 652–660, 2017. 6, 8

[36] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in neural information processing systems*, pages 5099–5108, 2017. 6

[37] Jean-Michel Roufosse, Abhishek Sharma, and Maks Ovsjanikov. Unsupervised deep learning for structured shape matching. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1617–1627, 2019. 2

[38] Samuele Salti, Federico Tombari, Riccardo Spezialetti, and Luigi Di Stefano. Learning a descriptor-specific 3d keypoint detector. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2318–2326, 2015. 1

[39] Tanner Schmidt, Richard Newcombe, and Dieter Fox. Self-supervised visual descriptor learning for dense correspondence. *IEEE Robotics and Automation Letters*, 2(2):420–427, 2016. 2

[40] Minhyuk Sung, Hao Su, Ronald Yu, and Leonidas J Guibas. Deep functional dictionaries: Learning consistent semantic structures on 3d models from functions. In *Advances in Neural Information Processing Systems*, pages 485–495, 2018. 2

[41] Supasorn Suwajanakorn, Noah Snavely, Jonathan J Tompson, and Mohammad Norouzi. Discovery of latent 3d keypoints via end-to-end geometric reasoning. In *Advances in Neural Information Processing Systems*, pages 2059–2070, 2018. 1

[42] Tatsunori Taniai, Sudipta N Sinha, and Yoichi Sato. Joint recovery of dense correspondence and cosegmentation in two images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4246–4255, 2016. 2

[43] Federico Tombari, Samuele Salti, and Luigi Di Stefano. Unique signatures of histograms for local surface description. In *European conference on computer vision*, pages 356–369. Springer, 2010. 7

[44] Matthias Vestner, Roee Litman, Emanuele Rodolà, Alex Bronstein, and Daniel Cremers. Product manifold filter: Non-rigid shape correspondence via kernel density estimation in the product space. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3327–3336, 2017. 8

[45] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. Dynamic graph cnn for learning on point clouds. *ACM Transactions on Graphics (TOG)*, 38(5):146, 2019. 6

[46] Wenxuan Wu, Zhongang Qi, and Fuxin Li. Pointconv: Deep convolutional networks on 3d point clouds. *CoRR*, abs/1811.07246, 2018. 6

[47] Li Yi, Vladimir G Kim, Duygu Ceylan, I Shen, Mengyan Yan, Hao Su, Cewu Lu, Qixing Huang, Alla Sheffer, Leonidas Guibas, et al. A scalable active framework for region annotation in 3d shape collections. *ACM Transactions on Graphics (TOG)*, 35(6):210, 2016. 1, 2

[48] Bolei Zhou, Hang Zhao, Xavier Puig, Tete Xiao, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Semantic understanding of scenes through the ade20k dataset. *International Journal of Computer Vision*, 127(3):302–321, 2019. 1

[49] Tinghui Zhou, Philipp Krahenbuhl, Mathieu Aubry, Qixing Huang, and Alexei A Efros. Learning dense correspondence via 3d-guided cycle consistency. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 117–126, 2016. 2