

# A Predictive and Generative Design Approach for Three-Dimensional Mesh Shapes Using Target-Embedding Variational Autoencoder

**Xingang Li**

Walker Department of Mechanical Engineering,  
University of Texas at Austin,  
Austin, TX 78712  
e-mail: xingang.li@utexas.edu

**Charles Xie**

Institute for Future Intelligence,  
Natick, MA 01760  
e-mail: charles@intofuture.org

**Zhenghui Sha<sup>1</sup>**

Mem. ASME  
Walker Department of Mechanical Engineering,  
University of Texas at Austin,  
Austin, TX 78712  
e-mail: zsha@austin.utexas.edu

*In this paper, we present a predictive and generative design approach for supporting the conceptual design of product shapes in 3D meshes. We develop a target-embedding variational autoencoder (TEVAE) neural network architecture, which consists of two modules: (1) a training module with two encoders and one decoder ( $E^2D$  network) and (2) an application module performing the generative design of new 3D shapes and the prediction of a 3D shape from its silhouette. We demonstrate the utility and effectiveness of the proposed approach in the design of 3D car body and mugs. The results show that our approach can generate a large number of novel 3D shapes and successfully predict a 3D shape based on a single silhouette sketch. The resulting 3D shapes are watertight polygon meshes with high-quality surface details, which have better visualization than voxels and point clouds, and are ready for downstream engineering evaluation (e.g., drag coefficient) and prototyping (e.g., 3D printing). [DOI: 10.1115/1.4054906]*

**Keywords:** data-driven design, generative design, product shape design, deep learning of design representation

## 1 Introduction

Sketching plays an essential role in sparking creative ideas to explore emerging design concepts [1]. For example, in car design, characteristic contour lines are often used to represent silhouettes in supporting conceptual design of car body shapes [2,3], which complements many other ideation approaches, such as freehand sketches, design analogies, and prototypes. Compared to freehand sketches, silhouettes regularize the sketching process and thereby makes sketching easier and more manageable. This is particularly useful for designers who lack professional sketching skills. However, silhouettes, as a specific type of 2D sketches, are often ambiguous and lack geometric details. In later design stages, such as embodiment design, a 3D computer-aided design (CAD) model is often required to more accurately evaluate the engineering

performance of a design concept. Three-dimensional shapes can also provide better visualization and thus help designers better understand the design, inspiring them to develop new shapes and refine geometric details. Therefore, the question is: can we build a system to predict and automatically generate 3D shapes just based on silhouettes?

Such a system will yield several benefits. First, it automates the 2D-to-3D reconstruction process, thereby saving labor and time. Designers can allocate more resources for better design iteration and ideation. Second, all silhouettes created during the conceptual design stage can be evaluated against the desired engineering performance in 3D form. So, designs that would have better performance will not be ruled out too early when performance-driven decisions (i.e., rational decisions) are not yet obtained. Third, ordinary people would not be discouraged to show their design ideas merely due to the lack of CAD experience or sketching skills. This may have significant educational implications for training novice designers and facilitate the democratization of design innovation. Lastly, enterprises may use this system to enable user interface soliciting consumer preferences for design customization.

However, automatically reconstructing 3D shapes directly from 2D sketches is a challenge because it is an ill-defined problem due to insufficient and imperfect information from simple strokes [4]. To tackle this challenge, inspired by the target-embedding autoencoder (TEA) network [5,6], we propose a novel target-embedding variational autoencoder (TEVAE) (see Fig. 1(a)). The TEVAE architecture consists of two modules: (1) a training module with an  $E^2D$  network that has two encoders and one decoder. (2) An application module performing two functions: *generative design* (GD) function, such as shape interpolation and random generation of new 3D shapes, and *predictive design* function (i.e., 3D shape prediction from silhouette sketches). The integration of generative and predictive functions is beneficial in that it makes the structure of the neural network compact, thus saving training costs. To demonstrate the utility and generalizability of the proposed approach, we apply it to two case studies in the design of 3D car body and mugs.

The contributions of this paper are summarized as follows:

- To the best of our knowledge, this is the first attempt of developing a system integrating both predictive and generative functions of 3D mesh shapes from silhouettes. The TEA has a classic autoencoder that can perform pseudo-generative tasks, but essentially is not a generative model. Our TEVAE applies variational autoencoders (VAEs) and becomes a true generative model. It can also learn a continuous and smooth latent representation of the data.
- Predicting 3D shapes from silhouettes is more challenging because a silhouette sketch provides less information (e.g., depth and normal maps) than traditional freehand sketches with inner contour lines. To that end, we introduce an intermediate step (e.g., extrusion or rotation) to first convert the silhouette to a 3D primitive shape. This transforms the original 2D-to-3D problem to a 3D-to-3D problem, which promotes a stable training process and the generation of reliable and viable 3D shapes.
- Building upon a graph convolutional mesh VAE [7], our approach can directly output high-quality 3D mesh shapes that are more storage-efficient for high-resolution 3D structure, compared to point clouds [8] and voxels [9]. Three-dimensional meshes also facilitate engineering analyses because they are compatible with existing computer-aided engineering (CAE) software.<sup>2</sup>

<sup>2</sup>“Meshes” are used for 3D representation here. In CAE software, there is a concept called “meshing.” Meshing is a process that breaks down the continuous geometric space of an object into a discrete number of shape elements. All 3D representations including meshes and native CAD data formats (e.g., initial graphics exchange specification (IGES), Drawing (DWG), and Standard Tessellation Language (STL) that can be directly input to CAE software have to go through the meshing process for analysis.

<sup>1</sup>Corresponding author.

Contributed by the Design Education Committee of ASME for publication in the JOURNAL OF MECHANICAL DESIGN. Manuscript received October 4, 2021; final manuscript received June 18, 2022; published online August 8, 2022. Assoc. Editor: Xiaoping Qian.

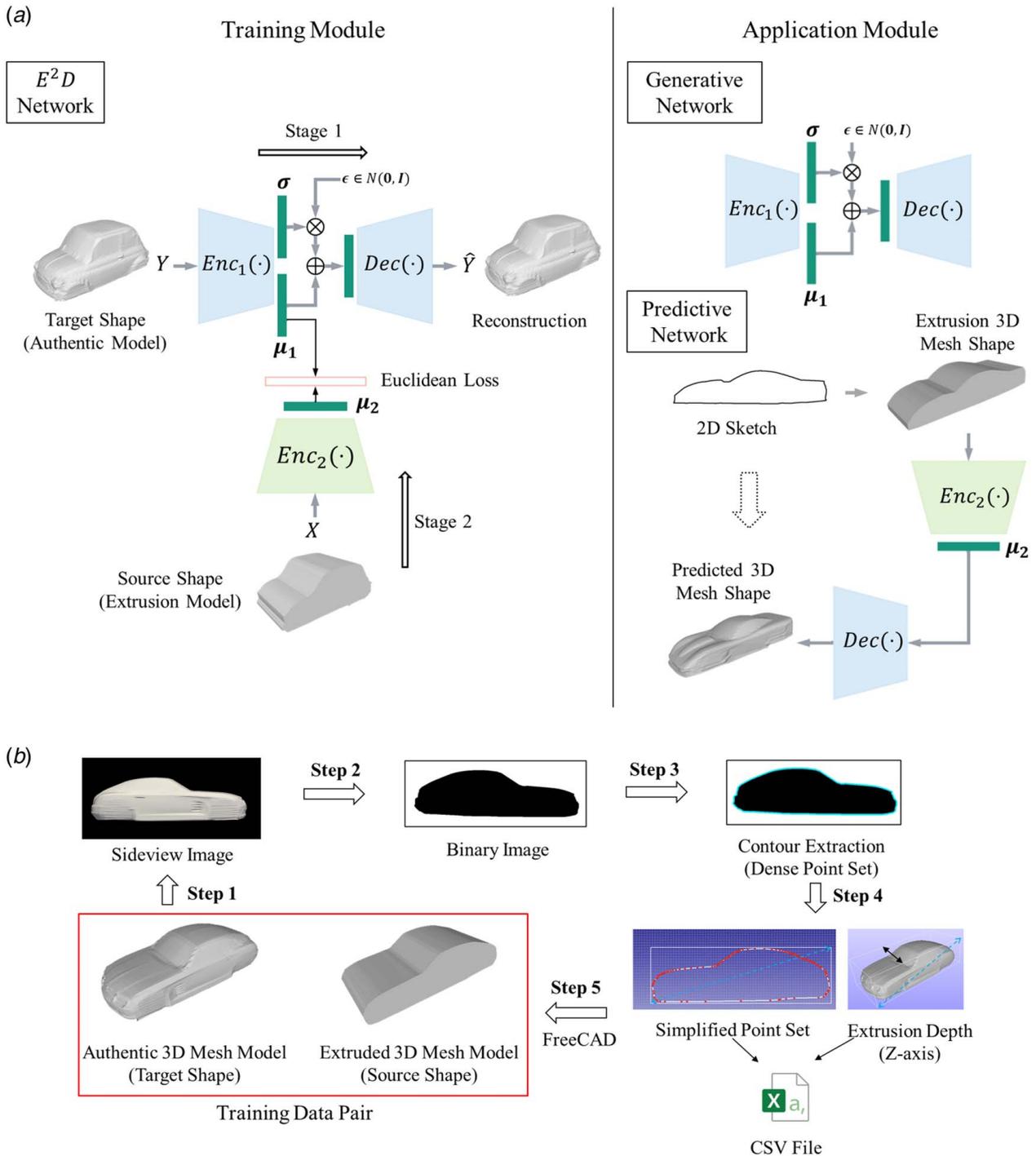


Fig. 1 (a) The proposed approach using TEVAE and (b) the preparation of data pairs

(d) A data automation program is developed for training data pairs of 3D shapes that can be used in any TEA-like neural networks for supervised learning problems.

## 2 Literature Review

In this section, we review the existing research that is most relevant to our work.

**2.1 Learning-Based Sketch-to-3D Generation Methods.** Point clouds and voxels have been widely used as 3D representations for sketch-to-3D generation [8–10]. These methods need to postprocess the resulting 3D shapes to meshes for better visualization, which still suffer from low surface quality. There are

studies attempting to directly produce high-quality mesh shapes. For example, concurrent with the development of our approach, Guillard et al. [11] propose a pipeline to reconstruct and edit 3D shapes from 2D sketches. They train an encoder/decoder architecture to regress surface meshes from freehand sketches. The method applies a differentiable rendering technique to iteratively refine the resulting 3D shapes. Similarly, Xiang et al. [12] integrate a differentiable rendering approach to an end-to-end learning framework for predicting 3D mesh shapes from line drawings.

All methods above are promising and have inspired us to explore a more challenging task, i.e., to predict a 3D shape from a simple silhouette sketch. Our approach is similar to Refs. [10,11,13,14]

in that we only need one single sketch as input, but we create a new neural network architecture that can predict a 3D shape from a single silhouette and simultaneously generate novel 3D shapes. The direct output shapes are 3D meshes with high-quality surface details, thus requiring no postprocessing.

**2.2 Learning-Based Generative Design Methods.** Learning-based GD methods have been primarily developed based on two techniques, generative adversarial networks (GANs) [15] and VAEs [16]. There are several approaches for 2D designs [17–19], but they are not appropriate for design applications that require 3D models. In 3D applications, Shu et al. [20] present a method that combines GAN and the physics-based virtual environment introduced in Ref. [18] to generate high-performance 3D aircraft models. Zhang et al. [21] propose a method using a VAE, a physics-based simulator, and a functional design optimizer to synthesize 3D aircrafts with prescribed engineering performance. Building upon Ref. [17], Yoo et al. [22] develop a deep learning-based CAD/CAE framework that can automatically generate 3D car wheels from 2D images. Gunpinar et al. [3] apply a spatial simulated annealing algorithm to generate various silhouettes of cars, which are then extruded to 3D car models. Those models can be further refined by sweeping a predefined cross-sectional sketch. However, simply extrusion does not guarantee satisfactory outcomes, and the resulting 3D car models look unreal.

**2.3 Target-Embedding Representation Learning.** Girdhar et al. propose a TL-embedding network [6] that is composed of a T-network for training and an L-network for testing (the architectures of the T-network and L-network are in the shape of alphabet T and L, respectively). The T-network contains an autoencoder (encoder–decoder) network and a convolutional neural network (CNN). After training, the L-network can be used to predict 3D shapes in voxel from images. Similarly, Mostajabi et al. [23] use an autoencoder and a CNN to perform the semantic segmentation task of images. Dalca et al. [24] apply a similar network structure as Refs. [6,23], consisting of a prior generative model to generate paired data (biomedical images and anatomical regions) to solve the scarcity of labeled image data for anatomical segmentation tasks. Jarrett and van der Schaar [5] categorize these studies as supervised representation learning methods. They observe that when the dimension of the target data space is higher or similar to the feature data space, a TEA can be more effective than a feature-embedding autoencoder. The authors verify that the TEA structure will guarantee the learning stability using a mathematical proof of a simple linear TEA and showing the empirical results from a complex non-linear TEA. Inspired by those existing works, we construct the TEVAE architecture.

### 3 Approach

The proposed TEVAE architecture, shown in Fig. 1(a), consists of two modules: a training module and an application module.

**3.1 The  $E^2D$  Network and the Two-Stage Training.** The key component of the training module is the  $E^2D$  network that consists of two encoders and one decoder, which is constructed by concatenating an encoder (labeled as  $\text{Enc}_2(\cdot)$ ) to a mesh VAE (an encoder–decoder network, labeled as  $\text{Enc}_1(\cdot)$  and  $\text{Dec}(\cdot)$ ) [7]. The  $\text{Enc}_1(\cdot)$  maps target shapes ( $S_s$ , i.e., the original authentic 3D mesh shapes) to a low-dimensional latent space, and the  $\text{Dec}(\cdot)$  maps latent vectors from that latent space to 3D mesh shapes. With the same network structure as  $\text{Enc}_1(\cdot)$ ,  $\text{Enc}_2(\cdot)$  takes source shapes ( $S_s$ , the 3D mesh shapes extruded from silhouette sketches of the target shapes) as the input and maps them to the same dimensional latent space as the mesh VAE.

We adopt the same loss function developed in Ref. [7] to train  $\text{Enc}_1(\cdot)$  and  $\text{Dec}(\cdot)$ . For  $\text{Enc}_2(\cdot)$ , we create a new loss function  $L_2$  as follows:

$$L_2 = \alpha D_{\text{Regress}} + D_R \quad (1)$$

where  $\alpha$  is the weight for the regression loss and

$$D_{\text{Regress}} = \frac{1}{2M} \sum_{i=1}^M \|\mu_2^i - \mu_1^i\|_2^2 \quad (2)$$

denotes the Euclidean loss, where  $\mu_2^i$  is the output of  $\text{Enc}_2(\cdot)$  using  $X^i$  as input and  $\mu_1^i$  is the mean vector obtained from the latent space of the mesh VAE using the input of  $Y^i$ .  $D_R$  is the regularization loss applied to improve the generalization ability of the  $\text{Enc}_2(\cdot)$ .

We apply a two-stage training strategy [23] to jointly train the mesh VAE and  $\text{Enc}_2(\cdot)$  from scratch. In stage 1, the mesh VAE is trained independently. In stage 2, we fix all the learning parameters of the mesh VAE and train  $\text{Enc}_2(\cdot)$  by minimizing  $L_2$ .

**3.2 The Predictive Network and Generative Network.** After the  $E^2D$  network is trained, we connect  $\text{Enc}_2(\cdot)$  to  $\text{Dec}(\cdot)$  to form the predictive network. It can take a 3D extrusion mesh shape as input and output a 3D mesh shape that is similar to the input shape but has finer geometric details, making it authentic and aesthetic. We use the trained mesh VAE as the generative network, which can perform generative design tasks, including shape reconstruction, interpolation, and random generation.

**3.3 Preparation of Data Pairs.** Data pairs  $\{S_s^i, S_t^i\}_{i=1}^N$  are needed to train the  $E^2D$  network. Figure 1(b) shows the process of obtaining one training data pair using a car body as an example. From the sideview image of an authentic 3D car model, we extract its contour points, from which we can obtain an extrusion model using the FREECAD PYTHON application programming interface (API). We develop a set of PYTHON scripts that fully automate the whole process, which are made open-source for the community.<sup>3</sup> We process  $N=1240$  car models obtained from Ref. [25] and  $N=203$  mug models from Ref. [26]. For car models, we keep only car bodies by removing all the other parts, such as mirrors, wheels, and spoilers.

**3.4 Shape Preprocessing and Feature Representation.** The  $E^2D$  network requires input mesh shapes in both the source shape set ( $\{S_s^i\}_{i=1}^N$ ) and the target shape set ( $\{S_t^i\}_{i=1}^N$ ) to have the same topology (i.e., the same number of vertices and the same mesh connectivity). However, the mesh topologies between the two datasets can be different. For simplicity, we use a uniform topology for both datasets and the non-rigid registration method [27] is applied to meet this requirement. Non-rigid registration is a widely used technique in the computer graphics field to map one point set (e.g., point cloud, mesh) to another. A uniform unit cube mesh with 19.2k triangles (9602 vertices) is used to register all mesh shapes. This makes all shapes have the same topology as the cube mesh, but remain the same as their original shapes and geometric details.

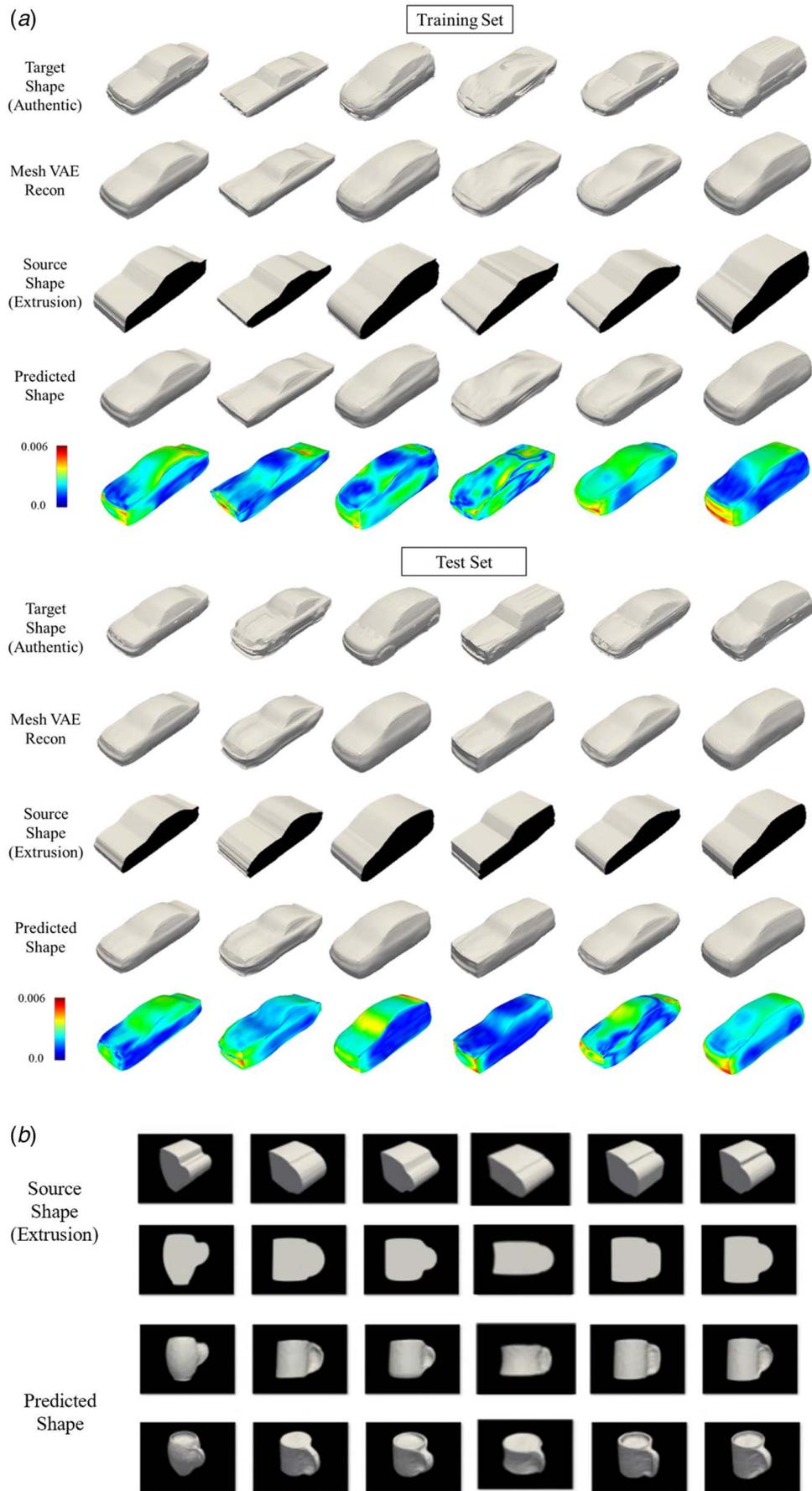
The as-consistent-as-possible method [28] is applied to extract features of a 3D shape to input to the  $E^2D$  network. We deform the aforementioned uniform cube mesh to a target 3D mesh shape by multiplying deformation matrices, from which nine unique numbers can be extracted for each vertex of the mesh shape. Thus, a shape with  $v$  vertices can be represented by a feature matrix  $M_f \in \mathbb{R}^{v \times 9}$ , where  $v=9602$  in our implementation. We can get the feature representations of the source shape dataset  $X = \{X^k\}_{k=1}^N$  and the target shape dataset  $Y = \{Y^l\}_{l=1}^N$ , where  $N=1240$  for the car models and  $N=203$  for the mug models, and  $\{X^i, Y^i\}$  forms the input feature of one data pair.

More details of the approach and the training of the  $E^2D$  network are provided in the [Supplementary Materials](#) on the ASME Digital Collection.

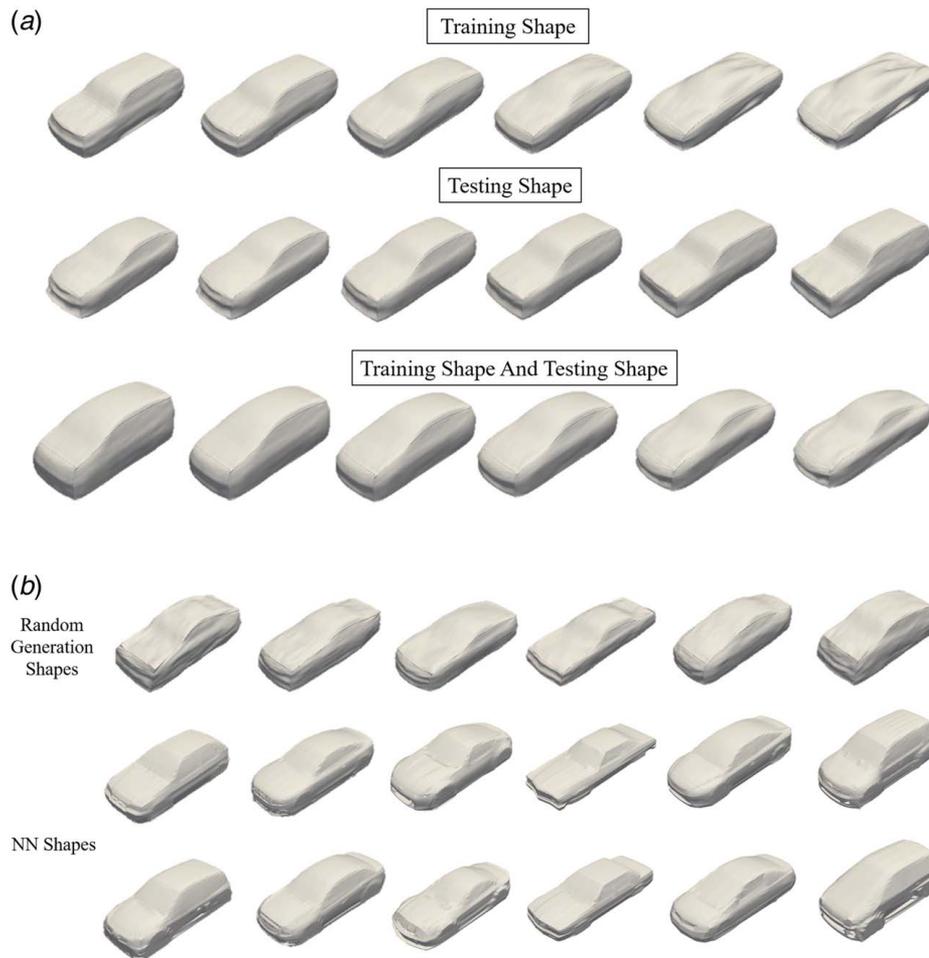
## 4 Case Studies and Results

**4.1 Implementation of the Two-Stage Training.** For training the mesh VAE in stage 1, the input target shape dataset  $Y = \{Y^l\}_{l=1}^N$

<sup>3</sup><https://github.com/Xingang1990/TEVAE>



**Fig. 2** (a) Results of car bodies: predicted shapes (in the fourth row) and reconstructed shapes (in the second row) and (b) the prediction results of mugs



**Fig. 3** (a) The results of shape interpolation for three cases and (b) the results of random generation shapes along with two NN shapes

is randomly divided into the training set (80%) and the test set (20%). For training the  $\text{Enc}_2(\cdot)$  network in stage 2, we also do an 80–20 split of the source shape dataset  $X = \{X^k\}_{k=1}^N$ , and meanwhile use the data pair to make sure the  $i$ th target shape ( $S_i^t$ ) corresponds to the  $i$ th source shape ( $S_s^i$ ) in both the training and test sets.

**4.2 The Predictive Network.** The predictive network aims to predict a 3D shape from an input silhouette sketch. We conducted experiments on the prediction of the training set and the test set. The results of the car models are shown in Fig. 2(a). For the result of the training set, the first row shows the target shapes, and the following rows are their corresponding reconstruction shapes from the mesh VAE, extrusion shapes (with silhouettes marked in dark), and the predicted shapes, respectively. The results indicate that, given an input extrusion shape from the corresponding silhouette sketch, the predictive network is capable of predicting an authentic 3D shape, as illustrated in the fourth row. It should be noted that even though we are targeting shapes (ground truth) in the first row, the best results that can be achieved from the predictive network are the reconstruction shapes in the second row. The reconstruction shapes and the corresponding predicted shapes look identical in terms of visual appearance but are different in geometric details. To show the difference, we compute the Hausdorff distance between those shapes and visualize the distance values in the fifth row. Similar results are also observed for the shapes in the test set, which indicates a good generalization of the network because the test set shapes are unseen data for the network. This is particularly important in real-world applications,

where user input often does not resemble existing shapes in a training dataset.

The prediction results of the mug models are shown in Fig. 2(b). The first two rows are the source shapes that are obtained from extruding the silhouettes, while the last two rows are the corresponding predicted shapes. Mugs are generally non-extrudable from sideview silhouettes, so the extrusion shapes look more like toast instead of mugs. However, our approach can still predict authentic mug shapes. Please note that, besides extruding, other 3D modeling techniques, such as revolving and sweeping, can also be used to obtain 3D primitive shapes. Extruding is adopted in this study for ease of implementation. In addition, it provides us with basic geometric features for 3D shape prediction.

**4.3 The Generative Network.** For the generative network, different generative operations, such as shape reconstruction, interpolation, and random generation, can be performed. The reconstructed 3D shapes are already shown in the second row for both the training set and the testing set in Fig. 2(a). For shape interpolation, new 3D shapes are synthesized by linearly interpolating two target 3D shapes through their encoded latent vectors. We demonstrate the results of shape interpolation in three cases using the case study of car models (see Fig. 3(a)): (1) interpolation between two training shapes, (2) between two test shapes, and (3) between a training shape and a test shape. In each case, the first and the last columns are the shapes to be interpolated, and the in-between columns are linearly interpolated shapes. It can be observed that there is a gradual transition of the shape geometry between the two target shapes.

For random shape generation, latent vectors are randomly sampled from the latent space of the mesh VAE, and decoded by the trained  $\text{Dec}(\cdot)$  to 3D mesh shapes. Figure 3(b) shows that the generative network can generate novel car models (in the first row) that are not seen in the original dataset. This is validated by finding their nearest neighbors (NNs) (the second and third rows) in the original dataset based on the Hausdorff distance. A quick visual comparison between the randomly generated car models and their NNs tells the differences, and they are indeed new shapes.

## 5 Conclusion

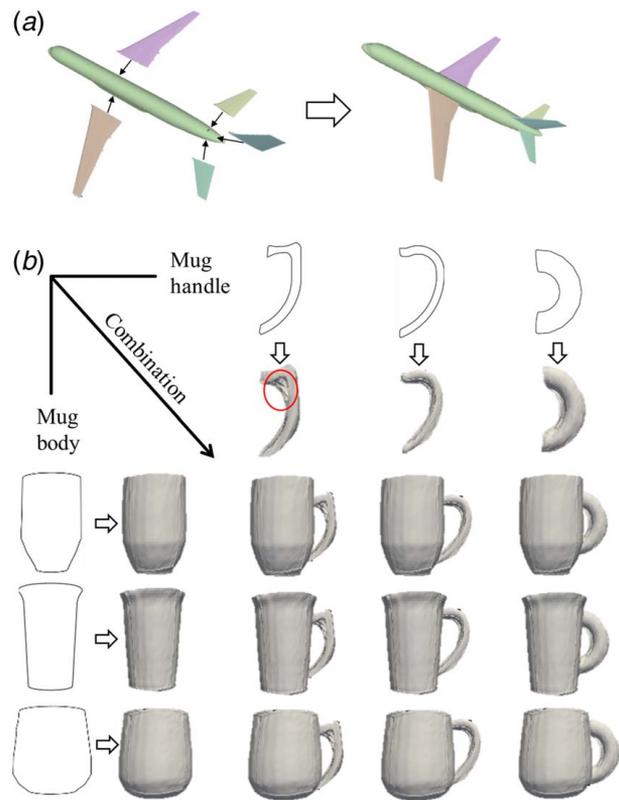
To tackle the challenge of predicting a 3D shape from a silhouette sketch, we present a novel TEVAE network that enables a 2D-to-3D design approach. Our approach can effectively predict a 3D shape from a silhouette sketch. The predicted 3D shape is consistent with the input sketch and is authentic with rich geometric details. Such a design transformation could greatly shorten the iteration between the design ideation and CAD modeling. The approach can also generate novel 3D shapes and thus could better inspire designers for their creative work. The resulting 3D shapes are represented in meshes, which are ready for downstream engineering analyses, evaluation, and prototype (e.g., 3D printing).

Quantity yields quality, and this can be achieved by broadening the initial pool of concept ideas [29]. We believe that the presented approach can help designers explore the design space more efficiently and stimulate creative design ideas in early design stages. From the methodology point of view, this new generative design approach is general enough to be applied in many applications where 3D shape modeling and rendering are necessary. As long as the sketch can provide a major perspective view of an object, like the frontview of a human body and the sideview of a bottle, the corresponding authentic 3D shape can be predicted and novel shape concepts can be generated. In addition, our approach is friendly to ordinary people who have few professional sketching skills, since it only requires a simple silhouette of an object as input.

There are a few limitations in the current study that the authors would like to share. First, the current model only handles genus-zero shapes and ignores any through holes (e.g., the hole between the body and the handle of a mug in Fig. 2(b)) in the original shape due to the non-rigid registration [27]. However, many design artifacts are usually non-genus-zero (e.g., a mug with a through hole between the body and the handle) or have more complex geometry consisting of many components, e.g., a plane model can have a body, two wings, and three tails, etc., as shown in Fig. 4(a).

To address this limitation, a part-aware method [26,30] may be a potential solution. We perform a quick experiment using a part-aware mug design problem (see Fig. 4(b)). In this particular application, users can first draw an outline sketch for an individual component (e.g., a mug body or a handle). Then, the corresponding 3D mesh shape can be predicted and new shapes can be generated. Lastly, the resulting individual components can be combined to a holistic structure allowing non-genus-zero topology. However, the part-aware strategy could not work for parts that are non-genus-zero and unable to be further decomposed into genus-zero components, e.g., a chair back with hollow-out structures and holes. In this experiment, we applied a cube mesh template to register mug models using non-rigid registration [27] as introduced previously. However, we observed that artifacts with a large curvature could not be perfectly registered (e.g., the mug handle in Fig. 4(b) highlighted by a circle). This issue can be alleviated using different templates of 3D primitives, e.g., a sphere or a cylinder.

In addition to the part-aware method, other methods based on new 3D representations could also be applied to address the first limitation. For example, primitive-based methods can use a set of primitive surfaces to represent a 3D shape [31–34]. Implicit 3D representation (e.g., signed distance fields [35,36]) can characterize 3D surfaces implicitly, and the resulting 3D geometries can be converted to mesh representation. These methods can capture the



**Fig. 4 Complex geometries (e.g., a toy plane with a body, two wings, and three tails) or non-genus-zero shapes (e.g., a mug with a body and a handle) can be partitioned to several genus-zero shapes. Then, the proposed approach can be applied to each component for shape exploration and synthesis.**

topology changes of 3D shapes without using a template mesh for data registration, thus deserving our future exploration.

Second, the constructed 3D shapes are consistent with the designer's sketch in terms of sideview, but they might not be the same as what the designer has in mind. To address this limitation, we plan to integrate interactive modeling techniques to build a graphical user interface for users to adjust the generated 3D shapes according to their preferences.

Third, the generative network performs well in shape reconstruction and shape interpolation, but the success rate of random shape generation is lower than one-third due to the sparsity of the training data. Therefore, the random shape generation function is not fully reliable in practice for now. This problem could be solved by obtaining more 3D shape data using data augmentation methods, such as the one presented in the study of Nozawa et al. [13], to improve the diversity and quality of the training dataset. These limitations motivate us to further improve the current model in the future.

## Acknowledgment

We thank Dr. Miaoqing Huang for granting us access to the computer with GPUs to train the  $E^2D$  network.

## Funding Data

- This study is supported by the U.S. National Science Foundation (NSF) Division of Undergraduate Education (DUE) through Grant No. #2207408.

## Conflict of Interest

There are no conflicts of interest.

## Data Availability Statement

The datasets generated and supporting the findings of this article are obtainable from the corresponding author upon reasonable request.

## References

- Pratt, M. J., Anderson, B. D., and Ranger, T., 2005, "Towards the Standardized Exchange of Parameterized Feature-Based CAD Models," *Comput Aided Des.*, **37**(12), pp. 1251–1265.
- Reid, T. N., Gonzalez, R. D., and Papalambros, P. Y., 2010, "Quantification of Perceived Environmental Friendliness for Vehicle Silhouette Design," *ASME J. Mech. Des.*, **132**(10), p. 101010.
- Gunpinar, E., Ovrur, S. E., and Gunpinar, S., 2019, "A User-Centered Side Silhouette Generation System for Sedan Cars Based on Shape Templates," *Optim. Eng.*, **20**(3), pp. 683–723.
- Schmidt, R., Khan, A., Kurtenbach, G., and Singh, K., 2009, "On Expert Performance in 3d Curve-Drawing Tasks," Proceedings of the 6th Euro Graphics Symposium on Sketch-Based Interfaces and Modeling, New Orleans, LA, Aug. 1–2, pp. 133–140.
- Jarrett, D., and van der Schaar, M., 2019, "Target-Embedding Autoencoders for Supervised Representation Learning," International Conference on Learning Representations, New Orleans, LA, May 6–9.
- Girdhar, R., Fouhey, D. F., Rodriguez, M., and Gupta, A., 2016, "Learning a Predictable and Generative Vector Representation for Objects," European Conference on Computer Vision, Amsterdam, The Netherlands, Oct. 11–14, Springer, pp. 484–499.
- Yuan, Y.-J., Lai, Y.-K., Yang, J., Duan, Q., Fu, H., and Gao, L., 2020, "Mesh Variational Autoencoders With Edge Contraction Pooling," Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, Virtual, June 16–18, pp. 274–275.
- Lun, Z., Gadelha, M., Kalogerakis, E., Maji, S., and Wang, R., 2017, "3d Shape Reconstruction From Sketches Via Multi-View Convolutional Networks," 2017 International Conference on 3D Vision (3DV), Qingdao, China, Oct. 10–12, IEEE, pp. 67–77.
- Jin, A., Fu, Q., and Deng, Z., 2020, "Contour-Based 3d Modeling Through Joint Embedding of Shapes and Contours," Symposium on Interactive 3D Graphics and Games, San Francisco, CA, Sept. 15–17, pp. 1–10.
- Nozawa, N., Shum, H. P., Feng, Q., Ho, E. S., and Morishima, S., 2021, "3D Car Shape Reconstruction From a Contour Sketch Using GAN and Lazy Learning," *Vis. Comput.*, **38**(4), pp. 1317–1330.
- Guillard, B., Remelli, E., Yvernay, P., and Fua, P., 2021, "Sketch2mesh: Reconstructing and Editing 3D Shapes From Sketches," Proceedings of the IEEE/CVF International Conference on Computer Vision, Virtual, June 19–25.
- Xiang, N., Wang, R., Jiang, T., Wang, L., Li, Y., Yang, X., and Zhang, J., 2020, "Sketch-Based Modeling With a Differentiable Renderer," *Comput. Animat. Virtual Worlds*, **31**(4–5), p. e1939.
- Nozawa, N., Shum, H. P., Ho, E. S., and Morishima, S., 2020, "Single Sketch Image Based 3d Car Shape Reconstruction With Deep Learning and Lazy Learning," Proceedings of the 15th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications, Valetta, Malta, Feb. 27–29, pp. 179–190.
- Han, X., Gao, C., and Yu, Y., 2017, "Deepsketch2face: A Deep Learning Based Sketching System for 3d Face and Caricature Modeling," *ACM Trans. Graph.*, **36**(4), pp. 1–12.
- Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y., 2014, "Generative Adversarial Nets," *Advances in Neural Information Processing Systems*, Palais des Congrès de Montréal, Montréal, Canada, Dec. 8–13, pp. 2672–2680.
- Kingma, D. P., and Welling, M., 2013, "Auto-Encoding Variational Bayes," preprint arXiv:1312.6114.
- Oh, S., Jung, Y., Kim, S., Lee, I., and Kang, N., 2019, "Deep Generative Design: Integration of Topology Optimization and Generative Models," *ASME J. Mech. Des.*, **141**(11), p. 111405.
- Dering, M., Cunningham, J., Desai, R., Yukish, M. A., Simpson, T. W., and Tucker, C. S., 2018, "A Physics-Based Virtual Environment for Enhancing the Quality of Deep Generative Designs," ASME 2018 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Quebec City, Quebec, Canada, Aug. 26–29.
- Fujita, K., Minowa, K., Nomaguchi, Y., Yamasaki, S., and Yaji, K., 2021, "Design Concept Generation With Variational Deep Embedding Over Comprehensive Optimization," International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Virtual, Aug. 17–19, p. V03BT03A038.
- Shu, D., Cunningham, J., Stump, G., Miller, S. W., Yukish, M. A., Simpson, T. W., and Tucker, C. S., 2020, "3D Design Using Generative Adversarial Networks and Physics-Based Validation," *ASME J. Mech. Des.*, **142**(7), p. 071701.
- Zhang, W., Yang, Z., Jiang, H., Nigam, S., Yamakawa, S., Furuhata, T., Shimada, K., and Kara, L. B., 2019, "3d Shape Synthesis for Conceptual Design and Optimization Using Variational Autoencoders," International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Anaheim, CA, Aug. 18–21, p. V02AT03A017.
- Yoo, S., Lee, S., Kim, S., Hwang, K. H., Park, J. H., and Kang, N., 2021, "Integrating Deep Learning Into CAD/CAE System: Case Study on Road Wheel Design Automation," *Struct. Multidisc. Opt.*, **64**(4), pp. 2725–2747.
- Mostajabi, M., Maire, M., and Shakhnarovich, G., 2018, "Regularizing Deep Networks by Modeling and Predicting Label Structure," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, June 19–21, pp. 5629–5638.
- Dalca, A. V., Gutttag, J., and Sabuncu, M. R., 2018, "Anatomical Priors in Convolutional Networks for Unsupervised Biomedical Segmentation," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, June 19–21, pp. 9290–9299.
- Umetani, N., 2017, "Exploring Generative 3d Shapes Using Autoencoder Networks," SIGGRAPH Asia 2017 Technical Briefs, Bangkok, Thailand, Nov. 27–30, pp. 1–4.
- Gao, L., Yang, J., Wu, T., Yuan, Y.-J., Fu, H., Lai, Y. K., and Zhang, H., 2019, "Sdm-net: Deep Generative Network for Structured Deformable Mesh," *ACM Trans. Graph.*, **38**(6), pp. 1–15.
- Zollhöfer, M., Nießner, M., Izadi, S., Rehm, C., Zach, C., Fisher, M., Wu, C., Fitzgibbon, A., Loop, C., and Theobalt, C., 2014, "Real-Time Non-Rigid Reconstruction Using an RGB-D Camera," *ACM Trans. Graph.*, **33**(4), pp. 1–12.
- Gao, L., Lai, Y.-K., Yang, J., Ling-Xiao, Z., Xia, S., and Kobbelt, L., 2019, "Sparse Data Driven Mesh Deformation," *IEEE Trans. Visual. Comput. Graph.*, **27**(3), pp. 2085–2100.
- Yang, M. C., 2003, "Concept Generation and Sketching: Correlations With Design Outcome," International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Chicago, IL, Sept. 2–6, pp. 829–834.
- Li, X., Xie, C., and Sha, Z., 2021, "Part-Aware Product Design Agent Using Deep Generative Network and Local Linear Embedding," Proceedings of the 54th Hawaii International Conference on System Sciences, Kauai, HI, Jan. 5–8, p. 5250.
- Paschalidou, D., Katharopoulos, A., Geiger, A., and Fidler, S., 2021, "Neural Parts: Learning Expressive 3d Shape Abstractions With Invertible Neural Networks," Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Virtual, June 19–25, pp. 3204–3215.
- Vasu, S., Talabot, N., Lukoianov, A., Baque, P., Donier, J., and Fua, P., 2021, "Hybridsdf: Combining Free Form Shapes and Geometric Primitives for Effective Shape Manipulation," preprint arXiv:2109.10767.
- Jones, R. K., Hanocka, R., and Ritchie, D., 2021, "The Neutrally-Guided Shape Parser: A Monte Carlo Method for Hierarchical Labeling of Over-Segmented 3D Shapes," preprint arXiv:2106.12026.
- Groueix, T., Fisher, M., Kim, V. G., Russell, B. C., and Aubry, M., 2018, "A Papier-Mâché Approach to Learning 3d Surface Generation," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, June 19–21, pp. 216–224.
- Chen, Z., and Zhang, H., 2019, "Learning Implicit Fields for Generative Shape Modeling," Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, June 16–20, pp. 5939–5948.
- Park, J. J., Florence, P., Straub, J., Newcombe, R., and Lovegrove, S., 2019, "Deepsdf: Learning Continuous Signed Distance Functions for Shape Representation," Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, June 16–20, pp. 165–174.