# Supplementary of Walking the Schrödinger Bridge: A Direct Trajectory for Text-to-3D Generation

**Ziying Li**
Zhejiang University
emmaleee@zju.edu.cn

**Xuequan Lu**
University of Western Australia
bruce.lu@uwa.edu.au

**Xinkui Zhao***
Zhejiang University
zhaoxinkui@zju.edu.cn

**Guanjie Cheng**
Zhejiang University
chengguanjie@zju.edu.cn

**Shuiguang Deng**
Zhejiang University
dengsg@zju.edu.cn

**Jianwei Yin**
Zhejiang University
zjuyjw@cs.zju.edu.cn

https://github.com/emmaleee789/TraCe.git

**Supplementary Overview** The supplementary includes the following sections:



*Corresponding author. zhaoxinkui@zju.edu.cn

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

## A Algorithm

We provide the algorithm for TraCe in Algorithm 1.

**Algorithm 1 Tra**jectory-**Ce**ntric Distillation (TraCe)

**Require:** LoRA params $\phi$, 3D params $\theta$, prompt $y$, camera pose $c$, renderer $g$, pre-trained model $\epsilon$, LoRA model $\epsilon_\phi$, noise schedule parameters ($\beta_t$, $\sigma_t$, $\bar{\sigma}_t$, $\bar{\alpha}_t$), weighting $w(t)$.

1: **for** $i \in \{0, \dots, N-1\}$ **do**
2: Render 3D structure $\theta$ at pose $c$ to get 2D view $x_{\text{rndr}} \leftarrow g(\theta, c)$
3: Randomly sample timestep $t' \sim \mathcal{U}(0.02, 0.7)$ for stage 1, and $t' \sim \mathcal{U}(0.02, 0.5)$ for stage 2
4: Calculate predicted target image $x_0^{\text{pred}} = \left(x_{\text{rndr}} - \sqrt{1-\bar{\alpha}_{t'}}\,\epsilon_{\text{pretrain}}(x_{\text{rndr}}, t', y)\right) / \sqrt{\bar{\alpha}_{t'}}$
5: Sample timestep $t \sim \mathcal{U}(0.02, 0.5)$ using scheduled $t$-Sampling
6: Sample intermediate state $x_t \sim q(x_t \mid x_0^{\text{pred}}, x_{\text{rndr}}) = \mathcal{N}(x_t; \mu_t, \Sigma_t I)$
7: $\theta \leftarrow \theta - \eta_1 \mathbb{E}_{\epsilon,t,c}\left[w(t)\left(\epsilon_\phi(x_t, t, y, c) - \frac{x_t - x_{\text{rndr}}}{\sigma_t}\right)\frac{\partial x_{\text{rndr}}}{\partial \theta}\right]$
8: $\phi \leftarrow \phi - \eta_2 \nabla_\phi \mathbb{E}_{\epsilon,t,c} \left\|\epsilon_\phi(x_t, t, y, c) - \epsilon\right\|_2^2$
9: **end for**

## B Experimental Setup

For [18] and [4], we use the official codebase; for [19], [15] and [8], we re-implement our algorithm with their implementations on threestudio [7] by using 3DGS instead of NeRF. We initialize 3DGS using Shap-E [2] following [18]. We use Stable Diffusion 2.1 [13] for our latent diffusion prior, and use Stable Diffusion 2.1-v [14] for predicting noise for LoRA U-Net. We use single NVIDIA A100 GPU for each generation and run each generation for 1700 steps with learning rate of $10^{-3}$ for our method. When generating predicted image as bridge's one end, we sample $t'$ with two stages: (1) for iteration 1 to 700: $t'$ is randomly sampled within $[0.02, 0.7]$; (2) for iteration 701 to 1700: $t'$ is randomly sampled within $[0.02, 0.5]$. We use camera distance range [1.5,4.0] for better generation results.

## C Gradient Analysis of SDS-based Methods

Existing text-to-3D methods often rely on Score Distillation Sampling (SDS) [12], which optimizes a 3D generator $g(\theta)$ by minimizing the diffusion model's training objective with respect to rendered images $x_{render} = g(\theta, c)$. The effective gradient is

$$\nabla_\theta \mathcal{L}_{\text{SDS}} \propto \mathbb{E}_{t,\epsilon}\left[w(t)\left((-\sigma_t' \nabla_{x_t} \log p_{\text{pretrain}}(x_t \mid y)) - (-\sigma_t' \nabla_{x_t} \log p(x_t \mid x))\right)\frac{\partial x}{\partial \theta}\right], \quad (1)$$

where $x_t$ is $x$ corrupted by noise $\epsilon$ at time step $t$. This gradient signal can be interpreted as the difference between the (scaled) score of the target distribution, $-\sigma_t' \nabla_{x_t} \log p_{\text{pretrain}}(x_t \mid y)$, approximated by the pre-trained model $\epsilon_{\text{pretrain}}$, and the score of the noisy data distribution, $-\sigma_t' \nabla_{x_t} \log p(x_t \mid x)$, corresponding to the added noise $\epsilon$. SDS thus distills the diffusion prior by aligning these score estimates evaluated on randomly noised samples $x_t$.

## D User Study

We conducted a user study to assess the perceptual quality of our method, TraCe, against other approaches SDS [12], CSD [19], VSD [15], ISM [4] and SDI [8]. We use 50 diverse text prompts, and 3D models generated by each method were rendered into videos. 60 volunteers then performed pairwise comparisons, evaluating the videos based on 3D consistency, prompt fidelity, and photorealism by following the evaluation setup in [3]. To evaluate 3D consistency, participants were instructed to choose the item that exhibited superior geometric coherence across multiple views, i.e., the one with a more consistent and accurate shape. For prompt fidelity, participants were asked to select the item that better aligned with the semantic content of the given text prompt. To assess photorealism, participants selected the item that displayed finer textures and a higher degree of visual realism,

resembling real-world 3D objects. Participants were also given the option to select "unknown" or "tie" if they were unable to make a confident decision. The order of presentation was randomized. Results in Table 1, 2, and 3 indicate a consistent user preference for the proposed TraCe, highlighting its superior coherence, faithfulness, and visual realism.

Table 1: **User Study: 3D Consistency.**

| Competitor Method | Ours Preferred | Tie | Competitor Preferred |
|---|---|---|---|
| SDS | 31.02% | **55.50%** | 13.48% |
| CSD | **42.64%** | 30.66% | 26.70% |
| VSD | 23.34% | **44.98%** | 31.68% |
| SDI | 24.80% | **42.26%** | 32.94% |
| ISM | 29.52% | **42.38%** | 28.10% |

Table 2: **User Study: Prompt Fidelity.**

| Competitor Method | Ours Preferred | Tie | Competitor Preferred |
|---|---|---|---|
| SDS | **41.26%** | 22.50% | 36.24% |
| CSD | **55.46%** | 24.50% | 20.04% |
| VSD | **58.75%** | 24.97% | 16.28% |
| SDI | **75.80%** | 16.04% | 8.16% |
| ISM | **42.68%** | 21.78% | 35.54% |

Table 3: **User Study: Photorealism.**

| Competitor Method | Ours Preferred | Tie | Competitor Preferred |
|---|---|---|---|
| SDS | **61.88%** | 15.16% | 22.96% |
| CSD | **72.25%** | 16.28% | 11.47% |
| VSD | **88.44%** | 5.40% | 5.16% |
| SDI | **76.70%** | 8.04% | 15.26% |
| ISM | **67.44%** | 12.92% | 19.64% |

## E Additional Quantitative Comparisons

To provide a more granular assessment beyond the overall GPTEval3D score, we present a detailed breakdown of its sub-metrics in Table 4. This qualitative comparison evaluates our 3D Gaussian Splatting results against baselines across five key categories. TraCe demonstrates superior performance in the majority of these detailed evaluations, achieving the top scores in Plausibility (1025.42), Texture Details (1031.98), Geometry Details (1014.03), and Text-Geometry Alignment (1024.45). While sds obtains a strong score in Text-Asset Alignment, TraCe remains highly competitive and exhibits the most robust profile across all metrics. This detailed analysis underscores TraCe's ability to generate 3D assets that are not only realistic but also excel in geometric fidelity and textural richness.

Table 4: **Quantitative comparisons of TraCe using 3D Gaussian Splatting under GPTEval3D evaluation.**

| Method | Plausibility | Texture Details | Geometry Details | Text-Asset Alignment | Text-Geometry Alignment |
|---|---|---|---|---|---|
| **sds (11min)** | 953.43 | 1022.40 | 1003.04 | 1038.08 | 1023.93 |
| **vsd (17min)** | 1020.81 | 985.59 | 996.29 | 993.04 | 1006.36 |
| **csd (11min)** | 1009.29 | 953.43 | 984.24 | 1024.24 | 950.82 |
| **sdi (10min)** | 1002.51 | 949.00 | 970.09 | 981.04 | 973.58 |
| **ism (20min)** | 1003.79 | 951.63 | 998.94 | 1007.07 | 989.78 |
| **TraCe (ours, 14min)** | **1025.42** | **1031.98** | **1014.03** | **1032.41** | **1024.45** |

Table 5: **Quantitative comparisons of TraCe using 3D Gaussian Splatting under GPTEval3D evaluation for new comparison methods Consistent3D and Dreamer XL.**

| Method | Plausibility | Texture Details | Geometry Details | Text-Asset Alignment | Text-Geometry Alignment |
|---|---|---|---|---|---|
| **Consistent3D [17] (14min)** | 981.48 | 1001.88 | 1013.99 | 986.38 | 995.42 |
| **Dreamer XL [9] (~1h)** | 1005.58 | 997.22 | 1002.94 | 1032.20 | 1012.37 |

Table 6: **Quantitative comparison under ImageReward evaluation for new comparison methods Consistent3D and Dreamer XL.**

| Method | ImageReward ↑ |
|---|---|
| **Consistent3D [17] (~1h)** | -0.8838 $\pm$ 0.5142 |
| **Dreamer XL [9] (~5h)** | -0.6737 $\pm$ 0.4098 |
| **TraCe (ours, ~3.5h)** | -0.4533 $\pm$ 0.4635 |

## F Further Studies on CFG Value

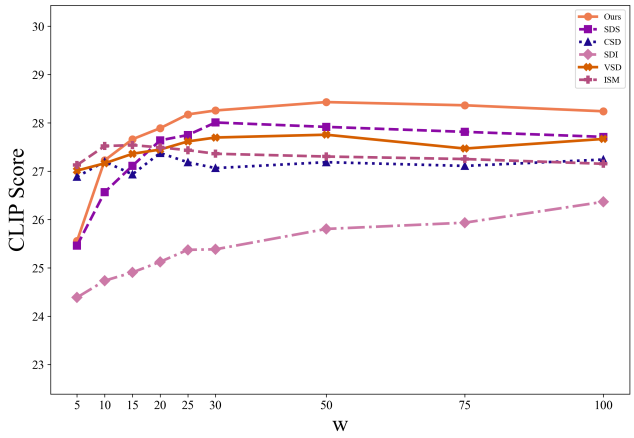


Figure 1: **Ablation Study of Classifier-Free Guidance (CFG) Value (w) on CLIP Score.** This figure illustrates the performance of TraCe, measured by CLIP Score (ViT-B/32), as the CFG value (w) is varied across four different text prompts. The plots indicate that optimal or near-optimal CLIP scores are generally achieved within a moderate CFG range (e.g., 10-30), with performance often plateauing or slightly decreasing at higher CFG values.

## G Generalization to NeRF

While the primary experiments in our main paper focus on 3D Gaussian Splatting (3DGS) for its high-fidelity rendering and fast optimization, our proposed Trajectory-Centric Distillation (TraCe) is a general optimization framework. It is readily applicable to other differentiable 3D representations, such as Neural Radiance Fields (NeRF) [10]. In this section, we demonstrate the efficacy of TraCe when applied to NeRF, using the same experimental setup as our 3DGS experiments but substituting the 3D representation.

**Quantitative Evaluation.** We conduct a quantitative comparison of TraCe (using NeRF) against the widely-used baselines, SDS [12] and VSD [15]. We evaluate all methods using GPTEval3D (leveraging GPT-4o) and ImageReward. As shown in Table 7, TraCe significantly outperforms both baselines across every metric in the GPTEval3D evaluation. Notably, TraCe achieves the highest Overall score (1179.86), surpassing VSD (1012.50) and SDS (1000.00). This trend holds for all sub-metrics, including Plausibility (1289.89), Texture Details (1127.06), and Geometry Details (1228.94), indicating a superior generation of realistic and detailed 3D assets. This superior performance is further validated by the ImageReward evaluation, presented in Table 8. TraCe achieves the best ImageReward score (-0.4533), a substantial improvement over VSD (-0.6737) and SDS (-0.8838). In terms of computational cost, TraCe (at $\sim$3.5 hours) provides a much better quality-to-time trade-off than VSD ($\sim$5 hours) and, while slower than the rapid SDS ($\sim$1 hour), it delivers vastly superior results as confirmed by both quantitative metrics.

**Qualitative Evaluation.** In Figure 2, we present a qualitative comparison for the prompt "a croissant". This visualization highlights the clear qualitative advantages of TraCe when applied to NeRF. Our method generates a high-fidelity rendering with well-defined geometry and a clean, coherent alpha mask. This stands in contrast to failure modes observed in baseline methods, such as the severe geometric noise and artifacting seen in the center example.

Table 7: **Qualitative comparisons of TraCe using NeRF under GPTEval3D evaluation.**

| | Plausibility | Texture Details | Geometry Details | Text-Asset Alignment | Text-Geometry Alignment | Overall |
|---|---|---|---|---|---|---|
| **SDS (~1h)** | 1000.00 | 1000.00 | 1000.00 | 1000.00 | 1000.00 | 1000.00 |
| **VSD (~5h)** | 1261.80 | 1058.73 | 1152.00 | 1246.37 | 1180.56 | 1012.50 |
| **Trace (ours, ~3.5h)** | 1289.89 | 1127.06 | 1228.94 | 1284.60 | 1245.06 | 1179.86 |

Table 8: **Quantitative comparison under ImageReward evaluation.**

| Method | ImageReward ↑ |
|---|---|
| **SDS (~1h)** | -0.8838 $\pm$ 0.5142 |
| **VSD (~5h)** | -0.6737 $\pm$ 0.4098 |
| **TraCe (ours, ~3.5h)** | -0.4533 $\pm$ 0.4635 |

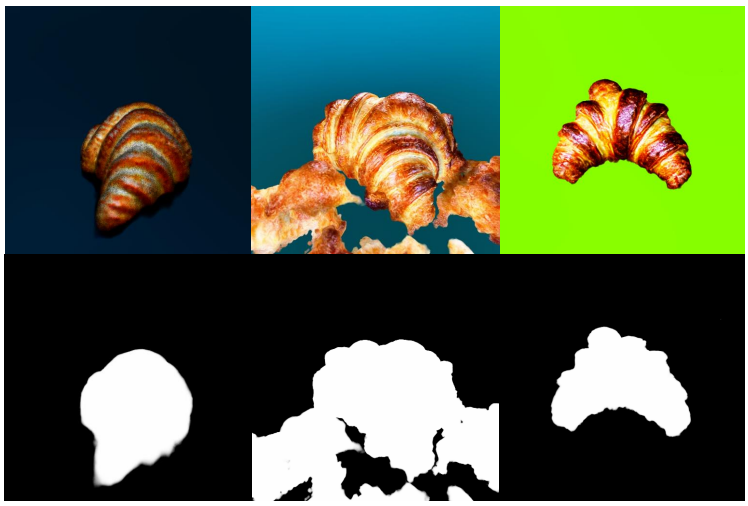

Figure 2: **Qualitative comparison on NeRF for "A delicious croissant".** Left: SDS; Middle: VSD; Right: TraCe.

## H 2D Experiments

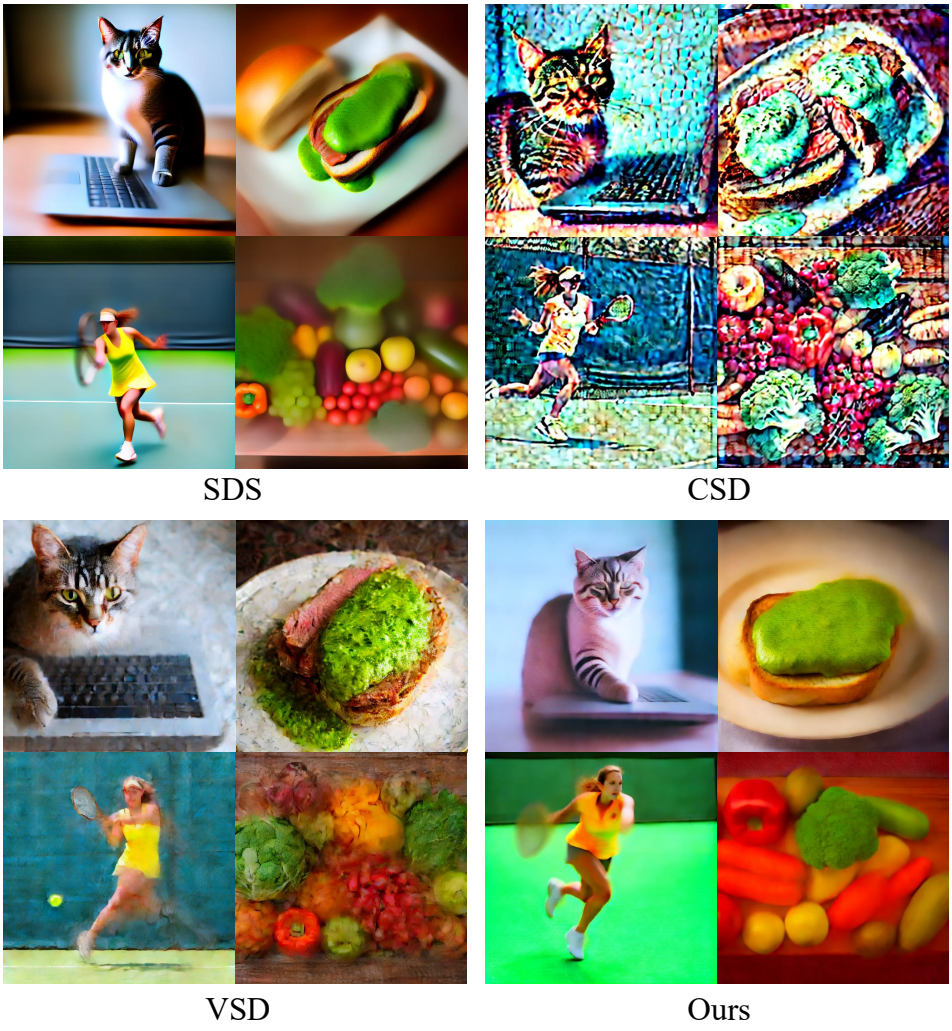


Figure 3: **Text-to-image generation results with COCO Captions.** Our method ("Ours") is compared with SDS [12], CSD [19], and VSD [15]. The visual examples demonstrate our model's improved image quality.

Figure 3 provides a qualitative comparison of our method against SDS [12], CSD [19], and VSD [15] on text-to-image generation using COCO Captions [2]. Visual inspection reveals that SDS [12] often introduces noise and lacks fine-grained detail. CSD [19] tends to produce stylized outputs, potentially sacrificing photorealism for artistic effect. While VSD [15] shows improved coherence, it can yield results with reduced sharpness and vibrancy. In contrast, our proposed method consistently generates images with enhanced visual fidelity, clarity, and detail. For example, across diverse subjects such as animals, food items, and human figures in scenes, our approach yields more realistic textures, sharper details, and more naturalistic overall compositions compared to other methods. These results suggest our model's superior capability in capturing complex image statistics and generating high-quality, coherent images from textual descriptions, beyond the 3D generation realm.

[2] https://cocodataset.org/#explore

# I More Results of Gradient and Intermediate Rendering Comparison

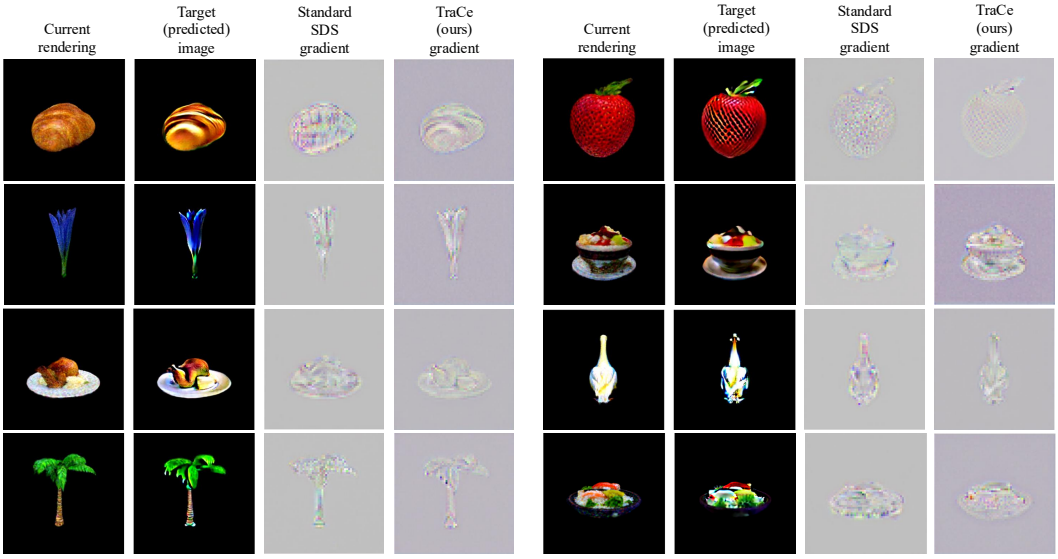


Figure 4: **Gradient and intermediate rendering comparison.** Note the reduced artifacts and potentially more coherent structure in the TraCe gradients and intermediate renderings.

To further investigate the optimization dynamics, we visualize and compare the gradients derived from our TraCe framework against those from standard SDS [18] in Figure 4. Across various examples, the gradients produced by SDS [18] (third column) often appear noisy and diffuse, lacking clear structural definition that aligns with the target (predicted) image (second column). In contrast, the gradients generated by TraCe (fourth column) consistently exhibit a notably cleaner and more discernible structure. These TraCe gradients demonstrate significantly enhanced coherence with the features of the target image, suggesting more direct and effective guidance. This improved gradient quality contributes to TraCe's ability to optimize 3D representations with greater fidelity and reduced artifacts, as the optimization is steered by more precise and relevant guidance at each step.

# J Diversity of the Generations

We assess the generation diversity of TraCe in Figure 5. For prompts like "a zoomed out DSLR photo of an amigurumi motorcycle" or "an overstuffed pastrami sandwich", TraCe produces multiple distinct 3D instances, showcasing variations in geometry, structure, and style. These results indicate TraCe's ability to sample varied, plausible 3D assets from the conditional distribution. Notably, while TraCe demonstrates useful diversity, this aspect is not presented as a significant advancement, which aligns with that typically reported by other score distillation-based methods [12, 15, 8].

Figure 5: **Generation diversity of TraCe.** Our method produces varied 3D assets from text prompts—"a zoomed out DSLR photo of an amigurumi motorcycle" (top) and "an overstuffed pastrami sandwich" (bottom).

## K Comparisons with Native 3D Generative Models

Recent advancements in native 3D generative models have enabled remarkable generation speeds, often producing 3D assets within seconds [5, 6, 16, 1]. However, despite this rapid synthesis, the output quality from these feed-forward models frequently remains limited, with textual coherence typically achieved only at a coarse level. Consequently, the 3D assets generated by such Native 3D Generative Models often cannot be directly used due to their low quality. Taking DiffSplat [5] as a prominent example, its generated 3D assets, while produced very quickly (e.g., in approximately 14 seconds as noted for results in Figure 6), tend to exhibit a visibly rough and low-fidelity appearance. This characteristic trade-off between speed and quality opens an important avenue for future investigation: the potential utility of these native 3D models as efficient initializers. Analogous to how foundational models like Shap-E [2] or Point-E [11] are employed to generate preliminary 3D structures, these fast native generators could provide a coarse but rapidly obtained starting point. Subsequently, optimization-based 3D generative frameworks, such as the TraCe methodology proposed herein, could be applied for further refinement to achieve higher levels of detail and overall fidelity. It is noteworthy that the concept of leveraging DiffSplat for initialization followed by a refinement stage (e.g., TraCe) is also an indicated direction in DiffSplat's official project [3], and could be a promising direction for future research.

[3] https://chenguolin.github.io/projects/DiffSplat/


Figure 6: **Comparisons with DiffSplat [5].** We employ three different pretrained DiffSplat models: DiffSplat (SD1.5), DiffSplat (PixArt-Sigma), and DiffSplat (SD3.5m). DiffSplat's assets show lower visual fidelity than TraCe's. For instance, DiffSplat's teacups are often simplistic or imperfect, while TraCe renders highly detailed, vibrant versions. Similarly, its hotdogs lack the textural richness of TraCe's. This trend of superior detail and complexity in TraCe's results is consistent across examples. While DiffSplat is faster, these comparisons highlight the enhanced quality achieved by TraCe's optimization-based, direct transport trajectory approach.

## L More Qualitative Comparisons


Figure 7: **Qualitative Comparisons with GaussianDreamer [18] (SDS) (left) and TraCe (right).**

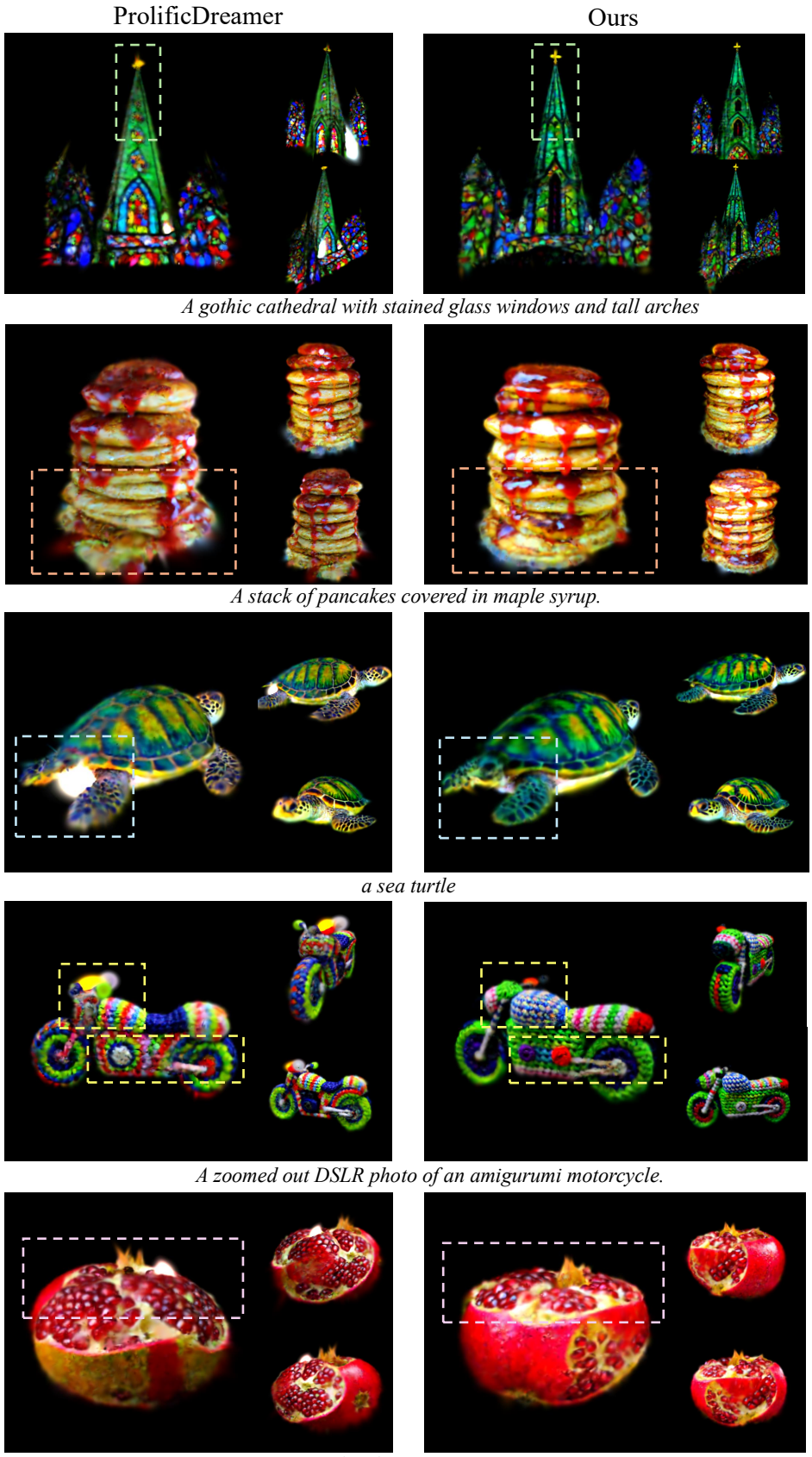


Figure 8: **Qualitative Comparisons with ProlificDreamer [15] (VSD) (left) and TraCe (right).**

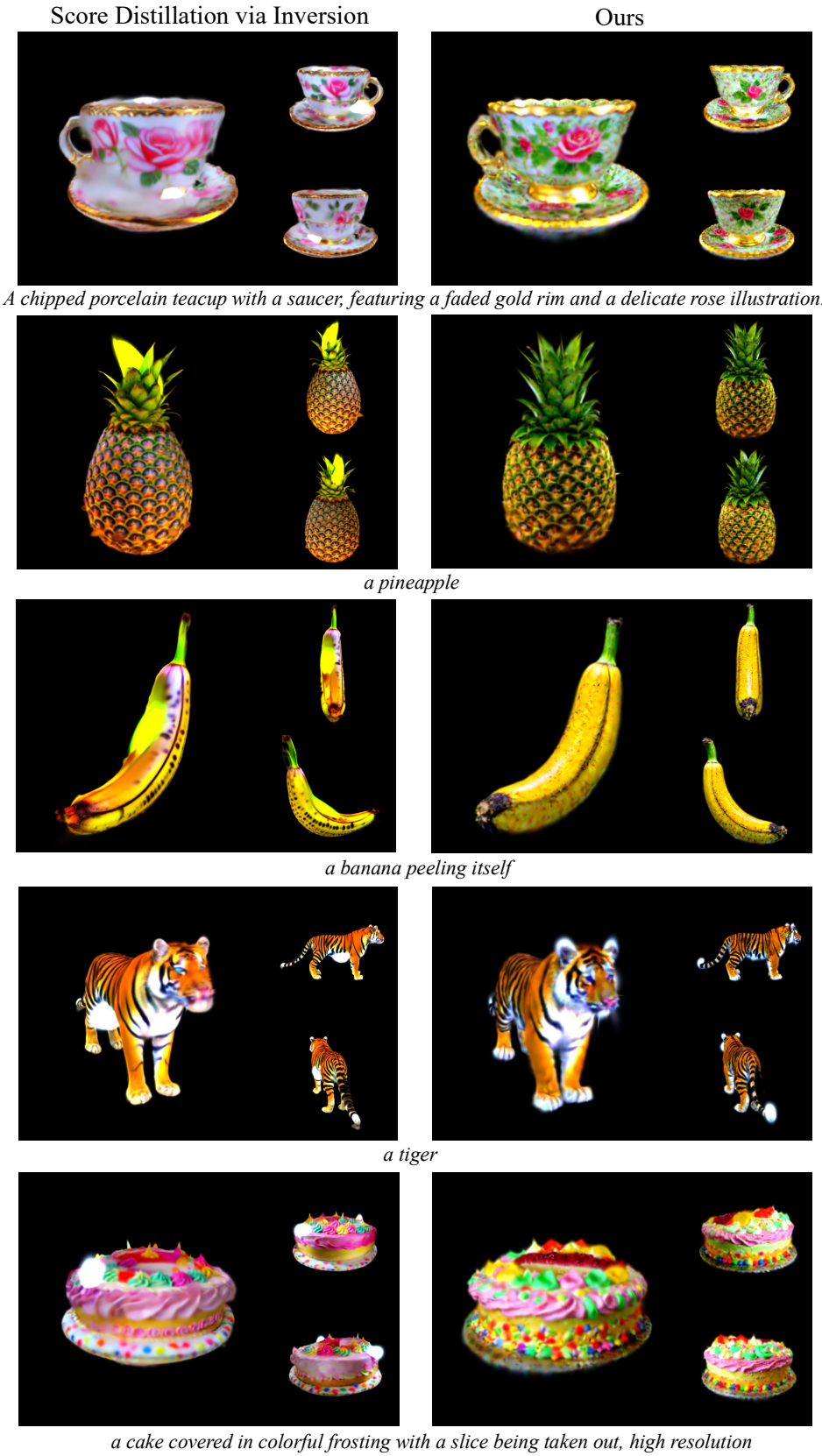


Figure 9: **Qualitative Comparisons with Score Distillation via Inversion [8] (SDI) (left) and TraCe (right).**

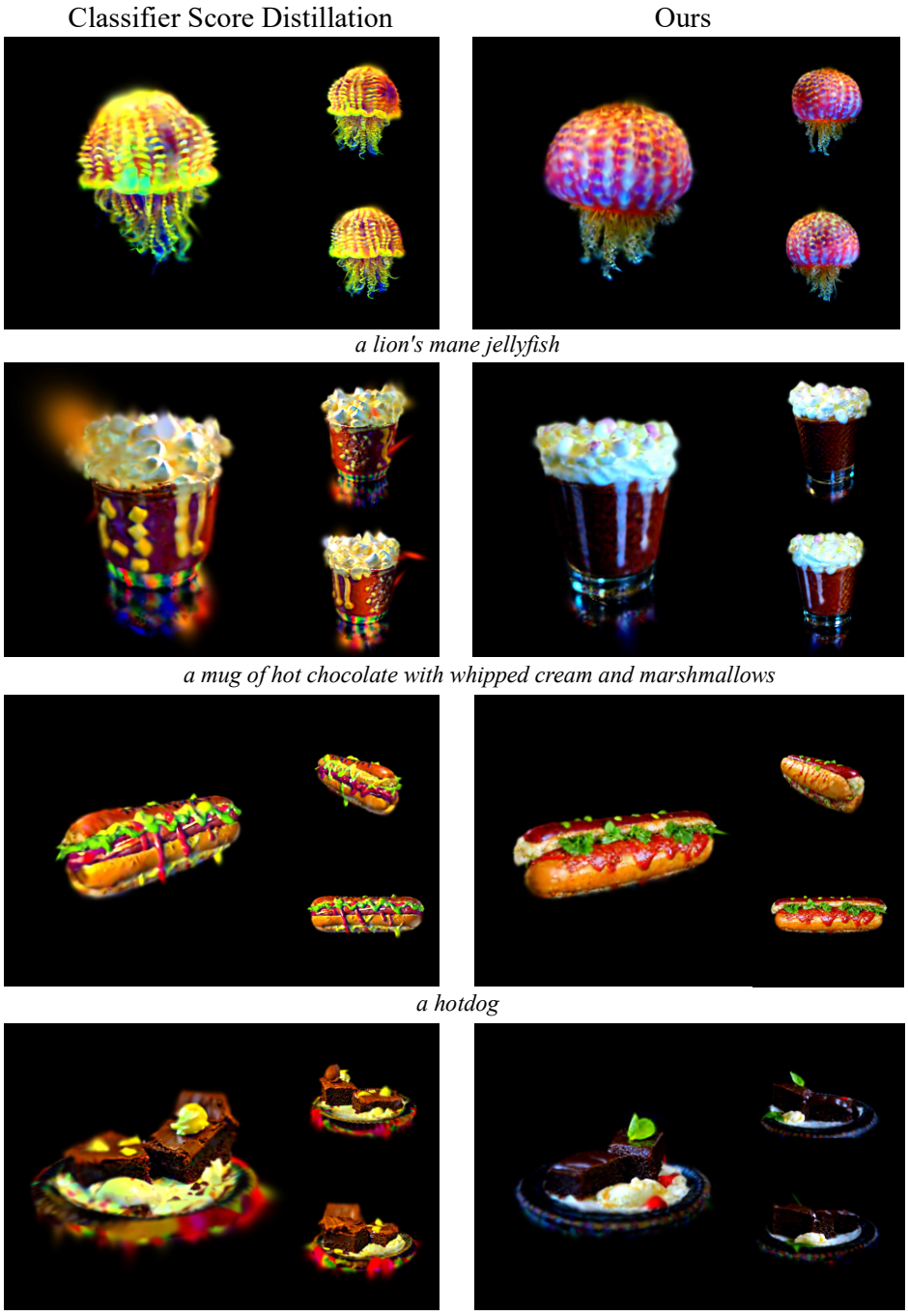


Figure 10: **Qualitative Comparisons with Classifier Score Distillation [19] (CSD) (left) and TraCe (right).**


Figure 11: **Qualitative Comparisons with LucidDreamer [4] (ISM) (left) and TraCe (right).**

## References


[1] Xianglong He, Junyi Chen, Sida Peng, Di Huang, Yangguang Li, Xiaoshui Huang, Chun Yuan, Wanli Ouyang, and Tong He. Gvgen: Text-to-3d generation with volumetric representation. In *European Conference on Computer Vision*, pages 463–479. Springer, 2024.

[2] Heewoo Jun and Alex Nichol. Shap-e: Generating conditional 3d implicit functions. *arXiv preprint arXiv:2305.02463*, 2023.

[3] Kyungmin Lee, Kihyuk Sohn, and Jinwoo Shin. Dreamflow: High-quality text-to-3d generation by approximating probability flow. *arXiv preprint arXiv:2403.14966*, 2024.

[4] Yixun Liang, Xin Yang, Jiantao Lin, Haodong Li, Xiaogang Xu, and Yingcong Chen. Luciddreamer: Towards high-fidelity text-to-3d generation via interval score matching. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6517–6526, 2024.

[5] Chenguo Lin, Panwang Pan, Bangbang Yang, Zeming Li, and Yadong Mu. Diffsplat: Repurposing image diffusion models for scalable gaussian splat generation. *arXiv preprint arXiv:2501.16764*, 2025.

[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.

[7] Ying-Tian Liu, Yuan-Chen Guo, Vikram Voleti, Ruizhi Shao, Chia-Hao Chen, Guan Luo, Zixin Zou, Chen Wang, Christian Laforte, Yan-Pei Cao, et al. threestudio: a modular framework for diffusion-guided 3d generation. *cg. cs. tsinghua. edu. cn*, 2023.

[8] Artem Lukoianov, Haitz Sáez de Ocáriz Borde, Kristjan Greenewald, Vitor Guizilini, Timur Bagautdinov, Vincent Sitzmann, and Justin M Solomon. Score distillation via reparametrized ddim. *Advances in Neural Information Processing Systems*, 37:26011–26044, 2024.

[9] Xingyu Miao, Haoran Duan, Varun Ojha, Jun Song, Tejal Shah, Yang Long, and Rajiv Ranjan. Dreamer xl: Towards high-resolution text-to-3d generation via trajectory score matching. *arXiv preprint arXiv:2405.11252*, 2024.

[10] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021.

[11] Alex Nichol, Heewoo Jun, Prafulla Dhariwal, Pamela Mishkin, and Mark Chen. Point-e: A system for generating 3d point clouds from complex prompts. *arXiv preprint arXiv:2212.08751*, 2022.

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

[13] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.

[14] Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. *arXiv preprint arXiv:2202.00512*, 2022.

[15] Zhengyi Wang, Cheng Lu, Yikai Wang, Fan Bao, Chongxuan Li, Hang Su, and Jun Zhu. Prolificdreamer: High-fidelity and diverse text-to-3d generation with variational score distillation. *Advances in Neural Information Processing Systems*, 36:8406–8441, 2023.

[16] Shuang Wu, Youtian Lin, Feihu Zhang, Yifei Zeng, Jingxi Xu, Philip Torr, Xun Cao, and Yao Yao. Direct3d: Scalable image-to-3d generation via 3d latent diffusion transformer. *arXiv preprint arXiv:2405.14832*, 2024.

[17] Zike Wu, Pan Zhou, Xuanyu Yi, Xiaoding Yuan, and Hanwang Zhang. Consistent3d: Towards consistent high-fidelity text-to-3d generation with deterministic sampling prior. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9892–9902, 2024.

[18] Taoran Yi, Jiemin Fang, Junjie Wang, Guanjun Wu, Lingxi Xie, Xiaopeng Zhang, Wenyu Liu, Qi Tian, and Xinggang Wang. Gaussiandreamer: Fast generation from text to 3d gaussians by bridging 2d and 3d diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6796–6807, 2024.

[19] Xin Yu, Yuan-Chen Guo, Yangguang Li, Ding Liang, Song-Hai Zhang, and Xiaojuan Qi. Text-to-3d with classifier score distillation. *arXiv preprint arXiv:2310.19415*, 2023.