KeypointDeformer: Unsupervised 3D Keypoint Discovery for Shape Control

Tomas Jakab[1,4*], Richard Tucker[4], Ameesh Makadia[4], Jiajun Wu[3], Noah Snavely[4], Angjoo Kanazawa[2,4]
[1]University of Oxford, [2]UC Berkeley, [3]Stanford University, [4]Google Research

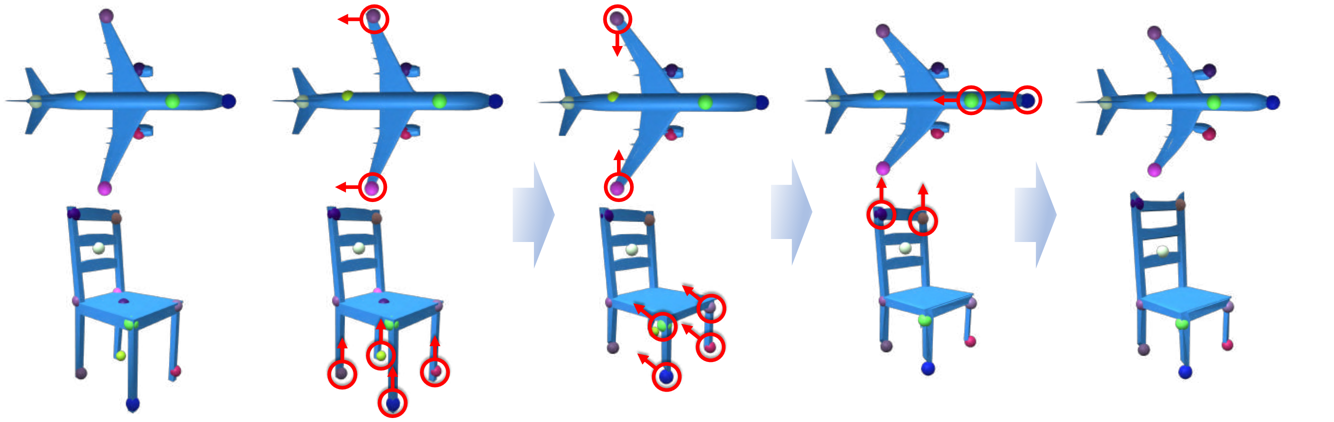


Figure 1: **Controlling shape deformation with unsupervised 3D keypoints.** We discover unsupervised 3D keypoints that allow for intuitive control of an object's shape. This figure shows individual steps of interactive control. The red arrows illustrate the direction in which the keypoints are manipulated. Note that the resulting deformations are localized and object parts are deformed in an intuitive manner—*e.g.*, moving keypoints at the tip of the wings backward moves the wings backwards—all while preserving the details of the original shape.

Abstract

We introduce KeypointDeformer, a novel unsupervised method for shape control through automatically discovered 3D keypoints. We cast this as the problem of aligning a source 3D object to a target 3D object from the same object category. Our method analyzes the difference between the shapes of the two objects by comparing their latent representations. This latent representation is in the form of 3D keypoints that are learned in an unsupervised way. The difference between the 3D keypoints of the source and the target objects then informs the shape deformation algorithm that deforms the source object into the target object. The whole model is learned end-to-end and simultaneously discovers 3D keypoints while learning to use them for deforming object shapes. Our approach produces intuitive and semantically consistent control of shape deformations. Moreover, our discovered 3D keypoints are consistent across object category instances despite large shape variations. As our method is unsupervised, it can be readily deployed to new object categories without requiring annotations for 3D keypoints and deformations. Project page: http://tomasjakab.github.io/KeypointDeformer.

1. Introduction

Given the vast number of 3D shapes available on the Internet, providing users with intuitive and simple interfaces for semantically manipulating objects while preserving their key shape properties has a wide variety of applications in AI-assisted 3D content creation. In this paper, we propose to automatically discover intuitive and semantically meaningful control points for interactive editing, enabling detail-preserving shape deformation for object categories.

Specifically, we identify 3D keypoints as an intuitive and simple interface for shape editing. Keypoints are sparse 3D points that are semantically consistent across an object category. We propose a learning framework for unsupervised discovery of such keypoints and a deformation model that uses the keypoints to deform a shape while preserving local shape detail. We call our model *KeypointDeformer*.

Figure 1 describes the inference-time use case of KeypointDeformer. Given a novel shape, KeypointDeformer predicts 3D keypoints on the surface. If a user manipulates a keypoint on a chair leg upwards, the entire leg is deformed in the same direction (bottom). Our approach optionally enables the use of a categorical deformation prior on these

[*] Work done while interning at Google Research.

edits, such that if a user moves one side of an airplane wing backwards, the opposite side of the wing is deformed symmetrically in the same direction (top)—while if the user wishes to only move one side of the wing, our approach also allows this. Our framework enables stand-alone shape edits or shape alignment between two shapes, and can also synthesize novel variations of shapes for amplifying stock datasets.

While 3D keypoints may be a good proxy for shape editing, obtaining explicit supervision for keypoints and deformation models is not only expensive but also ill-defined. As such, we propose an unsupervised framework for jointly discovering the keypoints and the deformation model. To solve our problem, we devise two components that operate in concert: (1) a method for discovering and detecting keypoints, and (2) a deformation model that propagates keypoint displacements to the rest of the shape. To achieve these, we set up a proxy learning task where the goal is to align a source shape with a target shape, where the two can represent very different instances of a category. We also propose a simple yet effective keypoint regularizer that encourages learning of semantically consistent keypoints that are well-distributed, lie close to the object surface and implicitly preserve underlying shape symmetries. The result of our training approach is a deformation model that deforms a shape based on automatically discovered 3D control keypoints. Since the keypoints are low-dimensional, we can further learn a category prior on these keypoints, enabling semantic shape editing from sparse user inputs.

Overall, our method has following key benefits:

1. It gives users an intuitive and simple way to interactively control object shapes.
2. Both the keypoint prediction and deformation model are unsupervised.
3. We show that keypoints discovered by our method are better for shape control than other kinds of keypoints, including manually annotated ones.
4. Our unsupervised 3D keypoints are semantically consistent across object instances of the same category giving us sparse correspondences.

We evaluate the semantic consistency of our unsupervised 3D keypoints on standard benchmarks, and achieve state-of-the-art results among unsupervised methods. We also demonstrate the suitability of our keypoints for shape deformation. Finally, we provide qualitative results of user-guided interactive shape control, and include videos of interactive shape control on our project page.

2. Related Work

Shape deformation. Our approach is closely related to detail-preserving deformations studied in geometric modeling, including Laplacian-based shape editing [21], As-Rigid-As-Possible shape deformation [22], and cages [13]. While these approaches enable shape editing via many forms of user-specified constraints (e.g., points or sets in an optimization framework), a major challenge is that they rely purely on geometric properties and do not consider semantic attributes or category-specific shape priors for deformation. Such priors can be obtained from artists painting the object surface with stiffness properties [1] or learned from a set of meshes with known correspondence [20]. However, such supervisions are prohibitively expensive to obtain and are not applicable to novel shapes. Yumer *et al*. [32] address this issue in a data-driven framework that provides a set of sliders that control the attributes of a given shape. However, this approach requires a set of predefined attributes obtained from expert annotations. We propose an unsupervised approach, and provide users with direct semantic deformation handles in the form of keypoints. Furthermore, our formulation can incorporate a category-specific deformation basis on the discovered 3D keypoints, allowing for semantically consistent user edits from sparse keypoints edits (such that if one side of an airplane wing is extended, the other opposite side also extends).

Another related problem is deformation transfer [23], which transfers the deformation exhibited by a source mesh onto a target mesh via known correspondences between shapes. Recent approaches employ deep learning to implicitly learn the shape correspondences to align two shapes [30, 8, 28]. While we also use a shape alignment objective to train our framework, we make our intermediate control explicit in the form of keypoints, which allows for stand-alone shape editing. In contrast, prior approaches always require a target shape to express the desired deformation.

User-guided shape editing. Our approach is related to recent deep learning–based methods that learn generative models of shapes for interactive editing. Tulsiani *et al*. [27] learn to abstract shapes in terms of primitives, which can be used to edit the shape by transferring primitive deformations to the surface. However, shape editing is not their primary focus, and it is unclear how well the direct transfer of primitive transformations preserve local shape detail. Recent approaches take this idea further by learning a generative model of primitives in the form of set of point-based primitives [9], shape handles [5], or disconnected shape manifolds [19]. These methods enable interactive editing by searching for latent primitive representations that best match user edits. However, they require an involved user interface via sketching or directly manipulating the underlying set of primitives. Most critically, as the edits are based on generative models, these approaches may change the local details of the original shape. In contrast, we directly deform the source shape, leading to better preservation of shape detail.

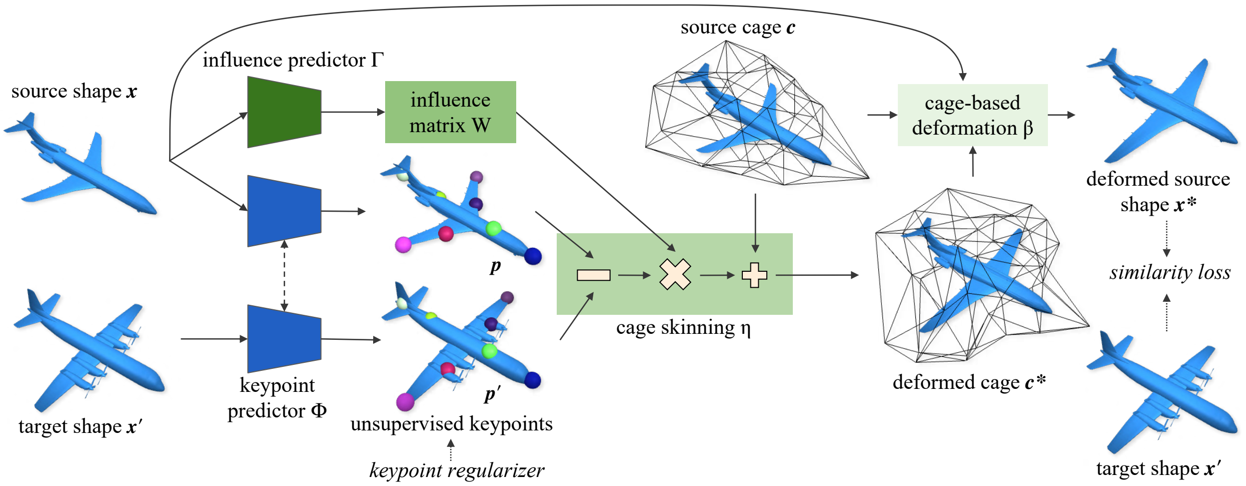


Figure 2: **Model.** Our model aligns the source shape $\boldsymbol{x}$ with the target shape $\boldsymbol{x}'$ using predicted unsupervised keypoints $\boldsymbol{p}$ and $\boldsymbol{p}'$. The unsupervised keypoints describe the objects pose and work as control points for the deformation. The model is trained end-to-end using a similarity loss between the deformed source shape $\boldsymbol{x}^*$ and the target shape $\boldsymbol{x}'$, as well as a keypoint regularization loss. During interactive shape manipulation at test time, a user can choose to input only the source shape $\boldsymbol{x}$ that the keypoint predictor Φ uses to estimate a set of unsupervised keypoints $\boldsymbol{p}$. The user can then manually control the keypoints $\boldsymbol{p}$ obtaining $\boldsymbol{p}'$ target keypoints that are fed into the deformation model to produce the deformed source shape $\boldsymbol{x}^*$ as demonstrated in Figure 1, Figure 9 and in the supplementary videos on our project page.

We qualitatively compare our approach to DualSDF [9] to illustrate this benefit.

Unsupervised keypoints. While the problem of unsupervised keypoint discovery is well studied in 2D [26, 33, 14, 10, 25, 11], this problem is relatively under-explored in 3D. Suwajanakorn *et al*. [24] detect 3D keypoints from a single image using 3D pose information as supervision. Here we focus on learning 3D keypoints on 3D shapes. Chen *et al*. [3] output a structured 3D representation to obtain sparse or dense shape correspondences. Closest to our approach in terms of 3D keypoint discovery is that of Fernandez *et al*. [4], which impose explicit symmetric constraints. In this work, we discover unsupervised keypoints for the purpose of shape control. While we focus on shape editing, our formulation results in state-of-the-art 3D keypoints for semantic consistency. Such unsupervised keypoints may be useful for robotics applications that use 3D keypoints as a latent representation for control [17, 6], and which currently require manually defined 3D keypoints as supervision.

3. Method

Our aim is to learn a keypoint predictor $\Phi : \boldsymbol{x} \rightarrow \boldsymbol{p}$ that maps a 3D object shape $\boldsymbol{x}$ to a sparse set of semantically consistent 3D keypoints $\boldsymbol{p}$. We also want to learn a conditional deformation model on keypoints $\Psi : (\boldsymbol{x}, \boldsymbol{p}, \boldsymbol{p}') \rightarrow \boldsymbol{x}'$ that deforms the shape $\boldsymbol{x}$ in accordance to the deformed control keypoints, where $\boldsymbol{p}$ describes the initial (source) keypoint locations and $\boldsymbol{p}'$ the target locations. Obtaining explicit supervision for keypoints and the deformation model is expensive and ill-defined. As such, we propose an unsupervised learning framework for training these functions. We do so by designing an auxiliary task of pair-wise shape alignment, where the key idea is to jointly learn keypoints and a deformation model that can bring two random shapes into alignment. Specifically, our model first predicts keypoint locations on the source and target shapes using a Siamese network. We then deform the source shape according to the correspondence provided by the discovered keypoints. In order to preserve local shape detail, we employ a cage-based deformation method, conditioned on keypoints. We devise a novel and highly effective, yet simple, keypoint regularization term that encourages keypoints to be well-distributed and lie close to the object surface. Figure 2 provides a schematic illustration of our framework.

3.1. Shape Deformation with Keypoints

We first predict keypoints from source and target meshes by representing each object as a point cloud $\boldsymbol{x} \in \mathbb{R}^{3 \times N}$, uniformly sampled from the object mesh. The keypoint predictor Φ takes the shape as an input $\boldsymbol{x}$ and outputs an ordered set of 3D keypoints $\boldsymbol{p} = (p_1, \ldots, p_K) \in \mathbb{R}^{3 \times K}$. The encoder is shared for both the source and target in a Siamese architecture. The shape deformation function Ψ takes the source shape $\boldsymbol{x}$ represented as a point cloud $\boldsymbol{x}$ as well as source keypoints $\boldsymbol{p}$ and target keypoints $\boldsymbol{p}'$. The keypoints $\boldsymbol{p}$ and $\boldsymbol{p}'$ are estimated by the keypoint predictor Φ. At test time, the user can input their own target keypoints $\boldsymbol{p}'$ for interactive shape deformation as illustrated in Figure 2.

In order to deform the object shape in a manner that preserves its local shape detail, we use the recently introduced differentiable cage-based deformation algorithm [30]. Cages are a classical shape modeling method [13, 12, 15] that use a coarse enclosing mesh that is associated with the shape. Deforming the cage mesh results in an interpolated deformation of the enclosed shape. The cage-based deformation function $\beta : (\boldsymbol{x}, \boldsymbol{c}, \boldsymbol{c}^*) \rightarrow \boldsymbol{x}^*$ takes a source control cage $\boldsymbol{c}$ and a deformed control cage $\boldsymbol{c}^*$, and deforms the input shape $\boldsymbol{x}$ that is in the form of a mesh or a point cloud. We automatically obtain the source cage $\boldsymbol{c}$ for each shape by starting with a spherical shape that is larger than the source shape $\boldsymbol{x}$ and iteratively pulling each of the cage vertices $\boldsymbol{c}_V$ towards the centre of the object until it is within a small distance from the object surface. The resulting cages are illustrated in Figure 2. While cages are a reliable method for shape-preserving deformation, modifying cages to achieve a desired deformation is not necessarily intuitive, particularly to novice users, because the cage vertices do not lie on the surface, do not have a coarse structure, and are not semantically consistent across different shapes. We propose keypoints as an intuitive handle to manipulate the cages.

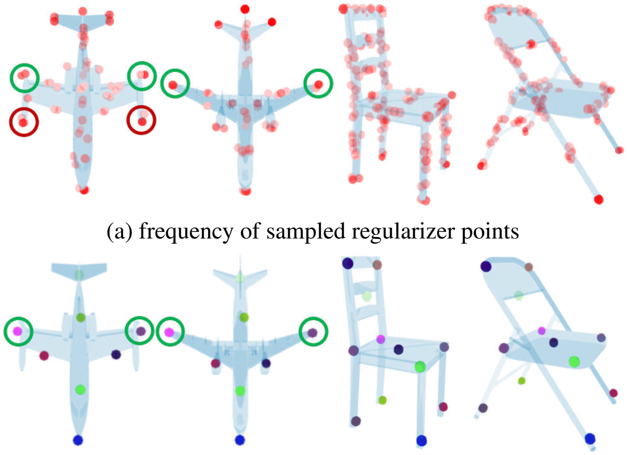
In order to control the object deformation using our discovered keypoints, we need to associate them with the cage vertices. We do so with a linear skinning function that takes the relative differences between the source and target keypoints $\Delta\boldsymbol{p} = \boldsymbol{p}' - \boldsymbol{p}$ and associates each of them with the source cage vertices $\boldsymbol{c}_V$ using an influence matrix $W \in \mathbb{R}^{C \times K}$ that we learn in an end-to-end manner, where C is the number of cage vertices and K is the number of discovered keypoints. The resulting deformed cage vertices $\boldsymbol{c}^*_V$ are then defined as

$$\boldsymbol{c}^*_V = \boldsymbol{c}_V + W\Delta\boldsymbol{p}. \qquad (1)$$

In order to adjust for the fact that cages are unique to each shape, we represent the influence matrix as a function of the input shape $\boldsymbol{x}$. Specifically, the influence matrix is a composition $W(\boldsymbol{x}) = W_C + W_I(\boldsymbol{x})$ of a canonical W_C matrix that is shared with all instances of the object category and an instance specific offset W_I that is predicted from the source shape $\boldsymbol{x}$ using an influence predictor $W_I = \Gamma(\boldsymbol{x})$. We regularize the instance specific W_I matrix by minimizing its Frobenius norm to prevent overfitting of the resulting influence matrix W. We denoted this regularizer as $\mathcal{L}_{\text{inf}}$. Finally, we limit the matrix W to only influence at most M nearest cage vertices per each keypoint to encourage locality.

3.2. Losses and Regularizers

Our KeypointDeformer is trained end-to-end with stochastic gradient descent by minimizing a similarity loss between the source and target shape, as well as a keypoint regularization term and instance-specific influence matrix regularization term.



(a) frequency of sampled regularizer points

(b) unsupervised keypoints predictions

Figure 3: **Farthest Point Keypoint regularizer.** We use farthest point sampling with a random starting point to regularize the predicted keypoints. **(a)** illustrates the frequency of a given point being sampled by the farthest point sampling algorithm. Darker colours indicate higher probability of a point being sampled. The expected locations of sampled points provide good coverage and inherently follow the symmetry of the original shape. Also, a subset of them tend to be semantically stable across different object instances. Using expected sample locations as a prior for keypoint location works well as the keypoint predictor will learn to be robust to noise in these sampled points. This can be seen in the example of the airplane where the tips on the fuel tanks (shown in **red** circle) are ignored, and the keypoints are instead predicted **(b)** at the wingtip (shown in **green** circle) location that is more consistent across the dataset (most planes have wings, but many lack fuel tanks).

Similarity loss. Ideally, we would like to compute the similarity between the deformed source shape $\boldsymbol{x}$ and the target shape $\boldsymbol{x}'$ using known correspondences between the meshes. However, such correspondence is not available since we aim to train on generic collections of object category CAD models. We approximate the similarity loss by computing the Chamfer distance between the deformed source $\boldsymbol{x}^*$ and the target shape $\boldsymbol{x}'$ represented as point clouds. We denote this loss as $\mathcal{L}_{\text{sim}}$.

Farthest Point Keypoint regularizer. We propose a simple, yet highly effective keypoint regularizer $\mathcal{L}_{\text{kpt}}$ that encourages predicted keypoints $\boldsymbol{p}$ to be well-distributed, lie on the object surface, and preserve the symmetric structure of the underlying shape category. Specifically, we devise a Farthest Sampling Algorithm to sample an unordered set of points $\boldsymbol{q} = \{q_1, \ldots, q_J\} \in \mathbb{R}^{3 \times J}$ from the input shape $\boldsymbol{x}$ represented as a point cloud. The initial point for sampling is chosen at random, and hence each time we compute this regularization loss a different set of sampled points $\boldsymbol{q}$ is used. Given these stochastic farthest points, the regularizer minimizes the Chamfer distance between the predicted keypoints

$\boldsymbol{p}$ and sampled points $\boldsymbol{q}$. In other words, the regularizer encourages the keypoint predictor Φ to place the discovered keypoints $\boldsymbol{p}$ at the expectation of the sampled farthest points $\boldsymbol{q}$. Figure 3 illustrates the properties of the sampled regularizer points. The sampled points provide equally spaced coverage of the input object shape $\boldsymbol{x}$, are relatively stable across different instances, and preserve the symmetric structure of the original input shapes.

Another intuition behind this regularization is that we can consider the sampled farthest points $\boldsymbol{q}$ as a noisy prior over keypoint locations. This prior is not perfect—it may miss important points in some models, or place spurious points in others—but the neural network keypoint predictor will learn keypoints in a way that is robust to such noise, and instead, prefer to predict keypoints at consistent locations, as demonstrated in Figure 3.

Full objective. In summary, our full training objective is

$$\mathcal{L} = \mathcal{L}_{\text{sim}} + \alpha_{\text{kpt}}\mathcal{L}_{\text{kpt}} + \alpha_{\text{inf}}\mathcal{L}_{\text{inf}} \tag{2}$$

where α_{kpt} and α_{inf} are scalar loss coefficients. Our method is simple and does not require additional shape specific regularization for shape deformation, such as the point-to-surface distance, normal consistency, and symmetry losses employed in [30]. This is due to the fact that keypoints provide a low-dimensional correspondence between shapes and that cage deformations are a linear function of these keypoints, preventing extreme deformations that result in unwanted local shape deformations.

3.3. Categorical Shape Prior

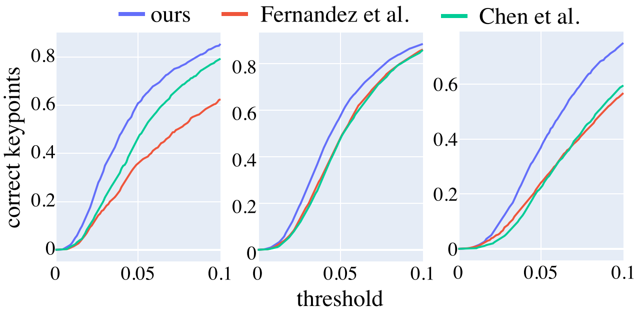
Since we represent an object shape as a set of semantically consistent keypoints, we can obtain a categorical shape prior by computing PCA on the keypoints predicted on the training set. This prior can be used to guide keypoint manipulation. For example, if user edits a single keypoint on an airplane wing, the remaining keypoints can be "synchronized" according to a prior by finding the PCA basis coefficients that best reconstruct the new position of the edited keypoint. The resulting reconstructed set of keypoints follow the prior defined by the data. This prior also allows sampling of novel shapes via sampling a new set of keypoints. This set of keypoints can be then used to deform the shape using our deformation model in order to, for instance, automatically augment libraries of stock 3D models.

4. Experiments

The main objectives of our experiments are to evaluate whether (1) our discovered keypoints are in general of good quality as keypoints (Section 4.2), (2) our discovered keypoints are better suited for shape deformation than other keypoints (Section 4.3), and (3) our method allows for intuitive shape control (Section 4.4). The supplementary material contains extended version of results and ablation studies.

	airplane	car	chair	motorbike	table
Chen *et al.* [3]	0.69	0.39	0.78	0.91	0.75
Fernandez *et al.* [4]	0.78	0.66	0.80	0.90	0.85
ours	**0.85**	**0.73**	**0.88**	**0.93**	**0.92**

Table 1: **Semantic part correspondence.** We report the average unsupervised keypoints correlation for each category. ↑ is better. Extended version with additional categories and detailed correlation tables can be found in the supplementary.



(a) airplane (b) car (c) chair

	airplane	car	chair
Chen *et al.* [3]	0.49	0.46	0.22
Fernandez *et al.* [4]	0.36	0.47	0.24
ours	**0.61**	**0.56**	**0.37**

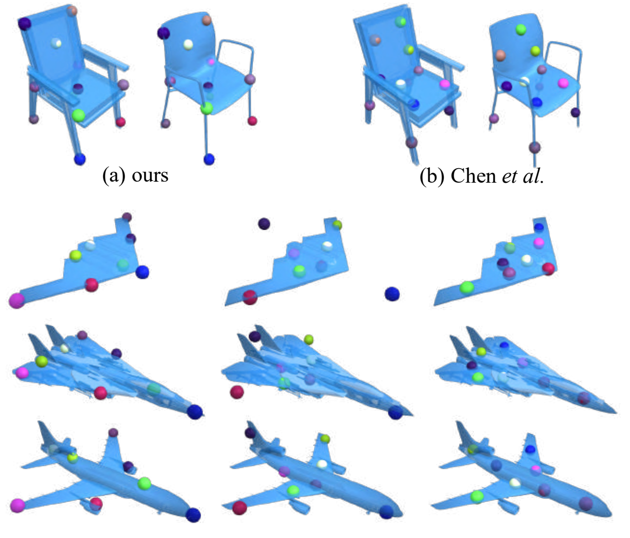
(d) Percentage of correct keypoints (PCK) @0.05

Figure 4: **Unsupervised 3D keypoints accuracy.** We measure the semantic consistency of keypoints following [26]. We train a linear regressor to predict manually annotated keypoints from unsupervised keypoints. The regressor accuracy on the test set estimates the semantic consistency of the underlying unsupervised keypoints. We show results in terms of PCK for airplane, car and chair category on the KeypointNet dataset [31].

4.1. Experimental Setup

Datasets. We train our KeypointDeformer using ShapeNet [2] following the standard training and testing split. We normalize all the shapes into a unit box. For evaluation, we use semantic part annotations for ShapeNet [29], as well as the KeypointNet [31] dataset, which contains semantic keypoint annotations for selected ShapeNet categories. Note that our method does not require any of these annotations for training. We also evaluate KeypointDeformer on real-world 3D scans of shoes from Google Scanned Objects dataset [7].

Implementation details. The keypoint predictor Φ and the influence predictor Γ are implemented as neural networks using a PointNet encoder and the whole model is optimized using the Adam optimizer. We use 1024 sampled points for the point cloud representation of shape $\boldsymbol{x}$. Unless otherwise mentioned, we predict 12 unsupervised keypoints for all



(a) ours

(b) Chen *et al.*

(c) ours

(d) Fernandez *et al.*

(e) Chen *et al.*

Figure 5: **Unsupervised 3D keypoints.** We compare our unsupervised 3D keypoints with Fernandez *et al.* [4] and Chen *et al.* [3]. Our keypoints are more semantically consistent despite large shape variations when compared to other methods. Keypoints obtained by Fernandez *et al.* [4] do not explain all the shapes well. Moreover, our keypoints are symmetrical without explicitly enforcing that in contrast with [4]. We show results on additional categories in the supplementary.

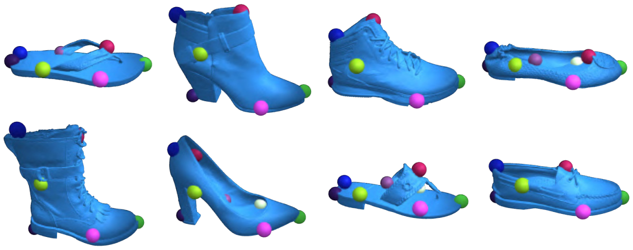
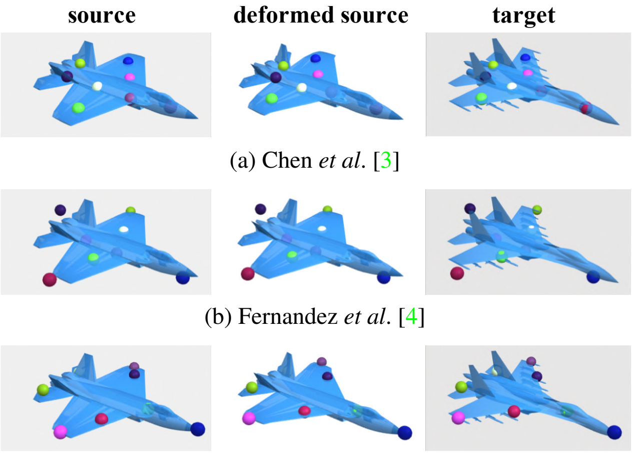


Figure 6: **Unsupervised 3D keypoints on real-world data.** We run our unsupervised keypoint detector on real-world scans of shoes [7]. The keypoints are semantically consistent across different shapes.

categories except for airplane and car where we use 8. The supplementary contains an ablation studying the effect of different number of unsupervised keypoints. We set the number of sampled farthest points $\boldsymbol{q}$ to the double of the number of keypoints. Detailed descriptions of network architectures and training details are in the supplemental material.

4.2. Semantic Consistency

We first demonstrate the quality of our unsupervised keypoints by evaluating their semantic consistency, *i.e.* whether they always correspond to the same semantic object parts or not. For instance, if a keypoint is predicted on the tip of the wing on one instance of an airplane, then that same keypoint should always correspond to the tip of the wing across different instances. For this task we compare with recently introduced methods for unsupervised keypoint discovery from Fernandez *et al.* [4] and Chen *et al.* [3].



(a) Chen *et al.* [3]

(b) Fernandez *et al.* [4]

(c) ours

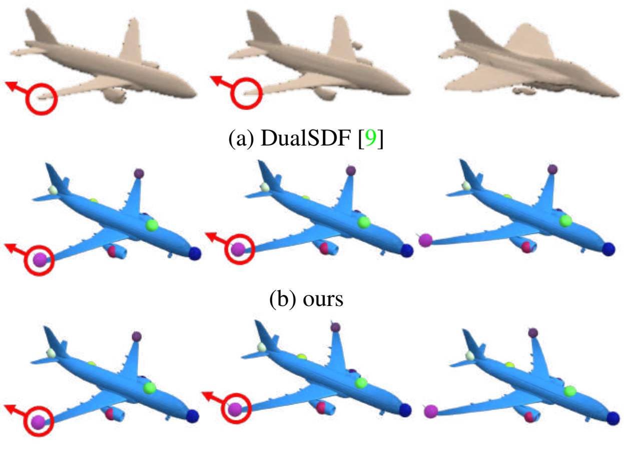
	Fernandez *et al.* [4]	Chen *et al.* [3]	annotations [31]	ours
CD	7.55	5.93	4.20	**3.02**

(d) Chamfer distance between deformed source and target

Figure 7: **Keypoints for shape deformation.** We replace our discovered keypoints in KeypointDeformer to compare with different keypoints detectors and manually annotated keypoints on keypoint-guided pairwise shape alignment for the airplane category. The degree of alignment is measured by the Chamfer distance between the deformed source and target shapes. Our discovered keypoints can align shapes better even when compared to manually selected keypoints from KeypointNet [31]. Keypoints from Fernandez *et al.* [4] and Chen *et al.* [3] fail to accurately align shapes as their keypoints are less precise. Data in the table are scaled by 10^3.

We evaluate semantic consistency using two protocols. First, we use an evaluation protocol of Fernandez *et al.* [4]. Since their evaluation is very coarse, we also follow an evaluation protocol for unsupervised keypoints established by Thewlis *et al.* [26].

The evaluation protocol of Fernandez *et al.* [4] employs the ShapeNet dataset with part annotations to measure the correlation between each keypoint and annotated semantic object parts across instances of the category. Each keypoint is associated with the nearest object part. This protocol has two limitations. First, a keypoint can be associated with an object part even if it lies far from the object (indicating a poor choice of keypoint). Second, this protocol does not account for boundary keypoints that are predicted just between two annotated object parts (which can still be high-quality, salient keypoints). To address these limitations, we propose a small modification to this protocol, in which we associate each keypoint with a given object part if it lies within its small



(a) DualSDF [9]

(b) ours

(c) ours with shape prior

Figure 8: **Comparison with DualSDF [9].** We move the wing tip in the direction of the red arrow. (a) DualSDF is a generative model and changing the position of the wing tip results in a change from an airliner to a jet fighter. In contrast, our method preserves the original structure of the mesh and allows for asymmetric manipulation when desired (b). Our method can also work in conjunction with a shape prior (Section 4.5) to achieve symmetrical manipulation (c).

neighborhood (0.05 from the object part when the object is normalized to unit box)—hence, a keypoint can be associated with multiple parts. For each keypoint, we compute its correlation with each object part. Since a keypoint can be associated with multiple parts, we consider only the most correlated part for a given keypoint in the final metric. The final metric then computes the average correlation over all the keypoints. We report semantic consistency results for ShapeNet categories in Table 1. Our keypoints show better average correlation when compared to Chen *et al*. [3] and Fernandez *et al*. [4].

Second, we adopt the standard unsupervised 2D keypoint evaluation protocol as in [26, 33, 10], since the semantic object parts are coarsely annotated (e.g., the airplane category comes with only 3 semantic parts). The objective of this protocol is to measure how predictive unsupervised keypoints are of semantic keypoints selected by humans. This is done by finding a linear mapping between the unsupervised keypoints and manually annotated ones. The linear mapping is established on the training set by fitting a linear regressor. The predictiveness of unsupervised keypoints is then measured in terms of this regressor's prediction error on the test set. We use the recent KeypointNet dataset [31], which contains semantic annotations on ShapeNet dataset. We report the performance in Figure 4. Our unsupervised keypoints are more predictive of manually annotated keypoints than other unsupervised keypoint. Figure 5 provides qualitative comparison of our unsupervised keypoints with those obtained by other methods.

Real world scans. We also demonstrate applicability of our unsupervised keypoint detector on real-world 3D scans of objects. We use the shoe category from Google Scanned Objects dataset [7]. We align the shapes using the automatical alignment method from [16]. We split the dataset into training and test sets with 219 and 36 samples respectively. We use the same hyper-parameters as done in experiments on ShapeNet. Figure 6 shows that our method learns semantically consistent 3D keypoints for shoes with largely different shapes.

4.3. Keypoints for Shape Deformation

To quantitatively demonstrate that controlled shape deformation is possible through unsupervised keypoints, we use the task of pairwise shape alignment, in which we deform a source shape into a target shape. In our case, the deformation is guided using keypoints. This task also evaluates that our discovered keypoints are more suitable for shape control than other keypoints. We modify our method by replacing our unsupervised keypoints with keypoints obtained from other methods. We then train our deformation model from scratch. We experiment with keypoints from [4], [3], and also manually annotated keypoints from [31]. Performance is evaluated by measuring the Chamfer distance between the deformed source shape and the target shape. We present results in Figure 7. The unsupervised keypoints obtained by other methods fail to capture the large variations in shapes in the dataset. Our keypoints, on the other hand, can follow the large changes in shapes. This ultimately leads to more accurate shape deformations.

4.4. Shape Control via Unsupervised 3D Keypoints

Our ultimate goal is to use automatically discovered keypoints to perform user-guided interactive shape deformation. Figure 9 shows interactive shape control using our unsupervised keypoints. Our method provides low-dimensional handles to control object shape. The control is intuitive as the deformation is semantically consistent, e.g., moving a keypoint on the leg of a chair or airplane wing results in movement of that object part in the same direction. Thus the user can easily edit shape meshes. Please refer to our project page for a demo video showcasing user-guided interactive shape control using keypoints.

The related work DualSDF [9] also allows for user-guided interactive shape deformation. However, the key distinction here is that DualSDF is a conditional generative model. Manipulating an object through its handle generates a new shape that respects the new position of the handle specified by the user, but the new generated shape can be very different from the original one. This aspect is illustrated in Figure 8, where DualSDF transforms an airliner to a jet fighter.

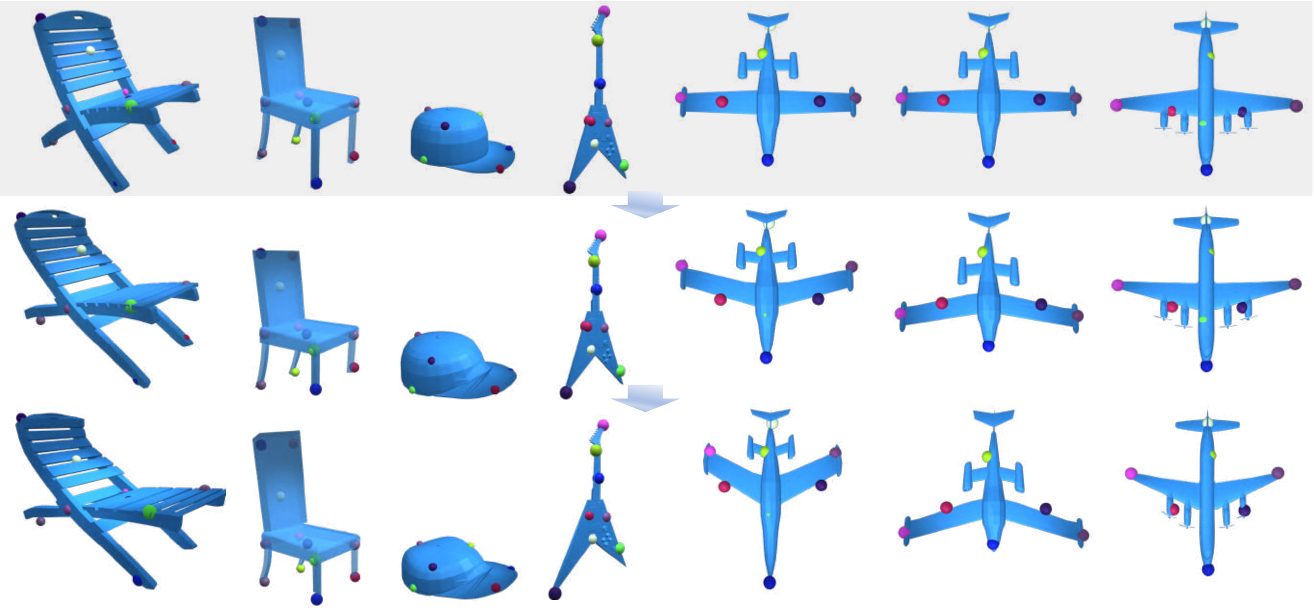


Figure 9: **Interactive shape control via 3D unsupervised keypoints.** We show iterative steps in user guided shape deformation using our discovered keypoint as handles. Top row shows initial state. Please refer to our project page for a demo video.

4.5. Categorical Shape Prior

Since our deformation model uses keypoints as its low-dimensional shape representation, we can compute categorical shape prior on them. We compute PCA on the set of predicted keypoints obtained from the training set. We set the number of basis to 8. As discussed in Section 4.5, we use the prior in two ways. First, we can use it during interactive shape control when the user manipulates only a single keypoint, to "synchronize" the rest of the keypoints according to the prior. This "synchronized" editing is used in Figure 8 where we drag only a single keypoint and the rest get automatically readjusted. Second, we can easily sample new deformations using sampled keypoints that we obtain by varying PCA basis coefficients. This can be applied to automatic dataset amplification as demonstrated Figure 10.

5. Conclusion

We present a method for controlling the shape of 3D objects through automatically discovered semantic 3D keypoints and a deformation model learned jointly with the keypoints. The resulting KeypointDeformer model provides users with a simple interface for interactive shape control. One limitation of the method is that our approach assumes aligned shape collections. However, in our experiments with real scans, automatic alignment method was sufficient. Another limitation is that the keypoint representation does not allow modeling of individual object part rotations. In this work we focused on the task of shape control and keypoint prediction, however 3D keypoints has various usage in other applications such as robotics [17, 18]. It would be interesting to explore the applicability of our unsupervised 3D keypoints to other tasks in the future.

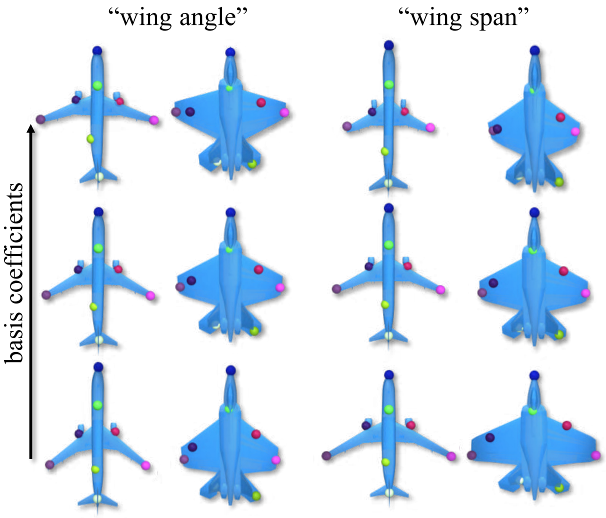


Figure 10: **Varying PCA basis coefficients for shape augmentation.** We sample new keypoints by varying its PCA basis coefficients. The sampled keypoints are used to deform the original shape obtaining a new set of shapes. The left two columns show results for a subspace that correlates with the wing angle. The right two columns show results for a subspace that correlates with the wing span.

References

[1] Mario Botsch, Mark Pauly, Markus H Gross, and Leif Kobbelt. PriMo: coupled prisms for intuitive surface modeling. In *Symposium on Geometry Processing*, pages 11–20, 2006. 2

[2] 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. 5

[3] Nenglun Chen, Lingjie Liu, Zhiming Cui, Runnan Chen, Duygu Ceylan, Changhe Tu, and Wenping Wang. Unsupervised learning of intrinsic structural representation points. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9121–9130, 2020. 3, 5, 6, 7

[4] Clara Fernandez-Labrador, Ajad Chhatkuli, Danda Pani Paudel, Jose J Guerrero, Cédric Demonceaux, and Luc Van Gool. Unsupervised learning of category-specific symmetric 3d keypoints from point sets. *European Conference on Computer Vision (ECCV)*, 2020. 3, 5, 6, 7

[5] Matheus Gadelha, Giorgio Gori, Duygu Ceylan, Radomir Mech, Nathan Carr, Tamy Boubekeur, Rui Wang, and Subhransu Maji. Learning generative models of shape handles. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 2

[6] Wei Gao and Russ Tedrake. kpam-sc: Generalizable manipulation planning using keypoint affordance and shape completion. *arXiv preprint arXiv:1909.06980*, 2019. 3

[7] GoogleResearch. Google scanned objects. https://fuel.ignitionrobotics.org/1.0/GoogleResearch/fuel/collections/Google%20Scanned%20Objects, September 2020. 5, 6, 7

[8] Rana Hanocka, Noa Fish, Zhenhua Wang, Raja Giryes, Shachar Fleishman, and Daniel Cohen-Or. Alignet: Partial-shape agnostic alignment via unsupervised learning. *ACM Transactions on Graphics (TOG)*, 38(1):1–14, 2018. 2

[9] Zekun Hao, Hadar Averbuch-Elor, Noah Snavely, and Serge Belongie. Dualsdf: Semantic shape manipulation using a two-level representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7631–7641, 2020. 2, 3, 7

[10] Tomas Jakab, Ankush Gupta, Hakan Bilen, and Andrea Vedaldi. Unsupervised learning of object landmarks through conditional image generation. In *Advances in neural information processing systems*, pages 4016–4027, 2018. 3, 7

[11] Tomas Jakab, Ankush Gupta, Hakan Bilen, and Andrea Vedaldi. Self-supervised learning of interpretable keypoints from unlabelled videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8787–8797, 2020. 3

[12] Pushkar Joshi, Mark Meyer, Tony DeRose, Brian Green, and Tom Sanocki. Harmonic coordinates for character articulation. *ACM Transactions on Graphics (TOG)*, 26(3):71–es, 2007. 4

[13] Tao Ju, Scott Schaefer, and Joe Warren. Mean value coordinates for closed triangular meshes. In *SIGGRAPH*, pages 561–566. 2005. 2, 4

[14] A Sophia Koepke, Olivia Wiles, and Andrew Zisserman. Self-supervised learning of a facial attribute embedding from video. In *BMVC*, page 302, 2018. 3

[15] Yaron Lipman, David Levin, and Daniel Cohen-Or. Green coordinates. *ACM Transactions on Graphics (TOG)*, 27(3):1–10, 2008. 4

[16] Ameesh Makadia and Kostas Daniilidis. Rotation recovery from spherical images without correspondences. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 28(7):1170–1175, 2006. 7

[17] Lucas Manuelli, Wei Gao, Peter Florence, and Russ Tedrake. kpam: Keypoint affordances for category-level robotic manipulation. *arXiv preprint arXiv:1903.06684*, 2019. 3, 8

[18] Lucas Manuelli, Yunzhu Li, Pete Florence, and Russ Tedrake. Keypoints into the future: Self-supervised correspondence in model-based reinforcement learning. *arXiv preprint arXiv:2009.05085*, 2020. 8

[19] Eloi Mehr, Ariane Jourdan, Nicolas Thome, Matthieu Cord, and Vincent Guitteny. Disconet: Shapes learning on disconnected manifolds for 3d editing. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3474–3483, 2019. 2

[20] Tiberiu Popa, Dan Julius, and Alla Sheffer. Material-aware mesh deformations. In *IEEE International Conference on Shape Modeling and Applications 2006 (SMI'06)*, pages 22–22. IEEE, 2006. 2

[21] Olga Sorkine. Differential representations for mesh processing. *Computer Graphics Forum*, 25(4):789–807, 2006. 2

[22] Olga Sorkine and Marc Alexa. As-rigid-as-possible surface modeling. In *Symposium on Geometry processing*, volume 4, pages 109–116, 2007. 2

[23] Robert W Sumner and Jovan Popović. Deformation transfer for triangle meshes. *ACM Transactions on graphics (TOG)*, 23(3):399–405, 2004. 2

[24] 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. 3

[25] James Thewlis, Samuel Albanie, Hakan Bilen, and Andrea Vedaldi. Unsupervised learning of landmarks by descriptor vector exchange. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 6361–6371, 2019. 3

[26] James Thewlis, Hakan Bilen, and Andrea Vedaldi. Unsupervised learning of object landmarks by factorized spatial embeddings. In *Proceedings of the IEEE international conference on computer vision*, pages 5916–5925, 2017. 3, 5, 6, 7

[27] Shubham Tulsiani, Hao Su, Leonidas J Guibas, Alexei A Efros, and Jitendra Malik. Learning shape abstractions by assembling volumetric primitives. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2635–2643, 2017. 2

[28] Weiyue Wang, Duygu Ceylan, Radomir Mech, and Ulrich Neumann. 3dn: 3d deformation network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1038–1046, 2019. 2

[29] Li Yi, Vladimir G Kim, Duygu Ceylan, I-Chao Shen, Mengyan Yan, Hao Su, Cewu Lu, Qixing Huang, Alla Sheffer, and Leonidas Guibas. A scalable active framework for region annotation in 3d shape collections. *ACM Transactions on Graphics (ToG)*, 35(6):1–12, 2016. 5

[30] Wang Yifan, Noam Aigerman, Vladimir G Kim, Siddhartha Chaudhuri, and Olga Sorkine-Hornung. Neural cages for detail-preserving 3d deformations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 75–83, 2020. 2, 4, 5

[31] Yang You, Yujing Lou, Chengkun Li, Zhoujun Cheng, Liangwei Li, Lizhuang Ma, Cewu Lu, and Weiming Wang. Keypointnet: A large-scale 3d keypoint dataset aggregated from numerous human annotations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13647–13656, 2020. 5, 6, 7

[32] Mehmet Ersin Yumer, Siddhartha Chaudhuri, Jessica K Hodgins, and Levent Burak Kara. Semantic shape editing using deformation handles. *ACM Transactions on Graphics (TOG)*, 34(4):1–12, 2015. 2

[33] Yuting Zhang, Yijie Guo, Yixin Jin, Yijun Luo, Zhiyuan He, and Honglak Lee. Unsupervised discovery of object landmarks as structural representations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2694–2703, 2018. 3, 7