KP-RED: Exploiting Semantic Keypoints for Joint 3D Shape Retrieval and Deformation

Ruida Zhang[1*], Chenyangguang Zhang[1*], Yan Di[2], Fabian Manhardt[3],
Xingyu Liu[1], Federico Tombari[2,3], Xiangyang Ji[1]
[1]Tsinghua University, [2]Technical University of Munich, [3]Google
{zhangrd23@mails, zcyg22@mails, xyji@}.tsinghua.edu.cn *

Abstract

*In this paper, we present **KP-RED**, a unified **KeyP**oint-driven **RE**trieval and **D**eformation framework that takes object scans as input and jointly retrieves and deforms the most geometrically similar CAD models from a pre-processed database to tightly match the target. Unlike existing dense matching based methods that typically struggle with noisy partial scans, we propose to leverage category-consistent sparse keypoints to naturally handle both full and partial object scans. Specifically, we first employ a lightweight retrieval module to establish a keypoint-based embedding space, measuring the similarity among objects by dynamically aggregating deformation-aware local-global features around extracted keypoints. Objects that are close in the embedding space are considered similar in geometry. Then we introduce the neural cage-based deformation module that estimates the influence vector of each keypoint upon cage vertices inside its local support region to control the deformation of the retrieved shape. Extensive experiments on the synthetic dataset PartNet and the real-world dataset Scan2CAD demonstrate that KP-RED surpasses existing state-of-the-art approaches by a large margin. Codes and trained models will be released in* https://github.com/lolrudy/KP-RED.

1. Introduction

Creating high-quality 3D models from noisy object scans has attracted wide research interest [21, 24, 39, 44, 48] due to its potential applications in 3D scene perception [32, 51], robotics [49] and artistic creation [10, 40]. Previous prior-free methods [32, 39] directly utilize deep neural networks to recover the object model. However, due to heavy (self-)occlusion and non-negligible noise, it is often infeasible to infer fine-grained geometric structures without prior knowledge. To address this issue, **Retrieval** and **Deformation** (**R**&**D**) methods [7, 19, 21, 31, 36, 40, 41, 44, 48] are proposed. These methods first retrieve the most geometrically similar source shape from a certain shape database and then deform the retrieved shape to tightly match the target, yielding a CAD model with fine-grained structural details inherited from the source shape.

Figure 1. **Top Two Rows:** Given the target point cloud, KP-RED first retrieves the most similar CAD model from the preprocessed database and deforms it to match the target using the keypoints for guidance. **Bottom Two Rows:** Given a scene scan, KP-RED reconstructs the CAD models of all objects and represents the scene by gathering the reconstructed models.

However, existing **R**&**D** methods typically suffer from two challenges, making them vulnerable to noise and occluded observations. First, when constructing the embedding space for retrieval, most methods [7, 11, 15, 21, 26, 41] resort to single global feature of the input point cloud, which is usually obtained via pooling of point-wise features. Unfortunately, this strategy inevitably causes the loss of local geometric information, leading to less accurate retrieval, and further deteriorates the deformation quality. Furthermore, such global feature based embedding is sensitive to

*Authors with equal contributions.

occlusion, making it infeasible to handle partial input. Second, most methods [11, 21, 41, 44] directly utilize dense point matching to control the shape deformation. However, several random outliers in observations may significantly mislead the matching process, resulting in undesired deformation results.

To tackle the aforementioned challenges, we propose KP-RED, a novel keypoint-driven joint **R**&**D** framework, which takes a full or partial object scan as input, and outputs the recovered corresponding CAD model via querying a pre-constructed database. Instead of directly leveraging dense point matching for **R**&**D**, we propose to utilize sparse keypoints as the intermediate representation, enabling our unified keypoint-based **R**&**D** framework. Due to lack of ground truth annotations of keypoints, we follow [19] to automatically detect the keypoints in an unsupervised manner by adopting semantically consistent control of shape deformation. As a result, the discovered keypoints are also proven to be semantically consistent, even under large shape variations across each category.

Specifically, KP-RED consists of two main modules: keypoint-based deformation-aware retrieval and keypoint-driven neural cage deformation. In the retrieval module (Fig. 2 Retrieval Block), keypoints are first detected via the keypoint predictor and point-wise features are extracted with PointNet [34]. For each keypoint, we aggregate the point-wise features in its support region (Fig. 2 (R-E)) to obtain the local retrieval tokens. Since the locations of keypoints are semantically consistent across each category, we concatenate all local tokens of each object in a uniform order to generate the global retrieval token, which is utilized to retrieve the most similar objects from the database. In the deformation module (Fig. 2 Deformation Block), unlike [19, 48] that employ global cage scaffolding, we propose to leverage self-attention to simultaneously encapsulate the local fine-grained details and global geometric cues among keypoints so to predict the influence vector upon the support region of each keypoint, which is then interpolated onto the source shape to control the deformation.

Compared with dense matching based baselines [11, 21, 41], our keypoint-based framework holds two main advantages. First, the extracted keypoints are semantically consistent across each category, allowing effective occlusion reasoning and noise suppression. Thereby KP-RED can better handle noisy partial object scans than competitors. Second, our keypoint-based feature aggregation approach preserves fine-grained local geometry information, yielding a more accurate embedding space for retrieval.

Our main contributions are summarized as follows,

- We present a unified network KP-RED for 3D shape generation from object scans, which learns category-consistent keypoints to jointly retrieve the most similar source shape from the pre-established database and control the shape deformation.
- We design a keypoint-driven local-global feature aggregation scheme to establish the shape embedding space for retrieval, which performs effectively for both full and partial object scans, enabling multiple real-world applications.
- We introduce a novel cage-based deformation scheme with self-attention that uses keypoints to control the local deformation of the retrieved shape.

2. Related Works

Neural Shape Generation. Recent advances in neural networks have led to the development of generative latent representations for 3D shapes. [6, 20, 27, 28, 33, 35, 46, 50, 52] model geometry as implicit functions, while [1, 29, 38, 43, 45, 47] generate point clouds, voxels or meshes to model shapes explicitly. Factorized representations are studied by [14, 25], decomposing shapes into different geometric parts and handling of the geometric variations of each part separately. However, while these methods exhibit impressive representation abilities, they may struggle to preserve structural details and to handle complex objects due to the lack of prior knowledge.

CAD Model Retrieval. Retrieving a CAD model that closely matches a 3D scan of a real-world object is a critical issue in 3D scene understanding. While many prior works directly retrieve the most similar CAD model by evaluating similarity in the descriptor space [4, 36] or the latent embedding space of neural networks [3, 7, 15, 26], the direct retrieval may not always yield satisfying results since the model database cannot contain all instances. To address this limitation, recent works propose extracting deformation-aware embeddings [40] or developing novel optimization targets [17] to better fit the details of the target shape after deformation. However, their deformation modules are fixed and non-trainable, leading to inferior performance.

3D Shape Deformation. One of the fundamental issues of geometry processing is to deform a source 3D model to tightly match a target shape. Traditional methods [13, 16, 37] directly optimize the deformed shapes to fit the targets. However, they are only applicable to complete target shapes and do not generalize well to the real-world scenarios since the object scans are typically partial due to (self-)occlusion. By modeling deformation as volumetric warps [18, 24], cage deformations [19, 48], vertex-based offsets [44], or flows [21], recent approaches attempt to learn deformation priors from a set of shapes by neural networks. Cage-based deformation [48] is particularly noteworthy for its ability to preserve geometry details, and [19] extends it by adopting automatically discovered semantic keypoints to enable human users to control the shape explicitly. However, most of these works do not study the retrieval process. Only a few works [21, 41] jointly

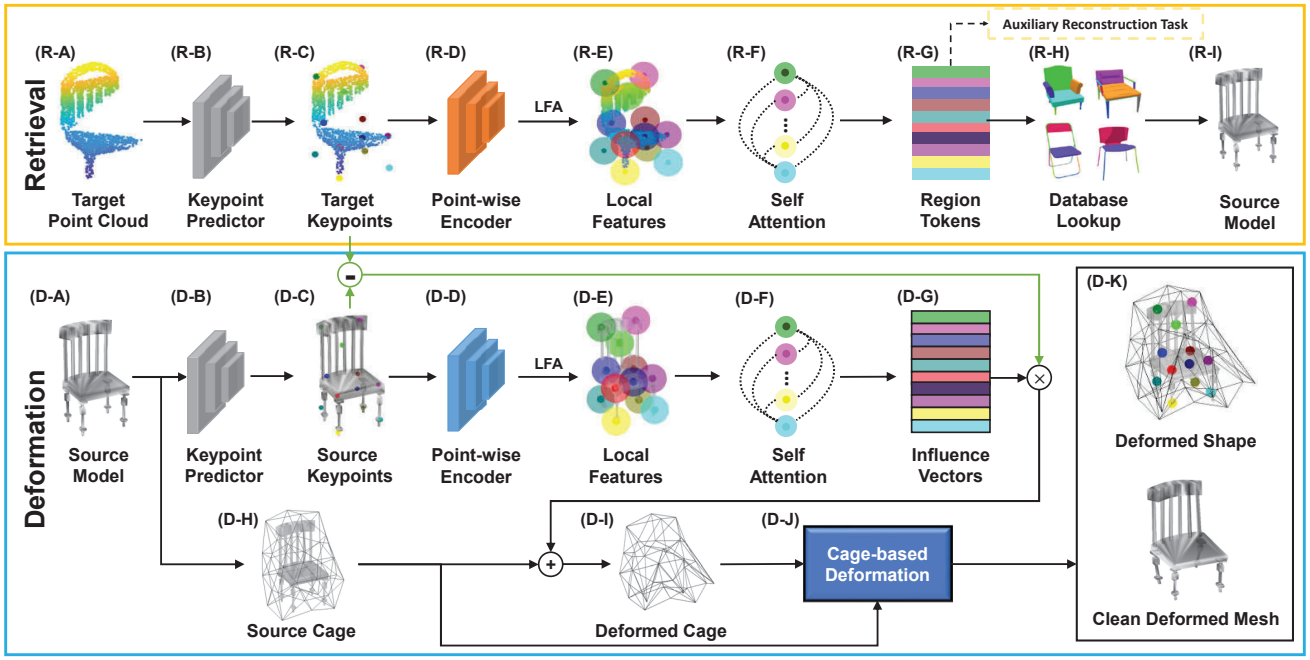

Figure 2. **Overview of KP-RED.** The target point cloud (R-A) is first canonicalized using the estimated pose obtained from an arbitrary pose estimator [9, 12, 53, 54], following which the keypoint predictor (R-B) is employed to forecast the target keypoints (R-C). An encoder (R-D) predicts point-wise features and Local Feature Aggregation (LFA) is used to obtain the features of each keypoint region (R-E). The self-attention module (R-F) extracts the local retrieval token of each region (R-G), which is then compared with the tokens of the database models (R-H). The region tokens are supervised with an auxiliary reconstruction task during training. The most similar shape to the target is chosen as the source model (R-I). The source keypoints are then predicted by the shared keypoint predictor and the local features are extracted via LFA (D-A - D-E). The self-attention module (D-F) predicts the influence vectors (D-G) which demonstrate how the displacements of keypoints inflect the cage. Given the cage of the source shape (D-H), the deformed cage (D-I) is derived from the influence vectors. Finally, the deformed point cloud and mesh (D-K) are finally computed by the cage-based deformation (D-J).

study **R**&**D**. Uy et al. [41] designs a novel training strategy to jointly optimize retrieval and deformation modules, but its deformation module depends on the part annotations of the database models which are labor intensive to obtain. U-RED [11] proposes point-wise residual-guided retrieval metrics and a one-to-many module to handle noisy and partial inputs. However, it also depends on the part annotations. ShapeFlow [21] constructs a flow-based deformation space and utilizes an auto-decoder to extract features for retrieval. Nevertheless, the auto-decoder needs to perform time-consuming online optimization during inference. Thus, we propose a keypoint-based joint **R**&**D** framework KP-RED which yields high-quality CAD models with no requirements of extra annotations and runs in real-time.

3. KP-RED

In this section, we first present the overview of KP-RED, and then introduce our **R**&**D** method for processing full shapes in detail in Sec. 3.1, Sec. 3.2. In Sec. 3.3, we demonstrate our confidence-based dynamic feature aggregation technique for partial shape. The **Overview** of KP-RED is shown in Fig. 2. Given an input full or partial object scan, KP-RED first constructs a keypoint-guided deformation-aware embedding space to retrieve the most similar model from the database, and then deforms the model to match the input shape via keypoint-driven cage deformation.

In the **Retrieval** module, given the target point cloud S_{tgt} as input, the keypoint detector (R-B) predicts N_K semantic keypoints $\mathbf{K}_{tgt} = \{K_{tgt}^{(1)}, K_{tgt}^{(2)}, ..., K_{tgt}^{(N_K)}\}$ on S_{tgt}. Then PointNet [34] is employed to extract point-wise features. For each keypoint $K_{tgt}^{(i)}$, we aggregate its corresponding deformation-aware local feature $l_{tgt}^{(i)}$ by pooling the point features within a ball region $\mathcal{R}_{tgt}^{(i)}$ centered at $K_{tgt}^{(i)}$, where $i = 1, ..., N_K$ (R-E). Self-attention is employed to discover the region-to-region relations and predict the local retrieval tokens of each keypoint region $\{\mathcal{T}_{tgt}^{(1)}, \mathcal{T}_{tgt}^{(2)}, ..., \mathcal{T}_{tgt}^{(N_K)}\}$ (R-G). Since the locations of keypoints are semantically consistent across each category, we concatenate all local tokens in a uniform order to generate the global deformation-aware token $\mathcal{T}_{tgt}$. We then utilize $\mathcal{T}_{tgt}$ to compare with the tokens in the database, so to re-

trieve the most similar shape as the source shape (R-I).

In the **Deformation** module, the source shape S_{src} (D-A) is fed into the identical keypoint detector to extract keypoints $\mathbf{K}_{src} = \{K_{src}^{(1)}, K_{src}^{(2)}, ..., K_{src}^{(N_K)}\}$ from S_{src}. Influence vectors $\{I_1, I_2, ..., I_{N_K}\}$ (D-G) are predicted via the self-attention module (D-F), describing how each keypoint influences its support cage vertices (Fig. 4 (a)). Finally, we obtain the deformed source shape $S_{src2tgt}$ from the deformed cage (D-I) by adopting the mean value coordinate based interpolation approach [22].

3.1. Keypoint-Driven Deformation

In the deformation module, we aim to deform the retrieved source shape $S_{src} \in \mathbb{R}^{N_P \times 3}$ to tightly match the input target shape $S_{tgt} \in \mathbb{R}^{N_P \times 3}$. Note that N_P denotes the number of points. We follow [19, 22, 48] and adopt the keypoint-driven neural-cage based deformation and extend it with self-attention to encapsulate local structural details and global geometric cues.

Neural Cage Deformation. To control the deformation of the source shape S_{src}, we adopt the sparse cage scaffolding strategy [19, 48]. First, a coarse control mesh (cage) with vertices $C_{src} \in \mathbb{R}^{N_C \times 3}$ is computed to enclose S_{src}, so that displacements of C_{src} can be interpolated to any point on S_{src}, via constructing mean value coordinates [48], enabling to control the deformation of S_{src}. Second, we predict the influence vector $I_i \in \mathbb{R}^{N_C \times 1}$ of each keypoint $K_{src}^{(i)} \in \mathbb{R}^{1 \times 3}$ in $\mathbf{K}_{src}$ (Fig. 2 (D-G), Fig. 4 (a2)). This vector describes how the cage vertices are influenced by the displacements of the keypoints. Finally, the differences between the keypoints $\mathbf{K}_{tgt}$ of the target shape S_{tgt} and $\mathbf{K}_{src}$ of S_{src} are compared to guide the algorithm moving C_{src}. The resulting deformed cage vertices $C_{src2tgt}$ are calculated as

$$C_{src2tgt} = C_{src} + \sum_{i=1}^{N_K} I_i (K_{tgt}^{(i)} - K_{src}^{(i)}). \quad (1)$$

By interpolating $C_{src2tgt}$ on S_{src} [19, 48], we obtain $S_{src2tgt}$ that tightly matches S_{tgt}.

Geometric Self-Attention. Previous methods [19, 48] directly utilize the global feature to predict the influence vectors, resulting in unsatisfactory performance due to inevitable loss of local information. We instead adopt local feature aggregation and self-attention mechanism to capture local and global information and to, thus, preserve more geometric details. To this end, we first predict the point-wise features via PointNet [34] and gather the local features $l_{src}^{(i)}$ of each keypoint $K_{src}^{(i)}$ by pooling the point-wise features inside its support ball region centered at $K_{src}^{(i)}$ with radius r (Fig. 2 (D-E), Fig. 4 (a1)). We then use a self-attention module to discover region-to-region relations and inject global information to $l_{src}^{(i)}$, whilst preserving the local information. Finally, the influence vector of each keypoint is derived from the feature of its own support region. We restrict the influence vector to only influence cage vertices inside the local support region of the keypoint. This ensures that the local information around the keypoint is fully exploited, leading to finer-grained deformation. Moreover, the region-to-region relations complement essential global information to the local features. For example, when facing a partial scan of a symmetric object (*e.g.* chair), the structure of the missing part can be inferred from its corresponding symmetric regions. Thus, the geometric self-attention not only preserves structural details, yet also enhances robustness towards partial inputs.

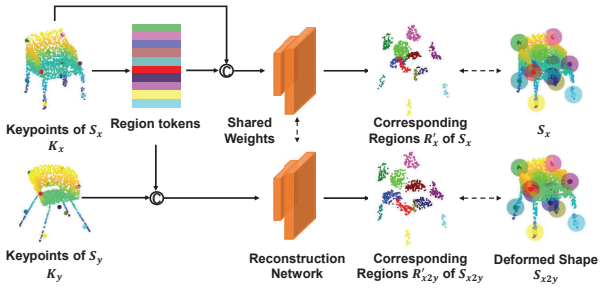

Figure 3. The training procedure of the retrieval module. Given the keypoints $\mathbf{K}_x$ and the region tokens extracted from the shape S_x, the reconstruction network reconstructs the corresponding regions R'_x of S_x. Meanwhile, the network reconstructs the regions of the deformed shape R'_{x2y} from the region tokens and $\mathbf{K}_y$.

Training. During training, our final objective is composed of two loss terms, used to simultaneously supervise the learning of keypoint extraction and shape deformation. The former term is supposed to enforce shape similarity $\mathcal{L}_{sim}$ by calculating the Chamfer Distance between $S_{src2tgt}$ and S_{tgt}. The other term is meant to regularize the keypoints [19]. This terms encourages keypoints to be well-distributed by minimizing the Chamfer Distance between different keypoints and N_K points sampled by means of Farthest Point Sampling on S_{src}. The overall loss function of the deformation module is thus defined as

$$\mathcal{L}_{def} = \mathcal{L}_{sim} + \lambda_{kpt} \mathcal{L}_{kpt}, \quad (2)$$

where λ_{kpt} weights the contribution of the keypoints regularization term. More details on the definitions of loss terms are provided in the Supplementary Material.

3.2. Deformation-Aware Retrieval

The retrieval module aims to retrieve the most geometrically similar source shape S_{src} for the input target shape S_{tgt} from a pre-constructed database. The retrieval task faces two main challenges. First, the retrieval process should be deformation-aware, meaning the retrieved shape should match the target shape tightly after deformation. Second,

the overall retrieval module should be lightweight and real-time with minimal additional computational cost. To address these challenges, we design a novel keypoint-based retrieval method.

Local-Global Feature Embedding. Unlike [41] that leverages a directly learned global feature for retrieval, we instead utilize local-global keypoint-based features. As shown in Fig. 2, given the target object S_{tgt}, we predict its keypoints $\mathbf{K}_{tgt}$ by the keypoint detector. Similar to the deformation module, we adopt local feature aggregation and self-attention to extract the local features $\mathcal{T}_{tgt}^{(i)}$ of each keypoint $K_{tgt}^{(i)}$ as the local retrieval tokens. Since the keypoints are, as aforementioned, semantically consistent across each category, we concatenate all local tokens $\{\mathcal{T}_{tgt}^{(1)}, \mathcal{T}_{tgt}^{(2)}, ..., \mathcal{T}_{tgt}^{(N_K)}\}$ in a uniform order to generate the global deformation-aware token $\mathcal{T}_{tgt}$. During inference, we choose the source model as

$$S_{src} = \arg\min_{\omega \in \Omega} f_{\mathcal{L}_1}(\mathcal{T}_{tgt}, \mathcal{T}_{\omega}), \tag{3}$$

where Ω denotes the pre-established model database, T_ω is the global token of the database model ω and $f_{\mathcal{L}_1}(\cdot,\cdot)$ computes the $\mathcal{L}_1$ distance between the two tokens.

Training. We illustrate the full training procedure in Fig. 3. To supervise the learning of retrieval tokens $\mathcal{T}_{tgt}$ and $\mathcal{T}_{src}$, we introduce a novel auxiliary reconstruction task. Given two randomly selected shapes S_x and S_y from the training set, we first extract their keypoints $\mathbf{K}_x$, $\mathbf{K}_y$ and retrieval tokens $\mathcal{T}_x$, $\mathcal{T}_y$, and then utilize the deformation module to deform S_x to match S_y, yielding S_{x2y}. Subsequently, we adopt an MLP-based reconstruction network Ψ, which processes two tasks in parallel, $\Psi_1 : (\mathcal{T}_x^{(i)}, \mathbf{K}_x) \rightarrow R_x^{(i)\prime}$ and $\Psi_2 : (\mathcal{T}_x^{(i)}, \mathbf{K}_y) \rightarrow R_{x2y}^{(i)\prime}$, where $R_x^{(i)\prime}$ and $R_{x2y}^{(i)\prime}$ denote the reconstruction results of the support region of the ith keypoint of S_x and S_{x2y} ($R_x^{(i)}$ and $R_{x2y}^{(i)}$) respectively. Therefore, our training objective is defined as

$$\mathcal{L}_{ret} = \frac{1}{N_K} \sum_{i=1}^{N_K} (f_{CD}(R_x^{(i)}, R_x^{(i)\prime}) + f_{CD}(R_{x2y}^{(i)}, R_{x2y}^{(i)\prime})), \tag{4}$$

with f_{CD} denoting the Chamfer Distance between two point clouds. The first reconstruction task Ψ_1 forces $\mathcal{T}_x$ to encapsulate the geometric information of S_x. On the other hand, comparing Ψ_2 with the deformation module that prescribes $(S_x, \mathbf{K}_y) \rightarrow S_{x2y}$ via neural cage deformation, where $S_x = \cup_{i=1}^{N_K} R_x^{(i)}$ and $S_{x2y} = \cup_{i=1}^{N_K} R_{x2y}^{(i)}$. Thereby, $\mathcal{T}_x$ is encouraged to also capture cues related to the deformation of S_x. We divide the reconstruction task into different regions to force each token to capture the local structural details of each support region and enhances the granularity of retrieval results.

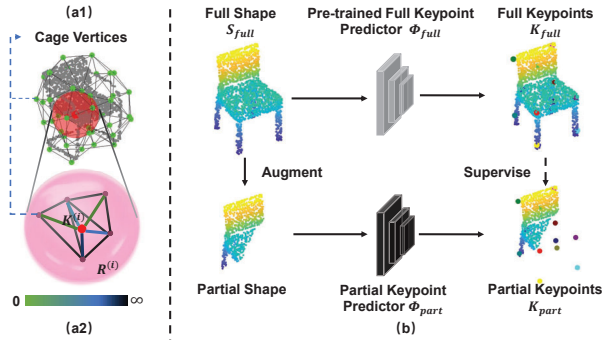

Figure 4. **(a1)**: The support region $R^{(i)}$ of the specific keypoint $K^{(i)}$. **(a2)**: The influence vectors I_i of the specific keypoint $K^{(i)}$. The color indicates the influence weight of the keypoint towards each cage vertex (as in Eq. 1). **(b)**: The training procedure of the keypoint predictor for partial shapes. We employ the keypoint predictor trained with full shapes for supervision.

3.3. Handling Partial Point Cloud

Due to (self-)occlusion, poor lighting conditions, viewpoints, *etc*, actual real-world scans are oftentimes just partially available. In order to simulate real-world partial condition, we thus augment full shapes in PartNet for generation of partial shapes by means of random slicing. Please refer to the Supplementary Material for details.

Compared with full shape **R**&**D**, handling partial shapes poses a new challenge. In particular, the observed point cloud is typically non-uniformly distributed in 3D space, rendering it difficult to reliably extract keypoints from poorly observed regions. To address this limitation, we thus propose a confidence-based dynamic feature extraction, to improve robustness towards partial inputs.

Without additional priors, we assume that the point density is a good measurement to understand the reliability of point observations in most cases. Specifically, if the density in the support region $R^{(i)}$ of keypoint $K^{(i)}$ is low, then $K^{(i)}$ can be considered unreliable and should contribute less for **R**&**D**. Conversely, $K^{(i)}$ should be assigned a larger weight. We define the density D_i at each keypoint $K^{(i)}$ as the normalized average density in its support region $R^{(i)}$, $D_i = \min(N_R^{(i)}/(\alpha V), 1)$, where $N_R^{(i)}$ denotes the number of points in $R^{(i)}$, V describes the region volume, and α is a constant for normalization. The density of all keypoints is $\mathbf{D} = \{D_1, D_2, ..., D_{N_K}\} \in \mathbb{R}^{N_K}$.

Retrieval. We use the aforementioned density $\mathbf{D}$ as the confidence weight to select the source model S_{src} from the database Ω via

$$S_{src} = \arg\min_{\omega \in \Omega} \sum_{i=1}^{N_K} D_i f_{\mathcal{L}_1}(\mathcal{T}_{tgt}^{(i)}, \mathcal{T}_{\omega}^{(i)}). \tag{5}$$

Thereby, keypoint regions with higher density contribute more to the final retrieval results.

Deformation. As shown in Fig. 4 (b), we introduce an additional keypoint predictor Φ_{part} to handle partial shapes. We use the keypoint predictor Φ_{full} (corresponding to (R-B) in Fig. 2) trained with full shapes to guide the learning of Φ_{part} in a teacher-student manner, where only the parameters in Φ_{part} are updated. In essence, given an augmented partial shape S_{part} and its corresponding full shape S_{full}, we obtain the keypoints of the partial shape $\mathbf{K}_{part} = \Phi_{part}(S_{part})$ and the full shape $\mathbf{K}_{full} = \Phi_{full}(S_{full})$. As S_{part} is augmented from S_{full}, they are required to also possess identical keypoints. Moreover, keypoints with higher confidence should contribute more to the deformation result and thus require stronger supervision signals. We use again the density $\mathbf{D}$ as the confidence weight and define the weighted keypoint loss as

$$\mathcal{L}_{wkpt} = \sum_{i=1}^{N_K} D_i f_{\mathcal{L}_1}(K_{full}^{(i)}, K_{part}^{(i)}). \tag{6}$$

Since the common Chamfer Distance (CD) as defined in Eq. 2 is bilateral, it is not a suitable metric to evaluate shape similarity for partial shapes. Therefore, we replace it with the Unilateral Chamfer Distance (UCD) from the target shape towards the deformed shape for the similarity loss, denoted as $\mathcal{L}_{usim}$. We refer again to the Supplemental Material for additional details.

Finally, the overall loss function of the deformation module, for handling partial shapes, is defined as

$$\mathcal{L}_{pdef} = \mathcal{L}_{usim} + \lambda_{wkpt}\mathcal{L}_{wkpt}, \tag{7}$$

where λ_{wkpt} is used as the weighting parameter.

4. Experiments

Datasets. To evaluate the effectiveness of our method, we leverage a synthetic datasets PartNet [30] and a real-world dataset Scan2CAD [2]. For PartNet, we adopt the same split of database, training set and test set as in [41]. The shapes in PartNet come from ShapeNet [5]. PartNet contains 1419 models in database, 11433 instances for training and 2861 for testing. Please note that we do not need the part annotations like in [41] and only use the mesh models for training. Scan2CAD [2] is a real-world dataset developed upon ScanNet [8] and provides the ground truth masks, poses and corresponding CAD models for 14225 objects. The input point clouds in Scan2CAD are generated by back-projecting the depth maps. We first centralize the point clouds and then follow [41] to canonicalize them by the provided rotations. We conduct experiments on the {chair, table, cabinet} categories on PartNet and Scan2CAD. To further evaluate the reconstruction quality when facing partial inputs under different occlusion ratios, we augment shapes in PartNet to generate partial shapes with random slicing (see Supp. Mat.). We generate partial target point clouds with occlusion ratio of 25%, 50% and 75% from the test split of Partnet as the test set.

Implementation Details. Following [41], $N_P = 2048$ points are sampled from each shape and normalized into a unit cube to serve as the input of the network. We first train the deformation module from scratch and then utilize the predicted keypoints to train the retrieval module. The source and target shapes are randomly selected from the model database and the training set in the above mentioned two steps. To handle the partial shape, we train the partial keypoint predictor while freezing other parameters learned from full shapes. During training, the target shape is augmented by random slicing with the occlusion ratio γ uniformly sampled from $\gamma \sim \mathcal{U}(25\%, 90\%)$. The baseline models [21, 41] trained with full shapes are fine-tuned with the same augmentation for partial shape evaluation. The original Chamfer Distance loss is also replaced by the Unilateral Chamfer Distance (UCD) loss for fair comparison. We directly utilize the models (KP-RED and [11, 21, 41]) trained on PartNet dataset to inference on the validation set of Scan2CAD, and the model database remains the same as the setting of PartNet. We use $N_K = 12$ keypoints for all categories. The radius of the support region is set to $r = 0.3$. The parameters for all loss terms are selected empirically and kept unchanged in experiments unless specified, with $\{\lambda_{kpt}, \lambda_{wkpt}\} = \{2, 20\}$. We run all experiments on a single NVIDIA 3090 GPU and employ the Adam optimizer [23] with batch size of 16 and base learning rate of 1e-3 for deformation module and 1e-2 for retrieval module. We train the two modules for 30 epochs each, about 300K iterations. Detailed descriptions of network architectures are provided in the Supplemental Material.

Evaluation Metrics. The typical Chamfer Distance (CD) between the reconstructed model and the object scanning is used for full shape evaluation, while the Unilateral Chamfer Distance (UCD) is reported for partial shapes. On Scan2CAD dataset, we use the ground truth model to compute the UCD metric. All methods retrieve the top 10 source candidates and choose the best **R**&**D** result to calculate metrics as in U-RED [11]. We also provide results using top 1, 5, 25 retrieval candidates in Sup. Mat.. The *average* metrics in all tables are obtained via averaging over all **instances**.

4.1. Experiments on Full Shapes

To demonstrate the joint **R**&**D** ability of KP-RED, we compare our method with the state-of-the-art [21, 41], and present the results in Tab. 1. KP-RED consistently outperforms other competitors in all categories and datasets under the Chamfer Distance metrics, demonstrating our superior ability of **R**&**D** under different conditions. In particular, on PartNet dataset, we obtain superior results with a realtive improvement of 85.7%, 86.6% and 84.2% under the

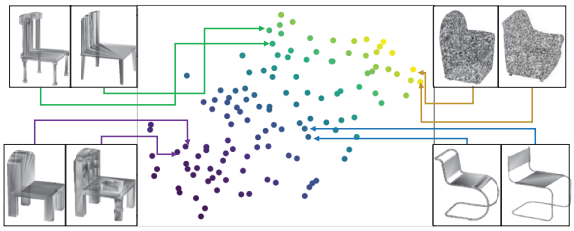

Figure 5. The visualization of the learned retrieval tokens of database shapes via t-SNE [42]. Objects whose tokens are close in the embedding space are considered similar in geometry.

Method	Chair	Table	Cabinet	Average
Uy *et al.* [41]	0.638	0.629	0.688	0.637
U-RED [11]	0.834	0.326	0.474	0.551
ShapeFlow [21]	0.238	0.400	0.514	0.340
Ours	**0.091**	**0.084**	**0.109**	**0.089**

Table 1. Chamfer Distance metrics for joint **R**&**D** results on full shapes on PartNet dataset [30]. Overall best results are **in bold**.

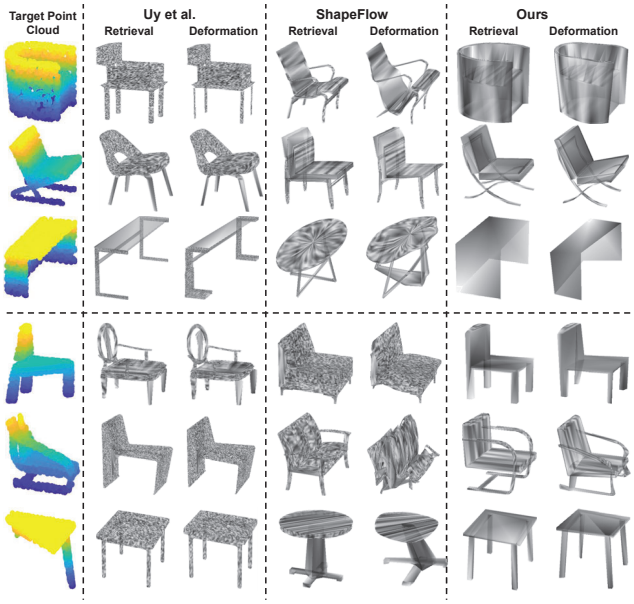

Figure 6. Qualitative **R**&**D** results on PartNet [30]. **Top Block:** Full shape **R**&**D**. **Bottom Block:** Partial shape **R**&**D**.

average Chamfer Distance, compared with Uy *et al.* [41] for three categories. When comparing with the second best method ShapeFlow [21], our improvement is still significant with an relative improvement of 73.8%. Moreover, the inference time of Uy *et al.* [41] reaches 0.7 seconds per instance, while ShapeFlow [21] adopts online optimization for better performance, which leads to non-negligible computational expenses and very long inference time, about 45 seconds per instance. In contrast, KP-RED maintains a real-time inference speed with about 30 ms per instance, whilst surpassing both by a large margin in terms of **R**&**D** quality.

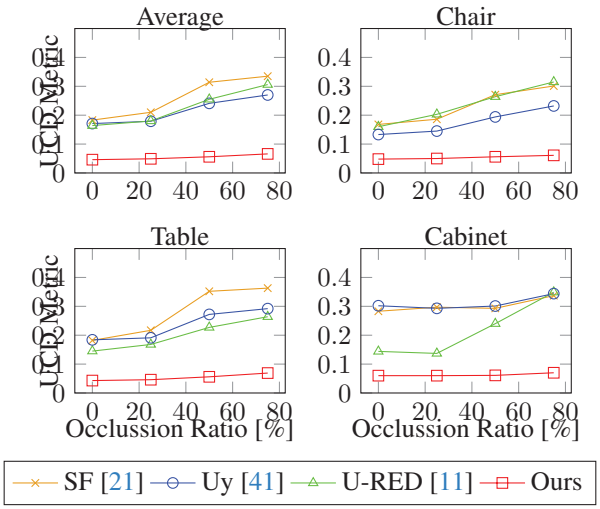

Figure 7. Unilateral Chamfer Distance metrics for joint **R**&**D** results on the augmented partial PartNet [30]. SF stands for ShapeFlow [21].

Method	Chair	Table	Cabinet	Average
Uy *et al.* [41]	0.158	0.190	0.676	0.210
U-RED [11]	0.227	0.132	0.316	0.207
ShapeFlow [21]	0.230	0.302	0.345	0.265
Ours	**0.059**	**0.057**	**0.073**	**0.060**

Table 2. Unilateral Chamfer Distance metrics for joint **R**&**D** results on Scan2CAD [2]. Overall best results are **in bold**.

Fig. 5 demonstrates that our unsupervised keypoint-driven retrieval module is capable of successfully establishing an embedding space for measuring the similarity among objects. As illustrated in Fig. 6, our **R**&**D** results are clearly more similar to the target compared to other methods. We attribute this to our well-designed retrieval module, employing local-global feature embedding for effective comparison among objects. Moreover, the cage-based deformation with geometric self-attention preserves structural details and boosts deformation quality. In Supplementary Material, we perform an oracle retrieval experiment to illustrate the effectiveness of our keypoint-guided deformation module.

4.2. Experiments on Partial Shapes

We conduct experiments on two datasets, namely real-world Scan2CAD [2] and synthetic augmented PartNet [30], to comprehensively demonstrate the robustness of KP-RED for handling partial point clouds. As shown in Tab. 2 and Fig. 7, KP-RED constantly outperforms the current state-of-the-art [21, 41] in both real-world and synthetic scenarios by a significant margin. On Scan2CAD dataset, under real-world occlusion, KP-RED exceeds Uy *et al.* [41], U-RED and ShapeFlow by 71.4% 71.0%, and 77.3% under

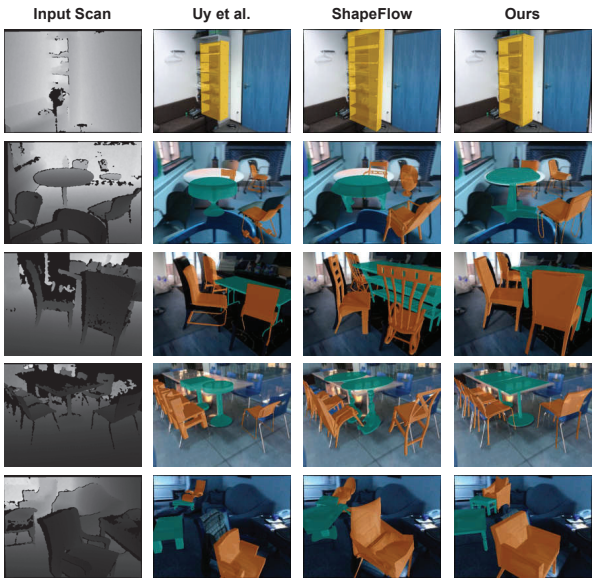

Figure 8. Qualitative results on Scan2CAD dataset [2]. The **R**&**D** results are rendered on the RGB images for better visualization.

	GSA	DAR	LGF	Chair	Table	Cabinet	Avg.
(a)	×	×	×	0.135	0.187	0.126	0.161
(b)	×	✓	×	0.128	0.187	0.124	0.158
(c)	✓	×	×	0.116	0.105	0.117	0.110
(d)	✓	×	✓	0.099	0.088	0.113	0.094
(e)	✓	✓	×	0.103	0.092	0.113	0.098
(f)	✓	✓	✓	**0.091**	**0.084**	**0.109**	**0.089**

Table 3. Ablation studies on full shapes of PartNet [30]. Avg. denotes the average CD metric. GSA denotes Geometric Self-Attention. Without GSA, we use PointNet [34] to extract the global feature of the input shape, like in [41]. DAR denotes Deformation-Aware Retrieval. We only use the reconstruction task Ψ_1 to train the retrieval network when it is ablated. LGF denotes Local-Global Feature Embedding for the retrieval process. When it is ablated, we directly utilize the global feature for retrieval.

the UCD error respectively. Fig. 8 shows superior visualization quality of KP-RED with accurate retrieval and accurate deformation. On partial PartNet, when the occlusion ratio increases from 25% to 75%, the *average* UCD error of KP-RED only increases by 0.015 (30.6%), while the error of U-RED and ShapeFlow increases 0.126 (70.0%) and 0.125 (59.5%), respectively. As can be seen KP-RED is more robust to incomplete input due to the dynamic feature extraction. More qualitative results are shown in the Supplementary Material.

4.3. Ablation Studies

We conduct ablation studies in both full shape (Tab. 3) and partial shape (Fig. 9) scenarios. The ablations on full shapes mainly aim at verifying the effectiveness of our proposed **G**eometric **S**elf-**A**ttention (GSA) in Sec. 3.1, **D**eformation-**A**ware **R**etrieval (DAR) and **L**ocal-**G**lobal **F**eature Embedding (LGF) in Sec. 3.2. While on partial shapes, **C**onfidence-**B**ased Dynamic Feature Extraction (CB) in Sec. 3.3 is ablated.

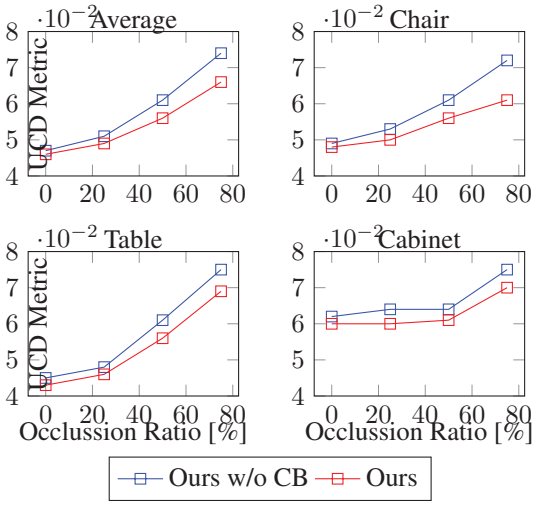

Figure 9. Ablation studies of **C**onfidence-**B**ased Dynamic Feature Extraction (CB) on partial PartNet [30].

Tab. 3 exhibits the ablations of GSA, DAR and LGF, conducting on full shapes of PartNet. Comparing (a) and (c) in Tab. 3, our proposed encoder with GSA contributes to an improved *average* performance by about 32%. The effectiveness of DAR is demonstrated by (c) and (e) with a reduction of *average* CD error by 10%. After incorporating LFA and DAR, utilizing LGF further enhances the *average* performance by 9% ((e) and (f)). As described in Sec. 3.3, the effectiveness of **R**&**D** with **C**onfidence-**B**ased Dynamic Feature Extraction (CB) is illustrated in Fig. 9. When ablating CB, we assume that the confidence weights of all keypoints are equal. There is a general trend that CB contributes more when the occlusion ratio is high. CB improves the *average* performance by 11% for up to 75% occlusion and 8% for less than 50% occlusion. This indicates that CB is an essential strategy for handling partial shape.

5. Conclusion

In this paper, we introduce KP-RED, a unified framework for 3D shape generation from full or partial object scans. Our approach employs category-consistent keypoints to jointly retrieve the most geometrically similar shapes from a pre-constructed database and deform the retrieved shape to tightly match the input. We propose a keypoint-driven local-global feature aggregation scheme to extract deformation-aware features for retrieval, and a neural cage-based deformation algorithm to control the local deformation of the retrieved shape. In the future, we plan to extend our technique to 3D scene understanding.

Acknowledgement. This work was supported by the National Key R&D Program of China under Grant 2018AAA0102801.

References

[1] Panos Achlioptas, Olga Diamanti, Ioannis Mitliagkas, and Leonidas Guibas. Learning representations and generative models for 3d point clouds. In *International conference on machine learning*, pages 40–49. PMLR, 2018. 2

[2] Armen Avetisyan, Manuel Dahnert, Angela Dai, Manolis Savva, Angel X Chang, and Matthias Nießner. Scan2cad: Learning cad model alignment in rgb-d scans. In *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition*, pages 2614–2623, 2019. 6, 7, 8

[3] Armen Avetisyan, Angela Dai, and Matthias Nießner. End-to-end cad model retrieval and 9dof alignment in 3d scans. In *Proceedings of the IEEE/CVF International Conference on computer vision*, pages 2551–2560, 2019. 2

[4] Frederic Bosche and Carl T Haas. Automated retrieval of 3d cad model objects in construction range images. *Automation in Construction*, 17(4):499–512, 2008. 2

[5] 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. 6

[6] Zhiqin Chen and Hao Zhang. Learning implicit fields for generative shape modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5939–5948, 2019. 2

[7] Manuel Dahnert, Angela Dai, Leonidas J Guibas, and Matthias Nießner. Joint embedding of 3d scan and cad objects. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8749–8758, 2019. 1, 2

[8] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 2017. 6

[9] Yan Di, Fabian Manhardt, Gu Wang, Xiangyang Ji, Nassir Navab, and Federico Tombari. So-pose: Exploiting self-occlusion for direct 6d pose estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12396–12405, 2021. 3

[10] Yan Di, Chenyangguang Zhang, Pengyuan Wang, Guangyao Zhai, Ruida Zhang, Fabian Manhardt, Benjamin Busam, Xiangyang Ji, and Federico Tombari. Ccd-3dr: Consistent conditioning in diffusion for single-image 3d reconstruction. *arXiv preprint arXiv:2308.07837*, 2023. 1

[11] Yan Di, Chenyangguang Zhang, Ruida Zhang, Fabian Manhardt, Yongzhi Su, Jason Rambach, Didier Stricker, Xiangyang Ji, and Federico Tombari. U-red: Unsupervised 3d shape retrieval and deformation for partial point clouds. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8884–8895, 2023. 1, 2, 3, 6, 7

[12] Yan Di, Ruida Zhang, Zhiqiang Lou, Fabian Manhardt, Xiangyang Ji, Nassir Navab, and Federico Tombari. Gpv-pose: Category-level object pose estimation via geometry-guided point-wise voting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6781–6791, 2022. 3

[13] Vignesh Ganapathi-Subramanian, Olga Diamanti, Soeren Pirk, Chengcheng Tang, Matthias Niessner, and Leonidas Guibas. Parsing geometry using structure-aware shape templates. In *2018 International Conference on 3D Vision (3DV)*, pages 672–681. IEEE, 2018. 2

[14] Lin Gao, Jie Yang, Tong Wu, Yu-Jie Yuan, Hongbo Fu, Yu-Kun Lai, and Hao Zhang. Sdm-net: Deep generative network for structured deformable mesh. *ACM Transactions on Graphics (TOG)*, 38(6):1–15, 2019. 2

[15] Can Gümeli, Angela Dai, and Matthias Nießner. Roca: Robust cad model retrieval and alignment from a single image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4022–4031, 2022. 1, 2

[16] Qi-Xing Huang, Bart Adams, Martin Wicke, and Leonidas J. Guibas. Non-Rigid Registration Under Isometric Deformations. *Computer Graphics Forum*, 2008. 2

[17] Vladislav Ishimtsev, Alexey Bokhovkin, Alexey Artemov, Savva Ignatyev, Matthias Niessner, Denis Zorin, and Evgeny Burnaev. Cad-deform: Deformable fitting of cad models to 3d scans. In *European Conference on Computer Vision*, pages 599–628. Springer, 2020. 2

[18] Dominic Jack, Jhony K Pontes, Sridha Sridharan, Clinton Fookes, Sareh Shirazi, Frederic Maire, and Anders Eriksson. Learning free-form deformations for 3d object reconstruction. In *Asian Conference on Computer Vision*, pages 317–333. Springer, 2018. 2

[19] Tomas Jakab, Richard Tucker, Ameesh Makadia, Jiajun Wu, Noah Snavely, and Angjoo Kanazawa. Keypointdeformer: Unsupervised 3d keypoint discovery for shape control. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12783–12792, 2021. 1, 2, 4

[20] Wonbong Jang and Lourdes Agapito. Codenerf: Disentangled neural radiance fields for object categories. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12949–12958, 2021. 2

[21] Chiyu Jiang, Jingwei Huang, Andrea Tagliasacchi, and Leonidas J Guibas. Shapeflow: Learnable deformation flows among 3d shapes. *Advances in Neural Information Processing Systems*, 33:9745–9757, 2020. 1, 2, 3, 6, 7

[22] Tao Ju, Scott Schaefer, and Joe Warren. Mean value coordinates for closed triangular meshes. *ACM Trans. Graph.*, 24(3):561–566, jul 2005. 4

[23] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6

[24] Andrey Kurenkov, Jingwei Ji, Animesh Garg, Viraj Mehta, JunYoung Gwak, Christopher Choy, and Silvio Savarese. Deformnet: Free-form deformation network for 3d shape reconstruction from a single image. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 858–866. IEEE, 2018. 1, 2

[25] Jun Li, Kai Xu, Siddhartha Chaudhuri, Ersin Yumer, Hao Zhang, and Leonidas Guibas. Grass: Generative recursive autoencoders for shape structures. *ACM Transactions on Graphics (TOG)*, 36(4):1–14, 2017. 2

[26] 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. *ACM trans-*

actions on graphics (TOG), 34(6):1–12, 2015. 1, 2

[27] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4460–4470, 2019. 2

[28] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021. 2

[29] Kaichun Mo, Paul Guerrero, Li Yi, Hao Su, Peter Wonka, Niloy Mitra, and Leonidas J Guibas. Structurenet: Hierarchical graph networks for 3d shape generation. *arXiv preprint arXiv:1908.00575*, 2019. 2

[30] 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/CVF conference on computer vision and pattern recognition*, pages 909–918, 2019. 6, 7, 8

[31] Liangliang Nan, Ke Xie, and Andrei Sharf. A search-classify approach for cluttered indoor scene understanding. *ACM Transactions on Graphics (TOG)*, 31(6):1–10, 2012. 1

[32] Yinyu Nie, Xiaoguang Han, Shihui Guo, Yujian Zheng, Jian Chang, and Jian Jun Zhang. Total3dunderstanding: Joint layout, object pose and mesh reconstruction for indoor scenes from a single image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 55–64, 2020. 1

[33] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 165–174, 2019. 2

[34] 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. 2, 3, 4, 8

[35] Edoardo Remelli, Artem Lukoianov, Stephan Richter, Benoît Guillard, Timur Bagautdinov, Pierre Baque, and Pascal Fua. Meshsdf: Differentiable iso-surface extraction. *Advances in Neural Information Processing Systems*, 33:22468–22478, 2020. 2

[36] Adriana Schulz, Ariel Shamir, Ilya Baran, David IW Levin, Pitchaya Sitthi-Amorn, and Wojciech Matusik. Retrieval on parametric shape collections. *ACM Transactions on Graphics (TOG)*, 36(1):1–14, 2017. 1, 2

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

[38] Yongbin Sun, Yue Wang, Ziwei Liu, Joshua Siegel, and Sanjay Sarma. Pointgrow: Autoregressively learned point cloud generation with self-attention. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 61–70, 2020. 2

[39] Maxim Tatarchenko, Stephan R Richter, René Ranftl, Zhuwen Li, Vladlen Koltun, and Thomas Brox. What do single-view 3d reconstruction networks learn? In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3405–3414, 2019. 1

[40] Mikaela Angelina Uy, Jingwei Huang, Minhyuk Sung, Tolga Birdal, and Leonidas Guibas. Deformation-aware 3d model embedding and retrieval. In *European Conference on Computer Vision*, pages 397–413. Springer, 2020. 1, 2

[41] Mikaela Angelina Uy, Vladimir G Kim, Minhyuk Sung, Noam Aigerman, Siddhartha Chaudhuri, and Leonidas J Guibas. Joint learning of 3d shape retrieval and deformation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11713–11722, 2021. 1, 2, 3, 5, 6, 7, 8

[42] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008. 7

[43] Nanyang Wang, Yinda Zhang, Zhuwen Li, Yanwei Fu, Wei Liu, and Yu-Gang Jiang. Pixel2mesh: Generating 3d mesh models from single rgb images. In *Proceedings of the European conference on computer vision (ECCV)*, pages 52–67, 2018. 2

[44] Weiyue Wang, Duygu Ceylan, Radomir Mech, and Ulrich Neumann. 3dn: 3d deformation network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1038–1046, 2019. 1, 2

[45] Haozhe Xie, Hongxun Yao, Xiaoshuai Sun, Shangchen Zhou, and Shengping Zhang. Pix2vox: Context-aware 3d reconstruction from single and multi-view images. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 2690–2698, 2019. 2

[46] Qiangeng Xu, Zexiang Xu, Julien Philip, Sai Bi, Zhixin Shu, Kalyan Sunkavalli, and Ulrich Neumann. Point-nerf: Point-based neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5438–5448, 2022. 2

[47] Guandao Yang, Xun Huang, Zekun Hao, Ming-Yu Liu, Serge Belongie, and Bharath Hariharan. Pointflow: 3d point cloud generation with continuous normalizing flows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4541–4550, 2019. 2

[48] 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. 1, 2, 4

[49] Guangyao Zhai, Xiaoni Cai, Dianye Huang, Yan Di, Fabian Manhardt, Federico Tombari, Nassir Navab, and Benjamin Busam. Sg-bot: Object rearrangement via coarse-to-fine robotic imagination on scene graphs. *arXiv preprint arXiv:2309.12188*, 2023. 1

[50] Guangyao Zhai, Evin Pınar Örnek, Shun-Cheng Wu, Yan Di, Federico Tombari, Nassir Navab, and Benjamin Busam. Commonscenes: Generating commonsense 3d indoor scenes with scene graphs. *Advances in Neural Information Processing Systems*, 36, 2024. 2

[51] Cheng Zhang, Zhaopeng Cui, Yinda Zhang, Bing Zeng, Marc Pollefeys, and Shuaicheng Liu. Holistic 3d scene understanding from a single image with implicit representation. In *Proceedings of the IEEE/CVF Conference on Computer*

Vision and Pattern Recognition, pages 8833–8842, 2021. 1

[52] Chenyangguang Zhang, Yan Di, Ruida Zhang, Guangyao Zhai, Fabian Manhardt, Federico Tombari, and Xiangyang Ji. Ddf-ho: Hand-held object reconstruction via conditional directed distance field. *Advances in Neural Information Processing Systems*, 36, 2024. 2

[53] Ruida Zhang, Yan Di, Zhiqiang Lou, Fabian Manhardt, Federico Tombari, and Xiangyang Ji. Rbp-pose: Residual bounding box projection for category-level pose estimation. In *European Conference on Computer Vision*, pages 655–672. Springer, 2022. 3

[54] Ruida Zhang, Yan Di, Fabian Manhardt, Federico Tombari, and Xiangyang Ji. Ssp-pose: Symmetry-aware shape prior deformation for direct category-level object pose estimation. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7452–7459. IEEE, 2022. 3