Supplementary Material: Cross-Modal 3D Shape Generation and Manipulation

Zezhou Cheng[1], Menglei Chai[2], Jian Ren[2], Hsin-Ying Lee[2], Kyle Olszewski[2], Zeng Huang[2], Subhransu Maji[1], and Sergey Tulyakov[2]

[1] University of Massachusetts, Amherst
[2] Snap Inc.

The following describes the content in each section in the supplementary material.

- § 1 describes additional implementation details.
- § 2 presents ablation study of the proposed model.
- § 3 illustrates more details of our baselines including encoder-decoder networks [4] and EditNeRF [8].
- § 4 details our experimental settings (*e.g.*, datasets, evaluations).
- § 5 provides detailed analysis on the limitations of the proposed model.
- § 6 provides more color editing results with diverse color scribbles.

1 Implementation Details

The implementation details of 3D shape and color networks are included in the main text. Here we provide additional implementation details.

- **Joint latent space.** The shape and color latent codes are both of dimension 128 throughout our experiments. We observe that lower-dimensional latent codes (*e.g.*, 32) lead to worse shape reconstruction.
- **2D sketch and renderings.** The image resolution of all 2D modalities is set to 128×128. We use the generator architecture from DCGAN [12] for all 2D modalities.
- **Few-shot shape generation.** We use the discriminator from DCGAN [12]. The mapping function $h_w(\boldsymbol{z})$ in the MineGAN framework [14] is a two-layer MLP with batch normalization [7] and a ReLU activation function.
- **Latent optimization.** In the task of shape and appearance manipulation, we conduct the latent optimization for 5 steps starting from a known initial latent code that corresponds to the initial 2D and 3D instances. The hyperparameter γ and β in Eqn.11 is 0.02 and 0.5 respectively by default. For the single-view shape generation tasks, we run multiple trials of latent optimization from different randomly sampled latent codes. The optimized code with minimal reconstruction loss is used as the final result. We observe that such multi-trial optimization significantly stabilizes the performance of 3D reconstruction (see Sec. 2 for more details).

2 Ablation study

The latent optimization is crucial to the performance of our shape reconstruction and manipulation tasks. In this section, we provide ablation studies on the regu-

larization loss (Eqn. 10 in the main text) and the multi-trial latent optimization method (as described in Sec. 1).

Regularization Loss. We apply the same regularization loss as DualSDF [5], *i.e.*, $\mathcal{L}_{\text{reg}} = \gamma \max(\|\boldsymbol{z}\|_2^2, \beta)$, where two hyperparameters γ and β control the strength of the regularization. The regularization term $L_{\text{REG}}(\boldsymbol{z})$ effectively constrains the optimization of the latent code $\boldsymbol{z}$ in the prior distribution of the pretrained MM-VADs. Without such regularization, we find that the single-view 3D reconstruction fails in most cases. Fig. 1 provides one example.

(a) Input (b) with $\mathcal{L}_{\text{REG}}$ (e) w/o $\mathcal{L}_{\text{REG}}$

Fig. 1. The effect of $\mathcal{L}_{\text{REG}}$. Without the regularization term, our model fails to reconstruct 3D shapes from a sketch image.

Multi-trial latent optimization for 3D reconstruction. Similar to other generative models (*e.g.*, GANs), the latent optimization with the proposed MM-VADs is a highly non-convex problem and prone to local minimal. To relieve this issue, we conduct the latent optimization for multiple rounds with different initial latent codes. We use the latent codes with minimal reconstruction loss in multiple trials as the final results of the latent optimization. We find this simple strategy significantly stabilizes our model in the 3D reconstruction task. For example, the mean Chamfer distance decreases from 5.50 to 1.73 in the task of 3D reconstruction from single-view sketch on ShapeNet airplanes and from 9.10 to 4.70 on ShapeNet chairs. In 3D shape manipulation, the latent optimization starts from a known latent code corresponding to the target shape to be edited, and we only run the latent optimization once.

3 Baselines

Here we present more details about the baselines used in our experiments.

– **Encoder-Decoder Networks** [4,13]. This model is originally designed for predicting 3D shapes from sketches, followed by a shape refinement step based on differentiable rendering. We re-purpose this model to reconstruct 3D shapes from RGB images by simply modifying the input channels in the first convolutional layer. We use the official implementations with default hyperparameter settings [3].

[3] https://github.com/cvlab-epfl/MeshSDF

– **EditNeRF** [8] edits a conditional radiance field representation of 3D scenes with sparse scribbles as input. The shape and color of 3D objects are edited by updating the neural network weights. We make qualitative comparisons with the EditNeRF using their pre-trained models[4]. Our model shares many similarities with EditNeRF (*e.g.*, network architecture, scribble-based interaction). However, the proposed model is significantly different from EditNeRF in terms of shape representation (SDFs [11] vs NeRF [9]), shape manipulation method (latent optimization vs network fine-tuning), and the way to bridge the 3D and 2D modalities (shared latent spaces vs differentiable rendering). Tab. 1 provides detailed comparisons between EditNeRF and our model.

Table 1. Comparisons with EditNeRF [8]. † The shape reconstruction and manipulation can be combined and interleaved with the proposed model. This enables us to edit novel instances (Fig. 10 in the main manuscript provides an example). ‡ The time cost of rendering a 256×256 image is included in the editing time

	EditNeRF [8]	Ours
Latent codes	Separate shape and color codes	
Network	A common network shared by all training instances	
Task	Shape/color manipulation with sparse scribbles	
Instance-specific sub-networks	✓	✗
Generative model	✗	✓
3D recon. from sketch or RGB	✗	✓
Editing novel instances †	✗	✓
Shape representation	NeRF [9]	SDFs [11]
Bridge of 2D/3D modalities	Differentiable rendering	Shared latent spaces
Editing method	Update network weights	Latent optimization
Estimated editing time ‡	60s	7s

4 More Experimental details

4.1 Training and Testing Dataset

We train the proposed multi-modal variational auto-decoders (MM-VADs) on the ShapeNet dataset [2]. The training and testing split is the same as DeepSDF [11] and DualSDF [5]. We use the same pre-trained MM-VADs throughout our experiments. For airplanes, there are 1780 shapes for training and 456 shapes for testing. For chairs, there are 3281 training shapes and 833 testing instances. For 3D shape manipulation, we present the results on known shapes (*i.e.*, shapes from training data), similar to EditNeRF [8] and DualSDF [5].

[4] https://github.com/stevliu/editnerf

4.2 3D reconstruction from Sketch or RGB modalities.

Table 2 presents quantitative evaluations of the 3D reconstruction from sketch and RGB inputs under different occlusion ratios, corresponding to the curves in Fig. 7 in the main manuscript. We report results on both vertically and horizontally occluded inputs. Since 3D shapes and their 2D views are generally symmetric horizontally, the proposed model has almost no performance drop when masking out the right-half regions of the inputs. In comparison, the encoder-decoder networks [4] that is trained on full-view inputs suffers from the input domain shift induced by the occlusion.

Table 2. Quantitative results of single-view reconstruction. We report the average Chamfer Distance (30,000 points) multiplied by 10^3 between the reconstructed 3D shapes and the groundtruth (*lower* is *better*). The performance of the proposed model is slightly worse than the encoder-decoder networks [4] trained on the full-view inputs. However, MM-VADs perform more robustly to the input domain shift (*e.g.*, only partial view of input is available). The first column presents the occlusion rate in the input, where "Full" means no occlusion in the input, "1/2-horizontal" the left half of the input is visible, and "3/4-vertical" the top 3/4 region of the object is available. Superscripts in the last row denote the performance drop under the input domain shift (*lower* is *better*). This table corresponds to Fig. 7 in the main text

View	Model	Airplane		Chair		Avg.
		Sketch	RGB	Sketch	RGB	
Full	Enc-Dec	**1.45**	**1.21**	**4.24**	**3.45**	**2.59**
	Ours	1.73	1.40	5.96	4.70	3.44
1/2-horizontal	Enc-Dec	3.30	6.18	16.34	7.61	$8.36^{+5.77}$
	Ours	**1.79**	**1.38**	**6.07**	**5.00**	$\mathbf{3.56}^{+0.12}$
3/4-vertical	Enc-Dec	2.33	1.94	13.10	6.99	$6.09^{+3.50}$
	Ours	**2.07**	**1.55**	**6.91**	**5.64**	$\mathbf{4.04}^{+0.60}$
1/2-vertical	Enc-Dec	3.97	3.56	24.31	10.13	$10.49^{+7.90}$
	Ours	**2.39**	**1.89**	**8.01**	**7.06**	$\mathbf{4.87}^{+1.43}$
1/4-vertical	Enc-Dec	4.28	4.77	27.64	9.77	$11.61^{+9.02}$
	Ours	**3.32**	**2.63**	**8.27**	**8.19**	$\mathbf{5.60}^{+2.16}$

4.3 Few-shot 3D Generation

Fig. 2 presents the 2D examples used in our few-shot shape generation experiments. For each category (*e.g.*, armchair), we randomly sample 10 images from our training data. We then adapt a pre-trained MM-VAD using these 2D examples based on the MineGAN framework [14]. We further collect 200 images per category from our training data for training binary classifiers and calculating FID scores. The classifiers are fine-tuned from a ResNet18 [6] pre-trained on ImageNet.

Fig. 2. 2D examples for few-shot shape generation. Each row presents the 10 2D examples used to adapt a pre-trained MM-VADs to generate armchairs, side chairs, and pink chairs respectively.

5 Limitations

3D reconstruction from 2D modalities. The proposed model fails to reconstruct *fine structures* of 3D shapes from sketches or RGB views, for example, the holes on the back of chairs (Fig. 3a, b, g, h), fine textures on the seat of chairs (Fig. 3e), or the wheelbase of desk chairs (Fig. 3c, f). The capability of modeling fine structures is mainly determined by the 3D shape representation (*i.e.*, SDFs [11]), training samples of SDFs, and the capacity of the proposed generative model. This issue can be potentially relieved by sampling more 3D training points surrounding the surface or increasing the capacity of the proposed model (*e.g.*, enlarging the dimension of the latent space, increasing the depth of 3D shape networks)

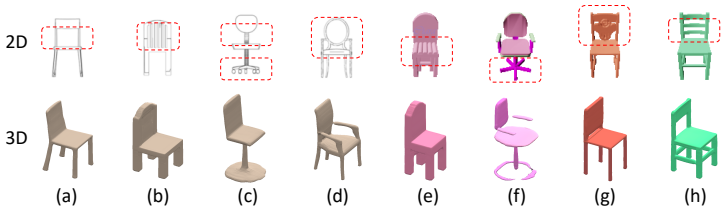

Fig. 3. Limitations of 3D reconstruction from 2D modalities. The proposed model fails to generate *fine structures* of 3D shapes from sketches (a-d) or RGB renderings (e-h). The red bounding boxes highlight the object parts where our model fails to reconstruct the 3D structures.

3D manipulation with 2D color scribble. Similar to GAN-based image manipulation models [15,3,1,10], we are only able to provide editing results within the prior distribution of a pre-trained MM-VAD. For example, in the task of editing shape with color scribbles, if there are multiple scribbles of different colors on the same part of a shape (*e.g.*, the seat of a chair), our model either

edits the shape based on one of the scribbles or generates a surface color that is completely different from all scribbles, as shown in Fig. 4. We notice that the editing results of EditNeRF [8] are similar to ours based on their released demo[5]. Our model may produce unexpected color editing results, for example, the edited 3D surface color may not match the 2D color scribbles provided by the user (Fig. 4d), probably due to bad initialization of the latent code or suboptimal hyperparameter settings. The multi-trial latent optimization described in Sec. 2 may relieve this issue.

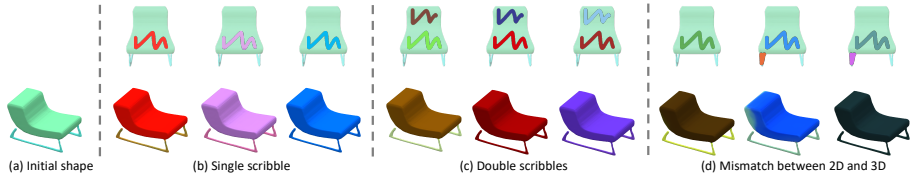

Fig. 4. Limitations of editing shape via color scribble. We are only able to provide color editing results in the prior distribution of the generative model. For example, if there are two scribbles of different color on the same part of a chair, our model either propagates one of the scribbles (*e.g.*, first two columns in (c)) or generates a surface color that are different from both scribbles (*e.g.*, last column in (c)). As a reference, we provide the editing results with single scribble in (b). Our model also produces 3D color editing results that do not match with the 2D input scribbles, as shown in (d).

3D manipulation via 2D sketch. In this task, the major issue is that editing one part of a shape usually leads to changes in other parts. For example, removing the engines on the wing of airplanes results in new engines on the tail in many cases, as shown in Fig. 3 in the main text. Fig. 6 in this section provides more examples. This is mainly because editing shapes via latent optimization can only produce new shapes in the prior distribution of the generative model. It is potentially useful to add more constraints upon the latent optimization, *e.g.*, enforcing the output of the 2D sketch generator to be as similar as possible to the original sketch. However, our preliminary experiments show that the latent optimization with such constraint typically under-fits the edited parts of the sketch and fails to achieve desired edits in 3D shape. In addition, the proposed model fails to add more complicated structures into the shape, for example, adding holes onto the back of chairs (Fig. 6c). We will investigate these issues further in our future work.

Few-shot shape generation. We are unable to adapt a pre-trained MM-VAD to generate shapes of *fine-grained categories* (*e.g.*, single-engine airplanes) using a few 2D RGB images. We also fail to adapt a pre-trained MM-VAD using a few

[5] https://github.com/stevliu/editnerf

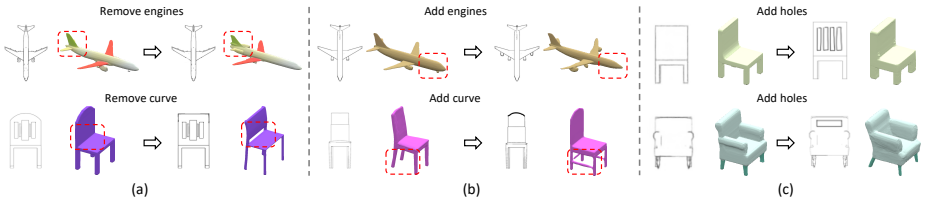

Fig. 5. Limitations of editing shape via sketch. **(a-b)** Editing one part via sketch leads to changes in other parts which are not edited in the sketch. The parts where our model fails to maintain are annotated in red bounding boxes. **(c)** The proposed method fails to add fine structures onto the shape via sketch (*e.g.*, adding holds onto the back of chairs).

2D sketches. We hypothesize that this is because the discriminator is trained from scratch and unable to learn discriminative representations among fine-grained categories or sparse inputs (*e.g.*, sketches) with limited 2D examples. These issues may be relieved by initializing the discriminator with a pre-trained classifier. We leave this in our future work.

6 Diverse color scribbles.

Fig. 6 shows more 3D color editing results with diverse color scribbles. Our method is robust to color scribbles of different shapes/amounts/positions.

Fig. 6. Diverse color scribbles. The first column presents the initial 2D and 3D modalities. The following columns present the color editing results with diverse scribbles.

References

1. Bau, D., Strobelt, H., Peebles, W., Zhou, B., Zhu, J.Y., Torralba, A., et al.: Semantic photo manipulation with a generative image prior. arXiv preprint arXiv:2005.07727 (2020) 5
2. Chang, A.X., Funkhouser, T., Guibas, L., Hanrahan, P., Huang, Q., Li, Z., Savarese, S., Savva, M., Song, S., Su, H., et al.: Shapenet: An information-rich 3d model repository. arXiv preprint arXiv:1512.03012 (2015) 3
3. Gu, J., Shen, Y., Zhou, B.: Image processing using multi-code gan prior. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3012–3021 (2020) 5
4. Guillard, B., Remelli, E., Yvernay, P., Fua, P.: Sketch2mesh: Reconstructing and editing 3d shapes from sketches. In: ICCV (2021) 1, 2, 4
5. Hao, Z., Averbuch-Elor, H., Snavely, N., Belongie, S.: Dualsdf: Semantic shape manipulation using a two-level representation. In: CVPR. pp. 7631–7641 (2020) 2, 3
6. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016) 4
7. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: International conference on machine learning. pp. 448–456. PMLR (2015) 1
8. Liu, S., Zhang, X., Zhang, Z., Zhang, R., Zhu, J.Y., Russell, B.: Editing conditional radiance fields. In: ICCV (2021) 1, 3, 6
9. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: Nerf: Representing scenes as neural radiance fields for view synthesis. In: European conference on computer vision. pp. 405–421. Springer (2020) 3
10. Pan, X., Zhan, X., Dai, B., Lin, D., Loy, C.C., Luo, P.: Exploiting deep generative prior for versatile image restoration and manipulation. In: European Conference on Computer Vision. pp. 262–277. Springer (2020) 5
11. Park, J.J., Florence, P., Straub, J., Newcombe, R., Lovegrove, S.: Deepsdf: Learning continuous signed distance functions for shape representation. In: CVPR. pp. 165–174 (2019) 3, 5
12. Radford, A., Metz, L., Chintala, S.: Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv preprint arXiv:1511.06434 (2015) 1
13. Remelli, E., Lukoianov, A., Richter, S.R., Guillard, B., Bagautdinov, T., Baque, P., Fua, P.: Meshsdf: Differentiable iso-surface extraction. arXiv preprint arXiv:2006.03997 (2020) 2
14. Wang, Y., Gonzalez-Garcia, A., Berga, D., Herranz, L., Khan, F.S., Weijer, J.v.d.: Minegan: effective knowledge transfer from gans to target domains with few images. In: CVPR. pp. 9332–9341 (2020) 1, 4
15. Zhu, J., Shen, Y., Zhao, D., Zhou, B.: In-domain gan inversion for real image editing. In: ECCV. pp. 592–608. Springer (2020) 5