Blended Point Cloud Diffusion for Localized Text-guided Shape Editing

Supplementary Material

Contents

We refer readers to the interactive visualizations at index.html. In this document, we provide implementation details (Section 1), additional experiments and comparisons (Section 2) and a discussion of limitations (Section 3).

1. Implementation Details

1.1. Training

We trained our model for each category using object-specific ShapeTalk and l-ShapeTalk subsets. To evaluate generalization, we also trained a unified model across all three categories (Chair, Table, and Lamp). For all models, we used a batch size of 6 and a learning rate of 11×10^{-4}. The number of epochs, iterations, and training hours for each category are detailed in Table 1. To encourage the network to rely more on structural guidance rather than text, we dropped the text guidance with a probability of 0.5, replacing the textual prompt with an empty string. As explained in Section 3.2.2 of the main paper, we also replaced the conditional point cloud input with the target point cloud with a probability of 0.1 to support our Inversion-Free Coordinate Blending mechanism. All training was conducted on a single RTX A5000 GPU (24GB VRAM) for each model.

Data. To generate the partial point clouds used as guidance during training and to construct the l-ShapeTalk dataset, we used the baseline Llama 3 [5] model provided by unsloth. Specifically, we instructed Llama 3 with the following prompt for chairs:

```
### Instruction: Return a single word
using only one of the following options.
Options: back, leg, arm, seat, unknown.
```

Model	Iterations	Epochs	Training Time (hours)
Chair	13.5 M	250	119
Table	12.1 M	250	107
Lamp	8.2 M	250	74
Airplane	0.9 M	250	52
Guitar	0.4 M	250	31
Knife	0.3 M	250	25
Cap	0.2 M	250	18
Skateboard	0.1 M	250	16
Unified	13.5 M	100	98

Table 1. **Training time.** We report the number of iterations, epochs and the overall training time for each category.

```
### Input: What part of a chair does the
next utterance describe?
If none of the parts are described in
the utterance return unknown. Utterance:
it has larger height.
### Response: unknown (EOT-token)
```

We adjusted the prompt for each category, instructing the language model to extract appropriate part names specific to that category. To enhance accuracy, we fine-tuned the model for this task using 200 manually labeled examples. Samples where the model returned *"unknown"* were excluded from the training set, leading to the creation of the l-ShapeTalk dataset.

For evaluation, we applied the same method but removed the option to return *"unknown"*, requiring the language model to extract the part name it deemed most suitable to convey the text prompt.

The part name in each l-ShapeTalk sample was provided as input to a segmentation model to extract binary masks. For segmentation, we used the PyTorch implementation of PointNet by ailia-models. These binary masks were inverted and then multiplied element-wise with the target point cloud to generate the guidance partial point cloud.

1.2. Evaluation

ShapeTalk prompts often describe relationships between objects (e.g., *"it has a taller backrest"*), which makes them unsuitable for evaluating individual instances. To address this, we use Llama 3 to translate ShapeTalk prompts into more *descriptive* prompts, such as converting *"it has a taller backrest"* to *"a chair with a tall backrest"*.

We use two CLIP-based [9] metrics, CLIP_{Sim} and CLIP_{Dir}, to evaluate edit fidelity. For the CLIP image encoder, we rendered a single image for each point cloud from a consistent viewpoint.

CLIP$_{Sim}$ evaluates the similarity between the final output and its textual description. We encoded the rendered image of the output point cloud and the descriptive prompt using their respective CLIP encoders and calculated the cosine similarity between the resulting encodings.

CLIP$_{Dir}$ assesses the semantic direction in CLIP space, capturing the relationship between both the guidance and output shapes. We encoded the rendered images of the input and output point clouds, along with the descriptive prompt and a simple prompt describing the general object shape (e.g., *"a chair"*). The differences between the two text encodings and the two image encodings were calculated, and their cosine similarity was used as the metric.

All experiments were conducted using the *"openai/clip-vit-base-patch32"* model, accessed through the Hugging Face library.

To calculate the rest of the metrics used in the quantitative analysis we used the evaluate_change_it_3d.py script available in ChangeIt3D's github repo.

User Study. As detailed in Section 4.2, our user study consisted of two forms, A and B, each containing 15 questions and answered by 60 different users. In each question, users were shown a text prompt and an input point cloud, then asked to select one of the editing results generated by ChangeIt3D, Spice-E, and our method, displayed in random order. Both the prompts and input point clouds were randomly selected from the l-ShapeTalk test set. The users were instructed:

"Your task is to select the target object that best represents the prompt while maintaining the shape of the source object as closely as possible. In other words, choose the most suitable target object that effectively embodies the editing of the source object based on the prompt. IMPORTANT - If none of the target objects align with the prompt, select the target object that best preserves the shape of the source object."

Baselines. We used the official Changeit3D implementation to access the ShapeTalk dataset and the pre-trained model weights. The script evaluate_change_it_3d.py was employed to reproduce their results. Similarly, we used the Spice-E implementation for obtaining their result. The outputs were converted into point clouds using marching cubes and random point sampling.

2. Additional Experiments

To better view 3D results, we recommended viewing the supplemental HTML page, which includs fly-through visualizations demonstrating the quality of our results from multiple views.

2.1. Comparison to Semantic Editing Paradigms

In this section, we complement our main paper's comparisons with a qualitative analysis of alternative approaches to text-guided semantic editing of 3D shapes.

Image Editing and Single View Reconstruction. Recent advancements in image editing and 3D reconstruction methods suggest another potential paradigm: perform image editing on the rendered image of a 3D shape and then use single-view reconstruction to generate an edited 3D shape. We tested this approach by using InstructPix2Pix [12] in conjunction with One-2-3-45++ [6]. Specifically, we rendered an image of the input shape, provided it along with an editing prompt to InstructPix2Pix, and used the edited image as input to One-2-3-45++ to reconstruct the edited 3D shape. As shown in Figure 3, InstructPix2Pix fail to perform fine-grained shape editing. To address this, we also fine-tuned InstructPix2Pix using images rendered from the ShapeTalk dataset, as also presented in Figure 3. While the fine-tuned model produces higher-quality results, it still fails to localize edits effectively, especially compared to our method. For single-view reconstruction, shapes generated from InstructPix2Pix results using One-2-3-45++ are shown in Figure 1. Beyond InstructPix2Pix's difficulty with precise localization, One-2-3-45++ introduces slight defects in the reconstructed shapes, such as an asymmetrical backrest (second row from the top) and slightly lopsided legs (third row from the top). However, we note that these defects are minor. The overall high visual quality of the results demonstrates promise for future research in this direction.

Optimization-Based Editing. A widely used approach for textual editing of 3D shapes is Score Distillation Sampling (SDS) [8], which employs inference-time optimization to modify a 3D representation based on a text prompt, using a pre-trained generator as a prior. A qualitative comparison with the SDS based methods Fantasia3D [3] and Vox-E [10] is presented in Figure 3. As shown, while Fantasia3D generates results that resemble the input shapes, it often fails to follow the fine-grained instructions in the editing prompts (e.g., the back is not rounded in the first row, and the chair in the second row lacks four legs). Vox-E also often does not follow the editing instruction (the chair on the third row does not have noticebly thinner legs, chair on the fourth row does not seem to have a taller backrest), but also often fails to correctly maintain the identity of the input objects (adding legs to the chairs on the first and last rows). Additionally, as is common with many SDS-based methods, both methods tend to exhibit noise.

2.2. Ablation Study for t_r Values

The parameter t_r controls the balance between coordinate blending steps and steps dedicated solely to shape reconstruction. Higher t_r values increase the number of coor-

ICCV #5344

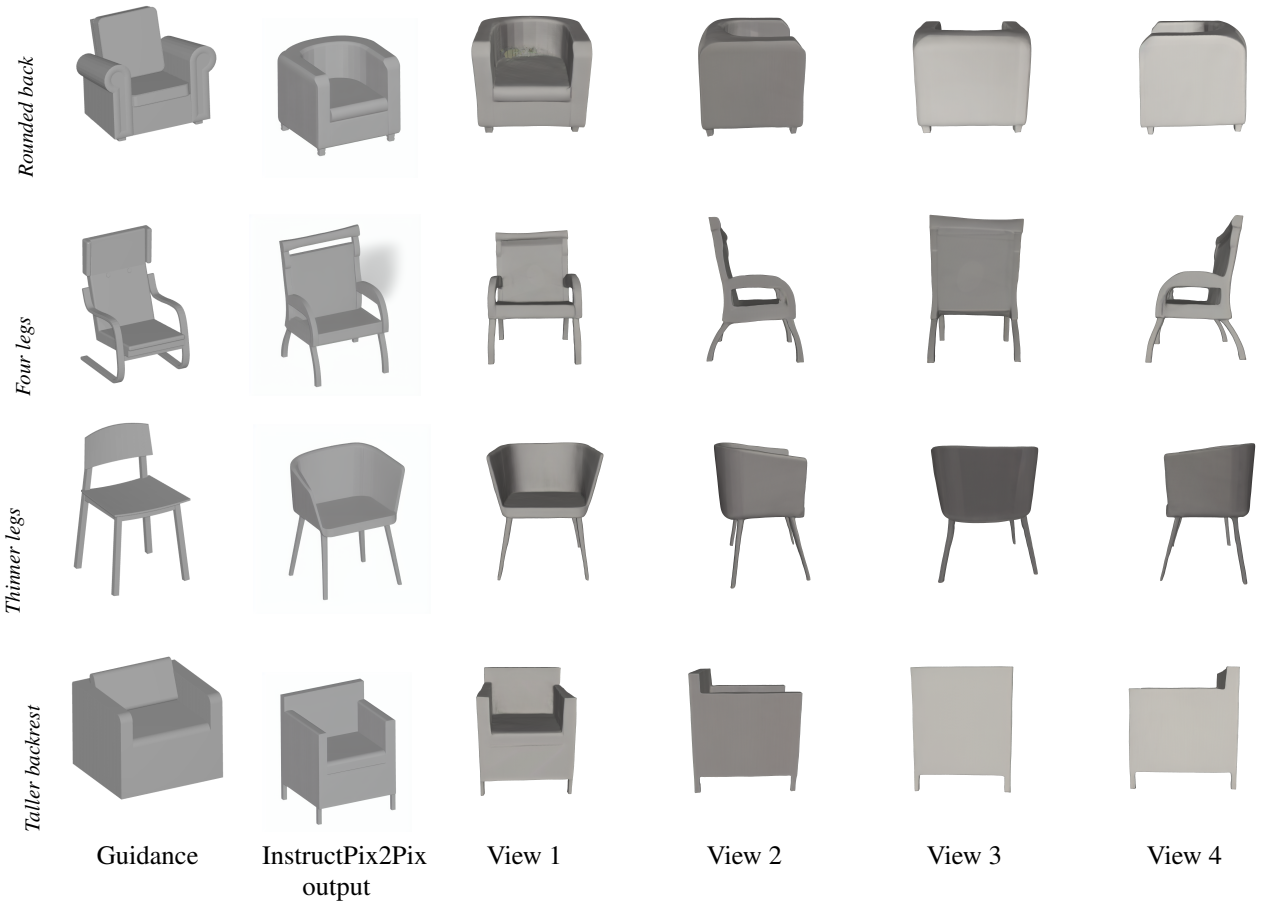

Figure 1. **Image Editing Followed by Single-View 3D Reconstruction.** We first use InstructPix2Pix to edit the rendered image and then apply One-2-3-45++ for single-view reconstruction. In addition to InstructPix2Pix's challenges with precise localization, One-2-3-45++ introduces minor defects in the reconstructed shapes.

dinate blending steps, providing greater editing freedom but at the cost of identity preservation. Conversely, lower t_r values improve identity preservation while reducing the model's ability to adhere closely to the text prompt.

Figure 2 illustrates the effect of different t_r values. As t_r decreases, the output aligns more closely with the input point cloud. While higher t_r values allow greater editing freedom, they can result in inconsistencies with the masked guidance point cloud. We set $t_r = 20$ in our experiments as it offers a good balance between identity preservation and edit fidelity.

2.3. Multi Category Training

Table 2 demonstrates that our technique is not restricted to single-category training. We trained our model on a unified dataset comprising all three categories combined. The results show that our method handles multiple categories effectively, with the unified model producing results comparable to those of the single-category models.

2.4. Per-Category Evaluation

Table 3 provides the evaluation results for each of the three object categories used for quantitative evaluation. The per-category scores align closely with the overall average scores reported in the main paper.

2.5. Extended Qualitative Results

In Figure 4 we present qualitative results from multiple shapeNet categories (*Guitar*, *Airplane*, *Chair*, *Lamp*, *Sword*, *Hat*, *Skateboard* and *Table*), demonstrating our

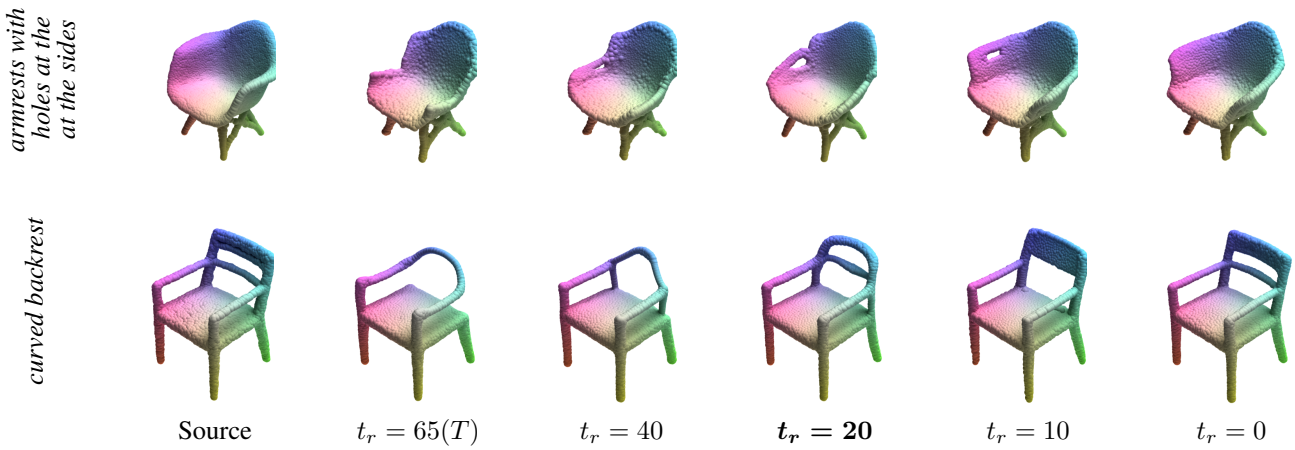

Figure 2. **Ablation Study for t_r Values.** Higher t_r values increase the number of coordinate blending steps, providing better editing freedom but at the cost of inferior identity preservation. Conversely, lower t_r values improve identity preservation while reducing the model's ability to follow the textual description.

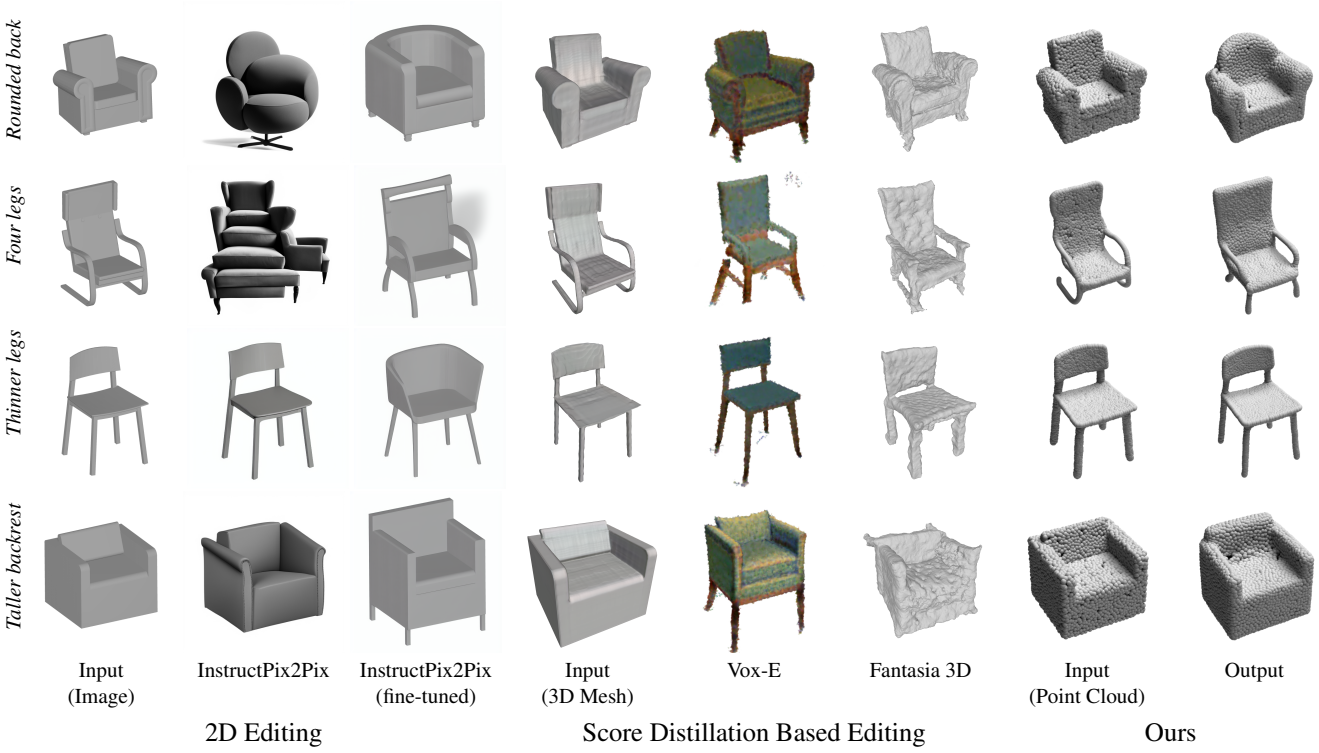

Figure 3. **Qualitative comparison.** We compare our method's outputs to those of the image editing method InstructPix2Pix [2] as well as the score distillation sampling optimization based 3D editing works Fantasia 3D [3] and Vox-E [10]. As illustrated above, our method outperforms these baselines in terms of edit fidelity, identity preservation and overall visual quality.

method's ability to make meaningful fine grained edits across a wide variety of shape types.

2.6. Comparison to RePaint

The free form inpainting method RePaint [7] has had a major impact on the field of diffusion based image inpainting. This work introduced an inference time algorithm which, somewhat similarly to our coordinate blending algorithm, blends noisy versions of the "known" regions of the image with the predicted denoised versions of the inpainted region according to an input binary mask. RePaint also proposes to resample noise and repeat the process for a given num-

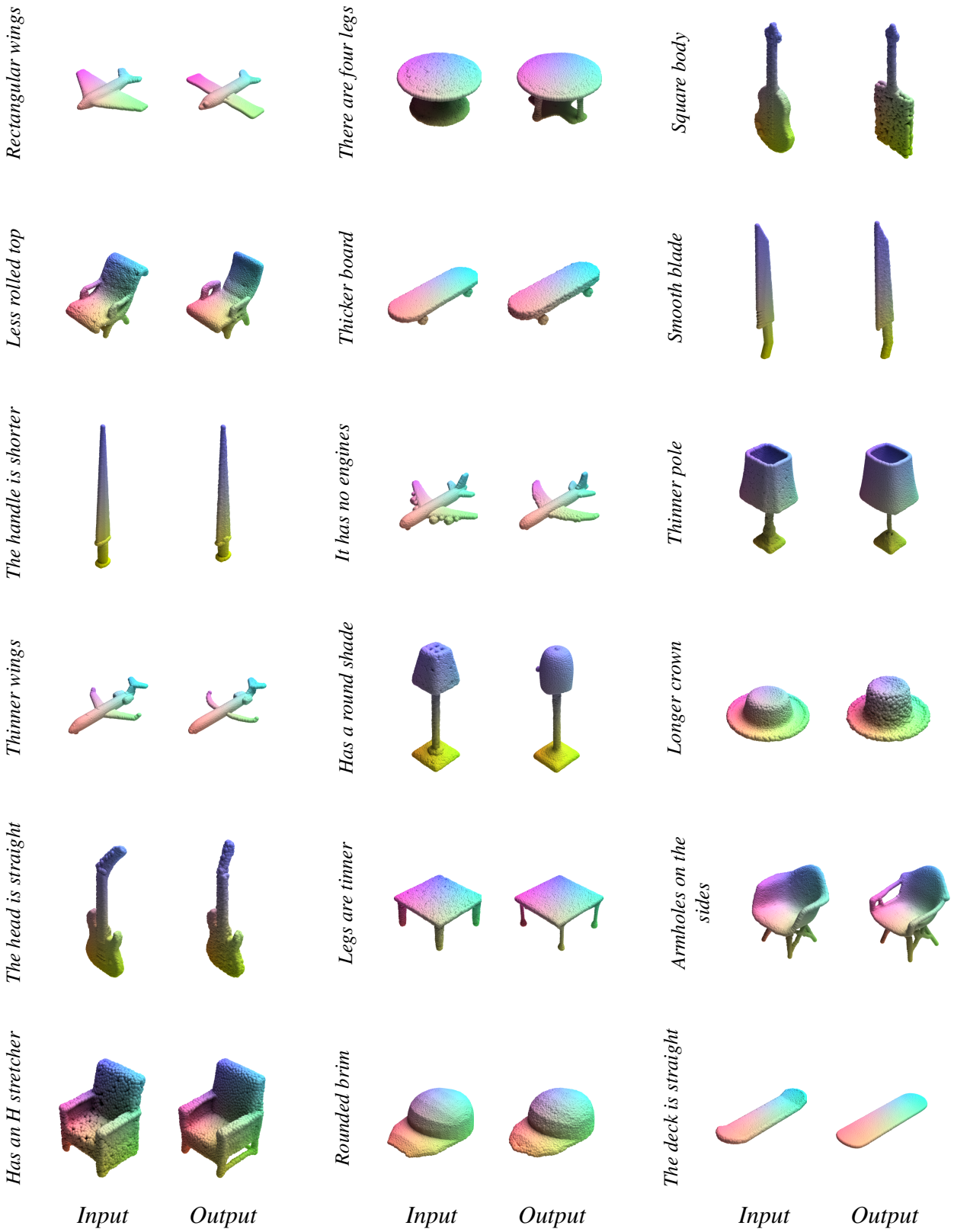

Figure 4. **Results Gallery.** Above we show results over various object categories including *chair*, *table*, *lamp*, *airplane*, *cap*, *guitar*, *skateboard* and *knife*.

	Shapetalk						l-Shapetalk					
Metric	$\text{CLIP}_{Sim}\uparrow$	$\text{CLIP}_{Dir}\uparrow$	GD↓	CD↓	FPD↓	l-GD↓	$\text{CLIP}_{Sim}\uparrow$	$\text{CLIP}_{Dir}\uparrow$	GD↓	CD↓	FPD↓	l-GD↓
Unified Model	**0.26**	0.00	**0.32**	**0.03**	43.05	0.68	0.26	0.00	0.36	0.09	138.26	0.82
Ours	**0.26**	**0.01**	**0.34**	**0.05**	**33.64**	0.78	**0.27**	**0.01**	**0.29**	**0.04**	**13.51**	**0.55**

Table 2. **Comparison to Unified Model.** Our technique performs effectively across multiple categories training, as evidenced by the comparable results.

		Shapetalk						l-Shapetalk					
	Metric	$\text{CLIP}_{Sim}\uparrow$	$\text{CLIP}_{Dir}\uparrow$	GD↓	CD↓	FPD↓	l-GD↓	$\text{CLIP}_{Sim}\uparrow$	$\text{CLIP}_{Dir}\uparrow$	GD↓	CD↓	FPD↓	l-GD↓
Chair	Changeit3D	0.22	-0.02	0.48	0.05	127.35	0.79	0.22	-0.02	0.68	0.07	156.11	0.87
	Spice-E	0.26	0.00	1.78	0.18	527.33	1.01	0.26	0.00	1.78	0.22	653.16	0.93
	Ours	**0.28**	**0.01**	**0.27**	**0.02**	**14.13**	**0.74**	**0.28**	**0.01**	**0.24**	**0.01**	**5.87**	**0.54**
Table	Changeit3D	0.22	-0.03	0.55	0.16	111.27	1.00	0.22	-0.04	0.66	0.13	141.12	1.04
	Spice-E	0.25	-0.01	1.85	0.40	489.32	0.88	**0.26**	-0.04	2.85	0.40	621.43	1.07
	Ours	**0.27**	**0.01**	**0.33**	**0.03**	**18.96**	**0.71**	**0.26**	**0.01**	**0.33**	**0.03**	**11.50**	**0.51**
Lamp	Changeit3D	0.19	-0.01	0.93	0.33	310.44	0.66	0.19	-0.02	1.01	0.38	354.63	0.75
	Spice-E	0.23	-0.02	1.88	0.14	153.41	0.94	0.23	-0.02	2.16	0.16	188.56	0.95
	Ours	**0.24**	**0.00**	**0.41**	**0.09**	**67.82**	**0.88**	**0.26**	**0.01**	**0.31**	**0.07**	**23.18**	**0.60**

Table 3. **Per-Category Evaluation** . We compare the performance of ChangeIt3D [1] and Spice-E [11] against ours over the three object categories in the ShapeTalk and l-ShapeTalk datasts.

Metric	$\text{CLIP}_{Sim}\uparrow$	$\text{CLIP}_{Dir}\uparrow$	GD↓	CD↓	FPD↓	l-GD↓
$\text{RP}_{r=1,j=1}$	0.22	-0.02	0.57	0.12	69.11	0.72
$\text{RP}_{r=10,j=1}$	0.19	-0.02	0.62	0.15	79.33	**0.55**
$\text{RP}_{r=1,j=10}$	0.19	-0.03	0.63	0.14	63.48	0.71
$\text{RP}_{r=10,j=10}$	0.21	-0.03	0.72	0.13	64.17	0.57
Ours	**0.27**	**0.01**	**0.29**	**0.04**	**13.51**	**0.55**

Table 4. **Quantitative Evaluation** against RePaint (RP) with different resampling (r) and jumping (j) values. Note that this implementation uses our InPaint-E model and our reconstructed noise on the non-edit regions as directly using Point-E and random noise resulted in highly noisy (and uninformative) outputs.

ber of iterations to further refine this inpainting process. By contrast, in addition to other core differences our work dedicates a significant portion of the inference process to reconstructing the full input shape before starting the inpainting process, as well as operating on a specific model tailored for this task instead of a more general text-to-3D model. To test the significance of some of these design choices we conducted a quantitative comparison against a baseline which resembles RePaint in its function. Specifically, in this baseline inpainting is performed at every inference step ($t_r = T$) and the edit region is initialized with random noise. This baseline also incorporates the "resampling" and "jumping" mechanisms introduced in RePaint. Unlike RePaint however, we used our reconstructed noise for the non-edit region and operated on Inpaint-E, as directly using Point-E and random noise resulted in highly noisy (and uninformative) outputs.

The results of this comparison are presented in Table 4 and clearly shows that our method outperforms this baseline across all metrics.

2.7. Shape Completion Comparison

Shape, or part, completion is a longstanding task involving completing a shape that is missing one or more parts in a way that maintains plausibility and optionally aligns with a text prompt. In Figure 5, we illustrate that these methods are not well suited for fine-grained shape editing as they are inherently blind to the missing parts, focusing on a comparison with SDFusion [4]. Note that we demonstrate this in a slightly different setting in comparison to the one addressed in our work—*unconditional* generation, and not *text-guided* part completion—as their official codebase does not include text-guided part completion.

Nonetheless, we performed a qualitative comparison in which we compared our results (text-guided) against unconditional part completion results of SDFusion. These results show that this method's ability to maintain identity is somewhat limited, even outside of the edit region (thicker legs on the chair in the second row, as well as adding arms to it). As SDFusion's part completion in this case is not guided by text it is somewhat hard to judge its quality. However, it is evident that the completed parts often don't match the general identity of the original shape particularly well (office chair type backrest in row 2, huge elaborate legs in row 1)

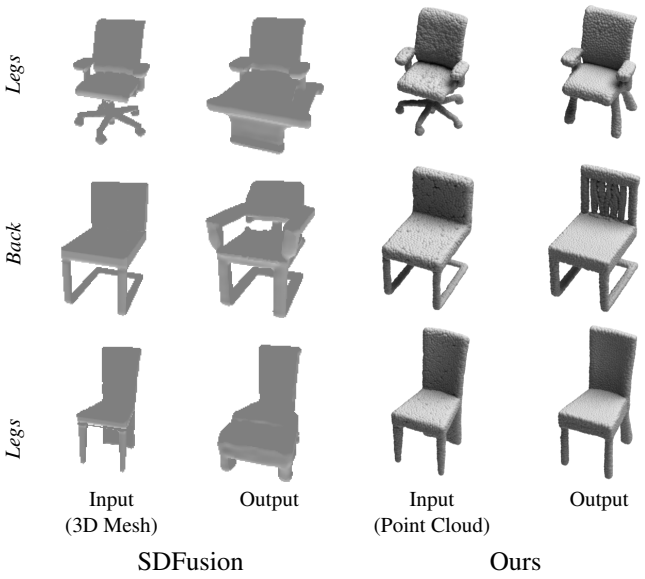

Figure 5. **Qualitative comparison against SDFusion [4] unconditional part completion.** We compare our method's outputs against the unconditional part completion results of the SDFusion [4] baseline. In each row we task the methods with completing (in SDFusion's case) or editing[1] (in our case) a different part. As these results show, SDFusion's ability to preserve identity across all regions of the shape is limited, in comparison to our approach that performs localized fine-grained editing of 3D shapes.

compared to our method.

3. Limitations

While our method performs well in most scenarios, it has certain limitations. First, our inpainting-based approach restricts the ability to generate entirely new objects or parts. BlendedPC is specifically designed for localized editing and struggles with global shape transformations. Second, as a supervised learning approach, the method's generalizability is constrained by the object categories in the training dataset. This limits performance when editing shape categories not encountered during training. These limitations present exciting opportunities for future research, such as developing techniques for generating object parts, enabling global shape modifications, and improving cross-category generalization.

References

[1] Panos Achlioptas, Ian Huang, Minhyuk Sung, Sergey Tulyakov, and Leonidas Guibas. Changeit3d: Languageassisted 3d shape edits and deformations. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, page 6, 2022. 6

[2] Tim Brooks, Aleksander Holynski, and Alexei A Efros. Instructpix2pix: Learning to follow image editing instructions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18392–18402, 2023. 4

[3] Rui Chen, Yongwei Chen, Ningxin Jiao, and Kui Jia. Fantasia3d: Disentangling geometry and appearance for high-quality text-to-3d content creation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 22246–22256, 2023. 2, 4

[4] Yen-Chi Cheng, Hsin-Ying Lee, Sergey Tulyakov, Alexander G Schwing, and Liang-Yan Gui. Sdfusion: Multimodal 3d shape completion, reconstruction, and generation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4456–4465, 2023. 6, 7

[5] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024. 1

[6] Minghua Liu, Ruoxi Shi, Linghao Chen, Zhuoyang Zhang, Chao Xu, Xinyue Wei, Hansheng Chen, Chong Zeng, Jiayuan Gu, and Hao Su. One-2-3-45++: Fast single image to 3d objects with consistent multi-view generation and 3d diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10072–10083, 2024. 2

[7] Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. Repaint: Inpainting using denoising diffusion probabilistic models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11461–11471, 2022. 4

[8] Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv preprint arXiv:2209.14988*, 2022. 2

[9] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 1

[10] Etai Sella, Gal Fiebelman, Peter Hedman, and Hadar Averbuch-Elor. Vox-e: Text-guided voxel editing of 3d objects. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 430–440, 2023. 2, 4

[11] Etai Sella, Gal Fiebelman, Noam Atia, and Hadar Averbuch-Elor. Spice· e: Structural priors in 3d diffusion using cross-entity attention. In *ACM SIGGRAPH 2024 Conference Papers*, pages 1–11, 2024. 6

[12] Jiale Xu, Xintao Wang, Yan-Pei Cao, Weihao Cheng, Ying Shan, and Shenghua Gao. Instructp2p: Learning to edit 3d point clouds with text instructions. *arXiv preprint arXiv:2306.07154*, 2023. 2

[1] The text guidance provided for our method (from top to bottom): *four legs*, *spindle backrest*, *four legs*.