DEEP PHYSICS-BASED DEFORMABLE MODELS FOR EFFICIENT SHAPE ABSTRACTIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

Efficient shape abstraction is challenging due to the complex geometries of natural objects. Recent methods learn to represent objects using a set of simple primitives or fit locally parameterized deformable models to the target shapes. However, in these methods, the primitives used do not always correspond to real parts or lack the geometric flexibility for interpretability. In this paper, we investigate salient and efficient primitive descriptors for accurate shape abstractions, and propose *Deep Physics-based Deformable Model (DPDM)*. DPDM employs global deformations with parameter functions and local deformations. These properties enable DPDM to abstract complex object shapes with significantly fewer primitives that offer broader geometry coverage and finer details. DPDM learning formulation is based on physics-based modeling (i.e., dynamics and kinematics) to enable multiscale explainable abstractions. Extensive experiments on *ShapeNet* demonstrate that DPDM outperforms the state-of-the-art (SOTA) methods in terms of reconstruction accuracy by a notable margin. We also demonstrate the robustness and abstraction ability of DPDM by visualizing the semantic consistency which offer interpretability without any part decomposition prior. Experiments on *ACDC*, *M&Ms*, and *M&Ms-2* further show the generalization ability of DPDM for object segmentation.

1 INTRODUCTION

Abstracting complex object shapes with few number of primitives that offer efficiency and explainability has been a long standing goal in computer vision, medical image analysis and graphics. It can be used in a variety of downstream tasks, such as shape reconstruction, object classification and segmentation. Recent methods utilize deep neural networks to decompose objects into primitives (Paschalidou et al., 2019; 2020; Tulsiani et al., 2017; Niu et al., 2018; Zou et al., 2017; Hao et al., 2020; Paschalidou et al., 2021; Deng et al., 2020). These primitive-based methods interpret shapes as a union of simple parts (e.g., cuboids, spheres, or superquadrics), offering explainable abstraction of object shapes. To achieve high reconstruction accuracy, these methods require joint optimization of a large number of primitives which does not correspond often to the object parts and therefore limits the interpretability of the output. Therefore, devising methods that can discover a fewer number of primitives for efficiency, robustness and improved abstraction of complex shapes is an active research area. The use of fewer primitives to estimate complex object shapes with abstraction requires discovery of primitives with broader and interpretable robust parametrization.

In this paper, we investigate salient and efficient primitive descriptors to address flexible and explainable shape abstractions for complex objects with a minimal number of primitives. We take our inspiration from the conventional physics-based deformable models (PDMs) (Metaxas, 2012; Nealen et al., 2006), which are capable of estimating and representing object shapes with strong abstraction ability and have been successfully applied to shape modeling in natural scenes, medical imaging and graphics. A major issue of PDMs is that they rely on prior knowledge (i.e., handcrafted parametric initialization and optimization) for specific shape abstractions, which limits the usage of PDMs for general automated shape modeling. In addition, they use global deformations with constant parameters, which limits the geometric flexibility. To address these limitations, we augment PDMs with strong abstraction ability and integrate it into a learning-based framework, named *Deep Physics-based Deformable Models (DPDM)*, as illustrated in Fig. 1.

Compared to the traditional PDMs, we make use of deep neural networks to learn geometric representations of object shapes and overcome the parametric initialization limitation. In addition, we generalize the constant global parameters in PDMs by using global parameter functions, which offer broader shape coverage and improve the shape abstraction accuracy (e.g., shape aspect ratio, tapering and bending functions). To further enhance the shape coverage of DPDM, we employ a diffeomorphic mapping which preserves shape topology to predict local non-rigid deformations for shape details beyond the coverage of global deformations.

DPDM also uses the PDM notion of "external forces" to minimize the divergence between the predicted primitives and the target shapes Metaxas (2012) during training. This allows us to use physics-based kinematic formulations and Jacobians to compute transformations between data space and the generalized latent parameter space for improved optimization.

To evaluate the proposed DPDM, we conducted extensive experiments covering various problem settings on shape abstraction tasks. Most noticeably, DPDM outperforms SOTAs by 3.4% using only a single primitive on the core thirteen shape categories of *ShapeNet*. We also show the improved abstraction accuracy, consistent semantic correspondence across the same shape category, and interpretable visualization results on *ShapeNet*, compared to SOTAs. Moreover, we demonstrate the generalization ability of DPDM to cardiac MR segmentation, where DPDM achieves significant improvement on *ACDC, M&Ms* and *M&Ms-2* by 4.0%, 2.3%, and 4.2%, respectively.

2 Related Work

3D shape representation can be categorized into several mainstreams: (1) voxel-based methods (Choy et al., 2016; Wu et al., 2016) leverage voxels to capture 3D object geometry. These methods usually require large memory and computation resources. Some methods reduce the memory cost (Maturana & Scherer, 2015). But the implementation complexity of these methods increases significantly. (2) Point Cloud methods (Fan et al., 2017; Qi et al., 2017) require less computation, but additional post-processing is necessary to address the lack of surface connectivity for mesh generation. (3) Mesh-based (Liao et al., 2018; Groueix et al., 2018) and (4) Implicit representation-based methods (Mescheder et al., 2019; Chen & Zhang, 2019; Park et al., 2019) can yield smooth shape surfaces, but most of them lack output interpretability (abstraction ability). (5) primitive-based methods (Tulsiani et al., 2017; Paschalidou et al., 2019) represent object shapes by deforming a number of primitives, each of which is fully defined by a set of shape parameters.

Primitive-based methods. Our approach falls into primitive-based shape abstractions which have been revisited in deep learning and have recently demonstrated promising results. Paschalidou et al. (2019) developed a method that combines superquadrics with deep networks. Given a prior decomposition of an object, it estimates sets of superquadric parts that enable 3D shape parsing. This method has been further extended to estimate hierarchical parts from 3D data (Paschalidou et al., 2020). Other shapes such as cuboids (Tulsiani et al., 2017; Niu et al., 2018; Zou et al., 2017), spheres (Hao et al., 2020; Paschalidou et al., 2021) and convexes (Deng et al., 2020) have also been used for primitive-based reconstruction. However, these basic parts only offer limited shape coverage and cannot address accurate estimation of complex shapes that require multiscale abstractions.

Parameterized deformable models. Prior research works developed parameterized deformable models that abstracted multiple shapes with relatively few parameters. A notable example is the work of Kass et al. (1988) which exploited computational physics in the modeling process and proposed snakes, a locally parameterized deformable model. The snake formulation employs a force field computed from data to fit the model. Nevertheless, snakes which use locally defined deformations does not intrinsically offer shape abstractions. Pentland (1987) addressed partially the problem of shape abstraction by using superquadric ellipsoids that can deform using a few global parameters. Terzopoulos & Metaxas (1991) developed a new physics-based framework offering multiscale global and local deformations, and demonstrated its power using deformable superquadrics. Although their physics-based framework was able to address complex shapes and motion estimations of objects, it relies on prior object segmentation (Jones & Metaxas, 1998). In addition, modeling global deformations with constant parameters offers limited geometric coverage.

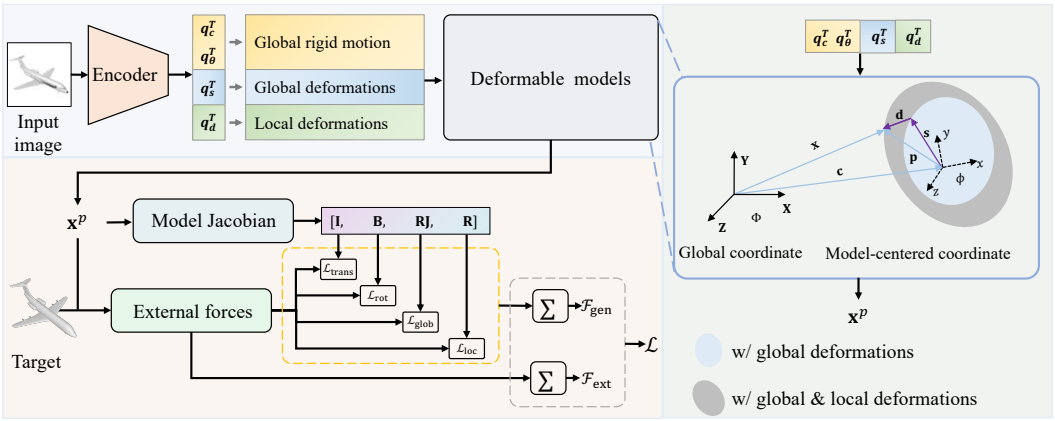


Figure 1: Overview of training Deep Physics-based Deformable Model (DPDM). Given an input $\mathcal{X}$, DPDM predicts a set of low dimensional latent representations $\mathbf{q}_c$, $\mathbf{q}_\theta$, $\mathbf{q}_s$, and $\mathbf{q}_d$ that describe global rigid motion, global and local deformations of the deformable model. For each reconstructed primitive of the deformable model $\mathbf{x}^p$, DPDM calculates the externally applied forces and the model Jacobian matrices which transform the external forces from the data space to the latent parameter space. The generalized force loss $\mathcal{F}_{\text{gen}}$ optimizes the deformable model parameters in terms of translation $\mathcal{L}_{\text{trans}}$, rotation $\mathcal{L}_{\text{rot}}$, global deformations $\mathcal{L}_{\text{glob}}$ and local deformations $\mathcal{L}_{\text{loc}}$. The training loss is a weighted summation of both the external force loss $\mathcal{F}_{\text{ext}}$ and the generalized force loss $\mathcal{F}_{\text{gen}}$.

3 METHOD

Given an input image (or 3D data from a scene) $\mathcal{X}$ to be segmented (or reconstructed), the goal of the proposed method is to incorporate a differentiable deformable model to predict P primitives that best describe the target shape. Each primitive is represented by a set of low dimensional parameters $\mathbf{q}$ with global and local deformations. We present the parameterization and network optimization of the proposed deformable model in Secs. 3.1 and 3.2, respectively. The training overview is given in Fig. 1.

3.1 DEFORMABLE MODEL GEOMETRY AND PARAMETERIZED DEFORMATIONS

We begin by summarizing the concept of physics-based deformable models and, along the way, introduce the notations. Geometrically, DPDM models a deformable primitive as a closed surface defined on a domain $\Omega \in \mathbb{R}^3$ with a model-centered coordinate ϕ. As shown in Fig. 1, given a point on the primitive surface, its location $\mathbf{x} = (x, y, z)$ *w.r.t.* the world coordinate Φ is:

$$\mathbf{x} = \mathbf{c} + \mathbf{R}\mathbf{p} = \mathbf{c} + \mathbf{R}(\mathbf{d} + \mathbf{s}), \tag{1}$$

where $\mathbf{c}$ and $\mathbf{R}$ represent the translation and rotation of the model's coordinate system ϕ *w.r.t.* the world coordinate system Φ; $\mathbf{p}$ denotes the relative position of the point on the primitive surface *w.r.t.* ϕ, which includes global deformation $\mathbf{s}$ and local deformation $\mathbf{d}$. Global deformations are expected to efficiently capture salient features of natural shapes using a minimum number of parameters, and therefore primitives, while local deformations allow the model to represent the fine-scale structures of complex real-world objects. We refer the readers to (Metaxas, 2012) for more details about the standard geometry formulation of PDMs.

Primitives with parameter functions. Instead of relying on the geometric coverage of such shapes using constant parameters, we create a novel shape parameterization that is based on parameter functions. This provides shapes with more flexibility, makes it possible to abstract complex objects with much fewer primitives and enhances shape abstration's explainability.

Due to their versatility in shape representation and abstraction, in this work, we generalize constant parameter superquadrics by defining a new class of deformable model primitives whose parameters are functions. These new primitives have improved shape coverage with explainable parameterization that is necessary for computer vision and medical imaging applications (Oblak et al., 2019;

Paschalidou et al., 2019; Park et al., 1995). Due to space limitations, the detailed definition of the parameterized primitive formulation $\mathbf{e}$ and its reference shape $\mathbf{s}$ are given in Appendix A.1.

Global deformations with parameter functions. To further improve the geometric coverage of these primitives, we introduce parameterized tapering and bending deformation functions. These additional global deformations are defined as continuously differentiable and commutative functions following (Barr, 1987). Specifically, due to their suitability for natural objects, we add linear tapering and bending to the primitives $\mathbf{e}$ and further express the novel deformation $\mathbf{q_s}$ with parameter functions as following:

$$\mathbf{q_s} = [\alpha(\mathbf{u}), \varepsilon(\mathbf{u}), \tau(\mathbf{u}), \beta(\mathbf{u})]^\top, \tag{2}$$

where $\mathbf{u} = (u, v)$ is the material coordinates of the primitive with $-\pi/2 \leq u \leq \pi/2$ and $-\pi \leq v \leq \pi$ (Wang et al., 2008); $\alpha(\mathbf{u})$ includes a scaling parameter function and three aspect ratio parameter functions; $\varepsilon(\mathbf{u})$ is the squareness function; $\tau(\mathbf{u})$ is the tapering parameter function; $\beta(\mathbf{u})$ includes the bending magnitude function, the location function, and the influence region function. See detailed formulations of these parameter functions in Appendix A.2. Note that our formulation can be applied to any primitive and its global deformation definition by replacing its constant parameters with differentiable parameter functions.

Diffeomorphic local deformations. We use local deformations to capture fine details beyond the coverage of global deformations. Previous approaches (Metaxas, 1992) adopted the finite element method (Zienkiewicz et al., 1977) to estimate local deformations. This requires handcrafted design of shape functions for the chosen fine elements with additional computational costs for accurate local deformation estimation. In this paper, we introduce a diffeomorphic mapping to estimate the local deformations $\mathbf{q}_d$. Due to the differentiable and invertible properties of diffeomophism, it preserves topology and guarantees one-to-one mapping during deformations (Dalca et al., 2018). In addition, since the global deformations used are invertibale, the composed deformation of global and local deformations in our model is invertible and smooth, which thus facilitates the learning of dense semantic correspondences for shape abstraction. To be specific, given the encoded local feature l from $\zeta_\mu^{-1}(\cdot)$, we first use a convolution layer to map l to a vector field v_0, and then map v_0 to a stationary velocity field (SVF) v using a Gaussian smoothing layer. v is defined via the ordinary differential equation Arsigny et al. (2006):

$$\frac{d\psi^{(t)}}{dt} = v(\psi^{(t)}), \tag{3}$$

where $\psi^{(t)}$ is the path of diffeomorphic non-rigid deformation field parameterized by $t \in [0, 1]$ and $\psi^{(0)} = Id$ is an identity transformation. To obtain the final local non-rigid deformation $\mathbf{q}_d = \psi^{(1)}$ at time $t = 1$, we follow (Arsigny et al., 2006; Dalca et al., 2018) and employ an Euler integration with a scaling and squaring layer (SS) to solve Eq. 3. Details are given in Appendix A.3.

Kinematics and Dynamics for DPDM. From Eq. (1), we can derive out the velocity of a point on the primitive surface as

$$\dot{\mathbf{x}} = \dot{\mathbf{c}} + \dot{\mathbf{R}}\mathbf{p} + \mathbf{R}\dot{\mathbf{p}} = \dot{\mathbf{c}} + \mathbf{B}\dot{\theta} + \mathbf{R}\dot{\mathbf{s}} + \mathbf{R}\mathbf{S}\dot{\mathbf{q}}_d, \tag{4}$$

where $\cdot$ denotes the first-order derivative; $\mathbf{B} = \partial\mathbf{R}\mathbf{p}/\partial\theta$, with θ the rotational coordinates and $\dot{\mathbf{s}} = [\partial\mathbf{s}/\mathbf{q}_s]\dot{\mathbf{q}}_s = \mathbf{J}\dot{\mathbf{q}}_s$, with $\mathbf{J}$ the Jacobian matrix of the model-centered coordinates ϕ *w.r.t.* the global deformation parameters at each point. We set the shape matrix $\mathbf{S}$ to identity matrix $\mathbf{I}$ in DPDM since we use one-to-one mapping for local deformation estimation. We note that the size of the Jacobian matrix is determined by the type of global deformations used. All the non-zero entries of the Jacobian matrix used are given in Appendix A.3. Eq. 4 can be further written in the form:

$$\dot{\mathbf{x}} = [\mathbf{I}, \mathbf{B}, \mathbf{R}\mathbf{J}, \mathbf{R}]\dot{\mathbf{q}} = \mathbf{L}\dot{\mathbf{q}}, \tag{5}$$

where $\mathbf{L}$ is the deformable model's Jacobian matrix that includes the Jacobians $\mathbf{J}$ for translation, rotation and deformations (Metaxas, 2012).

In the physics-based modeling paradigm, our deformable model continuously deforms from an initial shape (e.g., a sphere) to the target shape using the Lagrangian equations of motion given as:

$$\mathbf{M}\ddot{\mathbf{q}} + \mathbf{C}\dot{\mathbf{q}} + \mathbf{K}\mathbf{q} = g_q + f_q, \tag{6}$$

where $\cdot\cdot$ denotes the second-order derivative; $\mathbf{M}, \mathbf{C}, \mathbf{K}$ are the mass, damping and stiffness matrices, respectively; g_q is the inertial forces generated from the dynamic coupling between the local and

global deformations; f_q is the generalized forces which we will explain in the next. In this paper, we set $\mathbf{M} = \mathbf{0}$, $\mathbf{C} = \mathbf{1}$, $\mathbf{K} = \mathbf{0}$, $g_q = 0$ and use a simplified Lagrangian dynamic model given as:

$$\dot{\mathbf{q}} = f_q. \tag{7}$$

In this way, we formulate our deep physics-based deformable model. During training, by explicitly minimizing the generalized forces f_q, we can constrain the optimization of $\mathbf{q}$ in a physically regularized space, we term **lantent parameter space**, which ensures reaching the global minimum instead of a local one. In Sec. 3.2, we will show that, with the kinematics and Lagrangian dynamics, our physics-based deformable model can project the generalized force into the sub-optimization space corresponding to each deformation component, making it possible to explicitly supervise the learning of each deformation component, i.e. translation, rotation, global deformation and local deformation.

3.2 DPDM Training and Network Losses

Inspired by previous physics-based deformable models (Nealen et al., 2006; McInerney & Terzopoulos, 1996), we define the following loss function to train and optimize DPDM:

$$\mathcal{L} = \lambda_{\text{ext}}\mathcal{L}_{\text{ext}} + \lambda_{\text{gen}}\mathcal{L}_{\text{gen}}, \tag{8}$$

which is a weighted summation of the loss $\mathcal{L}_{\text{ext}}$ computed using external forces from the **data space** and the loss $\mathcal{L}_{\text{gen}}$ computed using generalized forces from the **latent parameter space**; λ_{ext} and λ_{gen} are their weights, respectively.

External model loss. To fit the primitives to the target shape, we train an encoder $\zeta_\mu^{-1}(\cdot)$ to optimize the loss $\mathcal{L}_{\text{ext}}$ computed using the external forces applied to the primitives:

$$\mathcal{L}_{\text{ext}} = \frac{1}{P}\sum_{p=1}^{P}\sum_{r} f_r^p = \frac{\gamma}{P\left|\hat{\mathcal{M}}_p\right|}\sum_{p=1}^{P}\sum_{r\in\hat{\mathcal{M}}_p}\mathcal{D}(\hat{\mathcal{M}}_p, \mathcal{T}). \tag{9}$$

$\mathcal{D}(\hat{\mathcal{M}}_p, \mathcal{T}) = \|\mathbf{x}^p - \mathcal{T}\|_2$ is the distance function of all points on the target shape $\mathcal{T}$ to all points r on p-th predicted primitive $\hat{\mathcal{M}}_p$, where P is the total number of used primitives and γ is the strength factor for the external forces f_r^p. The external force loss $\mathcal{L}_{\text{ext}}$ measures how well the primitives are deformed to fit the target shape in the data space during training.

Generalized model loss. Given Eq. 5, using the principle of virtual work [1], we can determine the relationship between the generalized forces and the external forces. In particular, the energy of the p-th primitive due to translation rotation and deformations, $\mathcal{E}_f^p$, is expressed as:

$$\mathcal{E}_f^p = \int (f^p)^\top d\mathbf{x}^p = \int (f^p)^\top \mathbf{L}^p d\mathbf{q}^p = \int f_q^p d\mathbf{q}^p, \tag{10}$$

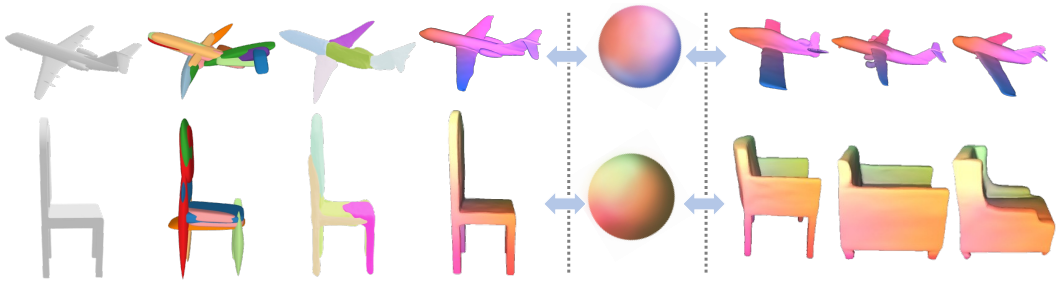
where f_q^p is the generalized forces applied to $\hat{\mathcal{M}}_p$ and is computed using the external forces f^p and the model Jacobian matrix $\mathbf{L}^p$. This allows us to employ the generalized forces f_q^p in the latent parameter space to facilitate the primitive prediction. Specifically, given the model Jacobian $\mathbf{L}^p = [\mathbf{I}^p, \mathbf{B}^p, \mathbf{R}^p\mathbf{J}^p, \mathbf{R}^p]$ (Metaxas, 2012), we express f_q^p as:

$$\begin{aligned} f_q^p = (f^p)^\top\mathbf{L}^p &= [(f^p)^\top, (f^p)^\top\mathbf{B}^p, (f^p)^\top\mathbf{R}^p\mathbf{J}^p, (f^p)^\top\mathbf{R}^p] \\ &= [(f_c^p)^\top, (f_\theta^p)^\top, (f_s^p)^\top, (f_d^p)^\top], \end{aligned} \tag{11}$$

where f_c^p and f_θ^p represent the generalized forces for the translation and rotation; f_s^p and f_d^p represent the generalized forces for the global and local deformations. In our learning framework, in addition to the external forces, we also train the encoder $\zeta_\mu^{-1}(\cdot)$ to optimize these four generalized force components and define the generalized model loss $\mathcal{L}_{\text{gen}}$ as:

$$\mathcal{L}_{\text{gen}} = \mathcal{L}_{\text{trans}} + \mathcal{L}_{\text{rot}} + \mathcal{L}_{\text{glob}} + \mathcal{L}_{\text{loc}}, \tag{12}$$

[1] In mechanics, virtual work is the total work done by the applied forces on a mechanical system as it moves through a set of virtual displacements.



(a) Target (b) Suq (c) NP (d) DPDM (e) Basic primitive (f) Shape correspondence

Figure 2: Visual results on *ShapeNet*. (a) Target meshes, (b) Suq (Paschalidou et al., 2019) with ∼20 primitives, (c) Neural Parts (Paschalidou et al., 2021) with 5 primitives, and (d) DPDM using one single primitive. Consistent semantic correspondences are illustrated in (e), (f) with color codes.

Table 1: Quantitative results on *ShapeNet*. We evaluate DPDM against Suq (Paschalidou et al., 2019), CvxNets (Deng et al., 2020), H-Suq (Paschalidou et al., 2020), and Neural Parts (NP) (Paschalidou et al., 2021). We report IoU and Chamfer-L_1 distance for comparison.

Category	IoU (↑)					Chamfer-L_1 (↓)				
	Suq	CvxNets	H-Suq	NP	DPDM	Suq	CvxNets	H-Suq	NP	DPDM
airplane	0.456	0.598	0.529	0.611	**0.637**	0.122	0.093	0.175	0.089	**0.081**
bench	0.202	0.461	0.437	0.502	**0.529**	0.114	0.133	0.153	0.108	**0.098**
cabinet	0.110	0.709	0.658	0.681	**0.725**	0.087	0.102	0.087	0.083	**0.081**
car	0.650	0.675	0.702	0.719	**0.727**	0.117	0.103	0.141	0.127	**0.101**
chair	0.176	0.491	0.526	0.532	**0.549**	0.138	0.337	0.114	0.107	**0.096**
display	0.200	0.576	0.633	0.646	**0.662**	0.106	0.223	0.137	0.098	**0.092**
lamp	0.189	0.311	0.441	0.402	**0.449**	0.189	0.795	0.169	0.153	**0.148**
speaker	0.136	0.620	0.660	0.693	**0.721**	0.132	0.462	0.108	0.128	**0.096**
rifle	0.519	0.515	0.435	0.537	**0.541**	0.127	0.106	0.203	0.189	**0.101**
sofa	0.122	0.677	0.693	0.712	**0.735**	0.106	0.164	0.128	0.107	**0.097**
table	0.180	0.473	0.491	0.531	**0.552**	0.110	0.358	0.122	0.102	**0.085**
phone	0.185	0.719	0.770	0.810	**0.813**	0.112	0.083	0.149	0.076	**0.072**
vessel	0.471	0.552	0.570	0.605	**0.642**	0.125	0.173	0.178	0.119	**0.107**
Average	0.277	0.567	0.580	0.614	**0.637**	0.122	0.245	0.143	0.114	**0.097**

where

$$\mathcal{L}_{\text{trans}} = \sum_{p=1}^{P} (f_c^p)^\top = \sum_{p=1}^{P} \sum_{r} (f_r^p)^\top, \qquad \mathcal{L}_{\text{glob}} = \sum_{p=1}^{P} (f_s^p)^\top = \sum_{p=1}^{P} \sum_{r} (f_r^p)^\top \mathbf{R}^p \mathbf{J}^p,$$

$$\mathcal{L}_{\text{rot}} = \sum_{p=1}^{P} (f_\theta^p)^\top = \sum_{p=1}^{P} \sum_{r} (f_r^p)^\top \mathbf{B}^p, \qquad \mathcal{L}_{\text{loc}} = \sum_{p=1}^{P} (f_d^p)^\top = \sum_{p=1}^{P} \sum_{r} (f_r^p)^\top \mathbf{R}^p,$$

are the generalized model losses associated with the translation, rotation, global and local model degrees of freedom, respectively. Note that, by decomposing the generalized forces into different components and by minimizing each force component, we can directly optimize the corresponding deformation components, which in turn ensures a global optimum.

4 Experiments

4.1 Settings and Datasets

We first evaluate the performance of DPDM on *ShapeNet* (Chang et al., 2015), a richly-annotated, large-scale dataset of 3D shapes. A subset of *ShapeNet* including 50k models and 13 major categories are used in our experiments. We split the dataset into training and testing sets following (Choy et al., 2016). We also test the general applicability of DPDM on object segmentation. We test on the

challenging task of cardiac MR image segmentation due to its complex shape and ill-defined boarders of the heart. Three different datasets are used for evaluation, including *ACDC* (Bernard et al., 2018), *M&Ms*, and *M&Ms-2* (Campello et al., 2021). The acquired cardiac MR images from all the three datasets were delineated by experienced clinical doctors, including the left ventricle (LV), right ventricle (RV), and the left ventricular myocardium (Myo). They contain 100, 350, and 160 cases, respectively. We train and test our network separately on the three image sets with manual labels, following the strategy of five-fold cross-validation.

4.2 Implementation Details

In all experiments, Adam (Kingma & Ba, 2014) is employed for optimization and the learning rate is initialized as 10^{-4}. We use a batch size of 32 and train the model for 300 epochs. All experiments are implemented with PyTorch and run on a Linux system with eight Nvidia A100 GPUs.

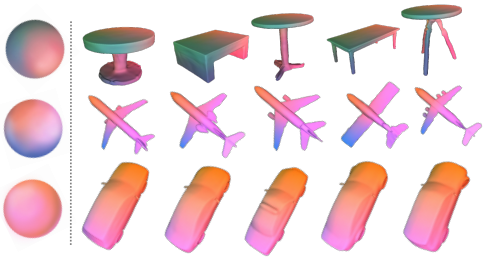


Figure 3: Illustration of interpretability of DPDM shape abstractions without any supervision or part prior. Consistent semantic correspondence is indicated by colors on reconstructed shapes (right) mapped from the initial primitives (left).

Similar to (Deng et al., 2020; Paschalidou et al., 2021; 2019), we draw 2k random samples from the surface of the target mesh, and sample 1k points for each generated primitive during training. During evaluation, we uniformly sample 100k points on the target/predicted meshes for the calculation of the volumetric Intersection over Union (IoU) and the Chamfer-$L1$ distance (CD). For cardiac MR segmentation, we resample all images in the three datasets to a spacing of 1.25 mm, and utilize data augmentation strategies, including random histogram matching, rotation, shifting, scaling, elastic deformation, and mirroring.

We empirically set the weights of the two losses in Eq. 8 to 0.6 and 0.4, respectively, which led to the best performance. The ablation study for the losses is given in Sec. 4.6.1. For a fair comparison with the other baselines, we use the standard ResNet18 (He et al., 2016) as the encoder $\zeta_{\mu}^{-1}(\cdot)$ for both tasks. The encoder output is followed by a fully connected layer to estimate four individual vectorized parameters that represent translation, rotation, global and local deformations.

4.3 Results on *ShapeNet*

We first compare DPDM to SOTAs (Paschalidou et al., 2019; 2020; 2021; Deng et al., 2020) on *ShapeNet*. We train our model with one **single** primitive and train other models following their reported experimental setups. Specifically, for Suq and H-Suq (Paschalidou et al., 2019; 2020), we use a maximum number of 64 primitives; for CvxNets (Deng et al., 2020) and Neural Parts (Paschalidou et al., 2021), we report the results using 50 and 5 primitives, respectively, which in their papers lead to the best performance. The quantitative results measured by IoU and Chamfer-L_1 distance are reported in Table 1. We observe that DPDM outperforms the other approaches for all the shape categories. Fig. 2 and Fig. 3 display a few qualitative examples using different methods. We find that while baseline methods use multiple primitives with obvious overlap to abstract partitions of shapes, DPDM captures the complete geometry of the chair and airplane using only one primitive. Moreover, DPDM demonstrates meaningful semantic correspondence among individual instances from the same category (see Fig. 2 (e), (f) and Fig. 3), corresponding partitions have similar color), indicating clear advantageous interpretability.

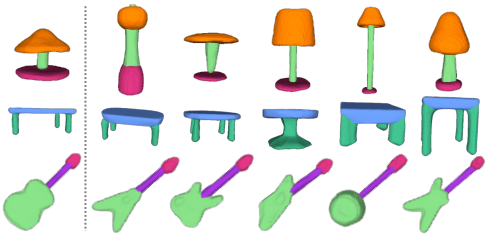


Figure 4: Results of part label transfer for abstraction capability evaluation (using a **single** primitive). The first column is the source shape, while the rest is the transferred labels from source shapes through learned dense correspondences.

4.4 Abstraction Capability

Part Label Transfer. We evaluate the abstraction capability through a part label transfer task (Wang et al., 2020) due to the lack of part decomposition prior for target shapes. Following (Deng et al., 2021), we use five labeled shapes from ShapeNet-Part (Yi et al., 2016) dataset as source shapes, and transfer their labels to other instances from the same category via the learned dense correspondences. The results in Fig. 4 illustrate the accurate semantic consistency across instances from the same category, e.g., the lampshades are always matched despite large variances of the object structures.

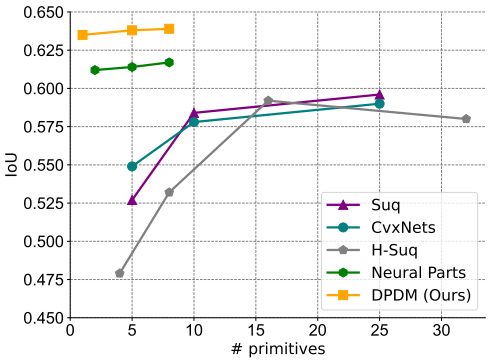


Figure 5: Analysis of reconstruction accuracy *v.s.* the number of primitives used. We compare the reconstruction performance with other primitive-based methods on *ShapeNet*.

Multi-primitive Fitting. DPDM is a general framework that can fit multiple primitives to the target shape. We provide the reconstruction accuracy on *ShapeNet* by varying the number of primitives and test the performance in terms of IoU. Note that our model shows leading reconstruction performance regardless of the number of primitives used as shown in Fig. 5. We also observe that the curve saturates when adding more primitives to our model. We attribute this to the efficient geometric coverage of our model where a single primitive is already sufficient for accurate abstractions. We also provide qualitative results in Fig. 7, where we use **three** and **two** primitives, respectively, for the abstraction of chairs and earphones, and visualize the results with colors. We observe that semantic consistency is still preserved using multiple primitives.

4.5 General Applicability

To demonstrate the general applicability of DPDM, we further test on cardiac MR segmentation and compare DPDM with TDAC (Hatamizadeh et al., 2020), the SOTA image segmentation method which uses learning-based deformable models. Note that to segment apical and basal regions in cardiac MR images is challenging, because the apical region is relatively small in the image and the basal region has unclear boundaries between the ventricles and the atria. We use three primitives to abstract the shapes of the left and right ventricles, and left ventricular myocardium. The results illustrate that the proposed DPDM outperforms the SOTA by a notable margin: 4.0% of Dice accuracy on *ACDC*, 2.3% on *M&Ms*, and 4.2% on *M&Ms-2*, respectively. Quantitative comparison with details is given in Appendix C.2.

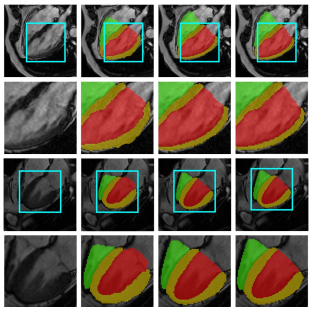


Figure 6: Results on cardiac MR segmentation. Zoom-in images are indicated by blue boxes in the previous images.

Fig. 6 shows that our predictions consistently capture both the anatomy and fine details of these challenging regions, while TDAC is less sensitive to the sharp edges. Although we observe that TDAC is able to recover the ventricle boundaries, the predicted details of both the ventricles (green & red) and the myocardium (yellow) are not smooth. More importantly, we note that our method can always preserve the topology changes of the objects, but TDAC only partially recovers the shape.

4.6 Ablation Studies

4.6.1 Impact of Loss Components

To evaluate the alignment performance of the predicted parameters, we train several partial variants of our method using different combinations of the loss functions. We first remove the external model loss $\mathcal{L}_{\text{ext}}$, and only keep the generalized model loss $\mathcal{L}_{\text{gen}}$ during training. We then experiment with the variant where only $\mathcal{L}_{\text{ext}}$ is employed, and the full model including both $\mathcal{L}_{\text{ext}}$ and $\mathcal{L}_{\text{gen}}$ for reconstruction. The results are shown in Table 2 and Fig. 8. We can see that our approach which uses multi-domain optimization performs better than those variants with partial components, suggesting the effectiveness of each technique component in DPDM.

Figure 7: Visual results on *ShapeNet* when fitting multiple primitives to the target shapes. The number of primitives used is given as prior. We employ 3 and 2 primitives for chairs and earphones, respectively. Semantic consistency within the same category is illustrated with colors.

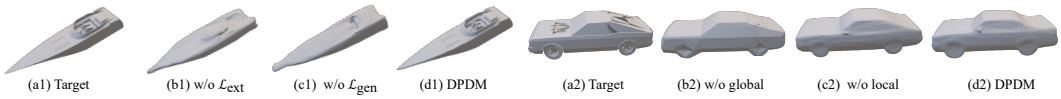


Figure 8: Ablation study on losses. We ablate the loss components as well as the global and local deformations, and show their impact qualitatively.

4.6.2 Impact of parameterized deformations

DPDM employs parameterized global deformations to abstract object shapes, and local deformations to estimate finer details. We study the effect of global and local deformations by training variants of our model by removing one component at a time. The results are reported in Table 2 and Fig. 8. We note that removing global parameter functions leads to significant performance drops. We hypothesize this is due to the major role of the global deformations, modeled as parameter functions, in capturing salient object structures. Compared to global deformations, local deformations focus on accurate estimation of shape details, especially for the object boundaries.

We also ablate the type of parameters of the primitive definitions and global deformations in Fig. 9. We observe that using parameter functions for both the primitive definitions and global deformations allows DPDM to capture the complex shape structure of the rifle with significant high fidelity.

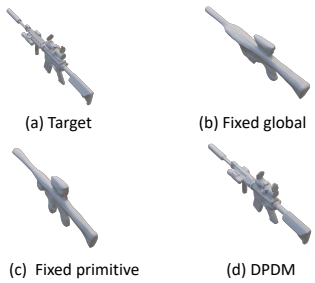


Figure 9: Ablation study on parameters of the primitives and deformations.

Settings				*M&Ms-2*		*ShapeNet*	
$\mathcal{L}_{\text{ext}}$	$\mathcal{L}_{\text{gen}}$	global	local	Dice (↑)	HD (↓)	CD (↓)	IoU (↑)
✗	✓	✓	✓	86.09	11.88	0.181	0.531
✓	✗	✓	✓	86.54	11.32	0.125	0.562
✓	✓	✗	✓	85.32	11.62	0.136	0.597
✓	✓	✓	✗	86.74	11.10	0.109	0.621
✓	✓	✓	✓	**87.24**	**10.51**	**0.097**	**0.637**

Table 2: Ablation studies on losses and deformations. We report Dice Score and Hausdorff Distance (HD) on *M&Ms-2* as well as Chamfer-L_1 Distance (CD) and IoU on *ShapeNet*.

5 Conclusion

In this work, we have introduced a novel and efficient physics-based learning approach for improved object shape abstractions. The generalized primitive formulation using parameter functions allows the proposed model to accurately capture the geometric structures of object shapes using significantly fewer shape components. Moreover, our physics-based modeling provides multiscale parameterized shape representation ability while preserving the semantic interpretation of the shape. Extensive experiments demonstrate that our automated approach yields both accurate and explainable shape abstractions on both shape reconstruction and object segmentation tasks. Our future work will consider including more primitive definitions (e.g., supertoroids, multigenous primitives) and global deformations (e.g., generalized shearing, twisting) to enhance the expressiveness of our primitives in more general and complex shape abstraction scenarios.

REFERENCES

Vincent Arsigny, Olivier Commowick, Xavier Pennec, and Nicholas Ayache. A log-euclidean framework for statistics on diffeomorphisms. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 924–931. Springer, 2006.

Alan H Barr. Global and local deformations of solid primitives. In *Readings in Computer Vision*, pp. 661–670. Elsevier, 1987.

Olivier Bernard, Alain Lalande, Clement Zotti, Frederick Cervenansky, Xin Yang, Pheng-Ann Heng, Irem Cetin, Karim Lekadir, Oscar Camara, Miguel Angel Gonzalez Ballester, et al. Deep learning techniques for automatic mri cardiac multi-structures segmentation and diagnosis: is the problem solved? *IEEE transactions on medical imaging*, 37(11):2514–2525, 2018.

Víctor M Campello, Polyxeni Gkontra, Cristian Izquierdo, Carlos Martín-Isla, Alireza Sojoudi, Peter M Full, Klaus Maier-Hein, Yao Zhang, Zhiqiang He, Jun Ma, et al. Multi-centre, multi-vendor and multi-disease cardiac segmentation: the m&ms challenge. *IEEE Transactions on Medical Imaging*, 40(12):3543–3554, 2021.

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.

Jieneng Chen, Yongyi Lu, Qihang Yu, Xiangde Luo, Ehsan Adeli, Yan Wang, Le Lu, Alan L Yuille, and Yuyin Zhou. Transunet: Transformers make strong encoders for medical image segmentation. *arXiv preprint arXiv:2102.04306*, 2021.

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*, pp. 5939–5948, 2019.

Christopher B Choy, Danfei Xu, JunYoung Gwak, Kevin Chen, and Silvio Savarese. 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. In *European conference on computer vision*, pp. 628–644. Springer, 2016.

Adrian V Dalca, Guha Balakrishnan, John Guttag, and Mert R Sabuncu. Unsupervised learning for fast probabilistic diffeomorphic registration. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 729–738. Springer, 2018.

Boyang Deng, Kyle Genova, Soroosh Yazdani, Sofien Bouaziz, Geoffrey Hinton, and Andrea Tagliasacchi. Cvxnet: Learnable convex decomposition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 31–44, 2020.

Yu Deng, Jiaolong Yang, and Xin Tong. Deformed implicit field: Modeling 3d shapes with learned dense correspondence. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10286–10296, 2021.

Haoqiang Fan, Hao Su, and Leonidas J Guibas. A point set generation network for 3d object reconstruction from a single image. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 605–613, 2017.

Thibault Groueix, Matthew Fisher, Vladimir G Kim, Bryan C Russell, and Mathieu Aubry. A papier-mâché approach to learning 3d surface generation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 216–224, 2018.

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

Ali Hatamizadeh, Debleena Sengupta, and Demetri Terzopoulos. End-to-end trainable deep active contour models for automated image segmentation: Delineating buildings in aerial imagery. In *European Conference on Computer Vision*, pp. 730–746. Springer, 2020.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Timothy N Jones and Dimitris N Metaxas. Image segmentation based on the integration of pixel affinity and deformable models. In *Proceedings. 1998 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No. 98CB36231)*, pp. 330–337. IEEE, 1998.

Michael Kass, Andrew Witkin, and Demetri Terzopoulos. Snakes: Active contour models. *International journal of computer vision*, 1(4):321–331, 1988.

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

Yiyi Liao, Simon Donne, and Andreas Geiger. Deep marching cubes: Learning explicit surface representations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2916–2925, 2018.

Daniel Maturana and Sebastian Scherer. Voxnet: A 3d convolutional neural network for real-time object recognition. In *2015 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pp. 922–928. IEEE, 2015.

Tim McInerney and Demetri Terzopoulos. Deformable models in medical image analysis: a survey. *Medical image analysis*, 1(2):91–108, 1996.

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*, pp. 4460–4470, 2019.

Dimitris Metaxas. Physics-based modeling of nonrigid objects for vision and graphics. *Technical Reports (CIS)*, pp. 444, 1992.

Dimitris N Metaxas. *Physics-based deformable models: applications to computer vision, graphics and medical imaging*, volume 389. Springer Science & Business Media, 2012.

Andrew Nealen, Matthias Müller, Richard Keiser, Eddy Boxerman, and Mark Carlson. Physically based deformable models in computer graphics. *Computer graphics forum*, 25(4):809–836, 2006.

Chengjie Niu, Jun Li, and Kai Xu. Im2struct: Recovering 3d shape structure from a single rgb image. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4521–4529, 2018.

Tim Oblak, Klemen Grm, Aleš Jaklič, Peter Peer, Vitomir Štruc, and Franc Solina. Recovery of superquadrics from range images using deep learning: A preliminary study. In *2019 IEEE International Work Conference on Bioinspired Intelligence (IWOBI)*, pp. 000045–000052. IEEE, 2019.

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*, pp. 165–174, 2019.

Jinah Park, Dimitri Metaxas, and Leon Axel. Volumetric deformable models with parameter functions: A new approach to the 3d motion analysis of the lv from mri-spamm. In *Proceedings of IEEE International Conference on Computer Vision*, pp. 700–705. IEEE, 1995.

Despoina Paschalidou, Ali Osman Ulusoy, and Andreas Geiger. Superquadrics revisited: Learning 3d shape parsing beyond cuboids. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10344–10353, 2019.

Despoina Paschalidou, Luc Van Gool, and Andreas Geiger. Learning unsupervised hierarchical part decomposition of 3d objects from a single rgb image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1060–1070, 2020.

Despoina Paschalidou, Angelos Katharopoulos, Andreas Geiger, and Sanja Fidler. Neural parts: Learning expressive 3d shape abstractions with invertible neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3204–3215, 2021.

Alex P Pentland. Perceptual organization and the representation of natural form. In *Readings in Computer Vision*, pp. 680–699. Elsevier, 1987.

Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30, 2017.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pp. 234–241. Springer, 2015.

Franc Solina and Ruzena Bajcsy. Recovery of parametric models from range images: The case for superquadrics with global deformations. *IEEE transactions on pattern analysis and machine intelligence*, 12(2):131–147, 1990.

Demetri Terzopoulos and Dimitri Metaxas. Dynamic 3 d models with local and global deformations: deformable superquadrics. *IEEE Transactions on pattern analysis and machine intelligence*, 13 (7):703–714, 1991.

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

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pp. 5998–6008, 2017.

Lingjing Wang, Xiang Li, and Yi Fang. Few-shot learning of part-specific probability space for 3d shape segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4504–4513, 2020.

Xiaoxu Wang, Dimitis Metaxas, Ting Chen, and Leon Axel. Meshless deformable models for lv motion analysis. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8. IEEE, 2008.

Haiping Wu, Bin Xiao, Noel Codella, Mengchen Liu, Xiyang Dai, Lu Yuan, and Lei Zhang. Cvt: Introducing convolutions to vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 22–31, 2021.

Jiajun Wu, Chengkai Zhang, Tianfan Xue, Bill Freeman, and Josh Tenenbaum. Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling. *Advances in neural information processing systems*, 29, 2016.

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

Olgierd Cecil Zienkiewicz, Robert Leroy Taylor, Perumal Nithiarasu, and JZ Zhu. *The finite element method*, volume 3. McGraw-hill London, 1977.

Chuhang Zou, Ersin Yumer, Jimei Yang, Duygu Ceylan, and Derek Hoiem. 3d-prnn: Generating shape primitives with recurrent neural networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 900–909, 2017.

A DETAILS OF THE DPDM FORMULATIONS

In this section, we provide more details of the formulation in the main paper and more examples of the global deformation functions we used in DPDM.

A.1 GENERALIZED SUPERQUADRIC-LIKE PRIMITIVE DEFINITION

DPDM employs superquadric-like primitive definitions with parameter functions that are useful in computer vision and medical imaging applications (Oblak et al., 2019; Paschalidou et al., 2019; Park et al., 1995). The definition of the parameterized primitive surface formulation $\mathbf{e}$ is given as:

$$\mathbf{e} = a_0(\mathbf{u}) \begin{bmatrix} a_1(\mathbf{u}) C_u^{\varepsilon_1(\mathbf{u})} C_v^{\varepsilon_2(\mathbf{u})} \\ a_2(\mathbf{u}) C_u^{\varepsilon_1(\mathbf{u})} S_v^{\varepsilon_2(\mathbf{u})} \\ a_3(\mathbf{u}) S_u^{\varepsilon_1(\mathbf{u})} \end{bmatrix}, \tag{13}$$

where $\mathbf{u} = (u, v)$ is the material coordinates with $-\pi/2 \leq u \leq \pi/2$ and $-\pi \leq v \leq \pi$ (Wang et al., 2008). The shape reference $S_\gamma^\varepsilon = \text{sgn}(\sin\gamma)|\sin\gamma|^\varepsilon$ and $C_\gamma^\varepsilon = \text{sgn}(\cos\gamma)|\cos\gamma|^\varepsilon$. Here, $a_0(\mathbf{u})$ is a scaling function, $a_1(\mathbf{u}), a_2(\mathbf{u}), a_3(\mathbf{u})$ are aspect ratio parameter functions, and $\varepsilon_1(\mathbf{u}), \varepsilon_2(\mathbf{u})$ are squareness parameter functions. In our definition, these parameters are no longer fixed values but extended to functional variables for improved expressivity and flexibility of the primitives. Note that $\mathbf{u} = (u, v)$ is not the input of the network, but the material (intrinsic) coordinates following the sampling strategy given in Sec. 4.2. This means, for each point in $\mathbf{u}$, there is a learned value for the parameter functions $a_i(\mathbf{u})$, $\varepsilon_i(\mathbf{u})$, $t_i(\mathbf{u})$ and $b_i(\mathbf{u})$, which provide broad geometry coverage for the primitives.

Without loss of generality, we assume that all the parameter functions used in this paper are functions of u, i.e., $a_i(\mathbf{u}) = a_i(u), \varepsilon_i(\mathbf{u}) = \varepsilon_i(u), t_i(\mathbf{u}) = t_i(u)$ and $b_i(\mathbf{u}) = b_i(u)$, allowing them to vary along one of the two material coordinates. We may also define the parameter functions as functions of v, which empirically lead to similar abstraction performance. Using parameters as functions of both u and v is not necessary for the experiments in this paper.

A.2 GLOBAL DEFORMATION FUNCTIONS

We provide detailed formulations for two examples with global deformation functions: 1) superquadric-like primitives with tapering and bending parameter functions; 2) superquadric-like primitives with twisting and bending parameter functions.

Superquadric-like primitives with tapering and bending. We follow the idea from Barr (1987); Solina & Bajcsy (1990) and define these global deformations as continuously differentiable as well as commutative. We integrate linear tapering and bending of the superquadric $\mathbf{e} = (e_1, e_2, e_3)^\top$ into one single parameterized deformation $\mathbf{T}$ and give the formulation of the reference shape as:

$$\begin{aligned} \mathbf{s}_{tp,b} &= \mathbf{T}(\mathbf{e}, t_1(u), t_2(u), b_1(u), b_2(u), b_3(u)) \\ &= \begin{pmatrix} (\frac{t_1(u)e_3}{a_0(u)a_3(u)w} + 1)e_1 + b_1(u)\cos(\frac{e_3(u)+b_2(u)}{a_0(u)a_3(u)w})\pi b_3(u) \\ (\frac{t_2(u)e_3}{a_0(u)a_3(u)w} + 1)e_2 \\ e_3 \end{pmatrix}, \end{aligned} \tag{14}$$

where $t_1(u), t_2(u)$ are the tapering parameter functions, $b_1(u)$ is the magnitude function, $b_2(u)$ is the location function, and $b_3(u)$ is the influence region function of bending. Here the material coordinate $w = 1$ since we use the primitive surface formulation in this paper instead of volumetric formulation.

Superquadric-like primitives with twisting and bending: We present another example of superquadric-like models with twisting and bending parameter functions. For a primitive $\mathbf{e} = (e_1, e_2, e_3)^\top$, the bending of the first axis along the other two axes is defined as a parameterized

deformation $\mathbf{T}_b$. Then the formulation of the reference shape is given as:

$$\begin{aligned}\mathbf{s}_b &= \mathbf{T}_b(\mathbf{e}, b_0(u), b_1(u), b_2(u), b_3(u)) \\ &= \begin{pmatrix} e_1 \\ e_2 + b_0(u)\cos(\frac{e_1+b_1(u)}{e_1}\pi) \\ e_3 + b_2(u)\cos(\frac{e_1+b_3(u)}{e_1}\pi) \end{pmatrix}.\end{aligned} \tag{15}$$

Here $b_0(u)$ and $b_2(u)$ are the bending magnitude functions, while $b_1(u)$ and $b_3(u)$ are the location functions where maximum bending performs.

Let $\mathbf{s}_b = (s_1, s_2, s_3)^\top$, we further express the formulation of the reference shape with twisting along the third axis as:

$$\begin{aligned}\mathbf{s}_{tw,b} &= \mathbf{T}_{tw}(\mathbf{s}_b; t_w(u)) \\ &= \mathbf{T}_{tw}(\mathbf{T}_b(\mathbf{e}, b_0(u), b_1(u), b_2(u), b_3(u)); t_w(u)) \\ &= \begin{pmatrix} s_1\cos(t_w(u)) - s_2\sin(t_w(u)) \\ s_2\cos(t_w(u)) + s_1\sin(t_w(u)) \\ s_3 \end{pmatrix},\end{aligned} \tag{16}$$

where the twisting parameter function $t_w(u)$ is defined on the third axis.

Note that in the main paper, we only train DPDM with the tapering and bending parameter functions because they offer sufficient geometric coverage for most cases in our study. In more general scenarios, such as 3D cardiac shape modeling, the twisting deformation will be included to capture a wider range of shape structures.

In this study, we only give a limited number of examples for the primitives as well as global deformation functions. However, global deformations are not restricted to only tapering, bending, and twisting. Any other deformations (e.g., shearing) that can be given as a continuous and parameterized function can be similarly integrated into our model. In addition, the type of primitives is not restricted to only superquadric-like shapes, and other primitive forms (e.g., spheres, convexes, supertoroids, etc.) can also be integrated into our unified framework, which opens up new possibilities for more downstream shape abstraction tasks.

A.3 JACOBIAN MATRIX

We employ physics-based modeling for multiscale shape representation, where the kinematics using Jacobian matrix $\mathbf{J}$ is essential for the external force transformation among multiple feature scales. Here, we provide details of the Jacobian matrix for the example used in the main paper. We also provide the Jacobian matrix for the pure definition of superquadric-like primitive without any global deformations, which is used for the ablation study in the main paper.

For the pure superquadric-like primitive, the Jacobian matrix $\mathbf{J}$ is derived as a 3×6 matrix, while for the superquadric-like primitive with tapering and bending parameter functions, $\mathbf{J}$ is a 3×11 matrix. All the non-zero entries of the two Jacobian matrices are given in Table 3, where we abbreviate all functions $a_i(\cdot), t_i(\cdot), b_i(\cdot)$ as a_i, t_i, b_i for simplicity.

For the p-th primitive, its model Jacobian matrix $\mathbf{L}^p = [\mathbf{I}^p, \mathbf{B}^p, \mathbf{R}^p\mathbf{J}^p, \mathbf{R}^p]$ is the overall Jacobian matrix for the deformable model, which includes the Jacobians for translation, rotation, global and local deformations (Metaxas, 2012). Specifically, $\mathbf{I}^p$ is the identity matrix, $\mathbf{R}^p$ is the rotation matrix, and the matrix $\mathbf{B}$ is defined using a closed formulation:

$$\mathbf{B} = -\mathbf{R}\hat{\mathbf{p}}\mathbf{G}, \tag{17}$$

where $\hat{p}$ is a 3×3 matrix of the position vector $\mathbf{p}$ in Eq. 1. Let $\mathbf{p} = (p^1, p^2, p^3)$, $\hat{\mathbf{p}}$ is expressed as:

$$\hat{\mathbf{p}} = \begin{bmatrix} 0 & -p^3 & p^2 \\ p^3 & 0 & -p^1 \\ -p^2 & p^1 & 0 \end{bmatrix}. \tag{18}$$

G is defined based on $\mathbf{q}_\theta = [s, \mathbf{v}_q]^\top$ which has unit magnitude, and is expressed as:

$$G = 2\begin{bmatrix} -v^1 & s & v^3 & -v^2 \\ -v^2 & -v^3 & s & v^1 \\ -v^3 & v^2 & -v^1 & s \end{bmatrix}, \tag{19}$$

Table 3: The non-zero entries of the Jacobian matrices for the two examples.

Pure Superquadric-like primitive	Superquadric-like primitive w/ tapering & bending
$\mathbf{J}_{11} = wa_1 C_u^{\varepsilon_1} C_v^{\varepsilon_2}$	$\mathbf{J}_{11} = (t_1 S_u^{\varepsilon_1} + 1) wa_1 C_u^{\varepsilon_1} C_v^{\varepsilon_2} + \frac{b_1 b_2 b_3}{a_0^2 w a_3} \pi \sin(r)$
$\mathbf{J}_{12} = a_0 w C_u^{\varepsilon_1} C_v^{\varepsilon_2}$	$\mathbf{J}_{21} = (t_2 S_u^{\varepsilon_1} + 1) wa_2 C_u^{\varepsilon_1} S_v^{\varepsilon_2}$
$\mathbf{J}_{15} = a_0 wa_1 \ln(\lvert\cos u\rvert) C_u^{\varepsilon_1} C_v^{\varepsilon_2}$	$\mathbf{J}_{31} = wa_3 S_u^{\varepsilon_1}$
$\mathbf{J}_{16} = a_0 wa_1 \ln(\lvert\cos v\rvert) C_u^{\varepsilon_1} C_v^{\varepsilon_2}$	$\mathbf{J}_{12} = (t_1 S_u^{\varepsilon_1} + 1) a_0 w C_u^{\varepsilon_1} C_v^{\varepsilon_2}$
$\mathbf{J}_{21} = wa_2 C_u^{\varepsilon_1} S_v^{\varepsilon_2}$	$\mathbf{J}_{23} = (t_2 S_u^{\varepsilon_1} + 1) a_0 w C_u^{\varepsilon_1} S_v^{\varepsilon_2}$
$\mathbf{J}_{23} = a_0 w C_u^{\varepsilon_1} S_v^{\varepsilon_2}$	$\mathbf{J}_{14} = \frac{b_1 b_2 b_3}{a_0 w a_3^2} \pi \sin(r)$
$\mathbf{J}_{25} = a_0 wa_2 \ln(\lvert\cos u\rvert) C_u^{\varepsilon_1} S_v^{\varepsilon_2}$	$\mathbf{J}_{34} = a_0 w S_u^{\varepsilon_1}$
$\mathbf{J}_{26} = a_0 wa_2 \ln(\lvert\sin v\rvert) C_u^{\varepsilon_1} S_v^{\varepsilon_2}$	$\mathbf{J}_{15} = t_1 \ln(\lvert\sin u\rvert) S_u^{\varepsilon_1} a_0 wa_1 C_u^{\varepsilon_1} C_v^{\varepsilon_2} + (t_1 S_u^{\varepsilon_1} + 1)$ $a_0 wa_1 \ln(\lvert\cos u\rvert) C_u^{\varepsilon_1} C_v^{\varepsilon_2} - b_1 b_3 \pi \ln(\lvert\sin u\rvert) S_u^{\varepsilon_1} \sin(r)$
$\mathbf{J}_{31} = wa_3 S_u^{\varepsilon_1}$	$\mathbf{J}_{25} = t_2 \ln(\lvert\sin u\rvert) S_u^{\varepsilon_1} a_0 wa_2 C_u^{\varepsilon_1} S_v^{\varepsilon_2} + (t_2 S_u^{\varepsilon_1} + 1)$ $a_0 wa_2 \ln(\lvert\cos u\rvert) C_u^{\varepsilon_1} S_v^{\varepsilon_2}$
$\mathbf{J}_{34} = a_0 w S_u^{\varepsilon_1}$	$\mathbf{J}_{35} = a_0 wa_3 \ln(\lvert\sin u\rvert) S_u^{\varepsilon_1}$
$\mathbf{J}_{35} = a_0 wa_3 \ln(\lvert\sin u\rvert) S_u^{\varepsilon_1}$	$\mathbf{J}_{16} = (t_1 S_u^{\varepsilon_1} + 1) a_0 wa_1 \ln(\lvert\cos v\rvert) C_u^{\varepsilon_1} C_v^{\varepsilon_2}$
	$\mathbf{J}_{26} = (t_2 S_u^{\varepsilon_1} + 1) a_0 wa_2 \ln(\lvert\sin v\rvert) C_u^{\varepsilon_1} S_v^{\varepsilon_2}$
	$\mathbf{J}_{17} = S_u^{\varepsilon_1} a_0 wa_1 C_u^{\varepsilon_1} C_v^{\varepsilon_2}$
	$\mathbf{J}_{28} = S_u^{\varepsilon_1} a_0 wa_2 C_u^{\varepsilon_1} S_v^{\varepsilon_2}$
	$\mathbf{J}_{19} = \cos(r)$
	$\mathbf{J}_{110} = -\frac{b_1 b_3}{a_0 w a_3} \pi \sin(r)$
	$\mathbf{J}_{111} = -b_1 \pi \sin(r) r$
$*S_\gamma^\varepsilon = \mathrm{sgn}(\sin\gamma)\lvert\sin\gamma\rvert^\varepsilon$	$*r = \frac{e_3 + b_2}{a_0 wa_3} \pi b_3$
$*C_\gamma^\varepsilon = \mathrm{sgn}(\cos\gamma)\lvert\cos\gamma\rvert^\varepsilon$	$w = 1$ for primitive surface formulation.

where $\mathbf{v}_q = [v^1, v^2, v^3]^\top$.

A.4 Diffeomorphic non-rigid deformation

To capture the finer local deformation beyond the coverage of global deformation, we employ a diffeomorphic mapping to estimate the local non-rigid deformation $\mathbf{q}_d$. Specifically, we first use a convolution layer to map the encoded feature l to a vector field v_0, and then map v_0 to a stationary velocity field (SVF) v using a Gaussian smoothing layer. v is defined through the ordinary differential equation (Arsigny et al., 2006; Dalca et al., 2018):

$$\frac{d\psi^{(t)}}{dt} = v(\psi^{(t)}) = v \circ \psi^{(t)}, \tag{20}$$

where $\circ$ denotes composition operator, $\psi^{(t)}$ is the path of diffeomorphic non-rigid deformation field parameterized by $t \in [0, 1]$ and $\psi^{(0)} = Id$ is an identity transformation. To obtain the final local non-rigid deformation $\mathbf{q}_d = \psi^{(1)}$ at time $t = 1$, we follow (Arsigny et al., 2006; Dalca et al., 2018) and employ an Euler integration with a scaling and squaring layer (SS) to solve Eq. 20. To be specific, starting with $\psi^{(1/2^T)} = v(k)/2^T + k$ where T is the scaling and squaring step and k is the spatial location of points on the primitive, we compute $\mathbf{q}_d = \psi^{(1)} = \psi^{(1/2)} \circ \psi^{(1/2)}$ using $\psi^{(1/2^t)} = \psi^{(1/2^{t+1})} \circ \psi^{(1/2^{t+1})}$. In general, the above diffeomorphic mapping makes the local non-rigid deformation $\mathbf{q}_d$ differentiable, invertible and topology-preserving.

B Network architecture

We use the standard ResNet18 (He et al., 2016) in the main paper to compare with the other baselines. The architecture using ResNet is given in Fig. 10. In this section, we also present a novel hybrid architecture with CNN and Transformer for the encoder $\zeta_\mu^{-1}(\cdot)$ to improve the prediction accuracy of the deformable model parameters (see Fig. 11). The proposed hybrid Trans-CNN

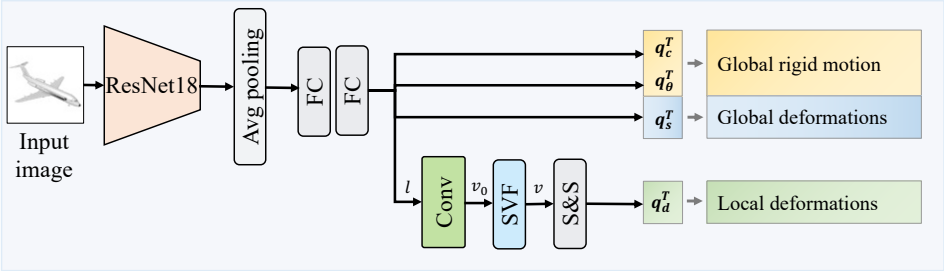


Figure 10: Network architecture of the DPDM encoder $\zeta_{\mu}^{-1}(\cdot)$ to estimate four vectorized parameters that describe the target object shape.

encoder combines the strength of both convolutional and attention mechanisms, which can leverage the inductive bias of input through convolutional layers to avoid large-scale pre-training, and also collect long-range dependencies through self-attention layers. We choose 2D residual and self-attention blocks for both the cardiac MR segmentation and 3D shape reconstruction tasks due to their same input form (i.e., images). Each residual block consists of two convolutional layers with pre-activation and batch normalization, as well as skip connections. Each self-attention block contains a Multi-Head Self-Attention (MHSA) Vaswani et al. (2017) and a convolutional layer with batch normalization and pre-activation. we apply self-attention blocks to each level of the encoder (except the first level to reduce computational cost), to draw long-range dependencies from multi-scale feature representations. Notably, efficient attention strategies Wu et al. (2021) by downsampling or reducing the size of the feature maps are not employed in our architecture, to avoid any information loss from multiple scales, especially for high-resolution feature maps. The output of the encoder is followed by two additional residual blocks to estimate four individual vectorized parameters that represent translation $\mathbf{q}_c$, rotation $\mathbf{q}_\theta$, global $\mathbf{q}_\mathbf{s}$, and local deformations $\mathbf{q}_\mathbf{d}$. Note that we employ the same diffeomorphic mapping as in Fig. 10 after the last residual block to estimate $\mathbf{q}_\mathbf{d}$. We evaluate the performance of the hybrid encoder in the Appendix C.3.1.

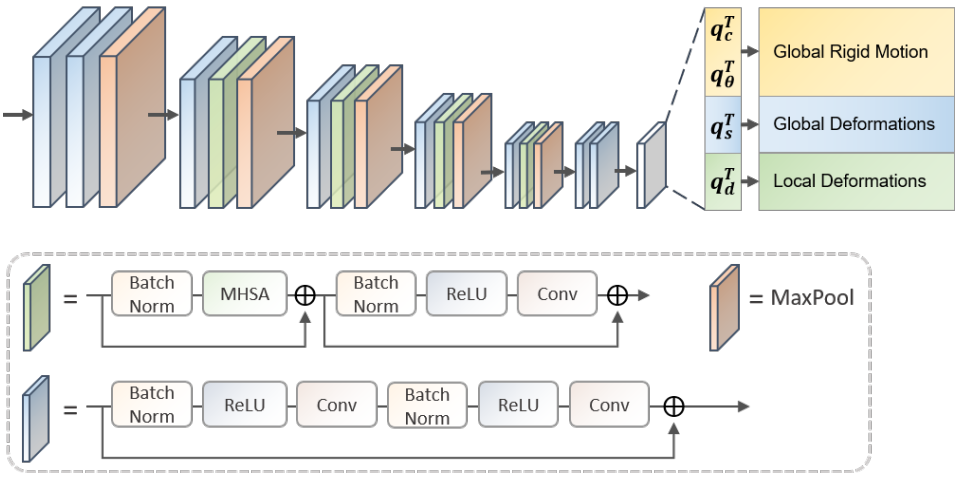


Figure 11: Network architecture of the hybrid Trans-CNN encoder $\zeta_{\mu}^{-1}(\cdot)$ to estimate four vectorized parameters that describe the target object shape. The structures of the self-attention block (green) and the residual block with pre-activation (blue).

Table 4: Additional results of Cardiac MR segmentation on *ACDC, M&Ms*, and *M&Ms-2*. We compare the proposed DPDM with more baselines which are not restricted to learning-based deformable models. The results are measured in terms of Dice Score and Hausdorff Distance (HD).

Methods	Datasets	Dice (↑)				HD (↓)			
		LV	RV	Myo	Avg	LV	RV	Myo	Avg
UNet	*ACDC*	94.92	87.10	80.63	87.55	13.91	12.23	12.98	13.04
ResUNet		95.21	88.32	82.78	88.77	13.73	11.98	12.21	12.64
TransUNet		95.73	88.86	84.53	89.71	13.28	11.32	12.05	12.22
TDAC		93.27	86.23	82.36	87.29	13.94	12.01	12.24	12.73
DPDM (ResNet18)		**95.96**	**90.76**	**85.72**	**90.81**	**10.48**	**9.49**	**10.73**	**10.23**
UNet	*M&Ms*	89.77	84.01	79.28	84.35	13.91	12.20	13.41	13.17
ResUNet		90.19	84.97	80.78	85.52	13.72	11.92	12.78	12.81
TransUNet		90.38	85.77	80.61	85.59	13.01	11.29	12.53	12.28
TDAC		89.71	84.54	81.32	85.19	13.75	11.94	12.97	12.89
DPDM		**92.31**	**86.78**	**82.37**	**87.15**	**12.19**	**10.03**	**11.98**	**11.40**
UNet	*M&Ms-2*	87.02	88.85	79.07	84.98	13.78	12.10	12.23	12.70
ResUNet		87.98	89.63	79.28	85.63	13.80	11.61	12.09	12.50
TransUNet		87.91	88.69	78.67	85.06	13.80	10.29	13.45	12.51
TDAC		86.44	87.23	77.61	83.76	11.95	10.24	12.05	11.41
DPDM (ResNet18)		**89.25**	**91.42**	**81.06**	**87.24**	**11.13**	**9.14**	**11.26**	**10.51**

C Additional experiments

C.1 Additional Results on *ShapeNet*

In this section, we provide additional qualtitative results on *ShapeNet*. We compare DPDM with Suq Paschalidou et al. (2019) which also uses superquadric-like definition for the primitive. We train DPDM with one **single** primitive, and train Suq with a maximum of 20 primitives. The results are given in Fig. 12. We observe that DPDM can capture the object shapes more accurately due to the broader geometry coverage the model provided. In addition, our model reconstructs the target shape with better details. This is because the global parameter functions allow object shape modeling with continuous and curved surfaces, and the local deformations further facilitate the fitting to the shape boundaries.

C.2 Additional Results on *ACDC, M&Ms* and *M&Ms-2*

We provide comprehensive experiments with more baselines for cardiac MR segmentation in Table 4. These baseline methods (Chen et al., 2021; Ronneberger et al., 2015) are classic learning approaches for medical image segmentation and are not restricted to learning-based deformable models as TDAC (Hatamizadeh et al., 2020).

C.3 Additional ablation studies

C.3.1 Impact of network architectures

We evaluate the impact of our hybrid Trans-CNN encoder through a comparison with ResNet18 He et al. (2016) on cardiac segmentation task (see Table 5). The result demonstrates that the hybrid Trans-CNN encoder outperforms ResNet18 (He et al., 2016) in terms of both metrics. We attribute this to the ability of the Transformer in capturing long-range dependencies for more accurate abstractions.

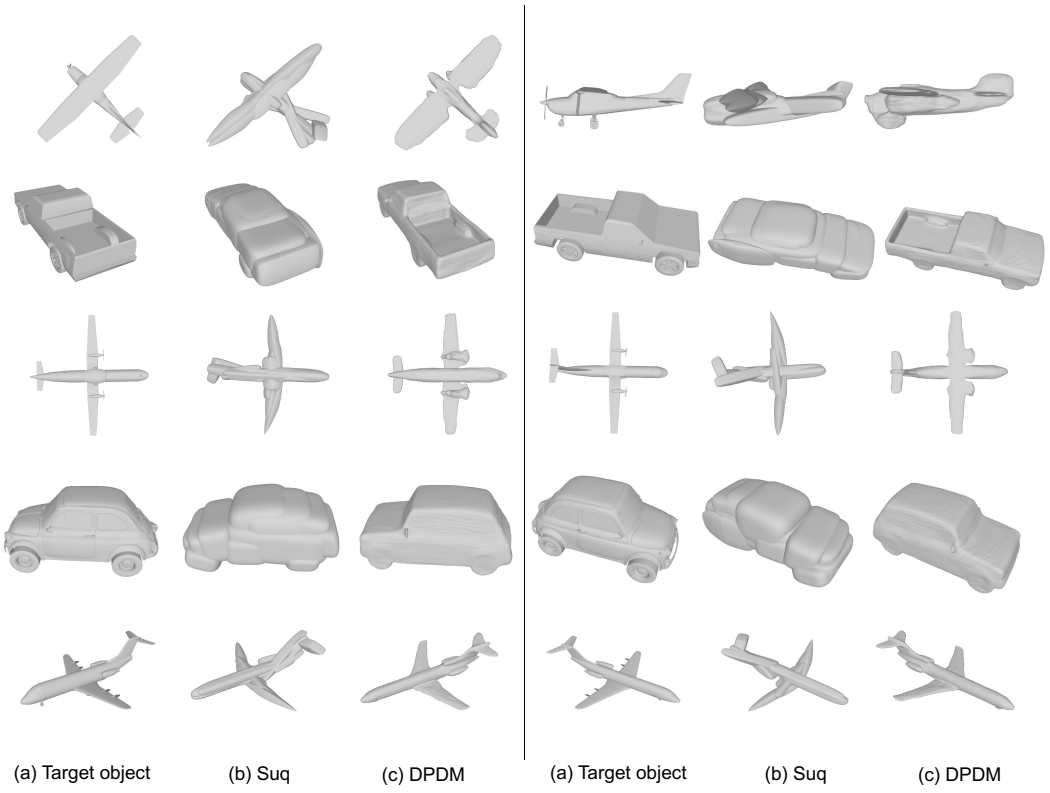


Figure 12: Additional visual results for 3D reconstruction on *ShapeNet*, including (a) target objects (ground truth), (b) Suq Paschalidou et al. (2019), and (c) DPDM.

Table 5: Ablation study on network architectures. We report Dice Score and Hausdorff Distance (HD) on the cardiac MR segmentation task.

Backbone	Datasets	Dice (↑)				HD (↓)			
		LV	RV	Myo	Avg	LV	RV	Myo	Avg
ResNet18	*ACDC*	95.96	90.76	85.72	90.81	10.48	9.49	10.73	10.23
Trans-CNN		**96.19**	**90.93**	**86.27**	**91.13**	**10.25**	**9.33**	**10.33**	**9.97**
ResNet18	*M&Ms*	92.31	86.78	82.37	87.15	12.19	10.03	11.98	11.40
Trans-CNN		**92.53**	**86.91**	**82.52**	**87.32**	**11.99**	**9.87**	**11.95**	**11.27**
ResNet18	*M&Ms-2*	89.25	91.42	81.06	87.24	11.13	9.14	11.26	10.51
Trans-CNN		**89.44**	**91.76**	**81.09**	**87.43**	**11.06**	**9.04**	**10.89**	**10.33**