

SVGDreamer: Text Guided SVG Generation with Diffusion Model

摘要

近年来，基于文本的可扩展矢量图形（SVG）生成在图标设计和草图绘制等领域展现出潜力。然而，现有的文本到 SVG 生成方法在可编辑性、视觉质量和结果多样性方面存在不足。为了解决这些问题，论文提出了一种新颖的文本引导矢量图形生成方法，称为 SVGDreamer。SVGDreamer 引入了一个基于语义驱动的图像矢量化（SIVE）过程，使生成过程能够分解为前景对象和背景，从而增强了可编辑性。具体而言，SIVE 过程引入了基于注意力的图形控制和一种注意力掩码损失函数，以实现对各个元素的有效控制和操作。

此外，论文提出了一种矢量化的基于粒子的得分蒸馏（VPSD）方法，旨在解决现有文本到 SVG 生成方法中的形状过度平滑、颜色过度饱和、结果多样性有限以及收敛速度慢等问题。该方法通过将 SVG 表示为控制点和颜色的分布来进行建模。同时，VPSD 利用一个奖励模型对矢量粒子进行重新加权，从而提升美学吸引力并加速收敛。

论文通过大量实验验证了 SVGDreamer 的有效性，结果表明，与基线方法相比，SVGDreamer 在可编辑性、视觉质量和多样性方面具有显著优势。

关键词：文本到矢量图；分数蒸馏采样；扩散模型；SVG

1 引言

最近，文本引导的可缩放矢量图形（SVG）合成在图像学和素描等领域显示出前景。然而，现有的文本到 SVG 的生成方法缺乏可编辑性，并且在视觉质量和结果多样性方面存在问题。因此，论文的研究难点是如何在不牺牲视觉质量的情况下提高生成 SVG 的可编辑性，以及如何解决现有方法中存在的形状过度平滑、颜色过度饱和、多样性有限和收敛速度慢的问题。

论文相关工作包括基于优化的矢量图形生成方法、文本到图像扩散模型以及分数蒸馏采样方法。现有的方法如 CLIPDraw [1]、VectorFusion [3] 和 DiffSketcher [15] 等在视觉质量和多样性方面仍存在不足。为解决上述存在的局限，论文提出了一种新的文本引导的矢量图形合成方法 SVGDreamer。

整体而言，论文的贡献主要有三点：(1) 引入了 SVGDreamer，这是一种用于文本到 SVG (Text-to-Vector, T2V) 生成的新模型。这种新模型能够在保持可编辑性的同时生成高质量的矢量图形；(2) 提出了语义驱动的图像矢量化 (Semantic-driven Image Vectorization, SIVE) 方法，该方法确保了生成的矢量对象是独立的和灵活的编辑。此外，提出了矢量化的基于粒子的分数蒸馏 (Vectorized Particle-based Score Distillation, VPSD) 损失，以保证生成的矢

量图形具有卓越的视觉质量和广泛的多样性；(3) 进行了全面的实验来评估提出的方法的有效性。结果表明，与基线方法相比，论文的方法具有优越性。此外，论文的模型在生成不同类型的矢量图形方面显示出强大的泛化能力。

2 相关工作

2.1 矢量图形生成

矢量图形（SVG）使用几何原语（如贝塞尔曲线、多边形和线条）来表示视觉概念。由于它们的固有特性，SVG 非常适合于海报和标志等视觉设计应用。与栅格图像相比，矢量图像可以保持紧凑的文件大小，更高效地进行存储和传输，并且具有更高的可编辑性。(1) 序列到序列模型：早期的矢量图形生成方法使用序列到序列（seq2seq）模型来生成 SVG 内容，但这些方法严重依赖于矢量格式的数据集，限制了它们的泛化能力和生成复杂矢量图形的能力。(2) 可微渲染器：Li 等人 [4] 引入了一种可微渲染器，将矢量图形和栅格图像领域连接起来。这种方法允许在评估时优化以匹配图像，克服了直接学习 SVG 生成网络的局限性。(3) 视觉文本嵌入对比语言图像预训练模型（CLIP [9]）：CLIP 的成功促使了一系列成功的草图合成方法的发展，如 CLIPDraw [1]。(4) VectorFusion [3] 和 DiffSketcher [15]：这些方法结合了可微渲染器和文本到图像扩散模型，用于矢量图形生成，取得了在图标设计、像素艺术和草图等领域的良好效果。

2.2 文本到图像的扩散模型

扩散概率模型（DDPMs），特别是那些基于文本的条件模型，在文本到图像合成中显示出良好的效果。例如，Classifier-Free Guidance (CFG) 提高了视觉质量，并在大规模文本条件扩散模型框架中得到广泛应用。(1) GLIDE [7]、Stable Diffusion [11]、DALL-E 2 [10] 和 Imagen [12]：这些模型利用 CLIP 作为监督源，生成高质量的图像。(2) 文本到图像：文本到图像扩散模型的进展也促进了文本引导任务的开发，如 text-to-3D [8]。

2.3 分数蒸馏采样

分数蒸馏采样（SDS [8]）损失从 2D 文本到图像扩散模型中派生出来，用于恢复 3D 对象结构。这种方法在文本到 3D 生成中表现出色。(1) DreamFusion [8]、Magic3D [5] 和 Score Jacobian Chaining [13]：这些方法利用 SDS 损失进行文本到 3D 生成，取得了令人深刻的结果。(2) 变分分数蒸馏：Wang 等人 [14] 扩展了 3D 模型的建模，将其视为随机变量而非常数，以解决文本到 3D 生成中的过平滑问题。

3 本文方法

3.1 本文方法概述

SVGDreamer 整体由两部分组成（如图1所示），即基于 SIVE 生成和文本提示匹配的矢量图形，目的是提高所生成的矢量图形的可编辑性，以及通过 VPSD 优化 SIVE 方法所合成的矢量图形，目的是提高所生成的矢量图形的视觉质量和结果多样性。

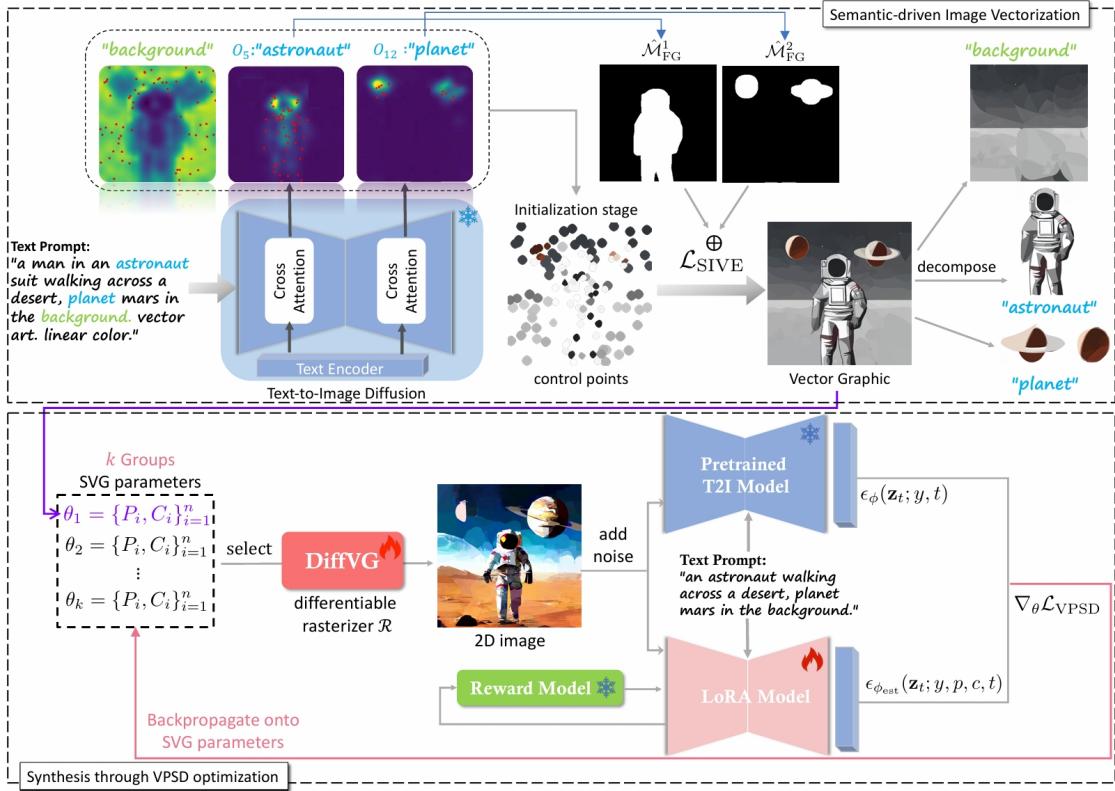


图 1. SVGDreamer 方法概览

3.2 SIVE

图像光栅化是计算机图形学中的一种成熟技术，而图像矢量化是光栅化的反向路径，仍然是一个重大挑战。已有办法 LIVE [6] 给定任意输入图像，通过添加新的可优化的封闭贝塞尔路径并优化所有这些路径来递归地学习视觉概念。然而，LIVE 很难在图像中捕捉和区分不同的主题，导致相同的路径被叠加到不同的视觉主题上。

因此，论文提出了一个语义驱动的图像向量化过程，将合成过程分解为前景对象和背景，以解决已有方法所存在的不足。该过程包括两个阶段：原始初始化和语义感知优化。

(1) 原始初始化：根据文本提示中不同对象的交叉注意力图来初始化每个前景对象和背景的控制点。

(2) 语义感知优化：将初始化阶段获得的注意力图转换为可重复使用的掩码，然后用掩码对指定的对象内的控制点进行优化。目的是利用掩码优化矢量元素，确保控制点在各自区域，即层次结构只存在指定对象，从而增强可编辑性。

3.3 VPSD

为了解决现有方法 SDS [8] 中存在的形状过度平滑、颜色过度饱和、多样性有限和收敛速度慢的问题，论文提出了向量化的粒子基分数蒸馏方法，如图 2.2 所示。该方法通过以下方面进行优化：

(1) 矢量粒子建模：与传统方法将 SVG 表示为固定的控制点和颜色集合不同，VPSD 将 SVG 建模为控制点分布和颜色分布。具体来说，给定文本提示 y ，假设存在一个概率分布 μ 表示所有可能的矢量图形表示。在矢量参数化表示 θ 下，该分布可被建模为 $\mu(\theta|y)$ 。

(2) 预训练扩散模型与 LoRA 网络：引入低秩适配（LoRA），通过对预训练扩散模型的高效调整估计控制点和颜色的分布。仅更新 LoRA 网络的参数，而不改变其他扩散模型的参数，从而降低计算复杂度。

(3) 奖励反馈学习：使用预训练的奖励模型（如 ReFL）为生成的矢量粒子分配奖励分数，过滤出高质量样本。具体来说，首先利用 LoRA 生成多组样本；然后使用奖励模型对每组样本评分，选择高分样本作为下一轮优化的基础；最后通过 DDIM 采样加速早期迭代，进一步减少优化时间。

(4) VPSD 的优化目标：VPSD 的最终损失函数由以下三部分组成。

$$\min_{\theta} r_{\theta} L_{\text{VPSD}} + L_{\text{LoRA}} + \lambda_r L_{\text{reward}} \quad (1)$$

其中， L_{VPSD} 通过对比真实噪声分布和预测噪声分布计算梯度，优化控制点和颜色的分布； L_{LoRA} 针对 LoRA 网络的参数调整，以适配特定任务； L_{reward} 引入奖励分数对样本进行筛选，进一步优化结果。

3.4 风格控制

除了文本提示外，SVGDreamer 还提供了多种矢量表示形式，用于风格控制。这些矢量表示通过限制图形原语类型及其参数实现。用户可以通过修改输入文本，或通过约束图形原语和参数的集合，来控制 SVGDreamer 生成的艺术风格，如图2所示。

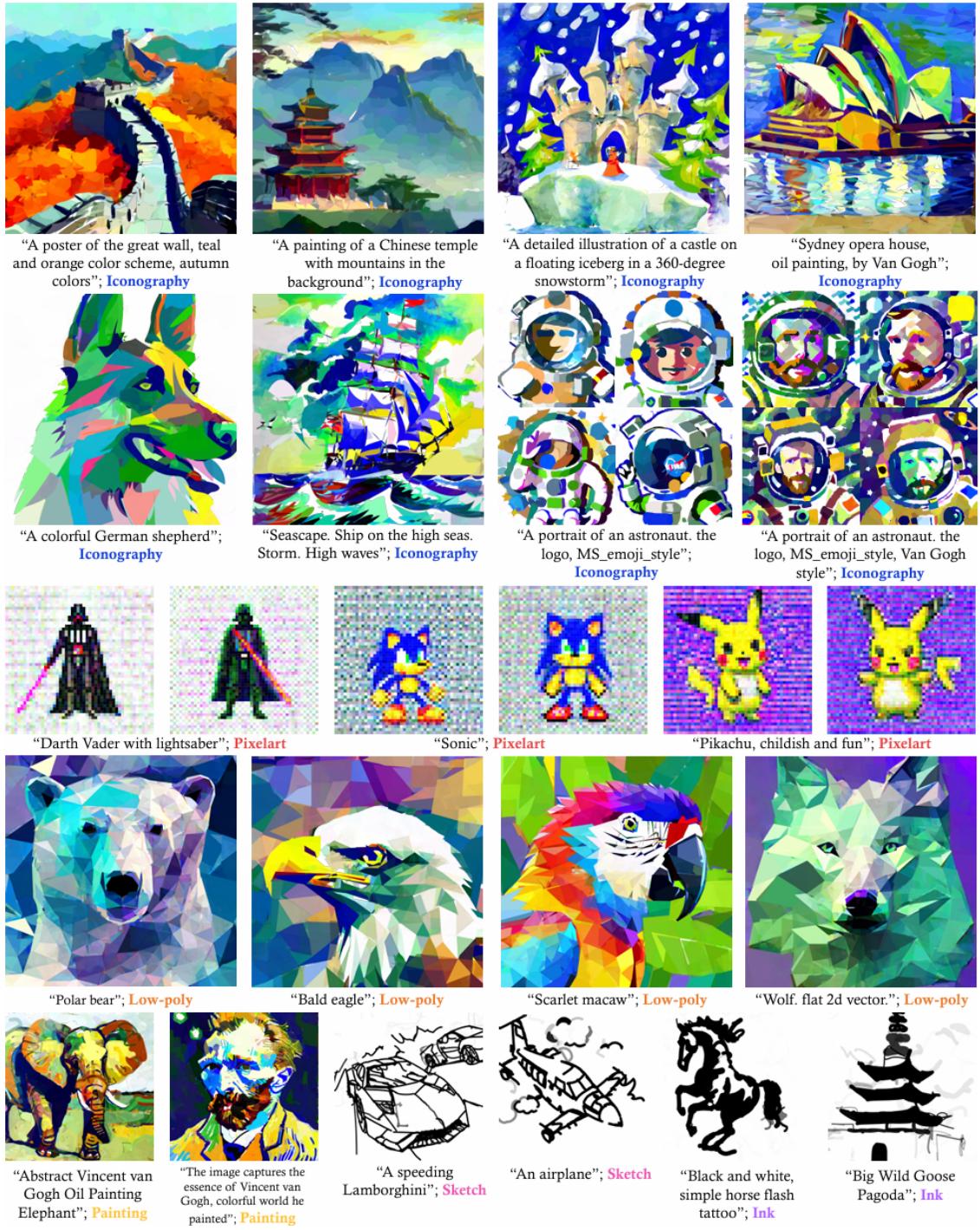


图 2. 给定一个文本提示，SVGDreamer 可以生成不同风格的矢量图形

4 复现细节

4.1 与已有开源代码对比

该论文已开源项目代码，链接如下：<https://github.com/ximinng/SVGDreamer>。本次复现是基于开源项目所生成的结果，从自底向上的思想出发，利用矢量编辑方法（Paper.js）来尝试对 SVGDreamer 生成的结果进行优化，以弥补结果中所存在的不足和局限。

4.2 实验环境搭建

该论文的项目适用在 Linux 系统环境下进行部署和运行，因此本次复现工作选择在 Ubuntu 20.04 版本下的环境进行部署。部署步骤参考项目主页的说明。

(1) 安装环境：在顶级目录中，运行命令“sh script/install.sh”，或者适用 docker 进行安装，即“docker run -name svgdreamer -gpus all -it -ipc=host ximingxing/svgrender:v1 /bin/bash”。

(2) 下载预训练的稳定扩散模型：首次运行时，通过在“/conf/config.yaml”中设置“diffuser.download=True”来下载预训练的 SD 模型，或者去模型库中手动下载，再存放到默认位置。

4.3 生成测试

在项目主页，作者提供了若干案例，包括文本提示（prompt）和参数设置。本次复现主要选取了最为常用的“Iconography”为生成风格，方法上分别选择了 SIVE 和 VPSD 以及单独的 SIVE 作为实验。此外，为降低训练成本将参数“n_particle”设为 4，表明在 VPSD 前生成 4 张用于采样的光栅图以及对应的 SIVE 结果。

4.3.1 SIVE 和 VPSD

首先，参考提供的 prompt 进行生成实验，即“an image of Batman. full body action pose, complete detailed body, white background, high quality, 4K, ultra realistic”，完整命令为“python svgdreamer.py x=iconography skip_sive=False ”prompt='an image of Batman. full body action pose, complete detailed body. white background. empty background, high quality, 4K, ultra realistic'” token_ind=4 x.vpsd.t_schedule='randint' result_path='./logs/batman'”。然后，对 prompt 进行修改以验证泛化效果，设置为“A pizza; Cartoon simple style; Center; The background is either transparent or blank”。进行了多轮生成，部分生成的结果如图3所示。



图 3. “蝙蝠侠”以及“披萨”的部分矢量图形生成结果 (SIVE 和 VPSD)

4.3.2 SIVE

参考项目主页提供的命令，即“`python svgdreamer.py x=iconography-s1 skip_sive=False`”`prompt='an astronaut walking across a desert, planet mars in the background, floating beside planets, space art'`”`token_ind=5 result_path='./logs/astronaut_sive'` `seed=116740`”，以验证 SIVE 方法单独的生成效果。部分生成结果如图4所示。

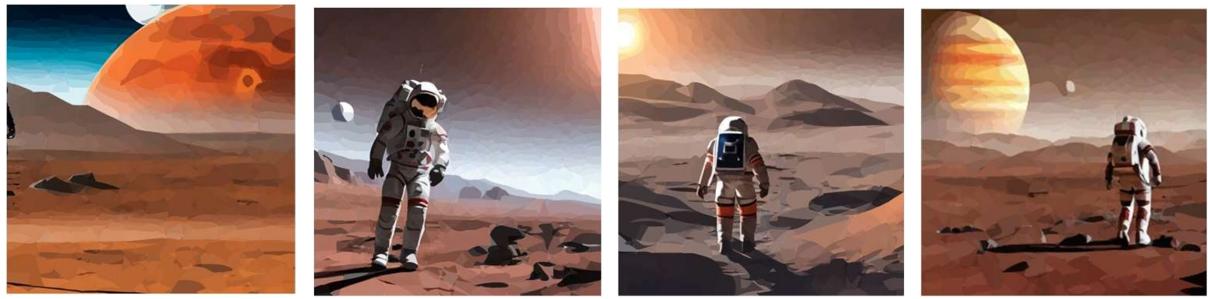


图 4. “宇航员”矢量图形生成结果 (SIVE)

4.4 创新点

SVGDreamer 在方法上属于基于图像的方法，从大量随机初始化的形状开始，使用可微渲染器 [4] 渲染矢量图形，并通过最小化文本图像相似度损失来迭代优化路径参数。尽管该类方法的结果看起来赏心悦目，但它们生成的 SVG 结果通常包含多个缺乏独立语义对应的碎片路径，这对于实际设计场景来说是不可取的。这就是 SVGDreamer 方法所存在的一个显著局限，即会产生具有有限语义的碎片路径。

针对 SVGDreamer 会产生形状重叠的缺陷，本文从自底向上的思想出发直接对 SVG-Dreamer 生成的结果进行优化，尝试利用矢量编辑方法对相似路径进行聚合，以提高生成结果的可编辑性。具体来说，本文基于 Paper.js 库对生成矢量图，即 SVG 中的路径进行处理和操作。操作逻辑为：(1) 遍历整个 SVG 图形结构，提取所有路径对象并存储在列表中，便于后续操作；(2) 对路径进行两两比较，判断是否邻近（边界框相交且最短距离在阈值范围内）以及颜色相似（通过 RGB 值计算相似度），如果两个路径满足邻近和颜色相似的条件，将其归为一组，并尝试合并路径，以及调整路径的颜色为组内路径颜色的平均值；(3) 遍历分组后的路径集合，对每个组中的路径进行颜色中和操作以及路径合并操作，并移除处理完成的路径，避免重复处理。

5 实验结果分析

5.1 论文实验结果

5.1.1 定性评估

SVGDreamer 和现有的文本到 SVG 方法之间进行了定性比较，如图5所示。和 CLIP-Draw [1] 相比较：强调了 SVGDreamer 解决形状过度平滑和颜色过饱和度等问题的能力，即具有更高的保真度和细节。和基于 SDS [8] 的方法 (VectorFusion [3] 和 DifferSketch [15]) 相比较：SVGDreamer 具有更优越的细节。此外，通过第 5 行可知，SIVE 只能解决语义解耦，SIVE 实现了语义解耦，但不能克服 SDS 方法固有的平滑特性。



图 5. 不同方法的定性比较

5.1.2 定量评估

从 6 个指标对 SVGDreamer 进行定量评估，如表1所示。其中，FID 和 PSNR 的结果表明 SVGDreamer 具有更大范围的多样性；CLIPScore 和 BLIPScore 的结果表明 SVGDreamer 在 SVG 和 prompt 的一致性的表现上更为优越；美学分数（Aesthetic）表明 SVGDreamer 具有更高的矢量图形感知质量；人类表现评分（HPS）表明 SVGDreamer 的结果在人类美学的角度上表现出色。

表 1. 不同方法的定量比较

| Method / Metric | FID | PSNR ↑ | CLIPScore ↑ | BLIPScore ↑ | Aesthetic ↑ | HPS ↑ |
|--------------------------------|--------------|--------------|---------------|---------------|---------------|---------------|
| CLIPDraw [1] | 160.64 | 8.35 | 0.2486 | 0.3933 | 3.9803 | 0.2347 |
| VectorFusion (scratch) [3] | 119.55 | 6.33 | 0.2298 | 0.3803 | 4.5165 | 0.2334 |
| VectorFusion [3] | 100.68 | 8.01 | 0.2720 | 0.4291 | 4.9845 | 0.2450 |
| DiffSketcher (RGB) [15] | 118.70 | 6.75 | 0.2402 | 0.4185 | 4.1562 | 0.2423 |
| SVGDreamer (from scratch) [16] | 84.04 | 10.48 | 0.2951 | 0.4311 | 5.1822 | 0.2484 |
| +Reward Feedback | 83.21 | 10.51 | 0.2988 | 0.4335 | 5.2825 | 0.2559 |
| SVGDreamer | 59.13 | 14.54 | 0.3001 | 0.4623 | 5.5432 | 0.2685 |

5.2 复现实验结果

观察实验结果图3和图4可知，SVGDreamer 会创建由多个锯齿状、不规则和碎片状或重叠的路径来组成对象，这些路径只有在作为一个整体来看时才会具有意义，而这阻碍了图形设计师能更方便地去对矢量图形进行编辑，即 SVGDreamer 所生成的结果具有低编辑性。此外，SVGDreamer 对象分割效果仍较一般，从而导致可能会存在内容上的缺失，如图6所示。

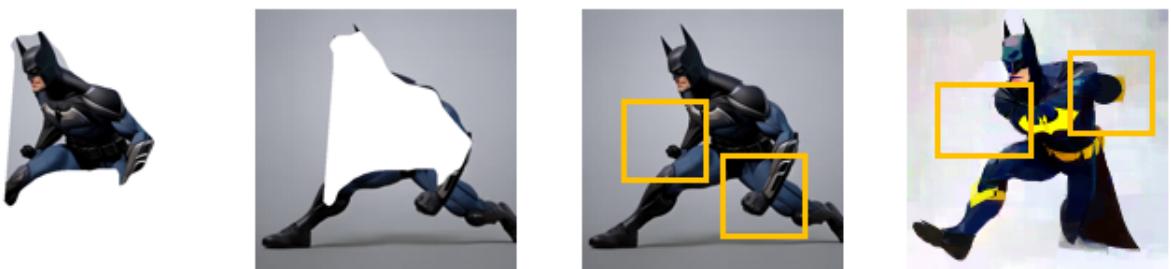


图 6. SVGDreamer 可能会存在的内容缺失

基于上文的优化逻辑，使用 Paper.js 对 SVGDreamer 生成的矢量图结果进行了优化，实验结果如图7所示。由结果可知，经过处理后的 SVG 图像在局部的路径重叠现象显著减少。

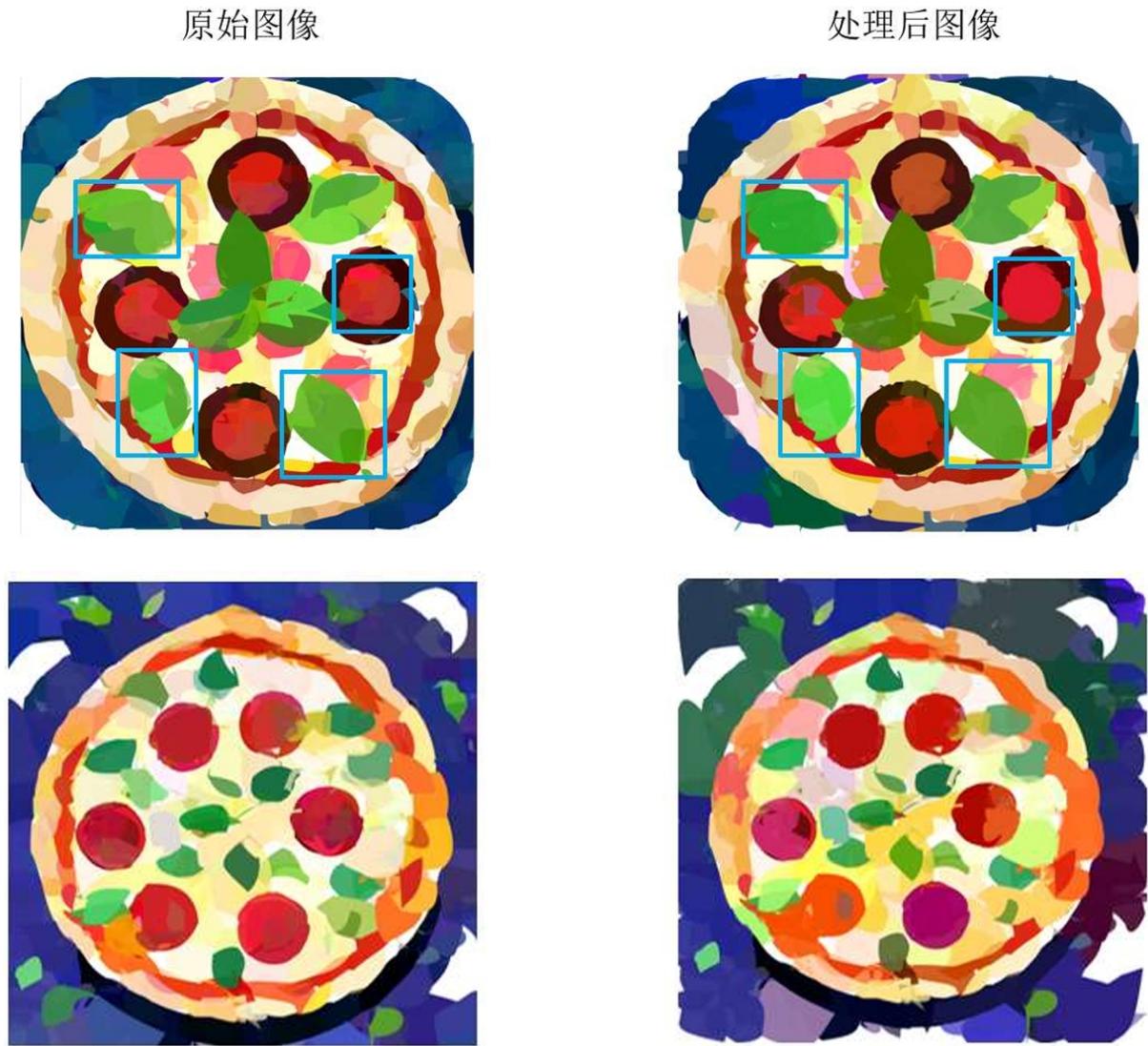


图 7. 矢量编辑优化实验结果

6 总结与展望

本文首先概述了复现论文的研究背景、相关工作以及提出的方法等基础内容；然后，基于复现论文的源代码进行了项目的复现和实验，并展示和分析了实验结果；最后，基于复现论文存在的局限提出了优化方案，并进行了优化实验以及结果分析。

由优化结果可知，目前方法仍存在明显缺陷，比如从整体上无法完全消除形状的碎片化效果，以及产生了颜色与原图像存在偏差的新问题，此外也无法解决 SVGDreamer 形状不规则的缺陷。现在已经有许多其它基于分数蒸馏的工作开始针对形状碎片化、不规则的局限进行了优化，例如 O&R [2] 以及 T2V-NPR [17]。T2V-NPR 提出了一个新的神经路径方法去解决路径优化问题，而 O&R 则是自顶向下逐步优化路径和删除冗余路径。两个方法都具有各自的特点和可参考价值，但其中 T2V-NPR 暂未公开代码，有待进一步的复现，而 O&R 则是基于矢量化的方法，有待后续迁移到 T2V 的方法中。

参考文献

- [1] Kevin Frans, Lisa Soros, and Olaf Witkowski. Clipdraw: Exploring text-to-drawing synthesis through language-image encoders. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 5207–5218. Curran Associates, Inc., 2022.
- [2] Or Hirschorn, Amir Jevnisek, and Shai Avidan. Optimize & reduce: A top-down approach for image vectorization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 2148–2156, 2024.
- [3] Ajay Jain, Amber Xie, and Pieter Abbeel. Vectorfusion: Text-to-svg by abstracting pixel-based diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1911–1920, 2023.
- [4] Tzu-Mao Li, Michal Lukáč, Michaël Gharbi, and Jonathan Ragan-Kelley. Differentiable vector graphics rasterization for editing and learning. *ACM Transactions on Graphics (TOG)*, 39(6):1–15, 2020.
- [5] Chen-Hsuan Lin, Jun Gao, Luming Tang, Towaki Takikawa, Xiaohui Zeng, Xun Huang, Karsten Kreis, Sanja Fidler, Ming-Yu Liu, and Tsung-Yi Lin. Magic3d: High-resolution text-to-3d content creation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 300–309, 2023.
- [6] Xu Ma, Yuqian Zhou, Xingqian Xu, Bin Sun, Valerii Filev, Nikita Orlov, Yun Fu, and Humphrey Shi. Towards layer-wise image vectorization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16314–16323, 2022.
- [7] Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. *arXiv preprint arXiv:2112.10741*, 2021.
- [8] Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv preprint arXiv:2209.14988*, 2022.
- [9] Alec Radford, JongWook Kim, Chris Hallacy, A. Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Askell Amanda, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. *Cornell University - arXiv, Cornell University - arXiv*, Feb 2021.
- [10] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 1(2):3, 2022.

- [11] 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.
- [12] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in neural information processing systems*, 35:36479–36494, 2022.
- [13] Haochen Wang, Xiaoyong Du, Jiahao Li, RaymondA. Yeh, and Greg Shakhnarovich. Score jacobian chaining: Lifting pretrained 2d diffusion models for 3d generation. *Cornell University - arXiv, Cornell University - arXiv*, Dec 2022.
- [14] 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, 2024.
- [15] Ximing Xing, Chuang Wang, Haitao Zhou, Jing Zhang, Qian Yu, and Dong Xu. Diffsketcher: Text guided vector sketch synthesis through latent diffusion models. *Advances in Neural Information Processing Systems*, 36:15869–15889, 2023.
- [16] Ximing Xing, Haitao Zhou, Chuang Wang, Jing Zhang, Dong Xu, and Qian Yu. Svdreamer: Text guided svg generation with diffusion model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4546–4555, 2024.
- [17] Peiying Zhang, Nanxuan Zhao, and Jing Liao. Text-to-vector generation with neural path representation. *ACM Transactions on Graphics (TOG)*, 43(4):1–13, 2024.