Text to shape generation with finer control

Akshai Srinivasan Sivakumar and Oliver van Kaick

Carleton University, Ottawa, Ontario, Canada
akshaisrinivasansiva@cmail.carleton.ca, oliver.vankaick@carleton.ca

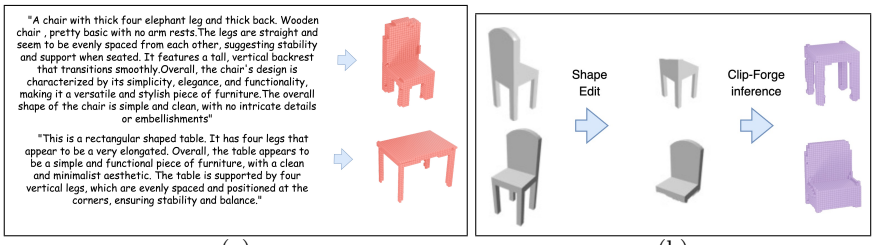
Abstract. Generating 3D shapes from text is challenging due to the limited availability of text-to-3D datasets. Existing methods train generative models by conditioning on multi-view CLIP image embeddings of shapes during training and use CLIP text embeddings for inference. However, due to the nature of the CLIP's training dataset, the supported text descriptions lack fine details. In addition, the CLIP text encoder has a limited token length, which restricts its ability to support rich captions. In this paper, we explore supervised fine-tuning of CLIP-based shape generation with richer captions that enhance control over the generation. We propose a fine-tuning approach that leverages diffusion models to map multiple text embeddings into the CLIP image embedding space, enabling more precise shape generation on the target dataset. The fine-tuned model adapts to the target dataset while remaining effective on open-set captions. We evaluate our method on chairs and tables from ShapeNet and test open-set prompts within these categories.

Keywords: Text-to-3D generation, CLIP-Forge, Diffusion models, 3D shape synthesis, Latent space mapping, Voxel-based representation

1 Introduction

Text-to-3D shape generation is an important research problem with a variety of potential applications, such as content generation for games, virtual reality (VR), and augmented reality (AR). However, unlike text-to-image generation, text-to-shape generation faces the challenge of limited availability of datasets consisting of shapes paired with textual descriptions. To address the data scarcity, CLIP-Forge [26] introduces the use of the CLIP [23] text-image joint embedding space to enable zero-shot 3D shape generation. Follow-up work such as CLIP-Sculptor [27] improves the speed of generation and the resolution of the shapes synthesized via CLIP, enabling the generation of higher-quality 3D shapes from various categories. However, one limitation of these zero-shot CLIP-based approaches is that only short textual descriptions can be input, since the CLIP text encoder has a capacity of 77 tokens. In addition, due to the nature of CLIP's training dataset, the supported shape descriptions lack fine details. Thus, the captions that can effectively guide shape generation are succinct, generic descriptions such as "a chair" or "a square table".

One possible approach for allowing more detailed captions for shape generation is to embed shape descriptions using a text encoder that allows for longer



(a) (b)

Fig. 1. (a) We map rich text descriptions of chairs and tables into CLIP image embeddings to generate 3D shapes that better conform to the descriptions. (b) Shape generation conditioned on CLIP image embeddings. Left: original images. Middle: edited images with parts removed. Right: shapes generated via Clip-Forge [27] reflecting the edits.

captions and then align the textual embeddings with the embeddings of 3D shapes. However, CLIP-Forge [26] showed the limited performance of this approach, where learning such an alignment is not easily achievable with training on only a few thousand examples.

In this work, we aim to enable shape generation from long and detailed captions that describe overall shape structure as well as part-level details and their spatial relationships. To this end, we propose an adaptation framework that maps multiple rich text embeddings into a single CLIP image embedding using a diffusion-based module. The model is trained on a dataset comprising both global captions generated by vision-language models and fine-grained, part-aware descriptions derived from shape segmentations. We integrate this adapter with Clip-Forge for shape generation (Fig. 1(a)).

Our method extends the core idea from DALL·E 2 [24], which involves mapping text embeddings to image embeddings to enhance image generation quality, into the domain of 3D shape synthesis. We observe that even minor image edits (e.g., masking chair parts) lead to significant changes in shapes generated via CLIP-Forge (Fig. 1(b)), emphasizing the importance of image embeddings. Motivated by this observation and the modality gap between text and image embeddings [12], our learned mapping improves control over the generated shapes within the trained categories while mitigating CLIP's token length limitations. We focus our experiments on chairs and tables from the ShapeNet dataset, given their diversity, and compare our approach against CLIP-Forge to demonstrate the effectiveness of our method in improving text-to-shape generation. To accommodate longer captions, we explore two approaches: (i) A multi-caption architecture with the CLIP text encoder, to overcome CLIP's limited token capacity. (ii) The language feature extractor BGE [30] that supports longer captions.

To summarize, our contributions are as follows:

1. We introduce a systematic approach for generating more descriptive captions for shapes by utilizing the PartNet [20] instance segmentation data and vision-language models such as Llava[14], Llama 3.2 [5] and Ovis [16].

2. We introduce a diffusion-based multi-caption mapping module for fine grained shape generation, learning the mapping from longer, more specific captions to the CLIP image space and utilizing only a small amount of training data.
3. We perform qualitative and quantitative evaluations, analyses, and a user study, on a dataset of chairs and tables to demonstrate the improvements provided by our approach.

2 Related work

In this section, we discuss the previous work most related to our method.

Few-shot text-to-image generation models. Recent advances in few-shot text-to-image generation address challenges in synthesis quality and diversity with minimal data. Lafite2 [32] generates pseudo-text features from image-only datasets, enabling effective pre-training for GANs and diffusion models. DomainGallery [4] fine-tunes Stable Diffusion using attribute-centric techniques to improve domain-specific generation. DreamBooth [25] fine-tunes diffusion models on 3–5 images to create personalized outputs while preserving key subject features. Our approach is particularly inspired by these prior studies in the image domain that explored fine-tuning diffusion models under limited data conditions.

NERF-based text-to-shape models. Neural Radiance Fields (NeRF) [19] serve as a key 3D representation, with several text-to-3D extensions. Dream Fields [10] optimizes NeRF via CLIP-guided multi-view consistency and transmittance regularization. Dream Fusion [22] replaces CLIP with a 2D diffusion-based SDS loss for better quality and alignment. In our work, we focus on volumetric shape generation using voxels, following CLIP Forge for compatibility. However, our adapter is decoder-agnostic and could be extended to use other shape representations such as implicit functions or meshes in future work.

Text-to-shape generation. Clip-Forge [26] generates shapes from text via CLIP's image-language alignment, using latent voxel grids in a two-stage process without paired text-shape data. Clip-Sculptor [27] refines shapes through coarse alignment and detailed enhancement. ShapeCrafter [6] takes a recursive approach, evolving shapes iteratively based on descriptive phrases. Liu et al. [15] decouple shape and color prediction and uses word-level alignment to guide shape generation. CLIP-Mesh [21] also uses image priors for shape generation. In contrast, our method enables the use of longer, richer captions for shape generation.

Visual guidance for image and shape generation. Methods like Spice·E [28] and ControlNet [31] enhance diffusion models using structural inputs (e.g., sketches, edges, or auxiliary shapes), enabling precise and controllable generation. While effective, these approaches rely on visual conditions, whereas we focus on richer textual guidance.

3 Approach

Our approach aims to effectively fine-tune text-to-3D generative models, especially when training data is scarce. Traditional methods pretrain on large text–3D shape corpora and fine-tune on smaller target datasets, but face two key challenges: the limited availability of text–3D shape data and the 77-token limit of the CLIP text encoder, which restricts detailed shape descriptions. To overcome these issues, we propose a two-step adaptation method.

In the first step, we pretrain a diffusion model to map multiple text embeddings such as CLIP [23] or richer ones like BGE [30], into the CLIP image embedding space. For CLIP, this circumvents the token limit by splitting long captions into segments, each producing a separate text embedding. For BGE, it enables mapping longer descriptions directly into CLIP image space. This pretraining uses a dataset of rich textual captions paired with object and scene images, teaching the model to translate these embeddings into corresponding CLIP image embeddings. In the second step, we fine-tune the diffusion model on a target dataset of long caption embeddings and multi-view shape image embeddings. Each 3D shape is represented by multi-view images, with CLIP image embeddings capturing comprehensive visual details. The fine-tuning aligns rich textual embeddings with these shape embeddings, allowing accurate mapping from text to 3D shapes via Clip-Forge's image embedding space. We describe each step of our method in more detail as follows.

3.1 Pretraining of the mapping model

The pretraining phase utilizes a large collection of text-image pairs based on the MSCOCO dataset [13] to establish a mapping between textual descriptions and visual embeddings. Since the original text labels accompanying these images lack detailed content, we employed the Llava [14] model to generate richer captions for the images. For each image, five sentences were randomly sampled from these enriched captions, and their corresponding text embeddings were created using the CLIP text encoder for each sentence or a rich BGE embedding representing all the sentences.

The text embeddings are then mapped to the CLIP image embedding space using a diffusion model. The diffusion model is designed to learn how to align detailed textual descriptions with CLIP image embeddings. For this, we utilized the standard Denoising Diffusion Probabilistic Model (DDPM) [8] implementation. The diffusion model is based on a one-dimensional U-Net architecture, which begins with a noised CLIP image latent and progressively reconstructs its unnoised version over several timesteps while conditioning on the text embeddings. We adopt classifier-free guidance [9] for conditioning. The diffusion loss L_D is a Mean Squared Error (MSE) between the predicted noise and the actual noise introduced.

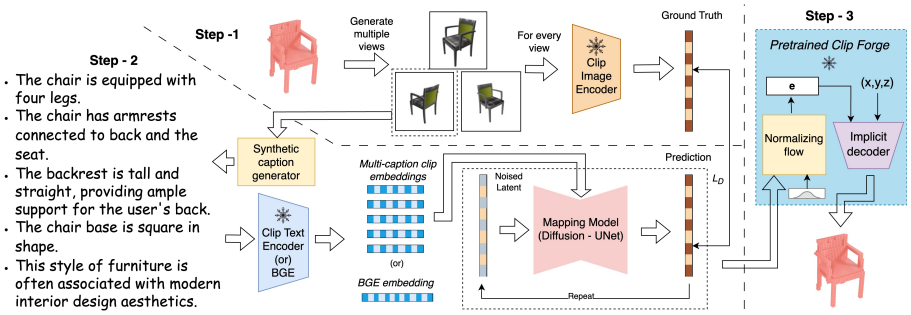


Fig. 2. Fine-tuning of the pretrained mapping model on the target dataset. Five sampled captions about a particular shape are first converted to text embeddings and then diffused to the CLIP image space with our mapping module. The generated CLIP image latent is then mapped to the 3D voxel space using CLIP-Forge [26]'s pretrained normalizing flow and voxel decoder.

3.2 Fine-tuning to the target dataset

After the pretraining phase, we adapt the mapping model to the target dataset. Since common shape datasets such as ShapeNet [2] lack rich textual descriptions of the shapes, we discuss as follows how we automatically created a synthetic dataset of shape descriptions. Then, we fine-tune the adapter model with the synthetic data.

Synthetic dataset of shape descriptions We create synthetic textual descriptions in three ways:

- **Caption generation using VLMs**: We employ Vision Language Models (VLMs) to obtain highly descriptive captions using multi-view shape images. We use the LLaMA3.2 [5], Ovis [16] and Llava [14] models.
- **Text2Shape dataset**: We use textual data from the Text2Shape dataset [11], which includes five different captions per object for chairs and tables.
- **PartNet-based synthetic caption generation**: We use the hierarchical part information available in the PartNet [20] dataset to create captions that capture relationships between parts. The captions are created by applying relationship templates to the part information. Specifically, based on the part information, we can identify the relative positions of parts, connections between parts, and summarize the hierarchy of parts. We generate three types of captions: (1) **Positional captions**: These captions describe the relative location of each part in relation to others, e.g, "The chair base is below the back and above the legs"; (2) **Connectivity captions**: These captions explain how different parts are connected to each other, e.g., "The seat is connected to the legs and the back of the chair"; (3) **Aggregate captions**: These captions describe how parts are compositionally grouped or encompassed, e.g., "The chair base is composed of four legs".

Fine-tuning of the mapping model Steps 1 and 2 in Fig. 2 illustrate how we fine-tune the adapter model using the synthetic data, so that it learns a mapping from text to image embeddings for the target dataset.

Specifically, we take the pretrained diffusion model from Section 3.1 and perform a fine-tuning on the target shape dataset. As shown in the training architecture of Fig. 2, the images of the shapes are sampled from multiple views of the 3D shapes and converted to image embeddings using CLIP. We then randomly select sentences from the union of all datasets of shape descriptions

during sampling. This sampling ensures a comprehensive representation of the various caption types. Mapping the embeddings of detailed textual descriptions to the image embedding space allows the model to effectively learn the alignment between textual and 3D shape views.

3.3 Inference with the mapping model

Steps 2 and 3 in Fig. 2 illustrate how we generate shapes from textual descriptions. Specifically, during the inference phase, a text query is split into 5 captions by dividing it at each period (`"."`). The captions are then embedded using the CLIP text encoder or the BGE model. After that, our diffusion adapter maps the embeddings of the five captions to the corresponding CLIP image embedding.

Moreover, following [26], a vector is sampled from a normal distribution and passed through the reverse path of the trained flow model, conditioned on the image embedding, to generate a shape embedding. Finally, the shape embedding is transformed into a 3D shape using the trained voxel decoder.

4 Experimental evaluation

In this section, we first describe the experimental setup and implementation details of our method, followed by a discussion of the qualitative and quantitative results.

Cap. Type	**Model**	**Eval**	**FID ↓**	**MMD (CD) ↓**	**COV (CD) ↑**	**CD ↓**	**EMD ↓**	**LFD ↓**
Short	Clip-Forge	SCap	2035.08	**2.344**	0.1692	**20.832**	**4.458**	6914.23
	MShortened	SCap	**951.43**	2.397	**0.2662**	22.033	4.506	**6314.65**
Long	*MRich-WoP*	RCap	1490.18	2.628	0.2259	26.551	4.900	6783.47
	MRich	RCap	**766.19**	2.591	**0.2799**	**21.817**	**4.441**	**6287.09**
	MBGE	RCap	1042.71	**2.580**	0.2675	21.930	4.468	6316.31
Longer	*MBGE+*	RCap+	848.94	2.650	0.2910	19.627	4.255	6062.44

Table 1. Quantitative evaluation of our adaptation method and comparison to Clip-Forge on the ShapeNet [2] test set. The best metric for each caption group of short or long captions is highlighted in bold.

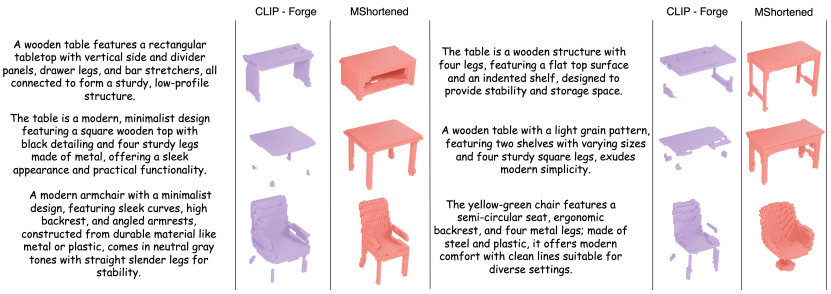


Fig. 3. Comparison of shapes generated from the test descriptions of the SCap dataset for Clip-Forge and our MShortened model (please zoom in to see the details)

.

4.1 Datasets

We use the following datasets in our experiments:

- **Pretraining dataset:** The MSCOCO dataset [13] was used for pretraining the adapter. It consists of a total of 591,743 image-text pairs. We utilized 95% of the data for training and the remaining 5% for testing.
- **Adaptation dataset:** For all experiments, we used the ShapeNetV2 dataset [2], focusing on chairs and tables so that we are able to compare to previous work. We utilized a processed version containing rendered images and voxel grids [18, 3], with train/validation/test splits of 4,745/676/1,354 for chairs and 5,957/849/1,700 for tables, totaling 10,692 training, 1,525 validation, and 3,054 test samples. For captions, we employed three variants: *RCap*, which includes five sampled captions per shape from LLava, Positional, Connection, Aggregate, and Text2Shape [11]; *RCap+*, an extended version of *RCap* incorporating LLama3.2 and Ovis captions with a total of 10 sampled captions per shape; and *SCap*, a single summarized caption generated by condensing all RCap sources using Qwen2.5:7b-instruct [29]. *RCap/RCap+* combine rule-based synthetic captions with more natural VLM captions. Further improving caption naturalness is left for future work.
- **Evaluation dataset**: We evaluate our trained models on two datasets. One is the test split from the adaptation dataset discussed above, which gives an idea about the level of control that the captions are contributing to the generated shape. The second dataset are the 44 text queries provided by Clip-Forge [26] for chairs and tables. This dataset provides insights into the model's performance when handling open-set queries within these two categories. Cross-category evaluation is left for future work.

4.2 Implementation details and model variations

We use the "ViT-B/32" variant of CLIP for all our experiments, consistent with the setup used in Clip-Forge. For the mapping model, we use a standard DDPM

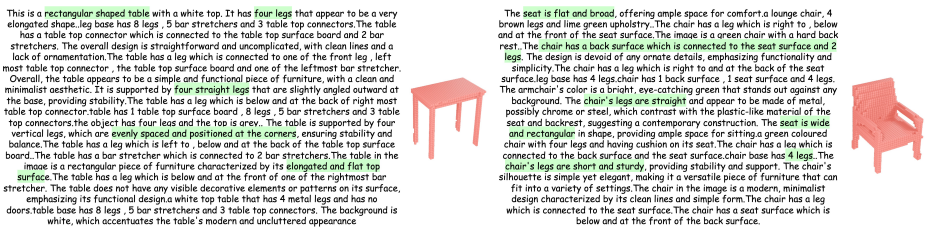


Fig. 4. Visualization of shapes generated from richer, longer captions in the RCap+ test set with our BGE+ model (please zoom in to see the details).

[8] with a linear beta schedule and 1000 timesteps. The backbone is a UNet1D model with a single input channel, 512 embedding dimensions, and conditional embeddings of 512×5 for CLIP text embeddings or 1024 for BGE embeddings. For pretraining, the model was trained for 50 epochs to establish a robust initial representation. During adaptation, the number of training epochs was set to 400.

We train five variants of the adapter model: A) **MRich**: A model trained on rich multi-CLIP embeddings using the RCap dataset; B) **MRich-WoP**: This is a similar model to MRich but adapted directly to the target dataset without the pretraining (Sec. 3.1), which serves as an ablation study on the effect of the pretraining; C) **MBGE**: A model trained on a single rich BGE embedding using the RCap dataset; D) **MBGE+**: A model trained on single BGE embedding using the RCap+ dataset; E) **MShortened**: A model trained on multi-CLIP embeddings using the SCap dataset. For the Mshortened model, since we have a single, short caption, we replicate it 5 times and reuse the same multi-conditional diffusion backbone, to avoid pretraining a separate model. We adopt the same replication when testing the Clip-Forge text queries on MShortened. As a baseline for comparison, we use Clip-Forge [26] without including our adapter.

4.3 Evaluation metrics

To evaluate our method, we take the caption of each shape in the test set, generate a 32^3 volume with our adapter based on the caption, and then compare the generated shape to the corresponding ground truth shape in the test set. Although this protocol is not fully deterministic, since multiple valid "ground truths" may exist for a given caption, it gives an indication of how close the results are to a valid shape.

To compare generated shapes to the ground truth, we use the following metrics: A) *Distributional metrics that compare entire sets of results*, such as Fréchet Inception Distance (FID) [7], calculated using a voxel classifier [26] pretrained on the ShapeNet dataset; Minimum Matching Distance (MMD) [1], using Chamfer distance to compare voxelized 3D shapes; Coverage (COV), based on the fraction of matched voxels; and B) *Pairwise metrics* comparing the predicted and ground truth volume shapes in the test set, such as Chamfer Distance (CD), Earthmover's Distance (EMD), and Light Field Distance (LFD).

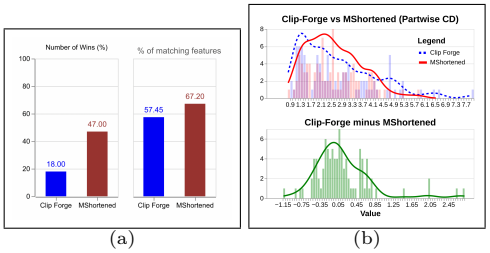
4.4 Results

Fig. 3 shows visual comparisons between the results of Clip-Forge and our method applied to Clip-Forge. Although our results may not fully capture the caption's details, the generated shapes have better geometry and more accurately reflect the test set captions of ShapeNet. For example, the caption in the last row mentions a "semi-circular" seat. Clip-Forge generates a typical chair. In contrast, since our method was adapted to the target ShapeNet dataset, it produces the correct type of seat. We also show shapes generated by the MBGE+ model in Fig. 4, which demonstrates that our method can support longer and richer captions that provide finer control over the shape generation.

Table 1 provides a quantitative comparison. We see that the Mshortened model has a very low FID, which is less than half the distance for Clip-Forge, hinting that our approach aligns the distribution closer to the target dataset. In addition, the trained model has a significantly higher COV value, suggesting that the predicted distribution is more representative of the target dataset. When we consider the pairwise metrics, our adapter has a better LFD than Clip-Forge, although CD and EMD are comparable. When analyzing the variations of our model, we see that MRich significantly outperforms MBGE in all the metrics, which shows that several simple multi-modal text embeddings more easily map to the CLIP image space than a consolidated rich text embedding. We can also see that the MRich model outperforms MRich-WoP on all metrics, showing that pretraining on an abundant text-to-image dataset improves performance of shape generation. MBGE+ has a better coverage and pairwise metrics among all three models and has the best performance, which shows that richer caption sets can further align the generated shapes with the target shape distribution. The successful mapping of BGE embeddings to the CLIP image space further highlights that our framework can seamlessly integrate various language feature extractors beyond CLIP, enabling greater flexibility and control over shape generation through rich and descriptive captions.

Moreover, the MShortened model performs relatively well on open set queries as indicated by Fig 5 (c). In fact, we obtain a better FID for the generated shapes. This strongly emphasizes that even though the mapping model is tuned to the target dataset, it still retains the global knowledge from the CLIP model.

We also evaluate whether the generated shapes have the same parts present in the ground truth shapes. We use the PartNet [20] dataset for this evaluation and calculate the one-way chamfer distance (CD) from the parts in the test set to the shapes generated with Clip-Forge and the MShortened model. The analysis included 95 leaf part labels, with 41 parts for chairs and 54 parts for tables. A total of 2,917 shapes out of 3,016 from the test set were used, since we were unable to find the part segmentation data for some of the shapes. The results in Fig. 5 (d) show that Clip-Forge does not generalize so well to all the different parts of the shapes. The fine-tuned model has a better overall part-wise CD for both chairs and tables. We perform further analysis by plotting in Fig. 5 (b) the partwise-CD distribution for the leaf parts for the two methods (top) and the distribution of their differences (bottom). The top graph shows that out of 96



(a) (b)

Model	FID ↓	MMD (CD)↓	COV (CD)↑
Clip-Forge	3576.74	**2.777**	**0.0140**
MShortened	**2966.95**	2.793	0.0134

(c)

Model	Chairs	Tables	Overall
Clip-Forge	2.2600	3.0739	2.7227
MShortened	**2.2293**	**2.8493**	**2.5817**

(d)

Fig. 5. (a) User study results of Clip-Forge versus MShortened models on the sampled SCap dataset. (b) Analyzing the performance of the part-wise chamfer distance ↓. (c) Quantitative results on the Clip-Forge test set using FID, MMD, and coverage metrics. (d) Chamfer Distance ↓ comparison for chairs and tables. Please zoom in for details.

parts, Clip-Forge misses some parts and hence the higher chamfer distance for those parts and a longer tail distribution. This can be further confirmed by the bottom graph, where the tail on the right indicates that some parts are not even considered by Clip-Forge for shape generation when compared to MShortened.

4.5 Human perceptual evaluation

We conducted a user study using 100 randomly selected shapes from the ShapeNet [2] test set. Shapes were generated based on SCap captions using the trained MShortened and CLIP-Forge models. Without viewing the generated shapes, we manually extracted both qualitative and numerical features from the SCap captions, compiling a feature list for each shape. The dataset was then randomly divided into two subsets, with 3 users assigned to each, with a total of 6 users in the study. Participants were tasked with matching the listed features to multi-view images of the corresponding shapes. We aggregated the results for all 100 shapes and calculated the overall percentage of matching features. Additionally, for each shape, we determined whether MShortened or CLIP-Forge had more matching features; the model with the higher count was considered the winner. Using this criterion, we computed the percentage of wins for both models, excluding ties. As shown in Fig. 5 (a), the fine-tuned MShortened model was preferred by users over CLIP-Forge on both the metrics.

5 Conclusion and future work

We showed that supervised adaptation from the text embedding space to the CLIP image space allows for shape generation with richer captions and also helps shorter captions perform better as indicated by Fig. 5 (a), without retraining the CLIP-Forge [26] model. We also showed how to synthetically generate enhanced captions using VLMs and a shape segmentation dataset, and showed that multi-caption adaptation is more effective than adaptation with a rich embedding.

Exploring the method on larger datasets, other categories and even deformable shapes would be a direction for future work. It would also be valuable to assess the performance of the method with alternative models like CLIP-Sculptor [27], which takes a CLIP image or text latent as a condition for shape generation. Utilizing more advanced 3D captioning models such as CAPS3D[17] may improve the synthetic dataset and thus the shape generation.

References

1. Achlioptas, P., Diamanti, O., Mitliagkas, I., Guibas, L.: Learning representations and generative models for 3d point clouds (2018), https://arxiv.org/abs/1707.02392
2. 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), https://arxiv.org/abs/1512.03012
3. Choy, C.B., Xu, D., Gwak, J., Chen, K., Savarese, S.: 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction (2016), https://arxiv.org/abs/1604.00449
4. Duan, Y., Hong, Y., Zhang, B., Lan, J., Zhu, H., Wang, W., Zhang, J., Niu, L., Zhang, L.: Domaingallery: Few-shot domain-driven image generation by attribute-centric finetuning (2024), https://arxiv.org/abs/2411.04571
5. Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., Letman, A., Mathur, A., Schelten, A., Yang, A., Fan, A., et al.: The llama 3 herd of models. arXiv preprint arXiv:2407.21783 (2024)
6. Fu, R., Zhan, X., Chen, Y., Ritchie, D., Sridhar, S.: Shapecrafter: A recursive text-conditioned 3d shape generation model (2023), https://arxiv.org/abs/2207.09446
7. 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), https://arxiv.org/abs/1706.08500
8. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models (2020), https://arxiv.org/abs/2006.11239
9. Ho, J., Salimans, T.: Classifier-free diffusion guidance (2022), https://arxiv.org/abs/2207.12598
10. Jain, A., Mildenhall, B., Barron, J.T., Abbeel, P., Poole, B.: Zero-shot text-guided object generation with dream fields (2022), https://arxiv.org/abs/2112.01455
11. Lee, H.H., Savva, M., Chang, A.X.: Text-to-3d shape generation (2024), https://arxiv.org/abs/2403.13289
12. Liang, W., Zhang, Y., Kwon, Y., Yeung, S., Zou, J.: Mind the gap: Understanding the modality gap in multi-modal contrastive representation learning (2022), https://arxiv.org/abs/2203.02053
13. Lin, T.Y., Maire, M., Belongie, S., Bourdev, L., Girshick, R., Hays, J., Perona, P., Ramanan, D., Zitnick, C.L., Dollár, P.: Microsoft coco: Common objects in context (2015), https://arxiv.org/abs/1405.0312
14. Liu, H., Li, C., Li, Y., Li, B., Zhang, Y., Shen, S., Lee, Y.J.: Llava-next: Improved reasoning, ocr, and world knowledge (January 2024), https://llava-vl.github.io/blog/2024-01-30-llava-next/
15. Liu, Z., Wang, Y., Qi, X., Fu, C.W.: Towards implicit text-guided 3d shape generation (2022), https://arxiv.org/abs/2203.14622

16. Lu, S., Li, Y., Chen, Q.G., Xu, Z., Luo, W., Zhang, K., Ye, H.J.: Ovis: Structural embedding alignment for multimodal large language model (2024), https://arxiv.org/abs/2405.20797
17. Luo, T., Rockwell, C., Lee, H., Johnson, J.: Scalable 3d captioning with pretrained models (2023), https://arxiv.org/abs/2306.07279
18. Mescheder, L., Oechsle, M., Niemeyer, M., Nowozin, S., Geiger, A.: Occupancy networks: Learning 3d reconstruction in function space (2019), https://arxiv.org/abs/1812.03828
19. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: Nerf: Representing scenes as neural radiance fields for view synthesis (2020), https://arxiv.org/abs/2003.08934
20. Mo, K., Zhu, S., Chang, A.X., Yi, L., Tripathi, S., Guibas, L.J., Su, H.: Partnet: A large-scale benchmark for fine-grained and hierarchical part-level 3d object understanding (2018), https://arxiv.org/abs/1812.02713
21. Mohammad Khalid, N., Xie, T., Belilovsky, E., Popa, T.: Clip-mesh: Generating textured meshes from text using pretrained image-text models. In: SIGGRAPH Asia 2022 Conference Papers. p. 1–8. SA '22, ACM (Nov 2022). https://doi.org/10.1145/3550469.3555392, http://dx.doi.org/10.1145/3550469.3555392
22. Poole, B., Jain, A., Barron, J.T., Mildenhall, B.: Dreamfusion: Text-to-3d using 2d diffusion (2022), https://arxiv.org/abs/2209.14988
23. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., Sutskever, I.: Learning transferable visual models from natural language supervision (2021), https://arxiv.org/abs/2103.00020
24. Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., Chen, M.: Hierarchical text-conditional image generation with clip latents (2022), https://arxiv.org/abs/2204.06125
25. Ruiz, N., Li, Y., Jampani, V., Pritch, Y., Rubinstein, M., Aberman, K.: Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation (2023), https://arxiv.org/abs/2208.12242
26. Sanghi, A., Chu, H., Lambourne, J.G., Wang, Y., Cheng, C.Y., Fumero, M., Malekshan, K.R.: Clip-forge: Towards zero-shot text-to-shape generation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 18603–18613 (2022)
27. Sanghi, A., Fu, R., Liu, V., Willis, K.D., Shayani, H., Khasahmadi, A.H., Sridhar, S., Ritchie, D.: Clip-sculptor: Zero-shot generation of high-fidelity and diverse shapes from natural language. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 18339–18348 (2023)
28. Sella, E., Fiebelman, G., Atia, N., Averbuch-Elor, H.: Spice-e : Structural priors in 3d diffusion using cross-entity attention (2024), https://arxiv.org/abs/2311.17834
29. Team, Q.: Qwen2.5: A party of foundation models (September 2024), https://qwenlm.github.io/blog/qwen2.5/
30. Xiao, S., Liu, Z., Zhang, P., Muennighoff, N.: C-pack: Packaged resources to advance general chinese embedding (2023)
31. Zhang, L., Rao, A., Agrawala, M.: Adding conditional control to text-to-image diffusion models (2023), https://arxiv.org/abs/2302.05543
32. Zhou, Y., Li, C., Chen, C., Gao, J., Xu, J.: Lafite2: Few-shot text-to-image generation (2022), https://arxiv.org/abs/2210.14124