Joint Learning of 3D Shape Retrieval and Deformation

Mikaela Angelina Uy[1] Vladimir G. Kim[2] Minhyuk Sung[3] Noam Aigerman[2]
Siddhartha Chaudhuri[2,4] Leonidas Guibas[1]
[1]Stanford University [2]Adobe Research [3]KAIST [4]IIT Bombay

Abstract

We propose a novel technique for producing high-quality 3D models that match a given target object image or scan. Our method is based on retrieving an existing shape from a database of 3D models and then deforming its parts to match the target shape. Unlike previous approaches that independently focus on either shape retrieval or deformation, we propose a joint learning procedure that simultaneously trains the neural deformation module along with the embedding space used by the retrieval module. This enables our network to learn a deformation-aware embedding space, so that retrieved models are more amenable to match the target after an appropriate deformation. In fact, we use the embedding space to guide the shape pairs used to train the deformation module, so that it invests its capacity in learning deformations between meaningful shape pairs. Furthermore, our novel part-aware deformation module can work with inconsistent and diverse part-structures on the source shapes. We demonstrate the benefits of our joint training not only on our novel framework, but also on other state-of-the-art neural deformation modules proposed in recent years. Lastly, we also show that our jointly-trained method outperforms various non-joint baselines.

1. Introduction

Creating high-quality 3D models from a reference image or a scan is a laborious task, requiring significant expertise in 3D sculpting, meshing, and UV layout. While neural generative techniques for 3D shape synthesis hold promise for the future, they still lack the ability to create 3D models that rival the fidelity, level of detail, and overall quality of artist-generated meshes [10]. Several recent techniques propose to directly retrieve a high-quality 3D model from a database and deform it to match a target image or point cloud, thereby approximating the target shape while preserving the quality of the original source model. These prior methods largely focus on one of two complementary subproblems: either retrieving an appropriate mesh from a database [5, 1], or training a neural network to deform a source to a target [2, 13, 14, 8]. In most cases, the static database mesh most closely matching the target is retrieved, and then deformed for a better fit [4]. The retrieval step is not influenced by the subsequent deformation procedure, and thus ignores the possibility that a database shape with different global geometry nevertheless possess local details that will produce the best match *after* deformation.

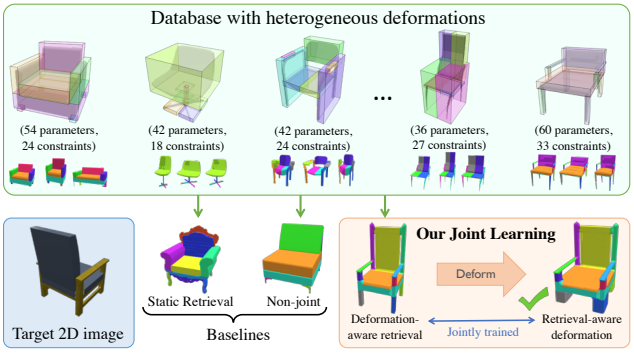

Figure 1. Given an input target we use jointly-learned retrieval and deformation modules to find a source model in a heterogeneous database and align it to the target. We demonstrate that our joint learning outperforms static retrieval and non-joint baselines.

In this paper, we argue that retrieval and deformation should be *equal citizens in a joint problem*. Given a database of source models equipped with some parametric representation of deformations, our goal is to learn how to retrieve a shape from the database and predict the optimal deformation parameters so it best matches a given target. A key feature of our method is that both retrieval and deformation are *learnable* modules, each influencing the other and trained jointly. While the benefit of deformation-aware retrieval [11] has been explored previously, we contribute the notion of *retrieval-aware deformation*: our learnable deformation module is optimized for fitting retrieved shapes to target shapes. Thus, the retrieval module is optimized to retrieve sources that the deformation module can fit well to the input target, and the deformation module is trained on sources the retrieval module predicts for the input target, thereby letting it optimize capacity and learn only meaningful deformations.

We further evise a *differentiable, part-aware deformation function* that deforms individual parts of a model while respecting the part-to-part connectivity of the original struc-

ture (Figure 1). Importantly, it accommodates varying numbers of parts and structural relationships across the database, and does not require part labels or consistent segmentations. This holistic view of joint retrieval and deformation is especially important when considering heterogeneous collections of shapes "in the wild" that often vary in their part structure, topology, and geometry. We evaluate our method by matching 2D image and 3D point cloud targets and demonstrate that our approach outperforms various baselines. The full paper can be found at [12].

2. Method

Overview. We assume to possess a database of parametric *source* models $\mathbf{s} \in \mathbf{S}$, and we aim to jointly train a deformation and retrieval module to choose a source and deform it to fit a given target $\mathbf{t}$ (an image or a point cloud), with respect to a fitting metric $\mathcal{L}_{\text{fit}}$ (we use chamfer in all experiments). Each source also has parameters defining its individual deformation space, that are optimized during training.

Our deformation module is designed to enable a different deformation function $\mathcal{D}_{\mathbf{s}}$ for each source $\mathbf{s}$, based on its parts. The retrieval module uses embeddings of the sources and the target into a latent space $\mathcal{R}$, where it retrieves based on a distance measure $\mathrm{d}_{\mathcal{R}}$, which enables the retrieval of the source shape that best fits to the target *after* deformation.

The training consists of optimizing the latent retrieval space $\mathcal{R}$ and the deformation functions $\{\mathcal{D}_{\mathbf{s}}\}$:

$$\min \mathcal{L}_{\text{fit}}\left(\mathcal{D}_{s'}(\mathbf{t}), \mathbf{t}_{\text{true}}\right),$$

where s' is the closest source to target $\mathbf{t}$ in latent space, w.r.t the distance measure $d_{\mathcal{R}}(\mathbf{s}', \mathbf{t})$, and $\mathbf{t}_{\text{true}}$ is the corresponding true shape.

2.1. Joint Deformation and Retrieval Training

It is critical for our approach to optimize the parameters of $\mathcal{R}$ and $\{\mathcal{D}_{\mathbf{s}}\}$ jointly. First, it enables the deformation module of each source to efficiently utilize its capacity and specialize on relevant targets that it could fit to. Second, it allows the retrieval module to create a deformation-aware latent space where sources are embedded closer to the targets they can deform to.

Soft Retrieval for Training. The retrieval module embeds the sources and the target in the latent retrieval space $\mathcal{R}$. The proximity in latent space is used to define a biased distribution that can be loosely interpreted as the probability of source $\mathbf{s}$ being deformable to $\mathbf{t}$:

$$\mathrm{P}_{\mathcal{R}}(\mathbf{s}, \mathbf{t}) = \mathrm{p}(\mathbf{s}; \mathbf{t}, \mathbf{S}, \mathrm{d}_{\mathcal{R}}, \sigma_0), \tag{1}$$

where

$$\mathrm{p}(\mathbf{s}; \mathbf{t}, \tilde{\mathbf{S}}, \tilde{\mathrm{d}}, \tilde{\sigma}) = \frac{\exp(-\tilde{\mathrm{d}}^2(\mathbf{s}, \mathbf{t})/\tilde{\sigma}^2(\mathbf{s}))}{\sum_{\mathbf{s}' \in \tilde{\mathbf{S}}} \exp(-\tilde{\mathrm{d}}^2(\mathbf{s}', t)/\tilde{\sigma}^2(\mathbf{s}))},$$

$\tilde{\mathrm{d}} : (\mathbf{S} \times \mathbf{T}) \to \mathbb{R}$ is a distance function between a source and a target ($\mathbf{T}$ is the target space), and $\tilde{\sigma} : \mathbf{S} \to \mathbb{R}$ is a potentially source-dependent scalar function. Though, $\sigma_0(\cdot) = 100$ is a constant set for all experiments.

Instead of choosing the highest-scoring source according to the probability $\mathrm{P}_{\mathcal{R}}$, we perform soft retrieval and *sample* $K = 10$ retrieval candidate sources from the distribution:

$$\mathbf{s}_i \sim \mathrm{P}_{\mathcal{R}}(\mathbf{s}, \mathbf{t}), \forall i \in \{1, 2, ..., K\}.$$

The candidates $\mathbf{S}_{\mathbf{t}} = \{\mathbf{s}_1, ..., \mathbf{s}_K\}$ sampled via our soft retrieval are then used to train both our retrieval module to learn $\mathcal{R}$ and deformation module for source-depedent deformation functions $\{\mathcal{D}_{\mathbf{s}}\}$.

The soft retrieval is crucial for our training: 1) adding randomness to the retrieval ensures that the latent space is optimized with respect to both high-probability instances and low-probability ones, that may reverse roles as the deformation module improves. 2) On the other hand, training the deformation module with a bias towards high-probability sources and not random ones ensures it is aware of the retrieval module and expands its capacity on meaningful matches.

Training. We train the two modules jointly in an alternating fashion, keeping one module fixed when optimizing the other, and vice versa, in successive iterations. To train the retrieval module, we deform the candidate sources and compute their fitting losses to the target. We update our latent space $\mathcal{R}$ by penalizing the discrepancy between the distances in the retrieval space $d_{\mathcal{R}}$ and the post-deformation fitting losses $\mathcal{L}_{\text{fit}}$ using softer probability measures estimated from the distances of the sampled candidates:

$$\mathcal{L}_{\text{emb}} = \sum_{k=1}^{K} |\, \mathbf{p}(\mathbf{s}_k, \mathbf{t}, \mathbf{S}_{\mathbf{t}}, d_{\mathcal{R}}, \sigma_0) - \mathbf{p}(\mathbf{s}_k, \mathbf{t}, \mathbf{S}_{\mathbf{t}}, d_{\text{fit}}, \sigma_k)|, \tag{2}$$

where

$$d_{\text{fit}}(\mathbf{s}, \mathbf{t}) = \mathcal{L}_{\text{fit}}(\mathcal{D}_{\mathbf{s}}(\mathbf{t}), \mathbf{t}_{\text{true}}), \tag{3}$$

and σ_k is a source-dependent scalar representing the predicted range of variations of each source model $\mathbf{s} \in \mathbf{S}$, which is also learned. For the deformation module, we update the deformation functions $\{\mathcal{D}_{\mathbf{s}_k}\}$ for the K biased samples by minimizing the post-deformation fitting losses weighted by their soft probability measures:

$$\mathcal{L}_{\text{def}} = \sum_{k=1}^{K} \mathbf{p}(\mathbf{s}_k, \mathbf{t}, \mathbf{S}_{\mathbf{t}}, d_{\mathcal{R}}, \sigma_0) \mathcal{L}_{\text{fit}}(\mathcal{D}_{\mathbf{s}_k}(\mathbf{t}), \mathbf{t}_{\text{true}}). \tag{4}$$

This weighting scheme puts greater weight on sources that are closer to the target in the embedding space, thus further making the deformation module aware of the retrieval module, and allowing it to specialize on more amenable sources with respect to the training target.

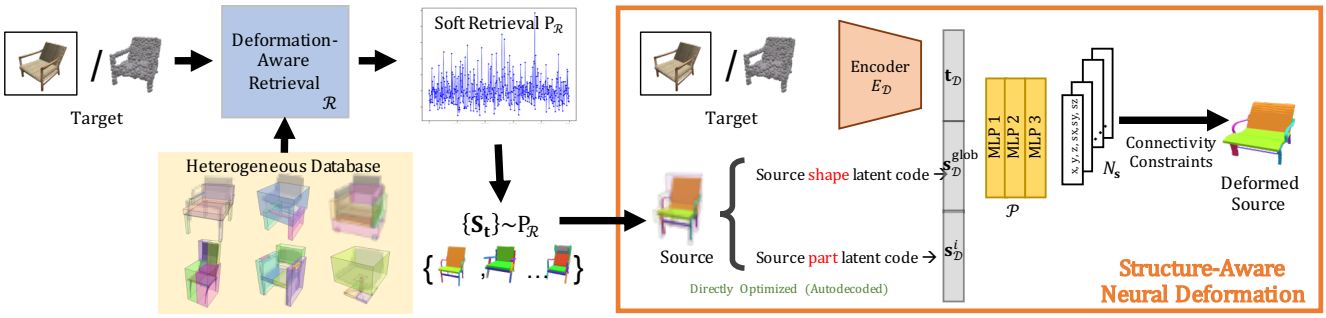

Figure 2. During training, given a target image or a point cloud and a database of deformable sources, we retrieve a subset of source models based on their proximity in the retrieval space, and use the structure-aware deformation module (right) to fit each source. Our deformation module uses encoded target, global and per-part source codes to predict per-part deformation parameters.

We also further run an *inner deformation optimization* to fit the source to the target shape, and directly run SGD on the deformation parameters until convergence of the fitting loss. See the supplementary for the full details.

2.2. Structure-Aware Neural Deformation

While our joint training approach described in Section 2.1 is generic and can work well with different parameterization of deformations, its greatest advantage is that it enables our deformation space to vary greatly between each source without having the deformation module learn subpar deformations. We thus devise a deformation module with a heterogeneous space of part-based deformations as shown in Figure 1, which vary per each source, a necessary feature if one wants to tailor the deformations to be restricted to preserve and adjust part structures.

To get meaningful parts, we use manual segmentations from PartNet [6] or automatic segmentations (preprocessing) of ComplementMe [9], produced by grouping connected components in raw meshes. Our deformation module predicts a simple deformation consisting of translation and axis-aligned scaling for each part in a source model. See supplementary for the details on the prediction. The number of parts for different sources vary, making the deformation functions source-dependent $\{\mathcal{D}_{\mathbf{s}}\}$. We abuse the notation a bit and let $\mathcal{D}$ denote our deformation module.

We propose to use a neural network which can be applied to each part separately, thus making it applicable to models with varying part-constellations, as opposed to previous methods. Namely, we assign to each source a global code $\mathbf{s}^{\text{glob}}_{\mathcal{D}} \in \mathbb{R}^{n_1}$, and for each part within the shape, we assign a local code $\mathbf{s}^{i=1\ldots N_s}_{\mathcal{D}} \in \mathbb{R}^{n_2}$. The target is encoded via an encoder (PointNet [7] for point clouds and ResNet [3] for images) into a latent vector $\mathbf{t}_{\mathcal{D}} = E_{\mathcal{D}}(\mathbf{t}) \in \mathbb{R}^{n_3}$. We set $n_1 = n_3 = 256$ and $n_2 = 32$ for all experiments. The global, local, and target codes are concatenated and fed to a lightweight 3-layer MLP (512, 256, 6), $\mathcal{P}$, which outputs the deformation parameters of the corresponding part. The deformation parameters of all parts are then used to obtain the final deformed source shape. Each source's global and local codes are optimized in an auto-decoder fashion during the training of the deformation module. Figure 2 (right) illustrates our module. We additionally add a symmetry loss in training our deformation module to enforce bilateral symmetry of the output deformed shapes as regularization, more details are found in the supplementary.

Figure 3. We test our trained method on online product images.

3. Results

In this section, we show results on the image-to-mesh set-up. Please see the full paper [12] or the supplementary material for more details and results.

3.1. Image-to-Mesh

We first test our system on product images "in the wild" as well as images from our test set and show qualitative results for retrieval and deformation in Figures 3 and 4. Note how retrieved results have compatible structure to the input, which then enables the deformation technique to match the source to the target. We quantitatively evaluate performance of our method and report chamfer distances in Table 1 (**Ours**) together with the chamfer distances with the inner deformation optimization (**Ours w/ IDO**). Since IDO step described significantly increases training time, we do not use it in ablations and comparisons.

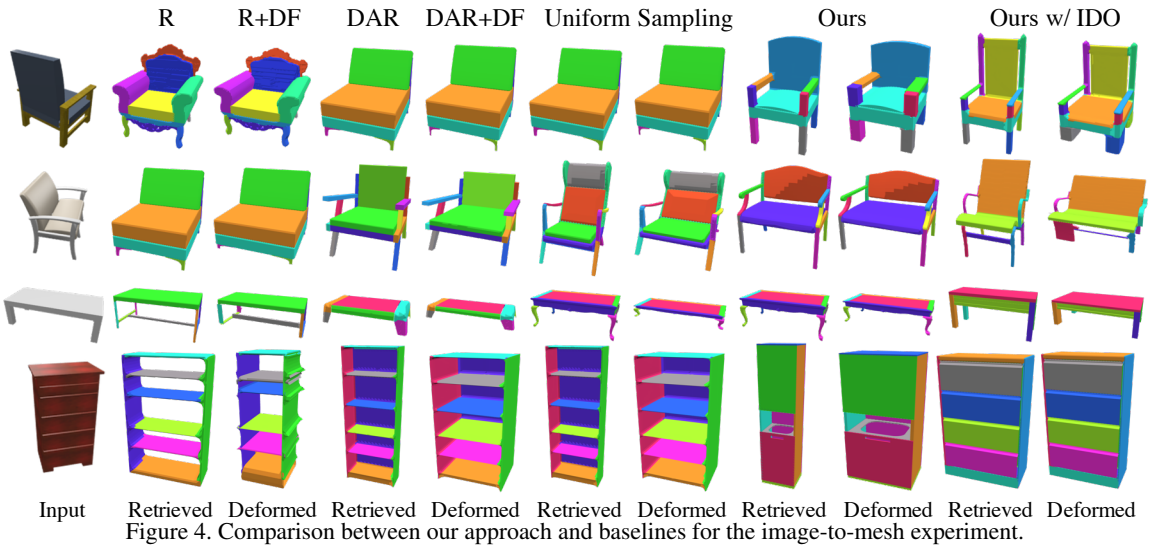

Figure 4. Comparison between our approach and baselines for the image-to-mesh experiment.

	Chair	Table	Cabinet
R	1.926	2.235	2.228
R+DF	1.969	2.705	2.035
DAR (Retrieval Only)	1.345	2.058	3.489
DAR+DF	1.216	1.621	1.333
Uniform Sampling	1.118	1.486	1.318
Ours	**1.005**	**0.970**	**1.220**
Ours w/ IDO	**0.976**	**0.935**	**1.141**

Table 1. Comparing our method to various baselines and ablations on image-to-mesh benchmark (chamfer distances, $\times 10^{-2}$).

Retrieval Baselines. We compare our method to a vanilla image-to-shape retrieval technique [5] (denoted by **R**). This baseline first constructs the latent space by projecting shape-to-shape chamfer distance matrix to 256-dimensional space via MDS, and then trains a ResNet [3] encoder to map images to that latent space with $L2$-loss. Since any retrieval baseline can also work with a pre-trained neural deformation, we also train our structure-aware deformation module on random pairs of shapes (i.e., ablating the joint training procedure) and report results with neural deformation applied to the retrieved results (**R+DF**). Since this vanilla baseline retrieves only based on geometric similarity and does not account for deformation, the retrieved shapes may not deform to targets well. Hence, there is no improvement when deforming with the pre-trained deformation function.

The second retrieval baseline is the deformation-aware retrieval [11], where we also use our structure-aware deformation module pre-trained on random pairs. For this baseline we report results for retrieval (**DAR**) as well as deformation (**DAR+DF**). Our results show that being deformation-aware is not sufficient, and it is important for deformation module to be trained with retrieved shapes.

Biased Sampling Ablation. Our joint training benefits from biasing sampling of retrieval targets (Eq. 1). To ablate this, we sample from a uniform distribution, i.e., each source is sampled with equal probability during training. In this setting, while the retrieval and deformation modules are still trained together, they are less aware of which samples are most relevant at inference time and thus yield higher errors (see **Uniform Sampling** in Table 1).

	Chair	Table	Cabinet
DF	0.748	0.702	0.706
Uniform Sampling	0.755	0.690	0.701
Ours	**0.681**	**0.584**	**0.675**
Ours w/ IDO	**0.669**	**0.533**	**0.689**

Table 2. Improvement in deformation module for image-to-mesh task with oracle retrieval due to joint training (chamfer $\times 10^{-2}$).

Improvement in Deformation Module. In addition to holistic improvement to the final output, we would like to evaluate the effect of joint training on deformation module. To do this, we use *oracle retrieval* where for each test target, we deform all sources and pick the one with the smallest fitting error. Our joint training allows the deformation module to specialize on targets that are a good fit. Thus, as shown in Table 2, our method achieves the lowest fitting error for the best-fit sources with respect to the deformation module trained on all pairs (**DF**), and the deformation module trained without the biased sampling (**Uniform Sampling**).

4. Conclusion

To summarize, we propose a joint training for retrieval-and-deformation problem, where the neural modules inform one another, yielding better matching results with respect to image and point cloud targets. Our joint training procedure offers improvements regardless of the choice of the neural deformation module. We further propose a novel structure-aware deformation module that is especially suitable for hetereogeneous datasets of source models with very diverse parameterizations of deformations. Our method does not require consistent manual segmentations or part labels and can work with imprecise automatic segmentations.

References

[1] Manuel Dahnert, Angela Dai, Leonidas Guibas, and Matthias Nießner. Joint embedding of 3d scan and cad objects. In *ICCV*, 2019. 1

[2] Thibault Groueix, Matthew Fisher, Vladimir G. Kim, Bryan C. Russell, and Mathieu Aubry. Deep self-Supervised cycle-Consistent deformation for few-shot shape segmentation. In *Eurographics Symposium on Geometry Processing*, 2019. 1

[3] Kaiming He, X. Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 3, 4

[4] Vladislav Ishimtsev, Alexey Bokhovkin, Alexey Artemov, Savva Ignatyev, Matthias Nießner, Denis Zorin, and Burnaev Evgeny. CAD-Deform: Deformable fitting of cad models to 3d scans. In *ECCV*, 2020. 1

[5] Yangyan Li, Hao Su, Charles Ruizhongtai Qi, Noa Fish, Daniel Cohen-Or, and Leonidas J. Guibas. Joint embeddings of shapes and images via cnn image purification. In *SIGGRAPH Asia*, 2015. 1, 4

[6] 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 *CVPR*, 2019. 3

[7] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. PointNet: Deep learning on point sets for 3D classification and segmentation. In *CVPR*, 2017. 3

[8] Minhyuk Sung, Zhenyu Jiang, Panos Achlioptas, Niloy J. Mitra, and Leonidas J. Guibas. DeformSyncNet: Deformation transfer via synchronized shape deformation spaces. In *SIGGRAPH Asia*, 2020. 1

[9] Minhyuk Sung, Hao Su, Vladimir G. Kim, Siddhartha Chaudhuri, and Leonidas Guibas. ComplementMe: Weakly-supervised component suggestions for 3D modeling. In *SIGGRAPH Asia*, 2017. 3

[10] Maxim Tatarchenko*, Stephan R. Richter*, René Ranftl, Zhuwen Li, Vladlen Koltun, and Thomas Brox. What do single-view 3d reconstruction networks learn? In *CVPR*, 2019. 1

[11] Mikaela Angelina Uy, Jingwei Huang, Minhyuk Sung, Tolga Birdal, and Leonidas Guibas. Deformation-Aware 3D model embedding and retrival. In *ECCV*, 2020. 1, 4

[12] Mikaela Angelina Uy, Vladimir G. Kim, Minhyuk Sung, Noam Aigerman, Siddhartha Chaudhuri, and Leonidas Guibas. Joint learning of 3d shape retrieval and deformation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 2, 3

[13] Weiyue Wang, Duygu Ceylan, Radomir Mech, and Ulrich Neumann. 3DN: 3D deformation network. In *CVPR*, 2019. 1

[14] Wang Yifan, Noam Aigerman, Vladimir Kim, Siddhartha Chaudhuri, and Olga Sorkine-Hornung. Neural cages for detail-preserving 3D deformations. In *CVPR*, 2020. 1