Improving Single-View Mesh Reconstruction for Unseen Categories via Primitive-Based Representation and Mesh Augmentation

Yu-Liang Kuo[1] hank.cs08g@nctu.edu.tw
Wei-Jan Ko[1] ts771164@gmail.com
Chen-Yi Chiu[1] charles.en07@nycu.edu.tw
Wei-Chen Chiu[1] walon@cs.nctu.edu.tw

***Abstract*— As most existing works of single-view 3D reconstruction aim at learning the better mapping functions to directly transform the 2D observation into the corresponding 3D shape for achieving state-of-the-art performance, there often comes a potential concern on having the implicit bias towards the seen classes learnt in their models (i.e. reconstruction intertwined with the classification) thus leading to poor generalizability for the unseen object categories. Moreover, such implicit bias typically stemmed from adopting the object-centered coordinate in their model designs, in which the reconstructed 3D shapes of the same class are all aligned to the same canonical pose regardless of different view-angles in the 2D observations. To this end, we propose an end-to-end framework to reconstruct the 3D mesh from a single image, where the reconstructed mesh is not only view-centered (i.e. its 3D pose respects the viewpoint of the 2D observation) but also preliminarily represented as a composition of volumetric 3D primitives before being further deformed into the fine-grained mesh to capture the shape details. In particular, the usage of volumetric primitives is motivated from the assumption that there generally exists some similar shape parts shared across various object categories, learning to estimate the primitive-based 3D model thus becomes more generalizable to the unseen categories. Furthermore, we advance to propose a novel mesh augmentation strategy, CvxRearrangement, to enrich the distribution of training shapes, which contributes to increasing the robustness of our proposed model and achieves better generalization. Extensive experiments demonstrate that our proposed method provides superior performance on both unseen and seen classes in comparison to several representative baselines of single-view 3D reconstruction.**

I. Introduction

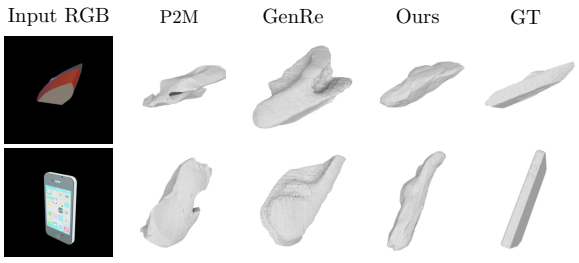

Fig. 1. Examples of reconstructing 3D shapes from unseen categories given the single-view 2D observations, where the comparison is made among the models from Pixel2Mesh (P2M) [1], GenRe [2] and our proposed method (in which they are all trained on *chair, car, airplane* classes). Please note that we visualize the 3D shapes with the viewpoints different from the ones of their corresponding 2D inputs for better comparing the holisticness and reasonableness among the results of different methods.

[1]All authors are with the Department of Computer Science, National Chiao Tung University

∗ Project page: https://hank-kuo-cs.github.io/VPSVR-Web

Reconstructing 3D shapes from single RGB images is a long-standing problem in computer vision and has wide applications for the machine perception, where impressive progress has been made in recent years thanks to the renaissance of deep learning techniques. Depending on the formats for representing 3D models, various approaches of single-view 3D reconstruction are proposed (e.g. [3], [4], [5], [6], [7], [8] and [9], [10], [11], [12] for voxel-based 3D shapes and 3D point clouds respectively, and [13], [14], [15], [16] for 3D models described by implicit functions). Among different 3D formats, the one based on 3D meshes [17], [18], [19], [20], [1] have garnered research attention these years, as the meshes (composed of points and surfaces) are capable of modelling more detailed shape geometry in comparison to other point-based ones (e.g. voxels or point clouds) and being easier to be deformed for performing the physics simulation thus becoming more desirable for numerous applications in computer animation and geometry processing. However, the benefit of adopting mesh-based 3D representations also comes with more challenges: as most existing methods aim to learn a mapping function to estimate the 3D meshes directly from 2D observations, the high complexity of the target 3D meshes usually results in making these methods inevitably learn the shape priors that are highly biased towards the training/seen classes (i.e. reconstruction is intertwined with the classification), hence having difficulty in generalizing to the classes which are not seen during training.

Moreover, another cause for the generalization issue for the unseen categories is that most of the current works reconstruct 3D shapes upon the object-centered coordinates, in which the 3D shapes are all aligned to the same canonical view: for example, no matter a 2D image of a chair is taken from its front view or the side view, the object-centered method always reconstruct the front-facing 3D chair. Despite the fact that such object-centered coordinate system is beneficial for simplifying the 3D reconstruction task thus boosting the performance on training categories, it however becomes problematic while tackling the shapes from unseen categories since there exists no plausible way to predefine the canonical pose for the novel/unseen categories and the learnt models in results tend to overfit upon the training categories, as clearly described in [21], [22]. In contrast to the object-centered coordinate, the **view-centered** coordinate system attempts to align the 3D objects to the same viewing perspective as their corresponding 2D input images: for example, given the 2D chair images taken from the front view and side view, the view-centered method

will reconstruct the front-facing chair model and the side-facing chair model respectively, thus being more suitable for addressing 3D reconstruction upon the unseen classes. Therefore, in this paper we explicitly tackle the task of view-centered single-view mesh reconstruction in order to advance the generalizability on unseen categories, and we focus on making comparison with respect to other works which are also built upon the view-centered coordinate system.

However, even though the view-centered coordinate is typically more generalizable, directly approximating the mapping function from 2D images to 3D shapes via a black-box model (e.g. a neural network) often results in the issue of overfitting. A pioneer work, GenRe [2], instead proposes to factorize the reconstruction procedure into three steps – depth estimation, spherical map inpainting, and voxel refinement. Such explicitly modelling the steps of the geometric projection from 2D image to 3D shape is more category-agnostic and beneficial for generalization, as every module only utilizes the cues from the previous step in which the potential issue of learning the biased shape priors is alleviated. However, while GenRe [2] successfully demonstrates better generalizability for the unseen classes comparing to other related works, it lacks the understanding on the local structures/details of the 3D shapes, which are also considered as another important perspective for assessing the generalizability [23], [24]. To this end, [23], [24] propose to reconstruct a 3D shape via the composition of volumetric primitives (e.g. cuboids), where such **primitive-based** representation is stemmed from the assumption that there generally exists some similar local parts/shapes shared across different object classes thus they are typically more category-agnostic. With adopting primitives to approximate these shared local parts, the resultant primitive-based representation not only respects the local structures of the 3D shape but also becomes more generalizable. Nevertheless, the model training of [23], [24] requires the supervision of part decomposition (i.e. the part segmentation) for 3D shapes, which are usually difficult and expensive to collect.

Motivated by the aforementioned works (i.e. GenRe [2] and [23], [24]), we propose an end-to-end framework for reconstructing view-centered 3D meshes from the single-view 2D observation, in which the framework is composed of three stages – depth estimation, primitive composition, and mesh deformation. In particular, the second stage for estimating the primitive-based composition of the target 3D shape is achieved in an unsupervised manner without requiring any groundtruth annotations of part segmentation. We note that, even the resultant volumetric primitives might not well aligned with the parts expected by humans, they are still beneficial to model the local structures of 3D shape and contribute to the better generalizability. Moreover, the primitive-based composition is further deformed in our third stage via the graph neural network [25] to better capture the surface details of the 3D shape and then achieves the final fine-grained 3D meshes, in which our primitive composition and mesh deformation stages together can be treated as a coarse-to-fine procedure and is able to provide better support on complicated meshes.

In addition to our proposed framework, we also introduce a novel data augmentation strategy for the 3D meshes, which is the first of its kind to the best of our knowledge, in order to enrich the training data distribution and improve the model's reconstruction capability across various categories. Basically, our augmentation strategy, **CvxRearrangement**, generates diverse and novel 3D meshes by rearranging the convex hulls obtained from the existing meshes via the technique of Approximate Convex Decomposition (ACD [26]).

II. Related Work

Single-View 3D Mesh Reconstruction. Here we focus on providing brief reviews on the single-view 3D reconstruction works where the output format is in the 3D meshes. Basically, most of the existing methods follow the similar process of deforming the template meshes (e.g. sphere) in accordance with the given 2D observation. For instance, [27] learns the deformation on a predefined mesh with view priors and utilizes the differentiable renderer [28] to reconstruct meshes without 3D supervision. [1] aggregates the global and perceptual features to train a graph convolution network for predicting the deformation on an ellipsoid mesh in a coarse-to-fine manner; [29] also applies a graph convolution network to deform the bounding boxes reconstructed by the decoder proposed from [30], which requires the groundtruth part annotations. While these above-mentioned studies improve the fidelity of reconstructed shapes, their model typically cannot generalize well for the unseen categories as all of them except Pixel2Mesh [1] reconstruct shapes in the object-centered coordinate. By contrast, our proposed model is built upon the view-centered coordinate system. Moreover, instead of starting from a predefined or template mesh, it adopts the primitive-based composition in terms of structural ellipsoids which are then deformed into the fine-grained meshes, without using any part label.

Single-View 3D Reconstruction for Unseen Classes. In comparison to the task of reconstructing 3D shapes of seen classes from single images, only few works investigate the model generalizability on the unseen classes. GenRe [2] firstly proposes to disentangle the geometric projections from shape reconstruction, where various representations (e.g. depth, silhouette, and spherical shapes) are explicitly involved to capture more generic and class-agnostic shape priors. [24] and [23] instead adopt the primitive-based representation for 3D shapes which is more category-agnostic thus benefiting the generalizability for the unseen classes. [24] proposes a structure-guided shape reconstruction framework to jointly learn the shape interpretation and the shape reconstruction, using a recursive neural network proposed by [31] which requires the pre-segmentation for training shapes. [23] factorizes the reconstruction procedure into several sub-problems (e.g. part segmentation, part orientation estimation, part size estimation) in which the training for each sub-problem requires the corresponding groundtruth labels. Although utilizing primitive-based representation brings [24], [23] superior reconstruction performance and better

generalizability than [2], the requirement of additional annotations on part segmentation or part orientation is usually harder to fulfill thus limiting their practical applicability. In contrast, our proposed method is similarly benefited from using primitive-based representation but does not rely on any additional part-level annotations.

Recently, we have witnessed the implicit-model-based work PixelNeRF [32] which learns to estimate the implicit 3D model from 2D image evidences thus being seeming to share the similar goal as our task. However, there exists significant differences: the implicit-model methods typically are good at rendering novel views (i.e. 2D images) instead of directly generating the 3D model, and it is non-trivial to extract from the implicit model the 3D points and the surfaces as what meshes can provide (since implicit model typically encodes the density). In contrast, the mesh-based 3D reconstruction is more preferred for directly producing 3D meshes which are able to model more detailed shape geometry and easier to be deformed for further applications.

Primitive-Based Representation. Treating the 3D shapes as a decomposition of parts is a crucial problem in the field of graphics and computer vision, thus has attracted plenty research attention. Various works [30], [33], [34] learn the part decomposition on the dataset with part labels (e.g. PartNet [35]) while recently several unsupervised works are proposed to represent 3D shapes as the volumetric primitives (e.g. cuboids [36], [37], ellipsoids [38], convex hulls [39], and superquadrics [40], [41]). While these works focus on abstracting the complex shapes into primitives instead of generating delicate shapes with details (as primitives typically are with regular/simple structure), our method treats primitives as the coarse estimation of shapes and advances to deform them into local-structure-aware fine-grained meshes with the geometry details of shapes being better captured.

III. Proposed Method

Our proposed framework learns to reconstruct the view-centered 3D mesh from the given 2D image and aims to provide better generalizability for unseen categories, in which it is composed of three stages: depth estimation, primitive composition, and mesh deformation.

A. Depth Estimation

Given an input RGB image I which shows the 2D observation of a 3D shape S with the clean background, the first stage of our proposed framework follows the similar idea as GenRe [2] to perform the depth estimation on I via a depth estimation network for obtaining the corresponding depth map D of I. The estimated depth map represents the preliminary 3D and geometric information (related to the visible surfaces of the target 3D shape) extracted from the 2D image, where the following stage will base on such (partial) information to proceed the complete reconstruction of the full 3D shape. The architecture of our depth estimation network is identical to the one used in [2], which is ResNet-18-based [42] encoder-decoder network with the U-Net structure [43]. The training objective $\mathcal{L}_{\text{depth}}$ of the depth estimation network is based on the mean square error (MSE) between the estimated depth map D and its corresponding groundtruth $\hat{D}$:

$$\mathcal{L}_{\text{depth}} = \text{MSE}(D, \hat{D}) \tag{1}$$

where $\hat{D}$ is rendered from the mesh $\hat{M}$ of the target 3D shape S with the same viewpoint as I (note that we adopt the Pytorch3D renderer [44]).

The estimated depth map D is then turned into the depth feature map F by a ResNet-18-based depth feature extractor $\mathcal{E}$ for performing the primitive prediction in the next stage.

B. Primitive Composition

Given the preliminary 3D information (i.e. $F = \mathcal{E}(D)$ where D is the predicted depth map of I) obtained from the previous depth estimation stage, the second stage of our proposed framework predicts from F the primitive-based representation of the target 3D shape, which is composed of K volumetric primitives. Basically, our primitive composition stage consists of three main components: (1) the *structure network* $\mathcal{D}_s$ which predicts the centroids of primitives in the 3D space, (2) the *geometry network* $\mathcal{D}_g$ which predicts the size and the rotation of each primitive, and (3) the *composing module* which assembles the ellipsoid meshes related to all the primitives into a unified mesh M_p.

Particularly, the configuration of centroids decides how the primitives are distributed in the 3D space to capture the **global structure** of the target 3D shape, while the setting of radius and rotation for each primitive instead determines how the primitive is transformed to fit the **local geometry** of the target 3D shape around the corresponding centroid.

The primitives used in our proposed framework are initialized from the unit 3D spheres and the k-th primitive is parameterized by a tuple (t_k, q_k, v_k), where $t_k = [t_k^x, t_k^y, t_k^z]$ indicates its 3D centroid (i.e. the center coordinates of the primitive), $q_k = [q_k^1, q_k^2, q_k^3, q_k^4]$ represents its rotation in terms of the quaternion vector, and $v_k = [v_k^x, v_k^y, v_k^z]$ indicates its radii on three dimensions (thus determining the volume/size of the primitive).

Structure Network. With taking the global depth feature z_g as input, which is obtained via applying average pooling on F, the structure network $\mathcal{D}_s$ predicts the centroids t_k for all the K primitives: $\{t_k | k = 1, ..., K\} = \mathcal{D}_s(z_g)$, where we provide some example results for visualizing the predicted centroids in Figure 3.

Geometry Network. Given the centroid t_k of k-th primitive predicted by D_s, we firstly adopt the perceptual feature pooling operation proposed by Pixel2Mesh [1] to extract the local depth feature z_k^l related to the k-th primitive. In details, we compute the 2D projection of t_k onto the depth feature map F using camera intrinsics, and then pool the depth feature from four nearby pixels using the bilinear interpolation to obtain z_k^l. Afterwards, the geometry network $\mathcal{D}_g$ takes z_k^l as input and predicts the parameters of size and rotation (i.e. v_k and q_k) for the k-th primitive:

$$\{(v_k, q_k) | k = 1, ..., K\} = \mathcal{D}_g(z_k^l) \tag{2}$$

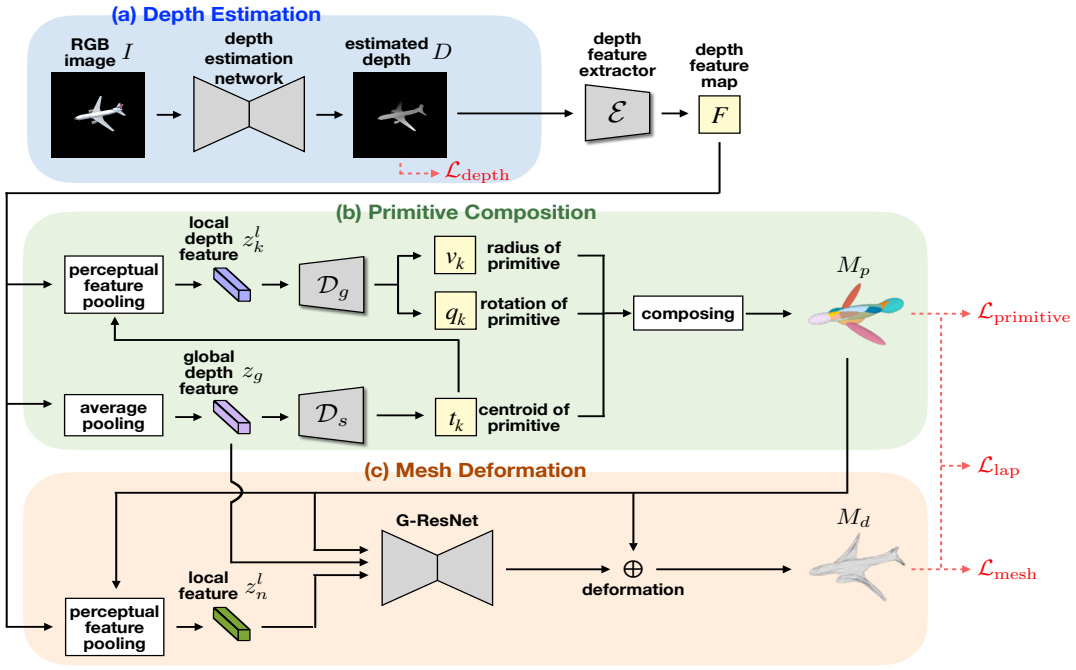

Fig. 2. Illustration of our proposed end-to-end framework for single-view 3D mesh reconstruction (cf. Section III).

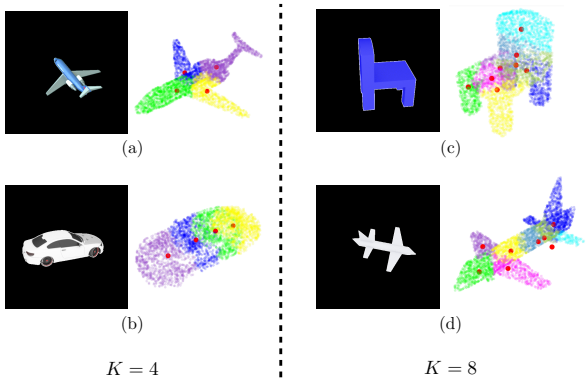

Fig. 3. Visualization for the example results (i.e. centroids t_k of K primitives, colored as red dots) produced by our structure network $\mathcal{D}_s$ within the primitive composition stage of our proposed framework. For every example we show a pair of input 2D image and the predicted centroids of the corresponding 3D shape. Please note that here we particularly represent the shape in canonical view by its sampled 3D points and colorize these points based on their closest centroids only for better visualization, while our proposed method actually reconstructs view-centered 3D meshes.

Composing Module. We first prepare K zero-centered unit sphere meshes $(M_1^\circ, ..., M_K^\circ)$ as the initialization for our volumetric primitives, where each sphere mesh has 128 vertices, then apply the spatial transformation on each mesh M_k° in accordance with the estimated (t_k, q_k, v_k) of its corresponding primitive to obtain the transformed mesh M_k':

$$M_k' = \mathcal{T}(\mathcal{R}(\mathcal{S}(M_k^\circ, v_k), q_k), t_k) | k = 1, ..., K \qquad (3)$$

where the translation $\mathcal{T}$, rotation $\mathcal{R}$, and scaling $\mathcal{S}$ transformations are based on t_k, q_k, and v_k respectively. Noting that all these transformations are differentiable thus they do not hinder our framework from being end-to-end trainable. Then, the ensemble of the transformed meshes (now becoming ellipsoids) results to a complete mesh $M_p = \bigcup_{k=1}^{K} M_k'$.

The training objective $\mathcal{L}_{\text{primitive}}$ of this primitive composition stage is defined upon the Chamfer Distance (CD) [45] to penalize the reconstruction error between the mesh M_p and the groundtruth target mesh $\hat{M}$.

$$\mathcal{L}_{\text{primitive}} = \text{CD}(\mathcal{P}_{M_p}, \mathcal{P}_{\hat{M}}) \qquad (4)$$

where $\mathcal{P}_{M_p}$ and $\mathcal{P}_{\hat{M}}$ denote the sets of 3D points sampled from M_p and $\hat{M}$, respectively.

C. Mesh Deformation

As the primitive-based mesh M_p is usually unable to well capture the surface details, we advance to utilize the G-ResNet (which is composed of graph convolution layers with skip connections, as proposed in Pixel2Mesh [1]) to deform M_p into a fine-grained mesh M_d. Basically, as there exists $128 * K$ vertices in M_p (i.e. 128 vertices per primitive), G-ResNet will predict the deformation for each vertex n with taking its coordinate, its local depth feature z_n^l, and the global depth feature z_g into account, where z_n^l is obtained via applying the perceptual feature pooling over the projection of vertex n on F. In details, the global depth feature z_g together with the coordinates and local depth features for all the vertices of M_p are jointly fed into G-ResNet to obtain the overall vertex-wise deformation for refining our M_p to obtain the resultant mesh M_d with surface details. Please note that, in comparison to other deformation-based methods [27], [1] where they deform only a single sphere or ellipsoid, our primitive-based deformation is more capable of handling fine-grained shape details such as holes and thin structures, and achieved without any part-level supervision.

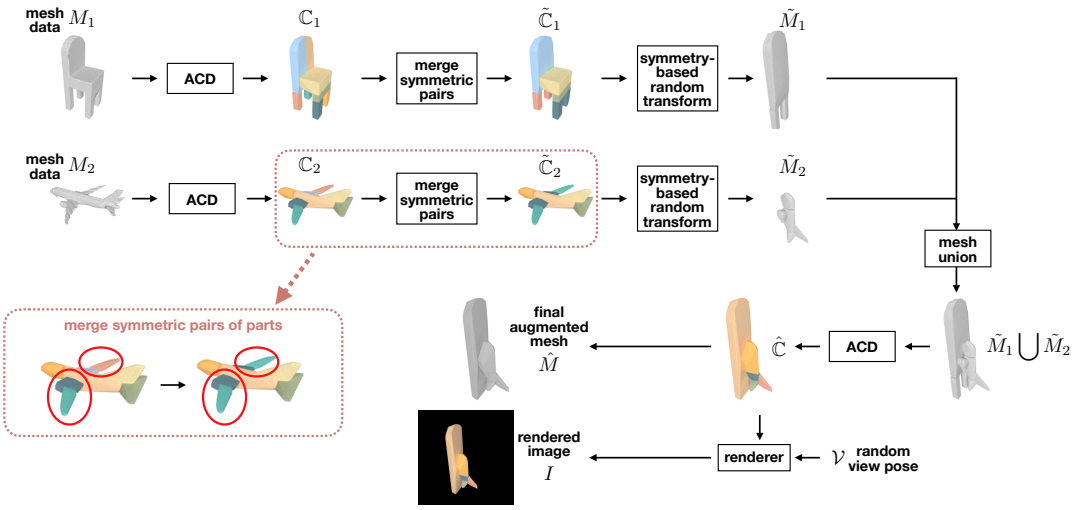

Fig. 4. Illustration of our proposed CvxRearrangement for performing mesh data augmentation (cf. Section III-D).

Similar to $\mathcal{L}_{\text{primitive}}$, we adopt the Chamfer Distance (CD) loss to penalize the reconstruction error between the final mesh M_d and the groundtruth target mesh $\hat{M}$.

$$\mathcal{L}_{\text{mesh}} = \text{CD}(\mathcal{P}_{M_d}, \mathcal{P}_{\hat{M}}) \tag{5}$$

where $\mathcal{P}_{M_d}$ denotes the vertices of M_d.

Moreover, we additionally introduce the Laplacian regularization term $\mathcal{L}_{\text{lap}}$, which is widely adopted in mesh deformation works [1], [46], to prevent M_d from overly deforming from M_p, i.e. encouraging the neighboring vertices to have the same movement thus tending to preserve the local geometry of M_p and alleviate the issue of self-intersection among meshes. Given a vertex $n \in M_p$ and its corresponding vertex $n' \in M_d$, the Laplacian coordinate of n is defined as $\delta_n = n - \sum_{\eta \in \mathcal{N}(n)} \frac{1}{\|\mathcal{N}(n)\|}\eta$ where $\mathcal{N}(n)$ denotes the neighboring vertices of n, then $\mathcal{L}_{\text{lap}}$ is defined as:

$$\mathcal{L}_{\text{lap}} = \sum_{n} \|\delta_{n'} - \delta_n\|_2 \tag{6}$$

The overall objective $\mathcal{L}_{\text{total}}$ for training our whole framework is the weighted sum over the aforementioned losses:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{depth}} + \mathcal{L}_{\text{primitive}} + \mathcal{L}_{\text{mesh}} + \lambda\mathcal{L}_{\text{lap}} \tag{7}$$

where $\lambda = 0.1$ in our experiments to balance among losses. In implementation, our full framework is trained for 20 epochs by the Adam optimizer with learning rate set to $1\text{e}{-3}$ for the depth estimation network and $1\text{e}{-4}$ for other reconstruction modules. All our trained models and source code will be publicly available upon paper acceptance.

D. Mesh Data Augmentation: CvxRearrangement

We now introduce our novel strategy for augmenting 3D mesh data, named CvxRearrangement, to enrich the training data distribution and further improve our model generalizability towards unseen categories, where the overview of CvxRearrangement is provided in Figure 4. This augmentation method is mainly built upon the technique of Approximate Convex Decomposition (ACD) [26], where ACD is able to decompose a mesh into multiple convex hulls which could be treated as preliminary part segmentation, moreover, ACD does not require any human intervention but utilizes only the geometric information of input mesh. Basically, given two meshes, we would like to apply random spatial transformation to rearrange their corresponding sets of convex hulls and assemble these rearranged parts for generating new mesh data, with our specific designs to ensure the generated meshes symmetric and reasonable.

Normalization. First, we prepare all our meshes aligned in the canonical view (i.e. facing the positive x-axis and having xy-plane the symmetry plane, noting that such canonical alignment is only for producing augmented meshes), zero-centered, and normalized to fit into a unit sphere. Given a normalized mesh M, ACD is applied to obtain the set of convex hulls $\mathbb{C} = \{c_i | i = 1, ..., C\} = \text{ACD}(M)$, where each convex hull c_i is treated as an object part.

Merging Symmetric Pair of Parts. The symmetry characteristic of object shapes, especially for the man-made objects, is commonly used for various works [37], [27] of 3D reconstruction and generation. Such symmetricity is also adopted in our augmentation strategy to merge the pair of symmetric convex hulls (parts) from a shape into one part, in which such operation will help our later procedure to generate symmetric augmented meshes. We check whether two convex hulls c_i and c_j from a mesh M are symmetric or not via the following steps: (1) we sample two sets of surface points X_i and X_j respectively from c_i and c_j, followed by reflecting X_j over the symmetry xy-plane to obtain $\tilde{X}_j$; (2) we compute the euclidean distance $\mathbf{d}_{\text{center}}$ between the centroids of X_i and $\tilde{X}_j$ as well as the Chamfer Distance $\mathbf{d}_{\text{chamfer}}$ between X_i and $\tilde{X}_j$; (3) c_i and c_j are considered to be symmetric and merged into one part by the union operation once $\mathbf{d}_{\text{center}}$ and $\mathbf{d}_{\text{chamfer}}$ are smaller than the thresholds τ_1 and τ_2 respectively (τ_1 and τ_2 are set to 0.15 in our experiments). We iterate over all possible pairs of convex hulls in $\mathbb{C}$ with aforementioned steps to merge all symmetric pairs, then obtain the new set of parts $\tilde{\mathbb{C}}$ for mesh M.

Symmetry-Based Random Transformation. We then apply the following four transformations on the set $\tilde{\mathbb{C}}$ of mesh M: (1) randomly *cutting-out* some parts from $\tilde{\mathbb{C}}$; (2) randomly *scaling* $\tilde{\mathbb{C}}$ along three dimensions; (3) randomly *translating* $\tilde{\mathbb{C}}$ via adding random offsets on x and y coordinates, where we do not perform translation for the z coordinate in order to maintain the shape symmetricity along the xy-plane; (4) randomly *rotating* $\tilde{\mathbb{C}}$ around z-axis by 90°, 180°, or 270°, where no rotation around x and y axes is perform for keeping the symmetricity. The mesh after applying these random transformations is denoted as $\tilde{M}$.

Given two meshes $\tilde{M}_1$ and $\tilde{M}_2$ which are already gone through the aforementioned operations (i.e. merging symmetric parts and applying random transformations), we are now able to generate a novel mesh by combining $\tilde{M}_1$ and $\tilde{M}_2$ into a unified one. In practice, for tackling the potential complex structure appearing during the naive combination between meshes, we first apply ACD on $\tilde{M}_1 \bigcup \tilde{M}_2$ then merge the resultant convex hulls $\hat{\mathbb{C}}$ to achieve the final augmented mesh $\hat{M}$, which is symmetric and reasonable.

Rendering. Given the augmented mesh $\hat{M}$, we sample a random pose $\mathcal{V}$ to transform $\hat{M}$ from the object-center coordinate into the view-center coordinate, and colorize each convex hull in $\hat{\mathbb{C}}$ with different colors. The 2D RGB image I is rendered by viewing $\hat{M}$ from the same viewing angle $\mathcal{V}$, where in results $\{I, \hat{M}\}$ becomes the augmented training pair for learning our 3D reconstruction framework.

IV. Experiments

Dataset & Evaluation Metric. We conduct our experiments by adopting the dataset[1] proposed by GenRe [2], in which this dataset uses the ShapeNet [47] renderings for model training and testing. In particular, the training set of GenRe contains only three categories (i.e. car, chair, and airplane), and there are nine classes (i.e. bench, vessel, rifle, sofa, table, phone, speaker, lamp, and display) in the testing set which are disjoint from the ones in the training, for the specific purpose of evaluating the model generalizability on unseen classes. The original 3D shapes provided from GenRe are in format of voxels, we therefore adopt the Marching Cube algorithm [48] to turn them into meshes for our experiments. Specifically, each mesh is rendered in 20 random view angles, hence there are 3,000 meshes (1,000 per training class) with their corresponding 60,000 images for training, and in total 285 meshes with their corresponding 5,700 images for testing. In addition to the meshes from the original shapes in GenRe dataset, we generate other 6,000 augmented meshes (similarly, each with 20 random views) via our proposed CvxRearrangement strategy, we hence have in total 9,000 meshes with 180,000 RGB images to train our proposed framework for single-view 3D mesh reconstruction. We follow GenRe to adopt the symmetric Chamfer Distance (CD) for evaluating the performance of shape reconstruction, where its computation is based on 1,024 points respectively sampled from the surfaces of the reconstructed mesh and the groundtruth target mesh.

[1] https://github.com/xiumingzhang/GenRe-ShapeHD

A. Evaluation on Unseen Categories

We make comparison with several view-centered single-view 3D reconstruction baselines: DRC [49] and MarrNet [50] reconstruct 3D shapes in terms of voxel representation; [21] (denoted as "Multi-View") represents the reconstructed 3D shapes via multi-view depth maps; Pixel2Mesh (P2M) [1] reconstructs 3D shape via deforming a single ellipsoid mesh according to the input 2D observation; and GenRe [2] is the baseline to provide state-of-the-art generalizability for unseen classes. Noting that, as aforementioned in the related works, although [24], [23] do provide better performance than GenRe, we do not include them into comparison due to their requirement of additional part-level annotations for their model training. For our proposed framework, we provide several variants: **Ours-P** denotes the intermediate primitive-based 3D reconstruction (K=16 primitives are used) produced by the primitive composition stage; **Ours** denotes the fine-grained shapes estimated by our full model; **Ours-Cvx** denotes the model variant which is trained without using any augmented data generated from CvxRearrangement; and **Ours-Oracle** denotes the results produced by our full models with accessing the groundtruth depth map of input image I.

	P2M	Ours
bench	.044	**.039**
vessel	.038	**.030**
riffle	.026	**.021**
sofa	.049	**.039**
table	.063	**.035**
phone	.038	**.032**
speaker	.061	**.045**
lamp	.051	**.036**
display	.046	**.036**
chair	.055	**.038**
airplane	.028	**.022**
car	.037	**.035**
cabinet	.049	**.039**
Average	.045	**.034**

TABLE I
RECONSTRUCTION ERRORS IN TERMS OF CD LOSS (LOWER THE BETTER) FOR ALL 13 CATEGORIES SEEN DURING TRAINING. OUR PROPOSED METHOD ACHIEVES BETTER RECONSTRUCTION ON ALL CATEGORIES THAN PIXEL2MESH (P2M) [1].

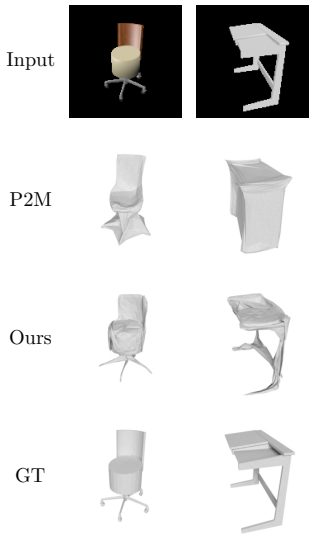

Fig. 5. Qualitative examples of reconstruction on seen classes. Each column from top to bottom: input RGB image, result from Pixel2Mesh (P2M), our result, and the groundtruth target shape.

As the quantitative results shown in Table II, our proposed method achieves the best performance in terms of the generalizability for reconstructing shapes on unseen classes, even without fine-grained deformation (i.e. Ours-P), and the improvement from Ours-Cvx to Ours clearly shows the benefit brought by our CvxRearrangement augmentation. Moreover, we also experiment to additionally provide our augmented meshes for training Pixel2Mesh (denoted as P2M+Cvx in

Table II), where we can see that our CvxRearrangement successfully boosts the performance of Pixel2Mesh on nearly all categories thus verifying the effectiveness of our novel strategy of mesh augmentation for different method. We also provide some qualitative examples in Figure 6. We can observe that, though Pixel2Mesh and GenRe seem to reconstruct reasonable shapes when viewing from the same view-angle as the input images, their reconstruction for the occluded parts (i.e. regions which are invisible on input images) are obviously inconsistent with the target shapes. In contrast, our proposed model produces more reasonable shapes even at the back view, as shown in column (i). Moreover, though our primitive-based representation might be not well aligned with the expectation by humans, such representation still contributes to improving the model generalizability for unseen categories, with all achieved without leveraging any part annotations.

B. Evaluation on Seen Categories

Here we also provide the comparison on reconstructing seen classes with respect to Pixel2Mesh [1], which is among the state-of-the-art methods of view-centered single-view mesh reconstruction for the classes seen during model training. Here we follow the same setting as Pixel2Mesh to conduct experiments on the 3D-R2N2 [3] dataset which contains 3D shapes from thirteen ShapeNet classes. Quantitative results provided in Table I show that our proposed method has superior reconstruction performance for all the classes in comparison to Pixel2Mesh. Some qualitative results are provided in Figure 5, where we can see that Pixel2Mesh cannot reconstruct the complex structure such as the thin chair legs in the left column and the hole structure in the right column. By contrast, our proposed framework, which actually models the coarse-to-fine reconstruction procedure, is capable of well handling these complicated shape as it regards the mesh as composition of primitives and further adopts the mesh deformation on each primitive to capture the fine-grained surface details.

V. Conclusion

We propose a primitive-based deformation framework to reconstruct view-centered meshes from single-view RGB images, in which it better generalizes to the object categories that are unseen during training. Moreover, we propose a novel strategy for 3D mesh augmentation to enrich and diversify the training data, which is shown to benefit the performance of different reconstruction models. Experiments conducted on both unseen and seen categories show the superior performance and generalizability of our proposed framework with respect to state-of-the-art baselines.

Acknowledgement. This project is supported by MOST 111-2636-E-A49-003 and MOST 111-2628-E-A49-018-MY4.

References

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

[2] X. Zhang, Z. Zhang, C. Zhang, J. B. Tenenbaum, W. T. Freeman, and J. Wu, "Learning to reconstruct shapes from unseen classes," in *NeurIPS*, 2018.

[3] C. B. Choy, D. Xu, J. Gwak, K. Chen, and S. Savarese, "3d-r2n2: A unified approach for single and multi-view 3d object reconstruction," in *ECCV*, 2016.

[4] M. Tatarchenko, A. Dosovitskiy, and T. Brox, "Octree generating networks: Efficient convolutional architectures for high-resolution 3d outputs," in *ICCV*, 2017.

[5] H. Xie, H. Yao, X. Sun, S. Zhou, and S. Zhang, "Pix2vox: Context-aware 3d reconstruction from single and multi-view images," in *ICCV*, 2019.

[6] H. Xie, H. Yao, S. Zhang, S. Zhou, and W. Sun, "Pix2vox++: multi-scale context-aware 3d object reconstruction from single and multiple images," *IJCV*, 2020.

[7] X. Yan, J. Yang, E. Yumer, Y. Guo, and H. Lee, "Perspective transformer nets: Learning single-view 3d object reconstruction without 3d supervision," in *NeurIPS*, 2016.

[8] S. Yang, M. Xu, H. Xie, S. Perry, and J. Xia, "Single-view 3d object reconstruction from shape priors in memory," in *CVPR*, 2021.

[9] H. Fan, H. Su, and L. J. Guibas, "A point set generation network for 3d object reconstruction from a single image," in *CVPR*, 2017.

[10] M. Gadelha, R. Wang, and S. Maji, "Multiresolution tree networks for 3d point cloud processing," in *ECCV*, 2018.

[11] A. Kurenkov, J. Ji, A. Garg, V. Mehta, J. Gwak, C. Choy, and S. Savarese, "Deformnet: Free-form deformation network for 3d shape reconstruction from a single image," in *WACV*, 2018.

[12] D. Novotny, D. Larlus, and A. Vedaldi, "Learning 3d object categories by looking around them," in *ICCV*, 2017.

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

[14] M. Niemeyer, L. Mescheder, M. Oechsle, and A. Geiger, "Differentiable volumetric rendering: Learning implicit 3d representations without 3d supervision," in *CVPR*, 2020.

[15] Q. Xu, W. Wang, D. Ceylan, R. Mech, and U. Neumann, "Disn: Deep implicit surface network for high-quality single-view 3d reconstruction," in *NeurIPS*, 2019.

[16] Y. Xu, T. Fan, Y. Yuan, and G. Singh, "Ladybird: Quasi-monte carlo sampling for deep implicit field based 3d reconstruction with symmetry," in *ECCV*, 2020.

[17] D. Du, Z. Zhang, X. Han, S. Cui, and L. Liu, "Vipnet: A fast and accurate single-view volumetric reconstruction by learning sparse implicit point guidance," in *3DV*, 2020.

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

[19] H. Li, W. Ye, G. Zhang, S. Zhang, and H. Bao, "Saliency guided subdivision for single-view mesh reconstruction," in *3DV*, 2020.

[20] J. Pan, X. Han, W. Chen, J. Tang, and K. Jia, "Deep mesh reconstruction from single rgb images via topology modification networks," in *ICCV*, 2019.

[21] D. Shin, C. C. Fowlkes, and D. Hoiem, "Pixels, voxels, and views: A study of shape representations for single view 3d object shape prediction," in *CVPR*, 2018.

[22] M. Tatarchenko, S. R. Richter, R. Ranftl, Z. Li, V. Koltun, and T. Brox, "What do single-view 3d reconstruction networks learn?" in *CVPR*, 2019.

[23] S. Han, J. Gu, K. Mo, L. Yi, S. Hu, X. Chen, and H. Su, "Compositionally generalizable 3d structure prediction," *ArXiv:2012.02493*, 2020.

[24] J. Wang and Z. Fang, "Gsir: Generalizable 3d shape interpretation and reconstruction," in *ECCV*, 2020.

[25] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *ICLR*, 2016.

[26] K. Mamou, E. Lengyel, and A. Peters, "Volumetric hierarchical approximate convex decomposition," in *Game Engine Gems 3*, 2016.

[27] H. Kato and T. Harada, "Learning view priors for single-view 3d reconstruction," in *CVPR*, 2019.

[28] H. Kato, Y. Ushiku, and T. Harada, "Neural 3d mesh renderer," in *CVPR*, 2018.

[29] A. Mao, C. Dai, L. Gao, Y. He, and Y.-j. Liu, "Std-net: Structure-preserving and topology-adaptive deformation network for 3d reconstruction from a single image," *ArXiv:2003.03551*, 2020.

[30] K. Mo, P. Guerrero, L. Yi, H. Su, P. Wonka, N. Mitra, and L. Guibas, "Structurenet: Hierarchical graph networks for 3d shape generation," *ACM Transactions on Graphics (TOG)*, 2019.

Models	bench	vessel	riffle	sofa	table	phone	speaker	lamp	display	Average
DRC [49]	.120	.109	.121	.107	.129	.132	.141	.131	.156	.127
MarrNet [50]	.107	.094	.125	.090	.122	.117	.123	.144	.149	.119
Multi-View [21]	.092	.092	.102	.085	.105	.110	.117	.142	.142	.109
P2M [1]	.096	.099	.114	.096	.122	.115	.122	.123	.144	.114
P2M+Cvx	.097	.091	.107	.093	.118	.109	.111	.115	.135	.108
GenRe [2]	.089	.092	.112	.082	**.096**	.107	.115	.124	.130	.105
Ours-P	.085	.078	.097	.082	.105	.090	.105	.103	.122	.096
Ours-Cvx	.084	.077	**.092**	.083	.105	.096	.109	.101	.130	.097
Ours	**.079**	**.075**	**.092**	**.077**	.098	**.087**	**.098**	**.097**	**.118**	**.091**
Ours-Oracle	.059	.050	.044	.065	.083	.063	.084	.062	.080	.066

TABLE II

QUANTITATIVE COMPARISON IN TERMS OF CD ERRORS FOR RECONSTRUCTING 9 NOVEL/UNSEEN CLASSES. OUR FULL FRAMEWORK ACHIEVES THE BEST PERFORMANCE ACROSS 8 OUT OF 9 CLASSES THUS SHOWING SUPERIOR GENERALIZABILITY WITH RESPECT TO OTHER BASELINES.

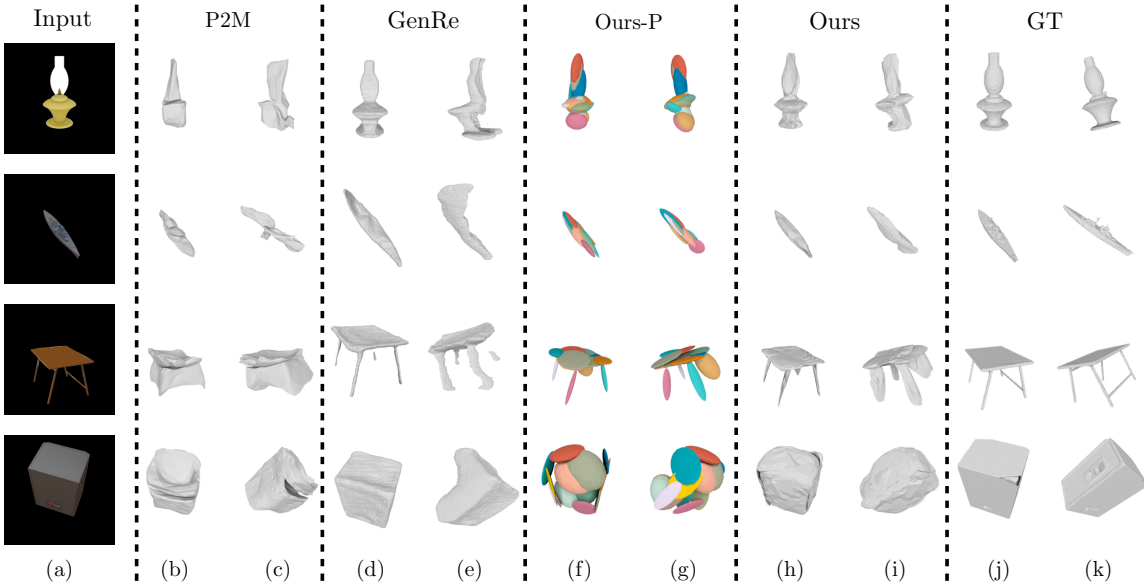

Fig. 6. Example results for single-view 3D reconstruction for the unseen classes. Each row from left to right: (a) the input RGB image, (b)(c) two views of Pixel2Mesh [1], (d)(e) two views of GenRe [2], (f)(g) two views of the primitive-based representation predicted by our primitive composition stage, (h)(i) two views of our full framework, and (j)(k) two views of the groundtruth target shape. Note that for (b)(d)(f)(h)(j) columns we visualize the shapes from the same viewing perspective as the corresponding input RGB images, while for (c)(e)(g)(i)(k) columns we particularly visualize the shapes from different view-angles to highlight the difference among various methods.

[31] J. Li, K. Xu, S. Chaudhuri, E. Yumer, H. Zhang, and L. Guibas, "Grass: Generative recursive autoencoders for shape structures," *ACM Transactions on Graphics (TOG)*, 2017.

[32] A. Yu, V. Ye, M. Tancik, and A. Kanazawa, "pixelNeRF: Neural radiance fields from one or few images," in *CVPR*, 2021.

[33] K. Mo, P. Guerrero, L. Yi, H. Su, P. Wonka, N. J. Mitra, and L. J. Guibas, "Structedit: Learning structural shape variations," in *CVPR*, 2020.

[34] R. Wu, Y. Zhuang, K. Xu, H. Zhang, and B. Chen, "Pq-net: A generative part seq2seq network for 3d shapes," in *CVPR*, 2020.

[35] K. Mo, S. Zhu, A. X. Chang, L. Yi, S. Tripathi, L. J. Guibas, and H. Su, "Partnet: A large-scale benchmark for fine-grained and hierarchical part-level 3d object understanding," in *CVPR*, 2019.

[36] C.-Y. Sun, Q.-F. Zou, X. Tong, and Y. Liu, "Learning adaptive hierarchical cuboid abstractions of 3d shape collections," *ACM Transactions on Graphics (TOG)*, 2019.

[37] C. Zou, E. Yumer, J. Yang, D. Ceylan, and D. Hoiem, "3d-prnn: Generating shape primitives with recurrent neural networks," in *ICCV*, 2017.

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

[39] B. Deng, K. Genova, S. Yazdani, S. Bouaziz, G. Hinton, and A. Tagliasacchi, "Cvxnet: Learnable convex decomposition," in *CVPR*, 2020.

[40] D. Paschalidou, A. O. Ulusoy, and A. Geiger, "Superquadrics revisited: Learning 3d shape parsing beyond cuboids," in *CVPR*, 2019.

[41] D. Paschalidou, L. V. Gool, and A. Geiger, "Learning unsupervised hierarchical part decomposition of 3d objects from a single rgb image," in *CVPR*, 2020.

[42] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016.

[43] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *MICCAI*, 2015.

[44] N. Ravi, J. Reizenstein, D. Novotny, T. Gordon, W.-Y. Lo, J. Johnson, and G. Gkioxari, "Accelerating 3d deep learning with pytorch3d," *ArXiv:2007.08501*, 2020.

[45] H. G. Barrow, J. M. Tenenbaum, R. C. Bolles, and H. C. Wolf, "Parametric correspondence and chamfer matching: Two new techniques for image matching," in *IJCAI*, 1977.

[46] W. Wang, D. Ceylan, R. Mech, and U. Neumann, "3dn: 3d deformation network," in *CVPR*, 2019.

[47] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su *et al.*, "Shapenet: An information-rich 3d model repository," *ArXiv:1512.03012*, 2015.

[48] W. E. Lorensen and H. E. Cline, "Marching cubes: A high resolution 3d surface construction algorithm," *ACM Transactions on Graphics (TOG)*, 1987.

[49] S. Tulsiani, T. Zhou, A. A. Efros, and J. Malik, "Multi-view supervision for single-view reconstruction via differentiable ray consistency," in *CVPR*, 2017.

[50] J. Wu, Y. Wang, T. Xue, X. Sun, W. T. Freeman, and J. B. Tenenbaum, "Marrnet: 3d shape reconstruction via 2.5 d sketches," in *NeurIPS*, 2017.