CLIP-Forge: Towards Zero-Shot Text-to-Shape Generation Supplemental Material

Aditya Sanghi[1] Hang Chu[1] Joseph G. Lambourne[1] Ye Wang[2]
Chin-Yi Cheng[1] Marco Fumero[1] Kamal Rahimi Malekshan[1]
[1]Autodesk AI Lab [2]Autodesk Research
{aditya.sanghi,hang.chu,joseph.lambourne,ye.wang,chin-yi.cheng,marco.fumero,kamal.malekshan}@autodesk.com
https://github.com/AutodeskAILab/Clip-Forge

1. Architecture and Experiment Details

For all our experiments in the ablation section of the main paper, we run the second stage network with 3 different seeds and report the mean in the main paper. We take the best seed for the experiment section in the main paper to report the qualitative and quantitative results. The text queries (or prompts) used for classification FID, MMD, and Acc. are shown in Table 1. Note that these text queries are mostly taken from WordNet [5] with added common synonyms and shape attributes. In Table 3, we show category-wise accuracy results of CLIP-Forge in the main paper's Table 1. For our visualizations, we output a shape with 64^3 resolution and use the rendering script inspired by [1]. We use a set of different thresholds values and pick the threshold for different category that yields the best visual result.

In the main paper, we refer to the batch normalization based voxel encoder as VoxEnc, whereas when we add residual connection to VoxEnc we refer it to as ResVoxEnc. For both of these encoders, we have 4 3D convolution layers followed by a linear layer. The input to these encoder is a 32^3 voxel representation based shape. We also experiment with a point cloud based encoder which is inspired by PointNet. The PointNet encoder has 5 linear layers followed by a max pooling operation. We then use an MLP followed by a final linear layer to project it to the latent size. The input to this encoder is a point cloud with 2048 sampled points. For the decoder, we refer to the residual connection based network as RN-OccNet. In this model, we concatenate the query locations with the latent code and pass it through a 5 block ResNet based decoder. We also experiment with conditioning the batchnorm of the decoder instead of concatenating it, which we refer to as CBN-OccNet. Both these decoders are inspired by OccNet [6]. For our point cloud based generation experiments, we use a FoldingNet [10] based decoder, where we use two folding based operations with a single square grid.

Finally, we use RealNVP [4] for the prior model. We use 5 blocks of coupling layer containing translation and scale along with batch norm, where the masking is inverted after each block. Each network comprises of a 2-layer MLP followed by a linear layer. A 1024 hidden vector size is used. For the MAF [8] model, we also use the same number of blocks and hidden vectors.

2. Comparison with Supervised Models

In this section, we provide a more detailed comparison between CLIP-Forge and supervised methods. Note, it is not clear how to compare our zero-shot model with supervised models. As our end goal is to generate shapes across categories and text queries, we decide to use our original text query subset (mentioned above) and Shapenet (v2) test dataset as the test set. This test set ensures we test on commonly described words for different shape category as mentioned in WordNet [5]. We consider two datasets: T2S is the annotated text-shape description dataset from text2shape [2] which mainly contains information regarding texture, and SN13 is the ShapeNet (v2) subset containing 13 categories from [3].

T2S dataset [2] has text labels only for chair and table-class, so we train a supervised baseline model that has a linear layer connecting a pre-trained CLIP text encoder [9] and a pre-trained occupancy network decoder [6] using an L2 loss in the latent space. We use the same text encoder and shape decoder in this baseline to ensure a fair comparison. We compare the baseline model with our model which does not have use any supervision from text labels and is also trained on T2S shape dataset (chair and table only). The results are shown in the first part of Table 2. It can be seen from the table that our model can generate shapes in chair and table categories based on common words with higher quality despite not using any text label information.

To test baseline models on all of ShapeNet subset (SN13), as there is no text label data, we create a simple supervision signal by directly using the category name as

“a triangular airplane”	“an airplane”	“a jet”	“a fighter plane”	“a biplane”
“a seaplane”	“a space shuttle”	“a supersonic plane”	“a rocket plane”	“a delta wing”
“a swept wing plane”	“a straight wing plane”	“a propeller plane”	“a boeing”	“an airbus”
“an f-16”	“a plane”	“an aeroplane”	“an aircraft”	“a commercial plane”
“a square bench”	“a round bench”	“a circular bench”	“a rectangular bench”	“a thick bench”
“a thin bench”	“a bench”	“a pew”	“a flat bench”	“a settle”
“a back bench”	“a laboratory bench”	“a storage bench”	“a park bench”	“a cuboid cabinet”
“a round cabinet”	“a rectangular cabinet”	“a thick cabinet”	“a thin cabinet”	“a cabinet”
“a garage cabinet”	“a desk cabinet”	“a dresser”	“a cupboard”	“a container”
“a case”	“a locker”	“a cupboard”	“a closet”	“a sideboard”
“a square car”	“a round car”	“a rectangular car”	“a thick car”	“a thin car”
“a car”	“a bus”	“a shuttle-bus”	“a pickup car”	“a truck”
“a suv”	“a sports car”	“a limo”	“a jeep”	“a van”
“a gas guzzler”	“a race car”	“a monster truck”	“an armored”	“an atv”
“a microbus”	“a muscle car”	“a retro car”	“a wagon car”	“a hatchback”
“a sedan”	“an ambulance”	“a roadster car”	“a beach wagon”	“an auto”
“an automobile”	“a motor car”	“a square chair”	“a round chair”	“a rectangular chair”
“a thick chair”	“a thin chair”	“a chair”	“an arm chair”	“a bowl chair”
“a rocking chair”	“an egg chair”	“a swivel chair”	“a bar stool”	“a ladder back chair”
“a throne”	“an office chair”	“a wheelchair”	“a stool”	“a barber chair”
“a folding chair”	“a lounge chair”	“a vertical back chair”	“a recliner”	“a wing chair”
“a sling”	“a seat”	“a cathedra”	“a square monitor”	“a round monitor”
“a rectangular monitor”	“a thick monitor”	“a thin monitor”	“a monitor”	“a crt monitor”
“a TV”	“a digital display”	“a flat panel display”	“a screen”	“a television”
“a telly”	“a video”	“a square lamp”	“a round lamp”	“a rectangular lamp”
“a cuboid lamp”	“a circular lamp”	“a thick lamp”	“a thin lamp”	“a lamp”
“a street lamp”	“a fluorescent lamp”	“a gas lamp”	“a bulb”	“a lantern”
“a table lamp”	“a torch”	“a square loudspeaker”	“a round loudspeaker”	“a rectangular loudspeaker”
“a circular loudspeaker”	“a thick loudspeaker”	“a thin loudspeaker”	“a loudspeaker”	“a subwoofer speaker”
“a speaker”	“a speaker unit”	“a tannoy”	“a thick gun”	“a thin gun”
“a gun”	“a machine gun”	“a sniper rifle”	“a pistol”	“a shotgun”
“an ak-47”	“an uzi”	“an M1 Garand”	“a M-16”	“a firearm”
“a shooter”	“a weapon”	“a square sofa”	“a round sofa”	“a rectangular sofa”
“a thick sofa”	“a thin sofa”	“a sofa”	“a double couch”	“a love seat”
“a chesterfield”	“a convertible sofa”	“an L shaped sofa”	“a settee sofa”	“a daybed”
“a sofa bed”	“an ottoman”	“a couch”	“a lounge”	“a divan”
“a futon”	“a square table”	“a round table”	“a circular table”	“a rectangular table”
“a thick table”	“a thin table”	“a table”	“a dressing table”	“a desk”
“a refactory table”	“a counter”	“an operating table”	“a stand”	“a billiard table”
“a pool table”	“a ping-pong table”	“a console table”	“an altar table”	“a worktop”
“a workbench”	“a square phone”	“a rectangular phone”	“a thick phone”	“a thin phone”
“a phone”	“a desk phone”	“a flip-phone”	“a telephone”	“a telephone set”
“a cellular telephone”	“a cellular phone”	“a cellphone”	“a cell”	“a mobile phone”
“an iphone”	“a square boat”	“a round boat”	“a rectangular boat”	“a thick boat”
“a thin boat”	“a boat”	“a war ship”	“a sail boat”	“a speedboat”
“a cabin cruiser”	“a yacht”	“a rowing boat”	“a watercraft”	“a ship”
“a canal boat”	“a ferry”	“a steamboat”	“a barge”	
0/9	1/9	2/9	3/9	4/9
5/9	6/9	7/9	8/9	9/9

Table 1. The full list of text queries. The colors show the results of the human perceptual evaluation study as described in section 10. Green indicates queries which gave rise to distinctive and recognisable shapes, while red indicates the shapes could not be distinguished from those generated using the ShapeNet category name. The key at the bottom shows the fraction of the nine crowd workers who found the generated shape recognisable. White indicates the query was not rated in the perceptual study.

method	*dataset*	**FID↓**	**MMD↑**	**Acc.↑**
sup.	T2S [2]	14881.96	0.1418	6.84
ours		14746.90	0.5412	30.77
sup.	SN13 [3]	19896.11	0.1805	14.10
ours		**2425.25**	**0.6607**	**83.33**

Table 2. Additional detailed comparisons with supervised models, where sup. stands for the supervised model.

the text for training the supervised model. The results are shown in second part of Table 2. It can be seen that our model outperforms the supervised method, demonstrating its stronger zero-shot generalization ability. This results also indicate that our model scales better with more data without requiring text-shape labels. In Figure 14, we show qualitative results of the supervised baselines, where the model fails to generate cars when trained on T2S, and fails to capture the details of sports car when trained with SN13.

3. Category-wise Accuracy Results

We also report category-wise accuracy results obtained from our classifier for our method in Table 3. It can be generally noted that our method can generate shapes across all categories of Shapenet. However, accuracy across some categories such as airplane and car are higher than other categories such as boat and loudspeaker. We hypothesize that this may be due to some categories having larger data points during training compared to others.

4. Comparison with Text2Img+Img2Shape

In this section, we compare our method to off-the-shelf networks that simply generate an image from text first and then generate a 3D shape from the image. We use pre-trained DALLE-mini for converting a text to image and use a pre-trained occupancy network with image encoder to convert an image to 3D shape. The results are shown in Fig. 1. It can be seen that the resulting shapes suffer from poor quality. This is mainly due to the domain gap between generated images and natural images such as distortion artifacts and unclean background.

5. Effect of Threshold Parameter

Our results are strongly affected by the threshold used to create the occupancy value. We use a constant threshold value of 0.05 for our metrics (Acc., FID and MMD) and human perceptual evaluations. However, for our visual results we do a grid search and choose the best threshold value. Figure 2, shows the visual results of different thresholds. It can be seen that different shapes require different threshold which depends on the category and local details of the shape. We believe that our metrics and human evaluation results can be further improved if a better technique is discovered for threshold tuning.

6. Out of Distribution Generation

We also conduct experiments to see if the network can generate shapes based on text queries which are out of distribution from its training data. The results are shown in Figure 3. It can be seen from the results that the method tries to generate the desired shape based on its training dataset. We believe extending our method to generalize on out of distribution samples might be interesting avenue to explore for future work.

7. Visual Results for Different Prefixes

In Figure 4, we show results for different prefixes. They indicate that for different prefixes there are small variations

Airplane	Bench	Cabinet	Car	Chair	Monitor	Lamp	Loudspeaker	Gun	Sofa	Table	Phone	Boat
95.00	64.29	87.50	96.88	96.15	92.86	93.33	45.45	92.86	89.47	75.00	60.00	61.11

Table 3. Category-wise accuracy results

.

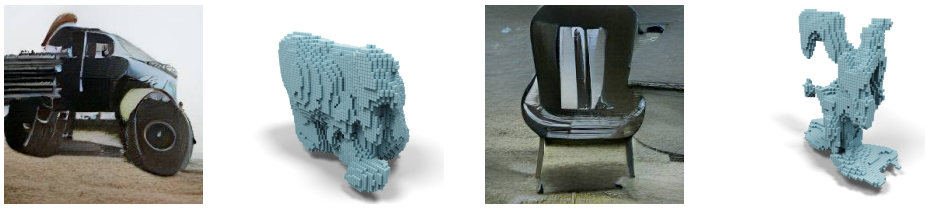

Figure 1. Text2Img+Img2Shape baseline intermediate and final results: “a monster truck”, “a round chair”.

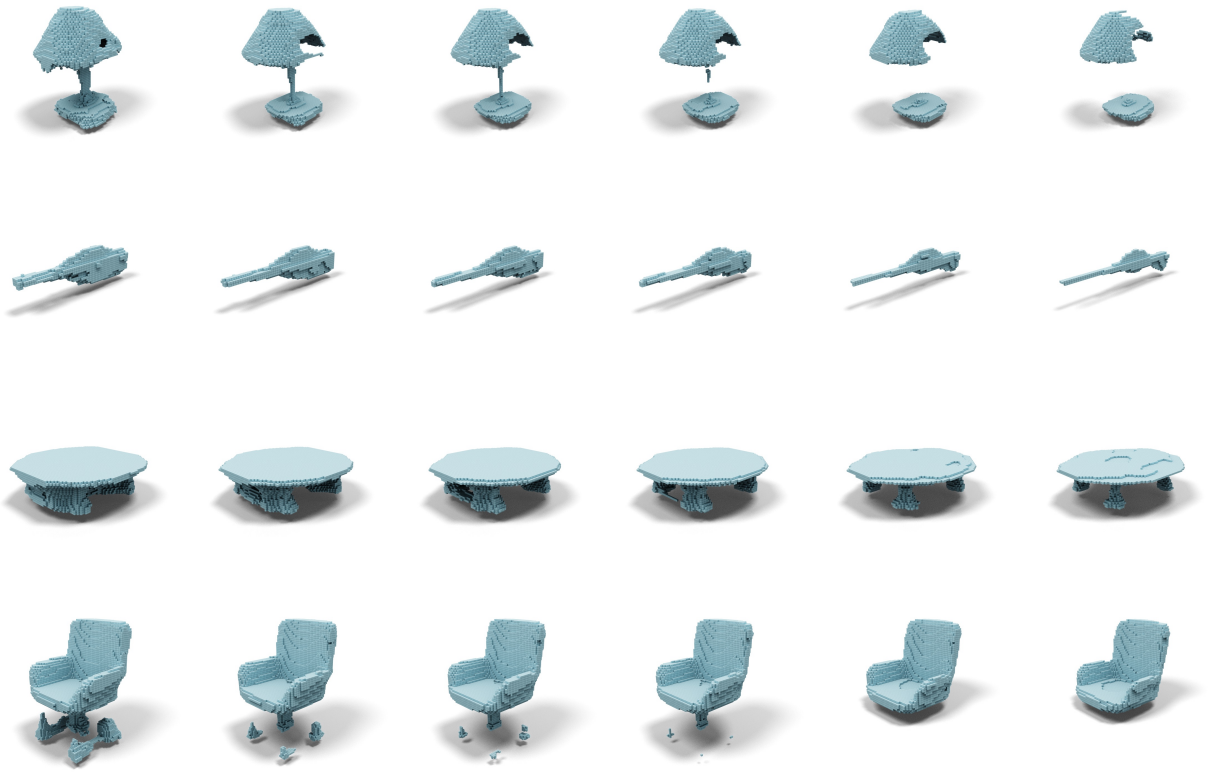

Figure 2. Effect of different thresholds for text: ”a lamp”, ”a sniper rifle”, ”a round table” and ”a swivel chair”.

in generated shape. Moreover, in some prefixes such as “a rendering of”, the visual results are worse. It would be interesting to investigate other prompts or do prompt tuning as future work.

8. Visual Results for more Descriptive Texts

We show additional results using text queries that are longer and more descriptive in Figure 13. It can be seen that CLIP-Forge is able to capture certain shape-related attributes. Non-shape related descriptions such as color is

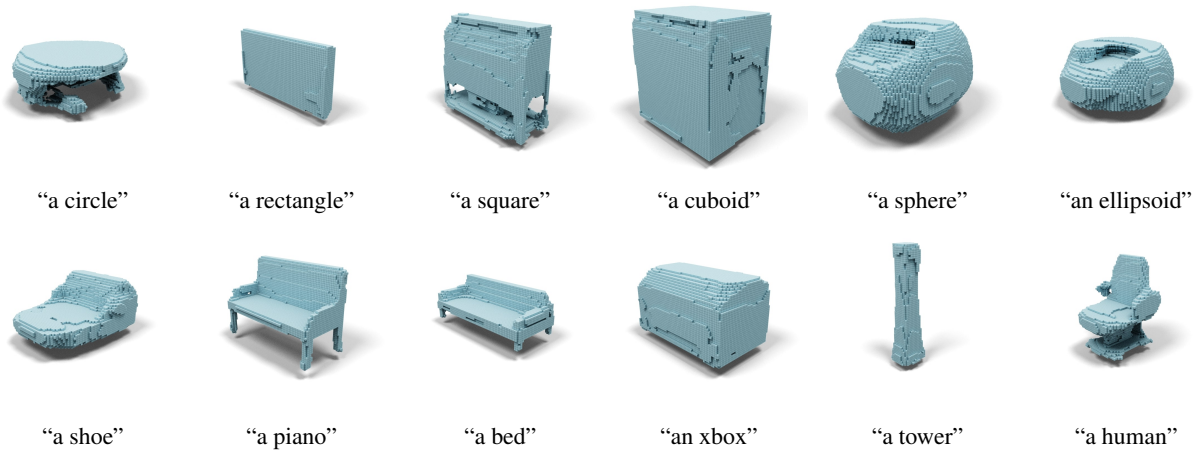

Figure 3. Results using text queries that are semantically outside the dataset.

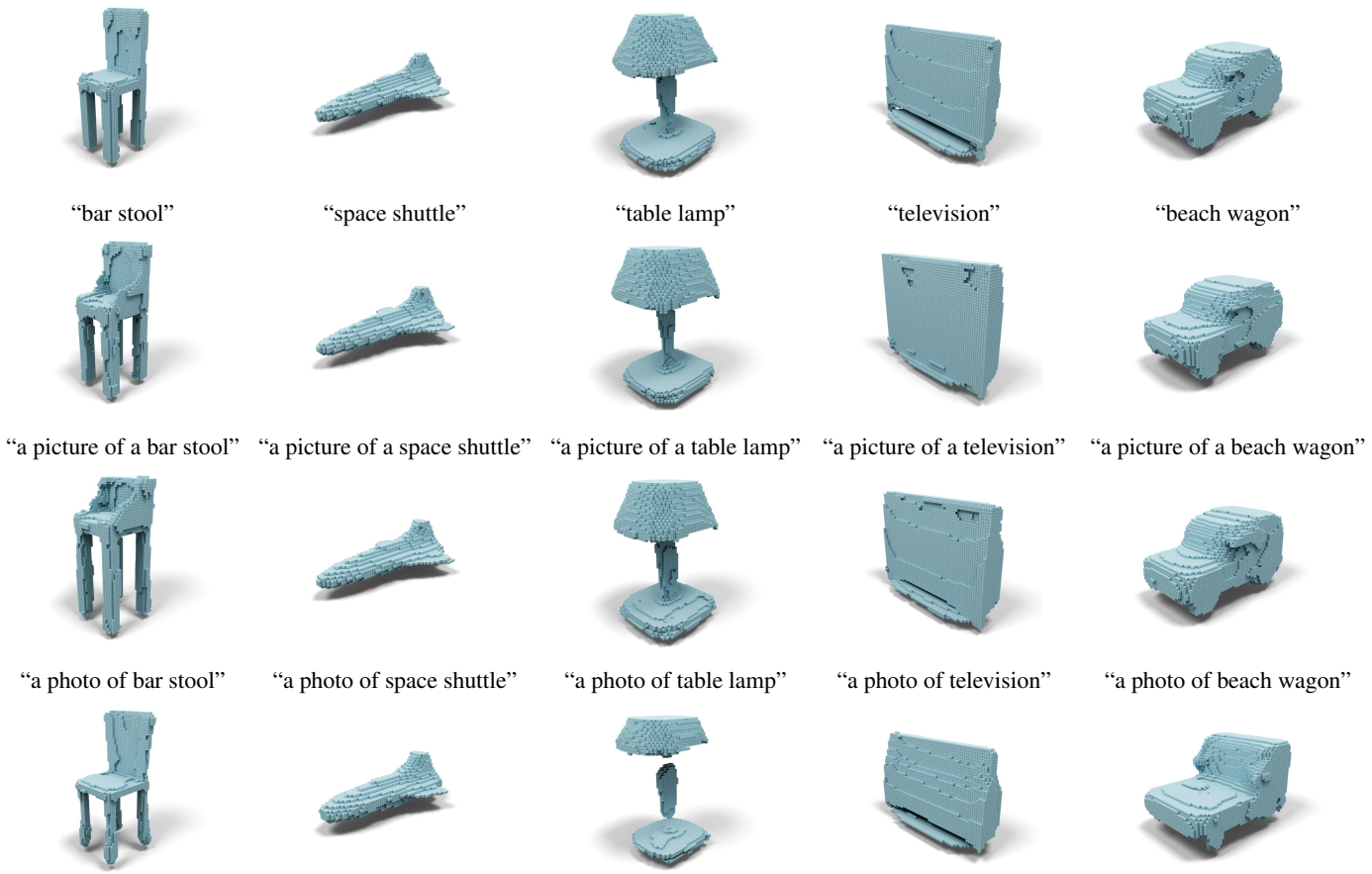

Figure 4. Results of varying the prefix for a given text query.

not captured but could potentially bias the generation. We believe that combining our method with semi-supervised learning can enable more fine control of shape generation using text.

9. Additional Qualitative Results

In this part, we show more visual results for shape generation conditioned with text based on sub-category (Figure 5 and Figure 6), synonyms (Figure 7), shape attributes (Figure 8 and Figure 9) and common names (Figure 10). Moreover, we also show more visuals for text based multiple shape generation (Figure 11) and interpolation (Figure 12). It can be seen from all these results that our method is good at generating 3D shapes based on text queries. However, in same cases for example "a swivel chair", it cannot construct all the details. Furthermore, on some sub-categories such as "an operating table" it cannot generate accurate shapes.

10. Human Perceptual Evaluation

In the human perceptual evaluation described in section 4.3 of the main paper, crowd workers recruited through Amazon Mechanical Turk [7] were shown pairs of images, one generated from the ShapeNet(v2) category name (see the first column of Figure 15) and the other from a detailed text prompt containing either subcategory or attribute information. The crowd workers were shown the detailed text prompt and asked to identify which of the two images it best describes. Nine crowd workers viewed each image pair and we record the number of times the model from the detailed text prompt is selected. For each detailed text prompt, this gives us a score from 0 to 9 indicating how effectively Clip-Forge can produce distinctive shapes which differ from the ShapeNet categories in a way which humans find semantically meaningful. In Table 1 the human evaluation scores are shown as colors for each query text for which the evaluation was conducted. Figure 15 shows a few examples in more detail. The second column of Figure 15 shows text prompts which produced distinctive shapes and the third column shows cases where the shapes were not as easily identified based on the text. We see that when the prompt elicited a very distinctive shape ('A monster truck", "A fighter plane") a high fraction of the human raters were able to identify the correct model. In some cases the low score reflects a lack of resolution (for example "A swivel chair", "A billiard table" and "A seaplane"). In the case of "A wheelchair", Clip-Forge was unable to generate round wheels, but as the bottom of the legs were joined up this gave enough of an impression of wheels for humans to select the model. In the case of "A muscle car" Clip-Forge attempted to create the shape of a low form of a sports car, however the shape was not far enough from the generic car for the crowd workers to select it.

References

[1] Blender Voxel Rendering. https://blender.stackexchange.com/questions/115847/algorithm-for-turning-voxels-into-triangle-mesh. 1

[2] Kevin Chen, Christopher B Choy, Manolis Savva, Angel X Chang, Thomas Funkhouser, and Silvio Savarese. Text2shape: Generating shapes from natural language by learning joint embeddings. In *Asian Conference on Computer Vision*, pages 100–116. Springer, 2018. 1, 2

[3] 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*, pages 628–644. Springer, 2016. 1, 2

[4] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real nvp. *arXiv preprint arXiv:1605.08803*, 2016. 1

[5] Christiane Fellbaum. *WordNet: An Electronic Lexical Database*. The MIT Press, 05 1998. 1

[6] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019. 1

[7] Abhishek Mishra. Machine learning in the aws cloud: Add intelligence to applications with amazon sagemaker and amazon rekognition, 2019. 5

[8] George Papamakarios, Theo Pavlakou, and Iain Murray. Masked autoregressive flow for density estimation. *arXiv preprint arXiv:1705.07057*, 2017. 1

[9] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision, 2021. 1

[10] Yaoqing Yang, Chen Feng, Yiru Shen, and Dong Tian. Foldingnet: Point cloud auto-encoder via deep grid deformation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 206–215, 2018. 1

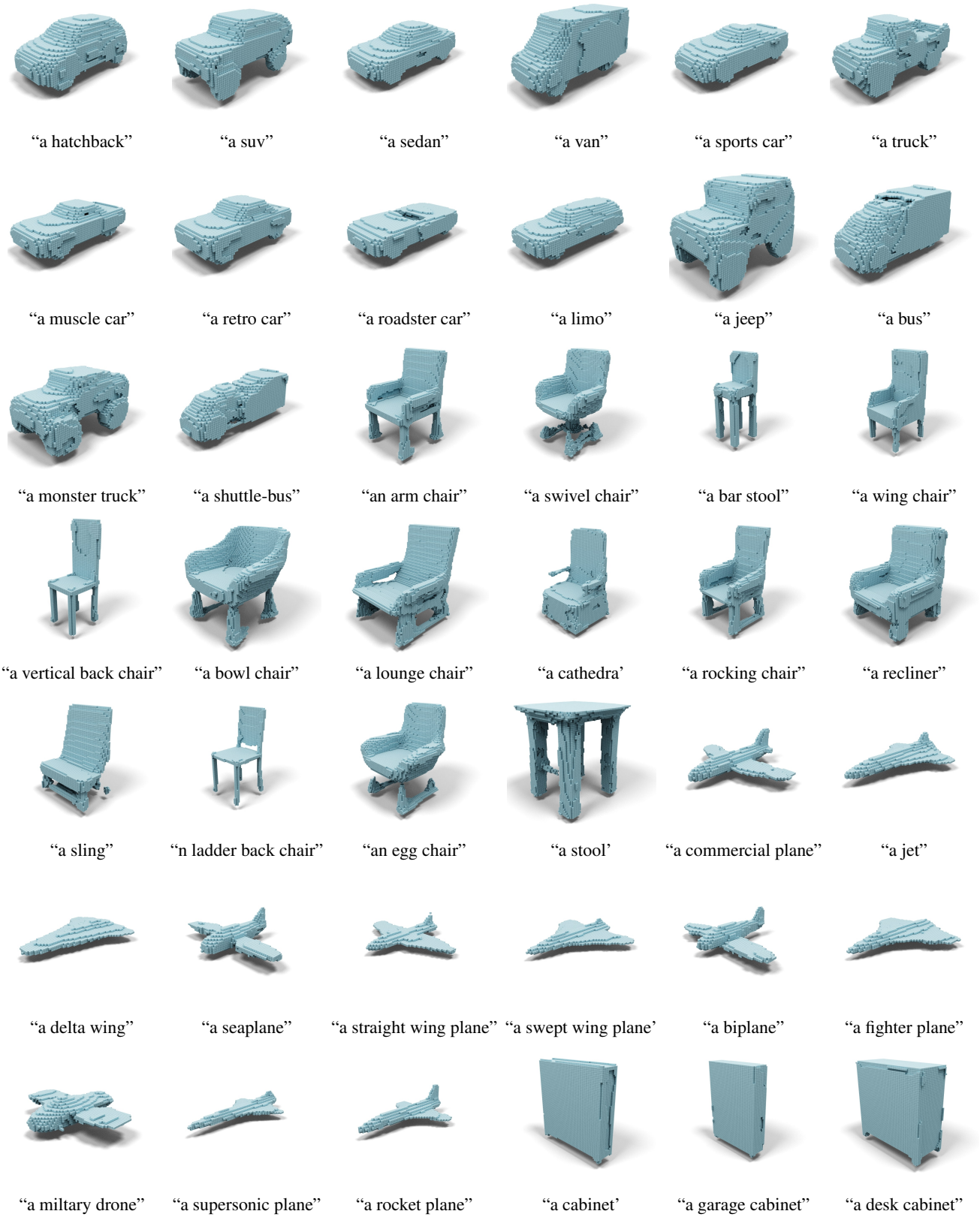

Figure 5. Additional shape generation results using sub-category text queries of CLIP-Forge.

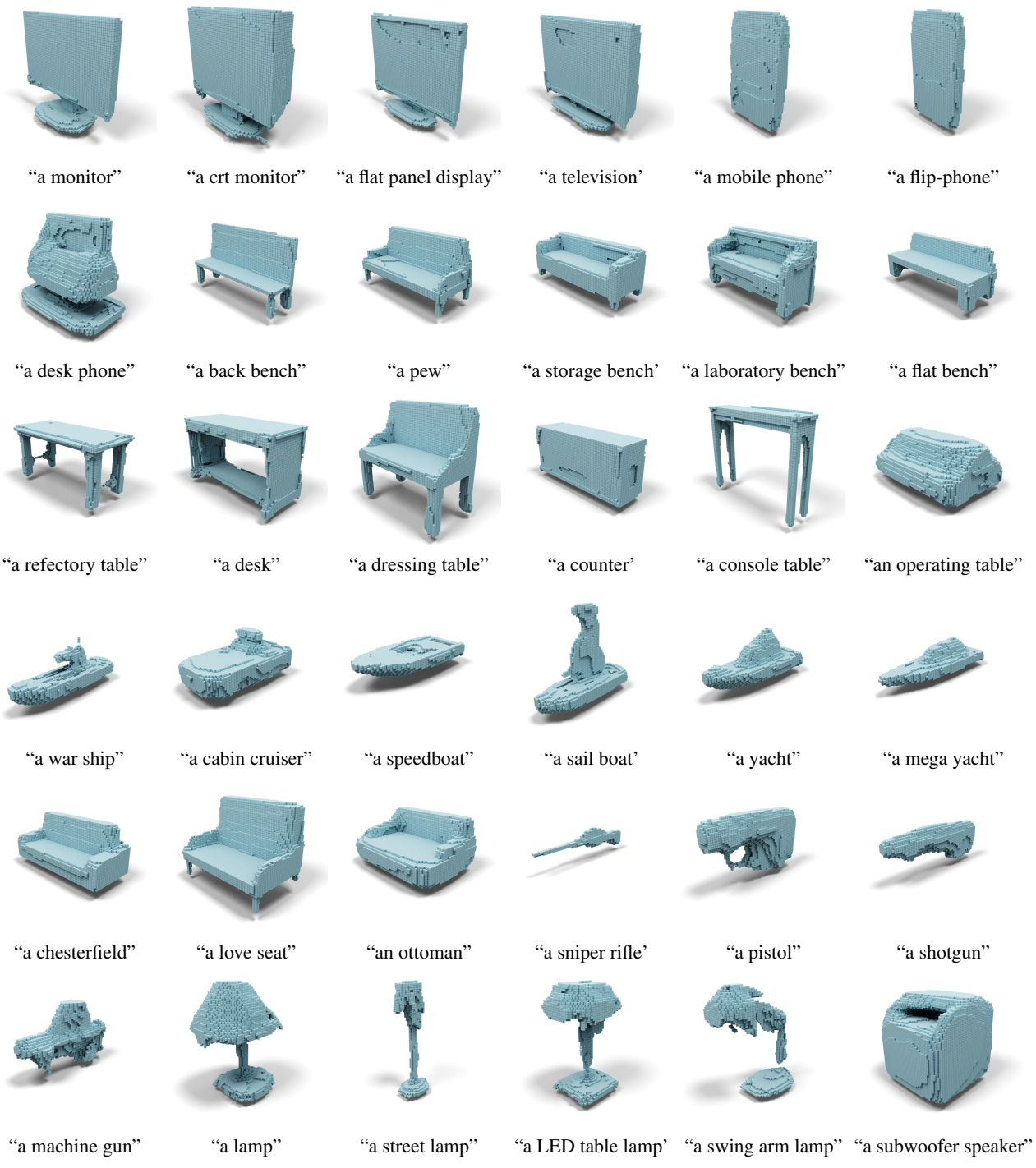

Figure 6. Additional shape generation results using sub-category text queries of CLIP-Forge (continued).

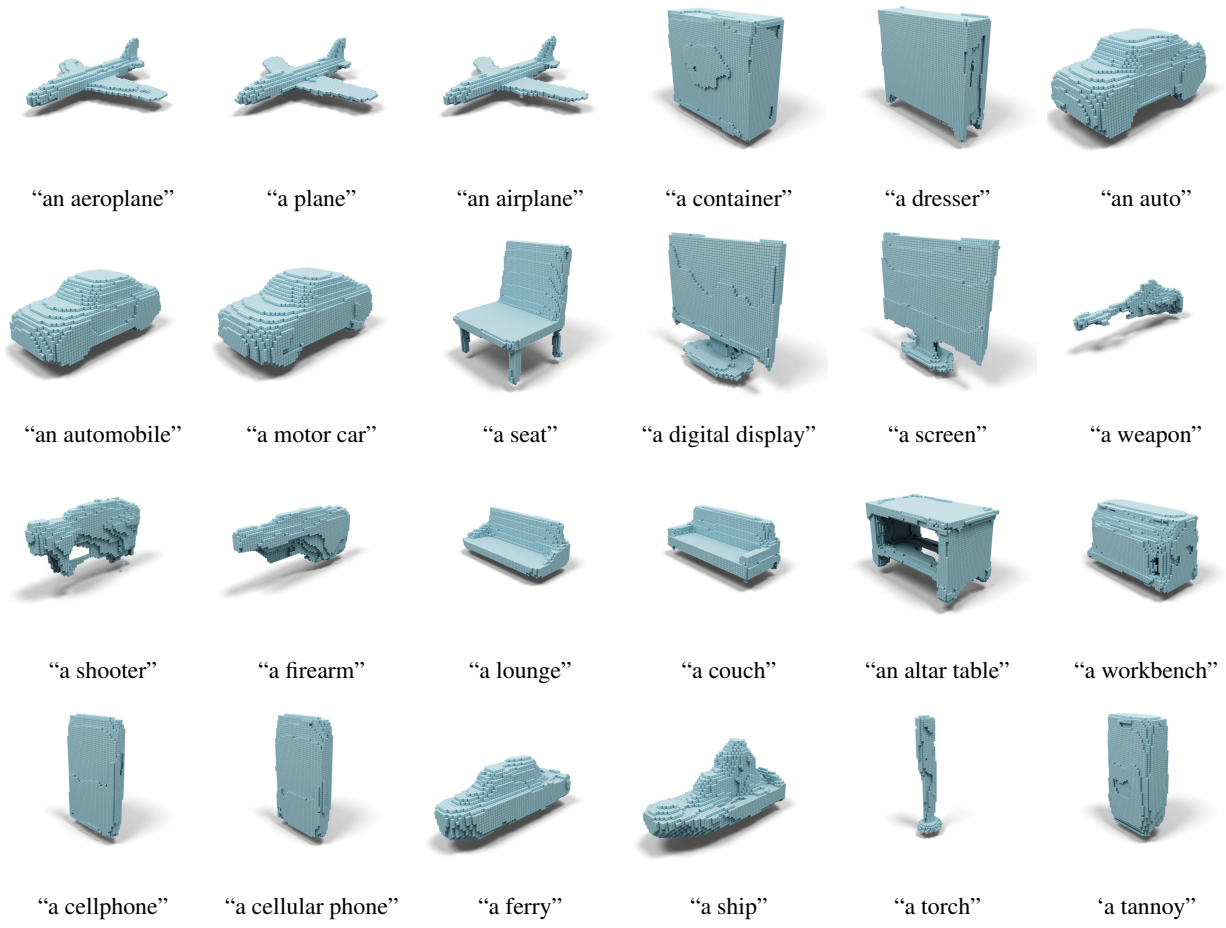

Figure 7. Additional shape generation results using category and synonyms based text queries of CLIP-Forge.

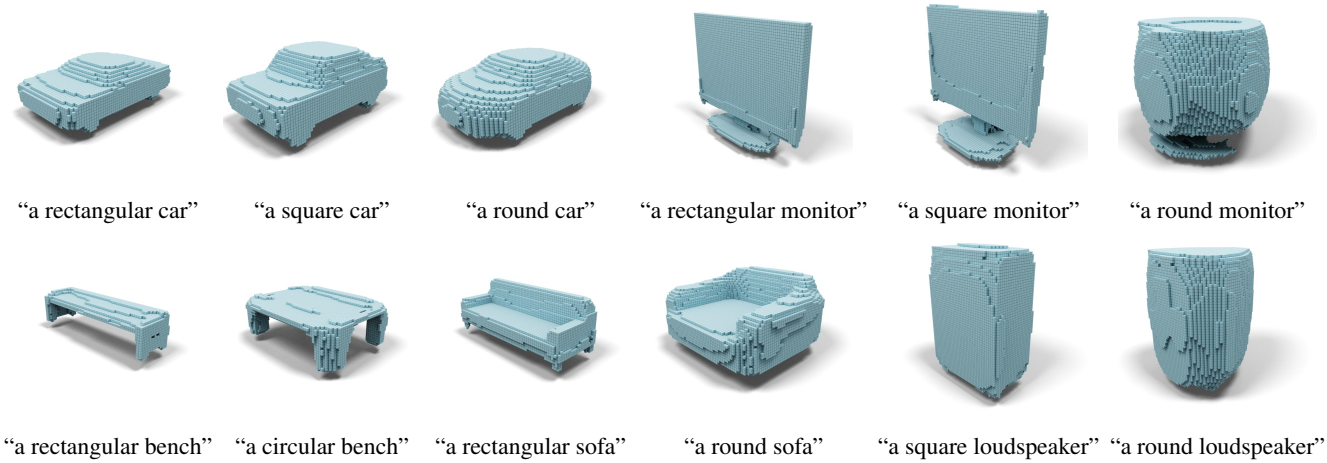

Figure 8. Additional shape generation results using attribute-based text queries of CLIP-Forge.

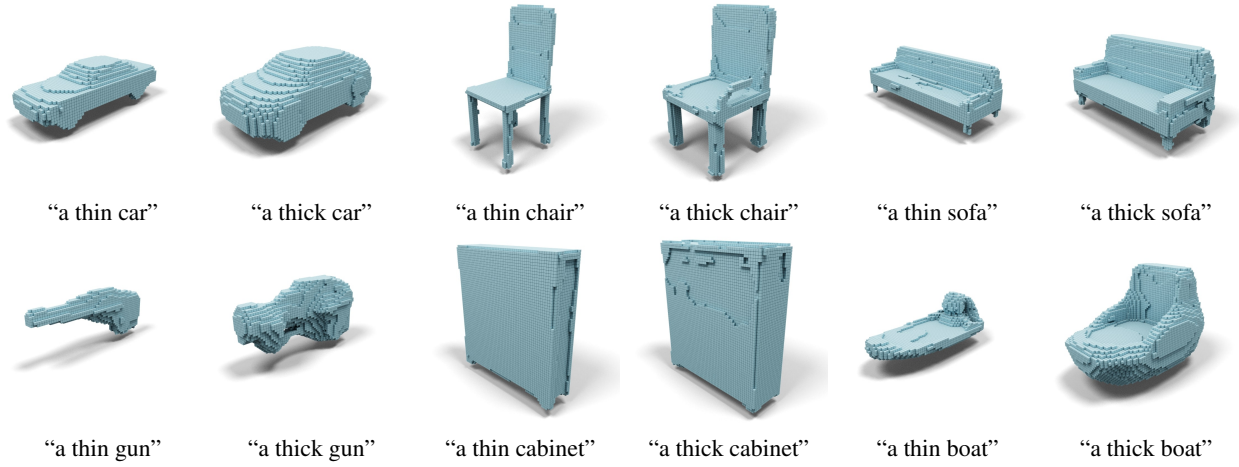

Figure 9. Additional shape generation results using attribute-based text queries of CLIP-Forge (continued).

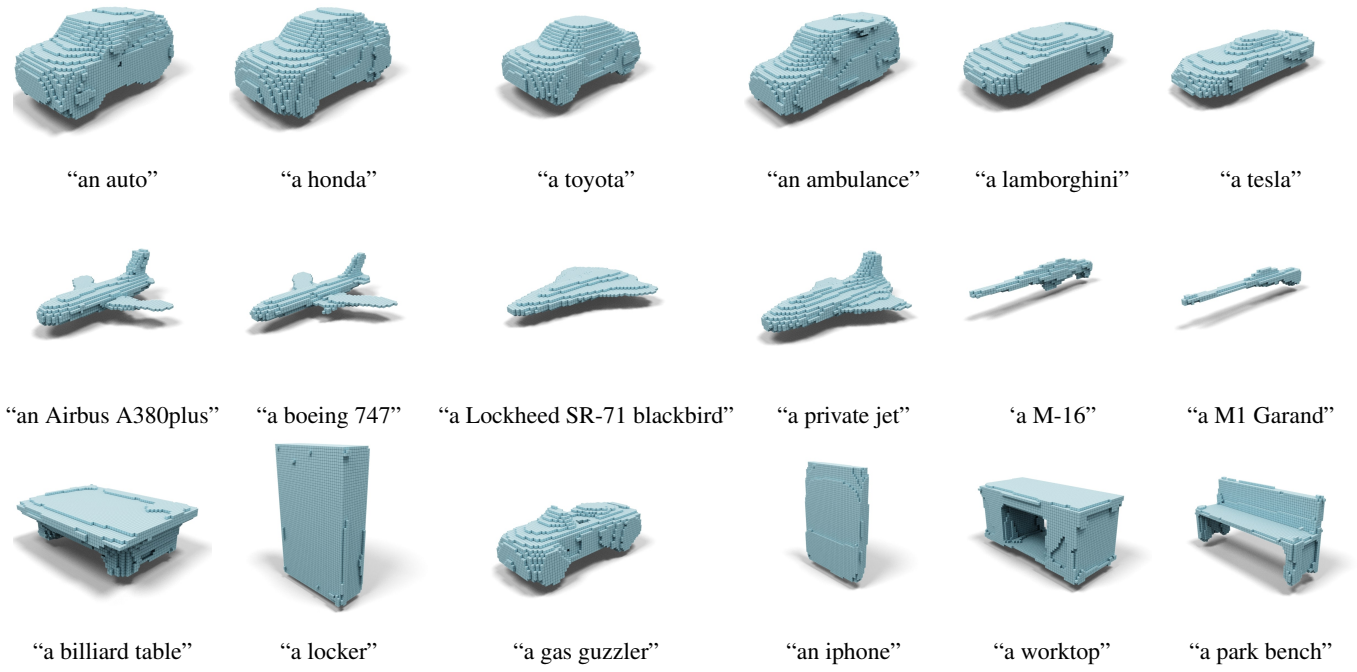

Figure 10. Additional shape generation results using common name text queries of CLIP-Forge.

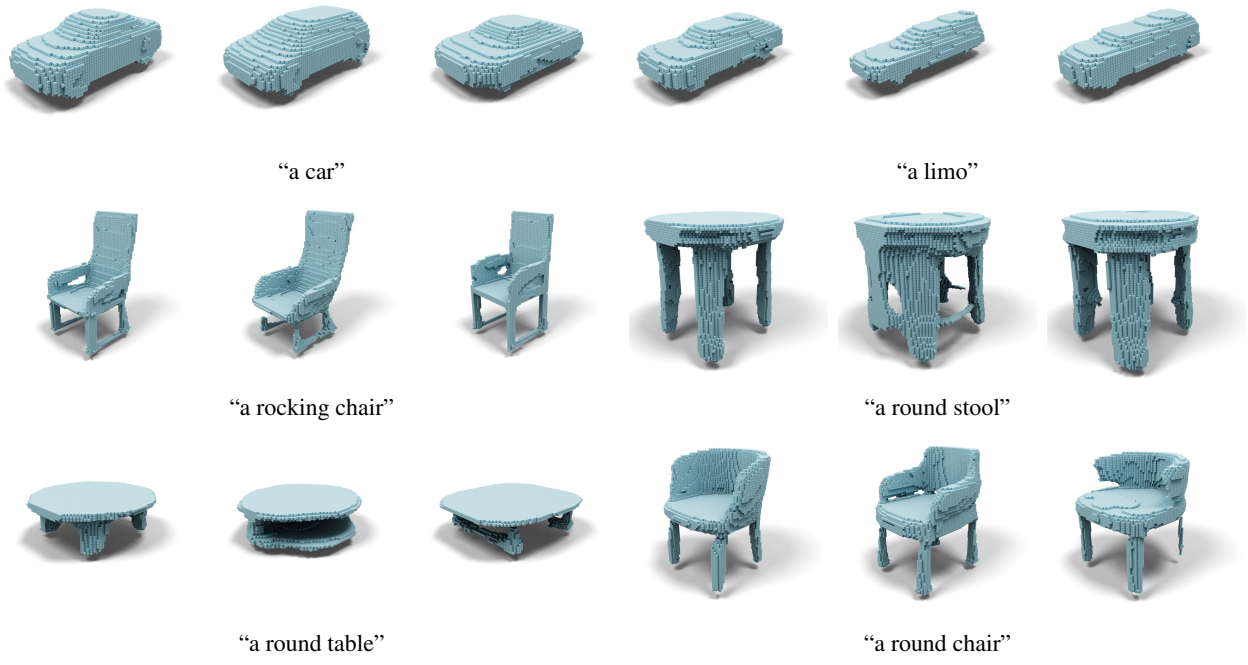

Figure 11. Additional results for multiple shapes generation.

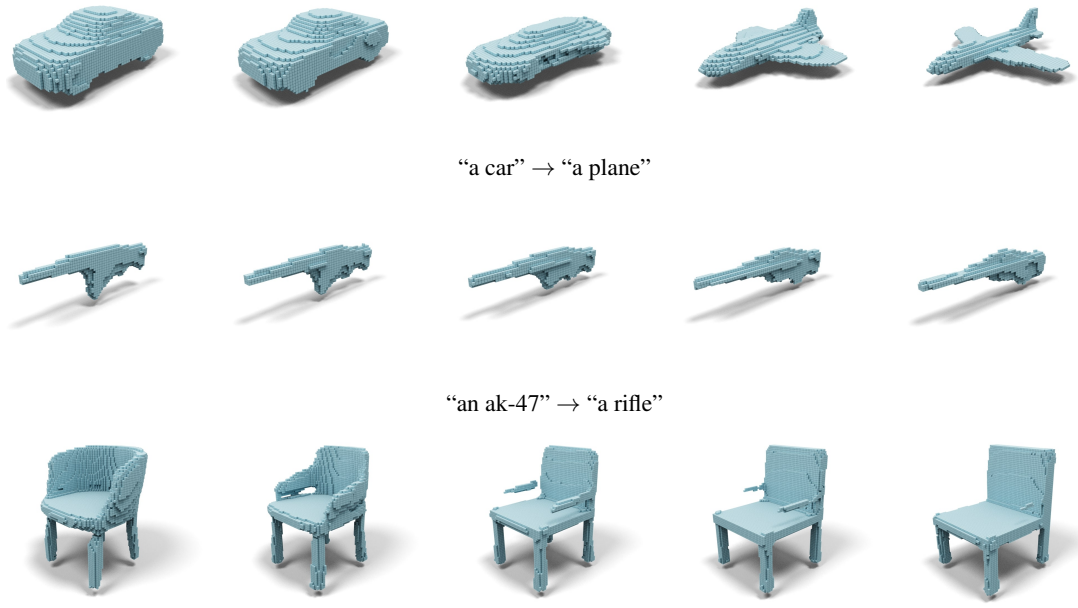

Figure 12. Additional Interpolation results between two text queries.

Figure 13. Descriptive CLIP-Forge results: “a brown table with four legs”, “an armless chair with curved rectangular back”, “an armed chair with curved rectangular back”, “big sofa having two legs of black color, backrest, armrest, and sitting of black color”.

CMA-T2S supervised-T2S supervised-SN13

Figure 14. Qualitative results for supervised baselines using “a sports car” and “a vertical back chair”.

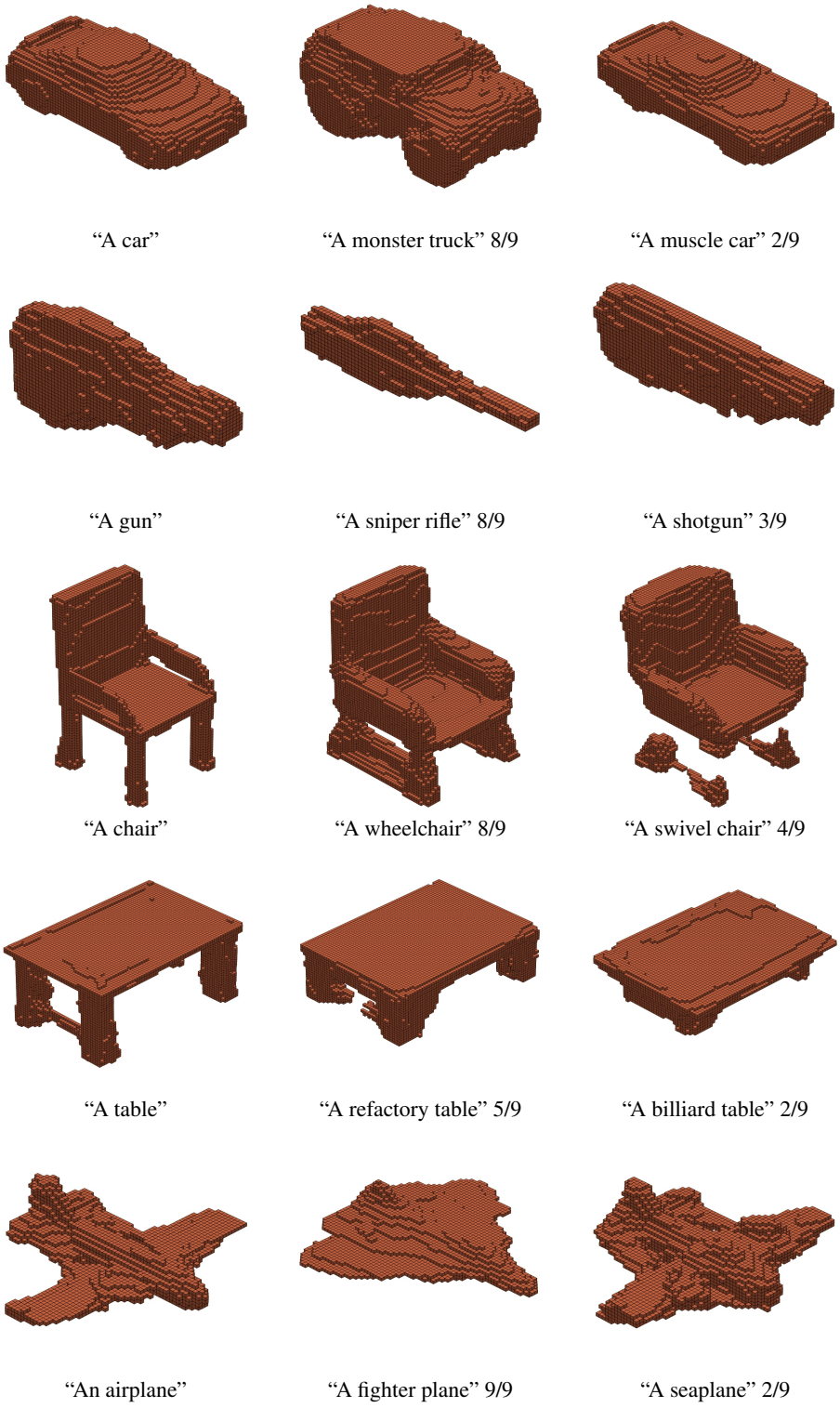

Figure 15. Images shown to the crowd workers in the human evaluation. The first column shows results generated using the ShapeNet(v2) category name. The second column shows examples of models which the crowd workers found easiest to identify based on the detailed text prompt and the third column shows the hardest. The fraction of the nine crowd workers who chose the model generated with the detailed text prompt is also shown.