EUROGRAPHICS 2024 / A. Bermano and E. Kalogerakis
(Guest Editors)

COMPUTER GRAPHICS *forum*
Volume 43 (2024), Number 2

Supplementary Material

Alexandre Binninger[1], Amir Hertz[2], Olga Sorkine-Hornung[1], Daniel Cohen-Or[2], Raja Giryes[2]

[1]ETH Zurich, Switzerland [2]Tel Aviv University, Israel

Abstract

We provide more details related to data preparation, implementation, training and evaluation of our method.

1. Network Architecture

The network is composed of three parts: a Vision Transformer encoder, a Transformer decoder and an implicit shape decoder (SPAGHETTI). The Vision Transformer encoder consists in a "sketch to visual embeddings" Transformer encoder. It takes as input a 256×256 grayscale image, decomposes it into 256 patches of size 16×16, uses a learnable position encoding, and maps each patch to a visual embedding of dimension $h_d = 512$. The Vision Transformer itself consists in 8 layers intertwining multi-head attention layers and feed-forward networks with layer normalization [DBK*20]. Then, we use a *Transformer decoder* as our "visual embedding to shape latent code" network. It maps the 256 visual embeddings to latent space code. The latent space code is composed of m vectors of dimensions d_{model}. Single-class SENS uses $m = 16$ and $d_{\text{model}} = 512$, while multi-class SENS uses $m = 32$ and $d_{\text{model}} = 768$. The Transformer decoder also takes as input m *learnable* part queries of dimension $1.5h_d$ that are optimized simultaneously with the weights of the network. It is composed of 12 cross-attention layers and feed-forward networks with layer normalization. The output of the Transformer decoder is then mapped to the latent code z_h of the shape decoder latent space via an MLP with ReLU activation.

2. Training

Single-class models are trained on an Nvidia RTX 3090 GPU for 850 epochs. We use a gradual warmup scheduler [GDG*17] to linearly increase the learning rate at each epoch. The learning rate starts at 10^{-7} and linearly increases to 10^{-6}. Our approach to training the multi-class model was based on a combined dataset from various classes, namely chairs, planes, and lamps. We include ShapeNet outline and partial outline renderings, as well as CLIPasso [VPB*22] abstract sketches, and ProSketch chair sketches [ZQG*21]. The training was based on 630 epochs, and the training duration for the multi-class model was 96 hours, which is longer than the 60 hours required for the single-class model due to the increased amount of data per epoch. The same learning rate and scheduler were used.

3. Evaluation

Our evaluation is performed on the AmateurSketch dataset [QGS*21], which contains 3000 freehand sketches of ShapeNet shapes [CFG*15] of medium abstraction level. We only compare with the chair class, because this is the only class ubiquitously supported by all the methods we compare with.

Table 1: *Performance comparison of shape reconstruction methods on the AmateurSketch dataset [QGS*21] using chamfer distance (CD), earth mover's distance (EMD), and Fréchet inception distance (FID). Lower values indicate better performance. Comparison is done with Pixel2Mesh [WZL*18], Sketch2Mesh [GRYF21], and DeepSketch [ZGZS22]. The notions "cropped" and "padded" refer to the differences in input normalization. DeepSketch results are shown with the network trained with their default training data and re-trained with our training data.*

Method	CD↓	EMD↓	FID↓
Pixel2Mesh	0.2191	0.1658	401.7
Sketch2Mesh (padded input)	0.2113	0.1573	368.4
Sketch2Mesh (cropped input)	0.2325	0.1635	305.8
DeepSketch (default dataset)	0.1520	0.1142	292.2
DeepSketch (our dataset)	0.1920	0.1417	317.4
SENS	**0.1186**	**0.0946**	**171.3**

3.1. Objective evaluation

Our quantitative evaluation is based on several metrics. We compare our results with different methods: Pixel2Mesh [WZL*18], Sketch2Mesh [GRYF21] and DeepSketch [ZGZS22]. The comparison results are shown in Table 1.

3.1.1. Chamfer distance (CD)

The chamfer distance calculates the average distance between each point in one set to its closest point in the other set and is an intuitive way to quantify the dissimilarity between two point clouds. It is

thus widely used for geometric comparison. The chamfer distance between two point sets A and B can be defined as follows:

$$d_{\text{chamfer}}(A,B) = \frac{1}{|A|}\sum_{a\in A}\min_{b\in B}\|a-b\|^2 + \frac{1}{|B|}\sum_{b\in B}\min_{a\in A}\|a-b\|^2.$$

For each sketch in the AmateurSketch dataset, we extract a mesh from the implicit shape produced by our network. Then, we sample $100,000$ points on the surface of our output and on the reference mesh, and compute the chamfer distance between the two produced point clouds using the Point Cloud Utils library [Wil22].

3.1.2. Earth mover's distance (EMD)

The earth mover's distance is a measure of dissimilarity between two probability distributions or point sets, and is often described as the minimum cost to transform one distribution into the other. The EMD between two point sets $A = \{a_i \in \mathbb{R}^3\}_{i=1}^n$ and $B = \{b_j \in \mathbb{R}^3\}_{j=1}^m$ can be formally defined as:

$$\text{EMD}(A,B) = \min_{\pi\in\Pi(A,B)}\sum_{i=1}^{n}\sum_{j=1}^{m}\pi_{i,j}\|a_i - b_j\|,$$

where π is a correspondence between A and B, *i.e.* $\Pi(A,B)$ is the set of $n\times m$ matrices, where rows and columns sum to one and $\pi_{i,j}\in[0,1]$ is the coefficient indicating how much points a_i and b_j correspond to each other. Due to the computational complexity of the EMD, we sample 1000 points on both meshes. We also use Point Cloud Utils library [Wil22] for the computation of the EMD.

3.1.3. Fréchet inception distance (FID)

To take visual perception into consideration, we use the Fréchet inception distance [HRU*18]. FID evaluates the similarity between two sets of images, generated and real, by computing the Fréchet distance between the Gaussian distributions of their respective features. A lower FID value signifies a greater resemblance between the two image sets. The shading image based FID has been described in SDF-StyleGAN [ZLWT22], for which the authors report that it yields relevant results for measuring the plausibility and similarity of two shapes. We sample 20 views and render the shape S_{out} produced by SENS and the reference shape S_{ref}. The features are then extracted from these image via the Inception-V3 network [SVI*15], an architecture trained over ImageNet [DDS*09], which maps an image to a probability distribution over 1000 classes. From this probability distribution, we can extract the mean μ_i and the covariance matrix Σ_i for each image i. The formula used to compute the FID is given by:

$$\text{FID} = \frac{1}{20}\sum_{i=1}^{20}\left(\|\mu_i^{\text{out}} - \mu_i^{\text{ref}}\|^2 + \text{Tr}\left(\Sigma_i^{\text{out}} + \Sigma_i^{\text{ref}} - 2\sqrt{\Sigma_i^{\text{ref}}\Sigma_i^{\text{out}}}\right)\right).$$

To compute the FID, we use the cleanFID library [PZZ22].

3.1.4. Interpretation

We report the results of our objective evaluation in Table 1. First, we note that Sketch2Mesh [GRYF21] fails to produce a shape in 112 cases when the input was cropped, and to provide a fair comparison we could not use their refinement because the camera view parameters are not an input of our method. We report the results for both cropped and padded input sketches, observing that the optimal method varies depending on the used metric. Because the training procedure is available for DeepSketch [ZGZS22], we train this method for our evaluation in two ways: (1) using their default dataset, which includes their synthetic renders and ProSketch [ZQG*21], and (2) using our training dataset which consists of our full outline rendering, ProSketch, and abstract CLIPasso [VPB*22] renders. We indicate results for both training procedures. The evaluation on the default DeepSketch is done on padded input. Because cropped inputs are used for retraining DeepSketch on our dataset, we crop and center the AmateurSketch input sketches for its evaluation. Pixel2Mesh [WZL*18] and our method are evaluated with cropped input sketches.

For both geometric and perceptual metrics, SENS performs substantially better than the state of the art. This indicates that SENS is particularly suitable for sketches with different levels of abstraction, and therefore is a relevant approach to allow people of various drawing skills to attempt sketch-based modeling. Since training DeepSketch on our dataset does not show any improvement on the metrics, this additionally indicates that the dataset is not the sole factor that explains the difference of performance between SENS and the state of the art.

Table 2: *Performance comparison of multi-class shape reconstruction methods on the AmateurSketch dataset [QGS*21] using chamfer distance (CD), earth mover's distance (EMD), and Fréchet inception distance (FID). Lower values indicate better performance. Comparison is done with LAS-diffusion [ZPW*23].*

Method	CD↓	EMD↓	FID↓
LAS-diffusion	0.2112	0.1585	209.2
SENS multi-class	0.1171	0.0940	171.0

3.1.5. Multi-class reconstruction

While LAS-Diffusion [ZPW*23] is targeted toward a view-aware setting, this sketch-to-shape method can run without camera parameters. Since the authors provide the multi-class pretrained network for this task, we compare multi-class SENS with LAS-Diffusion using the same evaluation metrics as for the single-class comparison. The results are reported in Table 2. We can see that our method performs better than LAS-diffusion on the AmateurSketch dataset. However, we emphasize that the multi-class LAS-diffusion has been trained on all the ShapeNet classes, while our method training was focused on only 3 classes. Moreover, while it is possible to run LAS-diffusion without input view information, the authors state in their ablation study that using a view-agnostic network tends to yield additional or wrong geometry. Therefore, no definitive conclusion can be drawn from this comparison.

Additionally, when comparing single-class and multi-class SENS, we notice that the metrics give very similar results. This shows that our multi-class setup has good generalization abilities.

3.2. Subjective evaluation (user study)

To perform a perceptual evaluation of our work, we conduct a user study. We *randomly* sample 24 sketches from the AmateurSketch

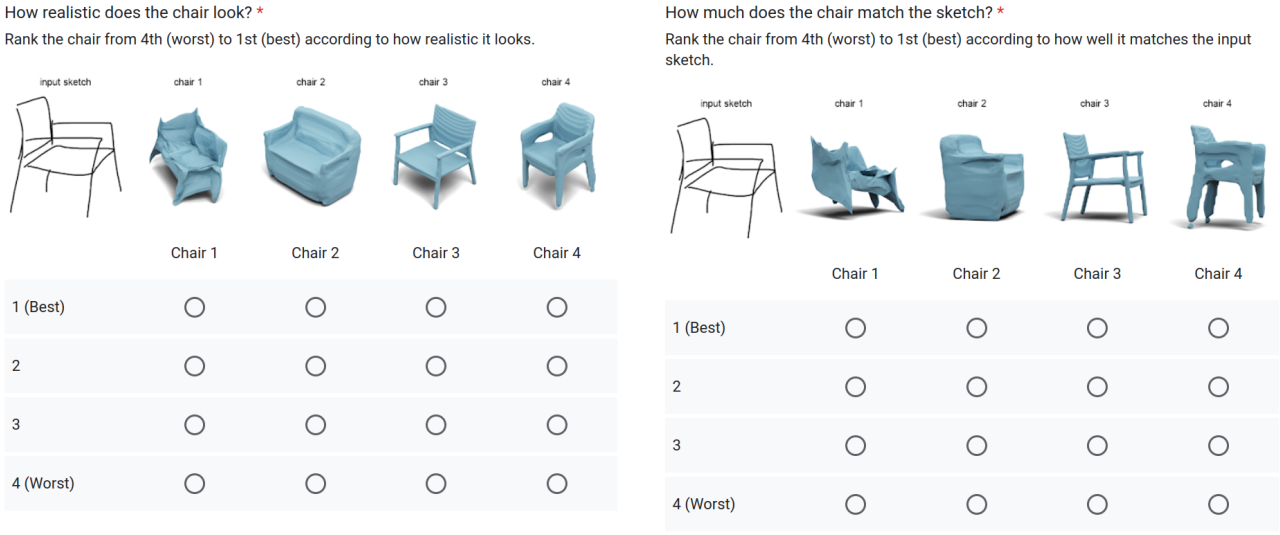

Figure 1: *The two types of questions asked in our user study. When asking for how realistic the shape looks, the same view is applied for rendering the shapes. When asking for similarity with the input sketch, shapes are rendered with the same azimuth angle as the input sketch. The azimuth angle is provided by the AmateurSketch dataset.*

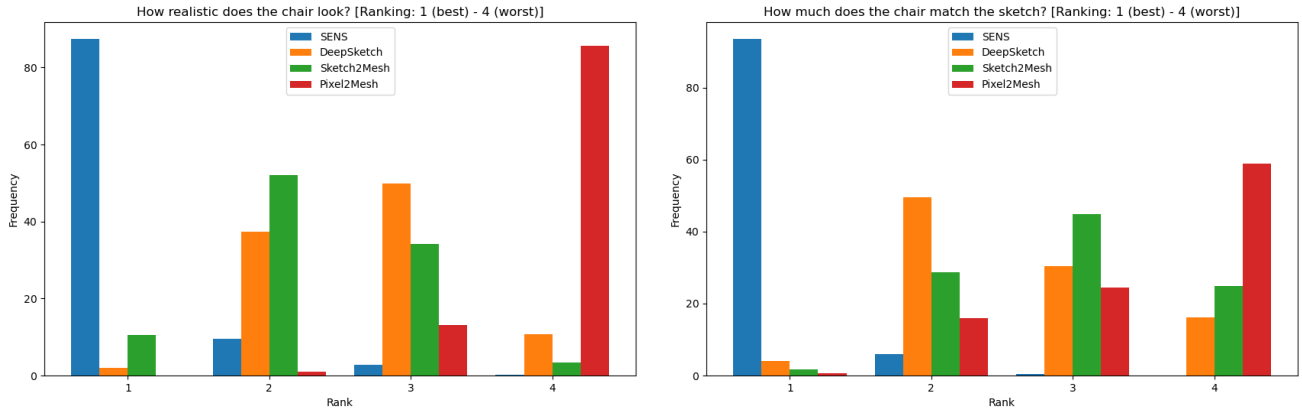

Figure 2: *Results of our user study, displayed as an histogram. The results highlight the performance of our method in comparison to Pixel2Mesh [WZL*18], Sketch2Mesh [GRYF21], and retrained DeepSketch [ZGZS22] in terms of realism and similarity to input sketches.*

dataset and render the output of SENS, Pixel2Mesh [WZL*18], Sketch2Mesh [GRYF21] (cropped input), and retrained DeepSketch [ZGZS22]. We show in Fig. 1 the exact format used for the user study. For each sketch, we ask participants to rank the four methods' output in two questions: how realistic and how close to the input sketch the resulting chair looks. For the second question, we align the rendering view of the shape with the same azimuth angle as given by the AmateurSketch dataset. The order of the methods is randomized across the sketches, but the same order is used for both questions for each sketch. We recruit 54 individuals of diverse backgrounds and ages to partake in the user study, including 15 women and 39 men.

The results are reported in Table 3 and Fig. 2. According to this study, SENS provides the most realistic shape in 87.9% of the cases and the most similar to the input sketch in 94% of the cases. Pixel2Mesh is often deemed to perform the worst, especially in terms of realism. Sketch2Mesh and DeepSketch both seem to perform equally well for both questions and rank second and third with nearly equal scores, as shown by the interquartile range in Table 4. Therefore, our user study is aligned with our objective evaluation.

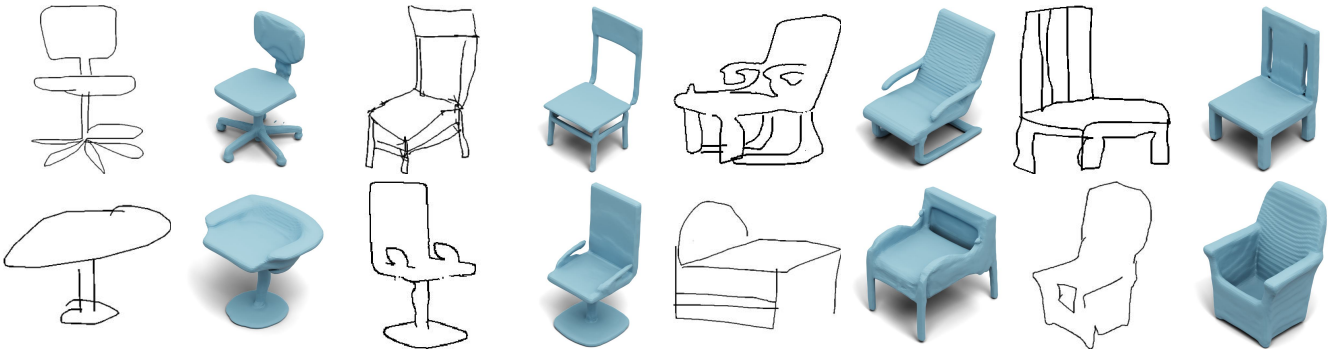

Figure 3: *Some sketches and shapes from the Task 1 of the usability study. The results come from each user (P1 to P8, ordered from left to right, top to bottom). Some sketches (P3, P6, and P8) are edited versions of the outline rendering from previously generated shapes. The displayed shapes are not solely generated by the input sketches, but might have been refined via part reconstruction or part-based modeling.*

Table 3: *Perceptual evaluation through a user study, highlighting the performance of our method in comparison to Pixel2Mesh [WZL*18], Sketch2Mesh [GRYF21] and retrained DeepSketch [ZGZS22] in terms of realism and similarity to input sketches. The ranking in each question is from 1 (best) to 4 (worst).*

Question	Realistic				Similar to sketch			
Rank	1	2	3	4	1	2	3	4
Pixel2Mesh	0.1	1.1	12.8	**86.0**	0.4	15.8	24.8	**59.0**
Sketch2Mesh	10.3	**53.1**	33.1	3.4	1.6	28.5	**45.1**	24.8
DeepSketch	1.7	36.6	**51.3**	10.3	4.0	**50.0**	29.8	16.2
SENS	**87.9**	9.1	2.7	0.3	**94.0**	5.7	0.3	0.0

Table 4: *Median and interquartile range (IQR) of the results of our user study, for both realism and similarity to input sketches.*

Method	Realistic		Similar	
	Median	IQR	Median	IQR
Pixel2Mesh	4.0	0.0	4.0	1.0
Sketch2Mesh	2.0	1.0	3.0	1.0
DeepSketch	3.0	1.0	2.0	1.0
SENS	1.0	0.0	1.0	0.0

3.3. Usability study

To evaluate the usability of our sketch-to-shape generation and editing methods, we carried out a usability study, drawing inspiration from the study presented in GA-Sketching [ZLY*23]. Eight participants from diverse backgrounds participated in the study. Among them, half were aged between 20 and 30, while the rest were above 30. The gender distribution was balanced, with 50% women and 50% men. In terms of 3D modeling experience, 25% reported having no experience, 50% had limited experience, and 25% identified as hobbyists. When it came to 2D sketching or drawing, half the participants had no experience, 25% reported limited experience, and 25% described themselves as hobbyists. Notably, none of the participants were professional 2D illustrators or 3D artists. The modeling session was divided into two phases. Initially, participants were introduced

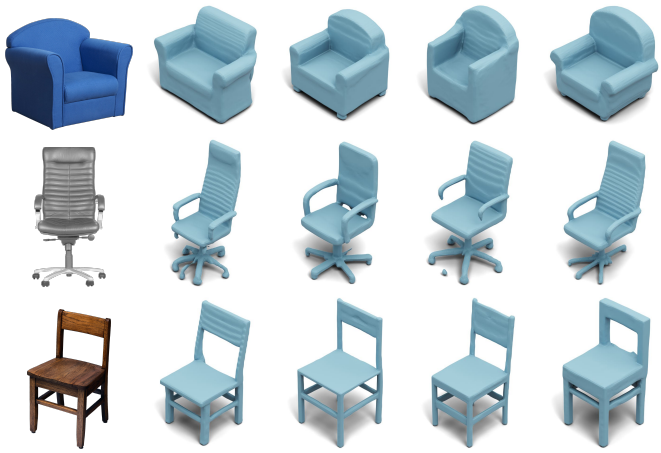

Figure 4: *The three target shape images are displayed in the first column, with four attempts to model them during Task 2 of the usability study. The target shapes are sourced from the public domain.*

to the software's operation and its various functionalities, which included sketch-to-shape generation, outline rendering, part-based modeling, and part refinement. Subsequently, participants undertook two tasks. In Task 1, they had the freedom to sketch any chair design; however, they were required to use each of the software's functionalities at least once during the session, ensuring they became familiar with all available options. Task 2 involved modeling three specific shapes provided as reference images. While their sketches did not need to align with the image's perspective, the resulting shapes should closely resemble the target. The outcomes from both tasks are depicted in Fig. 3 and Fig. 4. The outcomes of Task 1 underscore the system's resilience and adaptability. Even when participants, some of whom lacked advanced drawing skills, sketched rudimentary or imprecise chair designs, the algorithm consistently produced coherent 3D shapes. Often, only a few additional intuitive modeling steps were needed to refine the shape. Task 2 further demonstrates the system's ability to convert target ideas into concrete 3D models. Participants were able to transform target images into 3D chairs, even when the sketched perspectives differed from the reference images. This ease of transformation from a 2D reference image to a

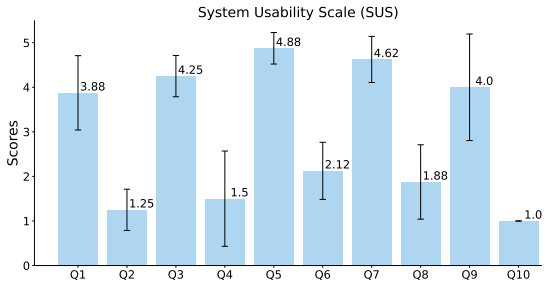

Figure 5: *The mean of SUS scores. The whiskers represent the standard deviation. For questions with odd index, higher scores indicate better performance; for even-numbered questions, lower scores are preferable.*

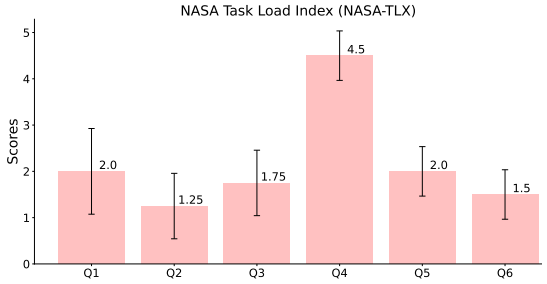

Figure 6: *The mean of the NASA-TLX scores, which asks the participant to rate their experience according to six criteria to assess the intensity of the effort. The whiskers represent the standard deviation. The lower the better, except for Q4.*

realistic 3D chair model accentuates the system's ability in bringing users' visions to realization.

After completing the modeling session, participants were invited to complete a feedback form including both the System Usability Scale (SUS) questionnaire [Bro96] and the NASA Task Load Index (NASA-TLX) questionnaire [HS88]. The SUS questionnaire contains ten questions which evaluate the system's usability, and gauge its usefulness, ease of use, and consistency. The NASA-TLX questionnaire is designed to measure task-related effort intensities, such as mental (Q1), physical (Q2), and temporal (Q3) demands, as well as performance (Q4), effort (Q5), and frustration levels (Q6). The results are shown in Fig. 5 and Fig. 6. Notably, the exceptionally low SUS scores for Q2 and Q4, combined with elevated scores for Q5 and Q7, and notably the unanimous score of 1 for Q10, suggest a high intuitiveness with the editing options. This observation is further corroborated by the low scores reflected in the NASA-TLX. The marginally subpar scores for Q6 and Q9 appear to align with the absence of very high-frequency details from sketches to the resulting shape, a limitation we acknowledge in the main paper. However, it is worth noting the significant elevation in the NASA-TLX Q4 score, implying participants' satisfaction with their performance. Participants could readily conceptualize an initial rudimentary shape, even from the most abstract sketches and for those with very limited experience.

3.4. Additional visual results

In addition to the quantitative and qualitative evaluations, we also provide further visual results. We *randomly* sample 128 sketches from the AmateurSketch dataset and present the result of SENS in Fig. 7, Fig. 8, Fig. 9, and Fig. 10.

References

[Bro96] BROOKE J.: *SUS – a quick and dirty usability scale*. 01 1996, pp. 189–194. 5

[CFG*15] CHANG A. X., FUNKHOUSER T., GUIBAS L., HANRAHAN P., HUANG Q., LI Z., SAVARESE S., SAVVA M., SONG S., SU H., XIAO J., YI L., YU F.: Shapenet: An information-rich 3d model repository, 2015. doi:10.48550/ARXIV.1512.03012. 1

[DBK*20] DOSOVITSKIY A., BEYER L., KOLESNIKOV A., WEISSENBORN D., ZHAI X., UNTERTHINER T., DEHGHANI M., MINDERER M., HEIGOLD G., GELLY S., USZKOREIT J., HOULSBY N.: An image is worth 16x16 words: Transformers for image recognition at scale, 2020. doi:10.48550/ARXIV.2010.11929. 1

[DDS*09] DENG J., DONG W., SOCHER R., LI L.-J., LI K., FEI-FEI L.: Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition* (2009), pp. 248–255. doi:10.1109/CVPR.2009.5206848. 2

[GDG*17] GOYAL P., DOLLÁR P., GIRSHICK R., NOORDHUIS P., WESOLOWSKI L., KYROLA A., TULLOCH A., JIA Y., HE K.: Accurate, large minibatch sgd: Training imagenet in 1 hour, 2017. doi:10.48550/ARXIV.1706.02677. 1

[GRYF21] GUILLARD B., REMELLI E., YVERNAY P., FUA P.: Sketch2mesh: Reconstructing and editing 3d shapes from sketches. *CoRR abs/2104.00482* (2021). arXiv:2104.00482. 1, 2, 3, 4

[HRU*18] HEUSEL M., RAMSAUER H., UNTERTHINER T., NESSLER B., HOCHREITER S.: Gans trained by a two time-scale update rule converge to a local nash equilibrium, 2018. arXiv:1706.08500. 2

[HS88] HART S. G., STAVELAND L. E.: Development of nasa-tlx (task load index): Results of empirical and theoretical research. In *Human Mental Workload*, Hancock P. A., Meshkati N., (Eds.), vol. 52 of *Advances in Psychology*. North-Holland, 1988, pp. 139–183. doi:10.1016/S0166-4115(08)62386-9. 5

[PZZ22] PARMAR G., ZHANG R., ZHU J.-Y.: On aliased resizing and surprising subtleties in gan evaluation. In *CVPR* (2022). 2

[QGS*21] QI A., GRYADITSKAYA Y., SONG J., YANG Y., QI Y., HOSPEDALES T. M., XIANG T., SONG Y.-Z.: Toward fine-grained sketch-based 3d shape retrieval. *Trans. Img. Proc. 30* (jan 2021), 8595–8606. doi:10.1109/TIP.2021.3118975. 1, 2

[SVI*15] SZEGEDY C., VANHOUCKE V., IOFFE S., SHLENS J., WOJNA Z.: Rethinking the inception architecture for computer vision, 2015. arXiv:1512.00567. 2

[VPB*22] VINKER Y., PAJOUHESHGAR E., BO J. Y., BACHMANN R. C., BERMANO A. H., COHEN-OR D., ZAMIR A., SHAMIR A.: Clipasso: Semantically-aware object sketching. *ACM Trans. Graph. 41*, 4 (jul 2022). doi:10.1145/3528223.3530068. 1, 2

[Wil22] WILLIAMS F.: Point cloud utils, 2022. https://www.github.com/fwilliams/point-cloud-utils. 2

[WZL*18] WANG N., ZHANG Y., LI Z., FU Y., LIU W., JIANG Y.-G.: Pixel2mesh: Generating 3d mesh models from single rgb images. In *ECCV* (2018). 1, 2, 3, 4

[ZGZS22] ZHONG Y., GRYADITSKAYA Y., ZHANG H., SONG Y.-Z.: A study of deep single sketch-based modeling: View/style invariance, sparsity and latent space disentanglement. *Comput. Graph. 106*, C (aug 2022), 237–247. doi:10.1016/j.cag.2022.06.005. 1, 2, 3, 4

[ZLWT22] ZHENG X.-Y., LIU Y., WANG P.-S., TONG X.: Sdf-stylegan: Implicit sdf-based stylegan for 3d shape generation. In *Comput. Graph. Forum (SGP)* (2022). 2

[ZLY*23] ZHOU J., LUO Z., YU Q., HAN X., FU H.: Ga-sketching: Shape modeling from multi-view sketching with geometry-aligned deep implicit functions, 2023. arXiv:2309.05946. 4

[ZPW*23] ZHENG X.-Y., PAN H., WANG P.-S., TONG X., LIU Y., SHUM H.-Y.: Locally attentional sdf diffusion for controllable 3d shape generation. *ACM Transactions on Graphics (SIGGRAPH) 42*, 4 (2023). 2

[ZQG*21] ZHONG Y., QI Y., GRYADITSKAYA Y., ZHANG H., SONG Y.-Z.: Towards practical sketch-based 3d shape generation: The role of professional sketches. *IEEE Transactions on Circuits and Systems for Video Technology 31*, 9 (2021), 3518–3528. doi:10.1109/TCSVT.2020.3040900. 1, 2

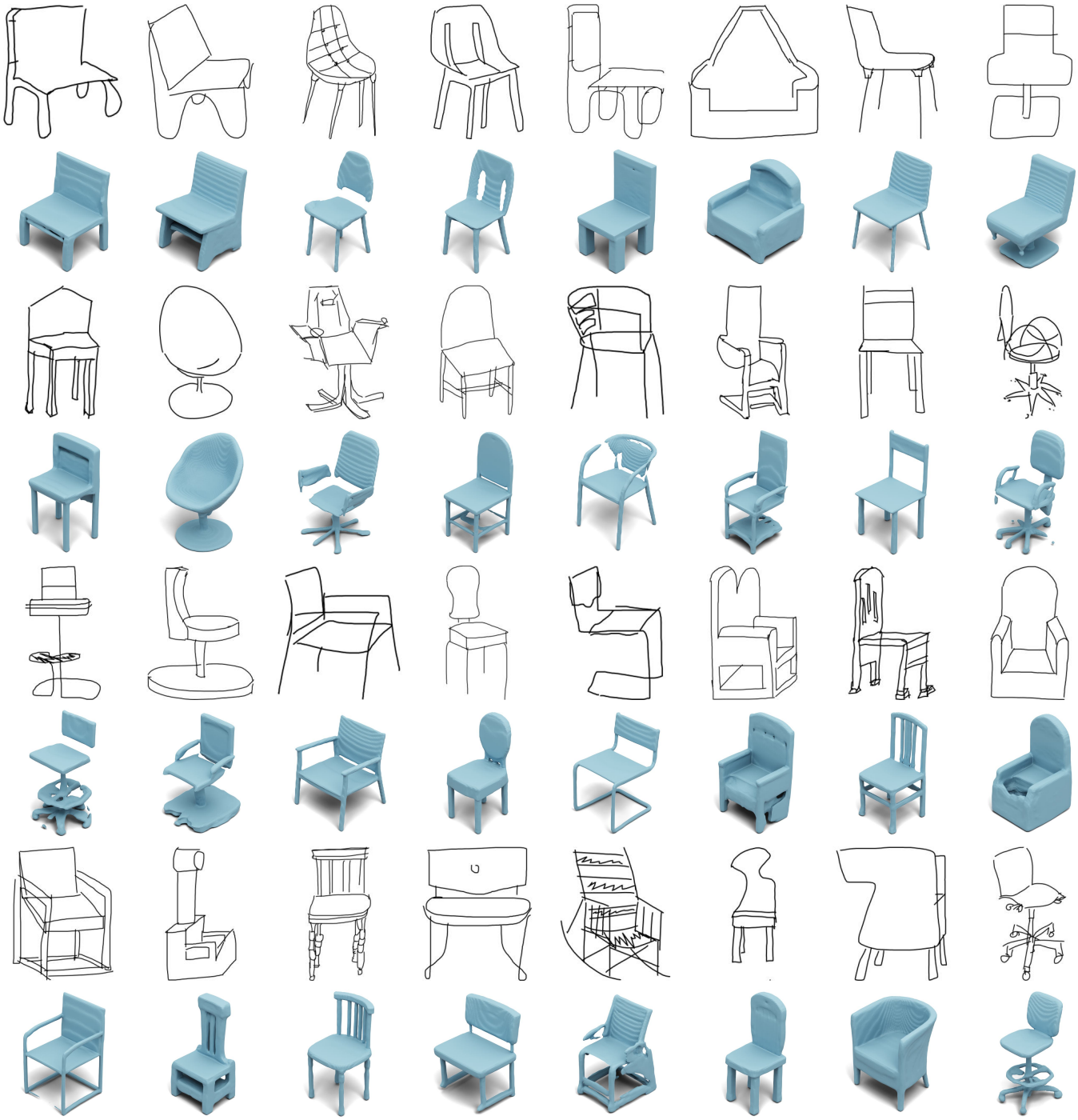

Figure 7: *We randomly sample sketches from the AmateurSketch dataset and showcase the results of our method.*

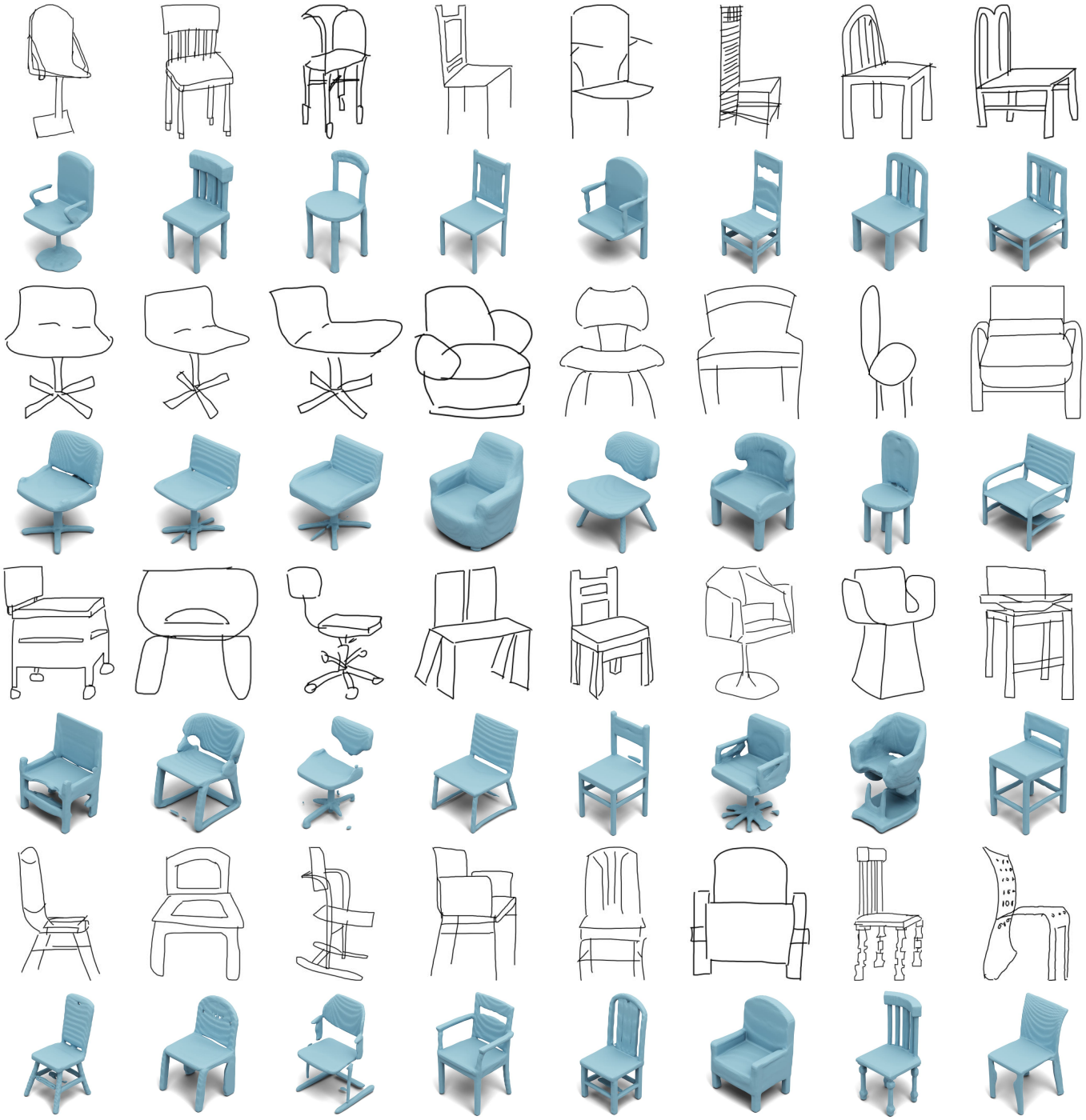

Figure 8: *We randomly sample sketches from the AmateurSketch dataset and showcase the results of our method.*

Figure 9: *We randomly sample sketches from the AmateurSketch dataset and showcase the results of our method.*

Figure 10: *We randomly sample sketches from the AmateurSketch dataset and showcase the results of our method.*