SVGDreamer: Text Guided SVG Generation with Diffusion Model

Ximing Xing, Haitao Zhou, Chuang Wang, Jing Zhang
Beihang University
{ximingxing, zhouhaitao, chuangwang, zhang_jing}@buaa.edu.cn

Dong Xu
The University of Hong Kong
dongxu@cs.hku.hk

Qian Yu*
Beihang University
qianyu@buaa.edu.cn

Abstract

Recently, text-guided scalable vector graphics (SVGs) synthesis has shown promise in domains such as iconography and sketch. However, existing text-to-SVG generation methods lack editability and struggle with visual quality and result diversity. To address these limitations, we propose a novel text-guided vector graphics synthesis method called SVGDreamer. SVGDreamer incorporates a semantic-driven image vectorization (SIVE) process that enables the decomposition of synthesis into foreground objects and background, thereby enhancing editability. Specifically, the SIVE process introduces attention-based primitive control and an attention-mask loss function for effective control and manipulation of individual elements. Additionally, we propose a Vectorized Particle-based Score Distillation (VPSD) approach to address issues of shape over-smoothing, color over-saturation, limited diversity, and slow convergence of the existing text-to-SVG generation methods by modeling SVGs as distributions of control points and colors. Furthermore, VPSD leverages a reward model to re-weight vector particles, which improves aesthetic appeal and accelerates convergence. Extensive experiments are conducted to validate the effectiveness of SVGDreamer, demonstrating its superiority over baseline methods in terms of editability, visual quality, and diversity. Project page: https://ximinng.github.io/SVGDreamer-project/

1. Introduction

Scalable Vector Graphics (SVGs) represent visual concepts using geometric primitives such as Bézier curves, polygons, and lines. Due to their inherent nature, SVGs are highly suitable for visual design applications, such as posters and logos. Secondly, compared to raster images, vector images can maintain compact file sizes, making them more efficient for storage and transmission purposes. More importantly, vector images offer greater editability, allowing designers to easily select, modify, and compose elements. This attribute is particularly crucial in the design process, as it allows for seamless adjustments and creative exploration.

In recent years, there has been a growing interest in general vector graphics generation. Various optimization-based methods [4, 12, 19, 28, 34, 40, 41, 48] have been proposed, building upon the differentiable rasterizer DiffVG [14]. These methods, such as CLIPDraw [4] and VectorFusion [12], differ primarily in their approach to supervision. Some works [4, 19, 28, 34, 40, 41] combine the CLIP model [23] with DiffVG [14], using CLIP as a source of supervision. More recently, the significantly progress achieved by Text-to-Image (T2I) diffusion models [20, 24, 26, 27, 37] has inspired the task of text-to-vector-graphics. Both VectorFusion [12] and DiffSketcher [48] attempted to utilize T2I diffusion models for supervision. These models make use of the high-quality raster images generated by T2I models as targets to optimize the parameters of vector images. Additionally, the priors embedded within T2I models can be distilled and applied in this task. Consequently, models that use T2I for supervision generally perform better than those using the CLIP model.

Despite their impressive performance, existing T2I-based methods have certain limitations. Firstly, the vector images generated by these methods lack editability. Unlike the conventional approach of creating vector graphics, where individual elements are added one by one, T2I-based methods do not distinguish between different components during synthesis. As a result, the objects become entangled, making it challenging to edit or modify a single object independently. Secondly, there is still a large room for improvement in visual quality and diversity of the re-

*Corresponding author

sults generated by these methods. Both VectorFusion [12] and DiffSketcher [48] extended the Score Distillation Sampling (SDS) [22] to distill priors from the T2I models. However, it has been observed that SDS can lead to issues such as color over-saturation and over-smoothing, resulting in a lack of fine details in the generated vector images. Besides, SDS optimizes a set of control points in the vector graphic space to obtain the average state of the vector graphic corresponding to the text prompt in a mode-seeking manner [22]. This leads to a lack of diversity and detailed construction in the SDS-based approach [12, 48], along with absent text prompt objects.

To address the aforementioned issues, we present a new model called SVGDreamer for text-guided vector graphics generation. Our primary objective is to produce vector graphics of superior quality that offer enhanced editability, visual appeal, and diversity. To ensure editability, we propose a semantic-driven image vectorization (SIVE) process. This approach incorporates an innovative attention-based primitive control strategy, which facilitates the decomposition of the synthesis process into foreground objects and background. To initialize the control points for each foreground object and background, we leverage cross-attention maps queried by text tokens. Furthermore, we introduce an attention-mask loss function, which optimizes the graphic elements hierarchically. The proposed SIVE process ensures the separation and editability of the individual elements, promoting effective control and manipulation of the resulting vector graphics.

To improve the visual quality and diversity of the generated vector graphics, we introduce Vectorized Particle-based Score Distillation (VPSD) for vector graphics refinement. Previous works in vector graphics synthesis [11, 12, 48] that utilized SDS often encountered issues like shape over-smoothing, color over-saturation, limited diversity, and slow convergence in synthesized results [22, 48]. To address these issues, VPSD models SVGs as distributions of control points and colors, respectively. VPSD adopts a LoRA [10] network to estimate these distributions, aligning vector graphics with the pretrained diffusion model. Furthermore, to enhance the aesthetic appeal of the generated vector graphics, we integrate ReFL [49] to fine-tune the estimation network. Through this refinement process, we achieve final vector graphics that exhibit high editability, superior visual quality, and increased diversity. To validate the effectiveness of our proposed method, we perform extensive experiments to evaluate the model across multiple aspects. In summary, our contributions can be summarized as follows:

- We introduce *SVGDreamer*, a novel model for text-to-SVG generation. This novel model is capable of generating high-quality vector graphics while preserving editability.
- We present the semantic-driven image vectorization (SIVE) method, which ensures that the generated vector objects are separate and flexible to edit. Additionally, we propose the vectorized particle-based score distillation (VPSD) loss to guarantee that the generated vector graphics exhibit both exceptional visual quality and a wide range of diversity.
- We conduct comprehensive experiments to evaluate the effectiveness of our proposed method. Results demonstrate the superiority of our approach compared to baseline methods. Moreover, our model showcases strong generalization capabilities in generating diverse types of vector graphics.

2. Related Work

2.1. Vector Graphics Generation

Scalable Vector Graphics (SVGs) offer a declarative format for visual concepts expressed as primitives. One approach to creating SVG content is to use Sequence-To-Sequence (seq2seq) models to generate SVGs [1, 5, 16, 25, 43, 44, 46]. These methods heavily rely on dataset in vector form, which limits their generalization ability and their capacity to synthesize complex vector graphics. Instead of directly learning an SVG generation network, an alternative method of vector synthesis is to optimize towards a matching image during evaluation time.

Li *et al.* [14] introduce a differentiable rasterizer that bridges the vector graphics and raster image domains. While image generation methods that traditionally operate over vector graphics require a vector-based dataset, recent work has demonstrated the use of differentiable renderers to overcome this limitation [17, 25, 28, 30, 36, 38, 39, 48]. Furthermore, recent advances in visual text embedding contrastive language-image pre-training model (CLIP) [23] have enabled a number of successful methods for synthesizing sketches, such as CLIPDraw[4], CLIP-CLOP [19], and CLIPasso [40]. A very recent work VectorFusion [12] and DiffSketcher [48] combine differentiable renderer with text-to-image diffusion model for vector graphics generation, resulting in promising results in fields such as iconography, pixel art, and sketch.

2.2. Text-to-Image Diffusion Model

Denoising diffusion probabilistic models (DDPMs) [8, 31, 33, 35], particularly those conditioned on text, have shown promising results in text-to-image synthesis. For example, Classifier-Free Guidance (CFG) [7] has improved visual quality and is widely used in large-scale text conditional diffusion model frameworks, including GLIDE [20], Stable Diffusion [26], DALL·E 2 [24], Imagen [27] and DeepFloyd IF [37]. The progress achieved by text-to-image diffusion models [20, 24, 26, 27] also promotes the devel-

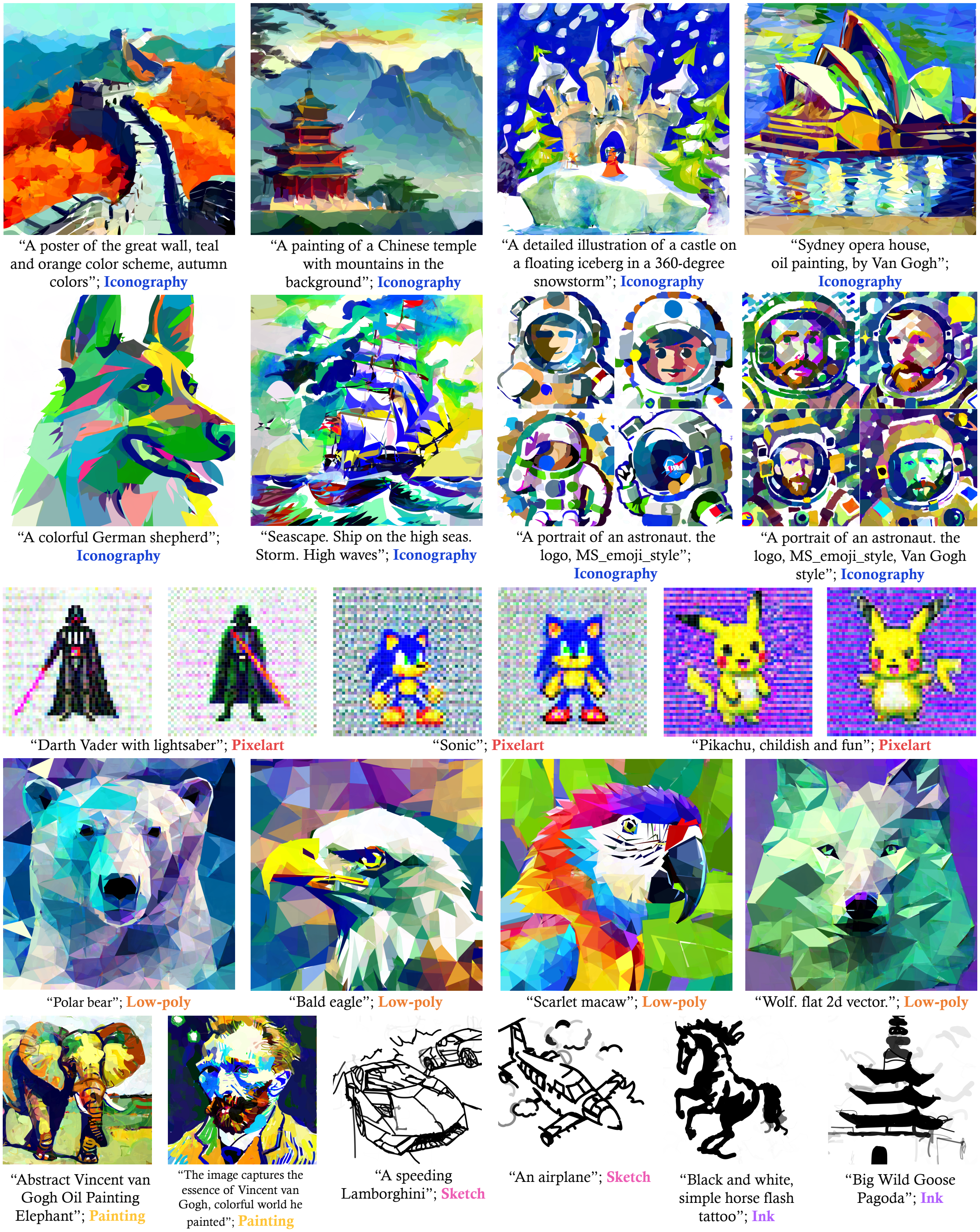

Figure 1. Given a text prompt, SVGDreamer can generate a variety of vector graphics. SVGDreamer is a versatile tool that can work with various vector styles without being limited to a specific prompt suffix. We utilize various colored suffixes to indicate different styles. The style is governed by vector primitives.

opment of a series of text-guided tasks, such as text-to-3D [22]. In this work, we employ Stable Diffusion model to provide supervision for text-to-SVG generation.

2.3. Score Distillation Sampling

Recent advances in natural image modeling have sparked significant research interest in utilizing powerful 2D pre-trained models to recover 3D object structures [15, 18, 21, 22, 42, 45]. Recent efforts such as DreamFusion [22], Magic3D [15] and Score Jacobian Chaining [42] explore text-to-3D generation by exploiting a score distillation sampling (SDS) loss derived from a 2D text-to-image diffusion model [26, 27] instead, showing impressive results. The development of text-to-SVG [12, 48] was inspired by this, but the resulting vector graphics have limited quality and exhibit a similar over-smoothness as the reconstructed 3D models. Wang *et al.* [45] extend the modeling of the 3D model as a random variable instead of a constant as in SDS and present variational score distillation to address the over-smoothing issues in text-to-3D generation.

3. Methodology

In this section, we introduce SVGDreamer, an optimization-based method that creates a variety of vector graphics based on text prompts. We define a vector graphic as a set of paths $\{P_i\}_{i=1}^n$ and color attributes $\{C_i\}_{i=1}^n$. Each path consists of m control points $P_i = \{p_j\}_{j=1}^m = \{(x_j, y_j)\}_{j=1}^m$ and one color attribute $C_i = \{r, g, b, a\}_i$. We optimize an SVG by back-propagating gradients of rasterized images to SVG path parameters $\theta = \{P_i, C_i\}_{i=1}^n$ via a differentiable renderer $\mathcal{R}(\theta)$ [14].

Our approach leverages the text-to-image diffusion model prior to guide the differentiable renderer $\mathcal{R}$ and optimize the parametric graphic path θ, resulting in the synthesis of vector graphs that match the description of the text prompt y. As illustrated in Fig. 2, our pipeline consists of two parts: semantic-driven image vectorization and SVG synthesis through VPSD optimization. The first part is **S**emantic-driven **I**mage **VE**ctorization (SIVE), consisting of two stages: primitive initialization and semantic-aware optimization. We rethink the application of attention mechanisms in synthesizing vector graphics. We extract the cross-attention maps corresponding to different objects in the diffusion model and apply it to initialize control points and consolidate object vectorization. This process allows us to decompose the foreground objects from the background. Consequently, the SIVE process generates vector objects which are independently editable. It separates vector objects by aggregating the curves that form them, which in turn simplifies the combination of vector graphics.

In Sec. 3.2, we propose the **V**ectorized **P**article-based **S**core **D**istillation (VPSD) to generate diverse high-quality text-matching vector graphics. VPSD is designed to model the distribution of vector path control points and colors for approximating the vector parameter distribution, thus obtaining vector results of diversity.

3.1. SIVE: Semantic-driven Image Vectorization

Image rasterization is a mature technique in computer graphics, while image vectorization, the reverse path of rasterization, remains a major challenge. Given an arbitrary input image, LIVE [17] recursively learns the visual concepts by adding new optimizable closed Bézier paths and optimizing all these paths. However, LIVE [17] struggles with grasping and distinguishing various subjects within an image, leading to identical paths being superimposed onto different visual subjects. And the LIVE-based method [12, 17] fails to represent intricate vector graphics consisting of complex paths. We propose a semantic-driven image vectorization method to address the aforementioned issue. This method consists of two main stages: primitive initialization and semantic-aware optimization. In the initialization stage, we allocate distinct control points to different regions corresponding to various visual objects with the guidance of attention maps. In the optimization stage, we introduce an attention-based mask loss function to hierarchically optimize the vector objects.

3.1.1 Primitive Initialization

Vectorizing visual objects often involves assigning numerous paths, which leads to *object-layer confusion* in LIVE-based methods. To address this issue, we suggest organizing vector graphic elements semantically and assigning paths to objects based on their semantics. We initialize O groups of object-level control points according to the cross-attention map corresponding to different objects in the text prompt. And we represent them as the foreground $\mathcal{M}_{\text{FG}}^i$, where i indicates the i-th token in the text prompt. Correspondingly, the rest will be treated as background. Such design allows us to represent the attention maps of background and foreground as,

$$\begin{aligned} \mathcal{M}_{\text{BG}} &= 1 - (\sum_{i=1}^{O} \mathcal{M}_{\text{FG}}^i); \\ \mathcal{M}_{\text{FG}}^i &= \text{softmax}(QK_i^T)/\sqrt{d} \end{aligned} \tag{1}$$

where $\mathcal{M}_{\text{BG}}$ indicates the attention map of the background. $\mathcal{M}_{\text{FG}}^i$ indicates cross-attention score, where K_i indicates i-th token keys from text prompt, Q is pixel queries features, and d is the latent projection dimension of the keys and queries.

Then, inspired by DiffSketcher [48], we normalize the attention maps using softmax and treat it as a distribution map to sample m positions for the first control point $p_{j=1}$ of each Bézier curve. The other control points ($\{p_j\}_{j=2}^m$) are sampled within a small radius (0.05 of image size) around $p_{j=1}$ to define the initial set of paths.

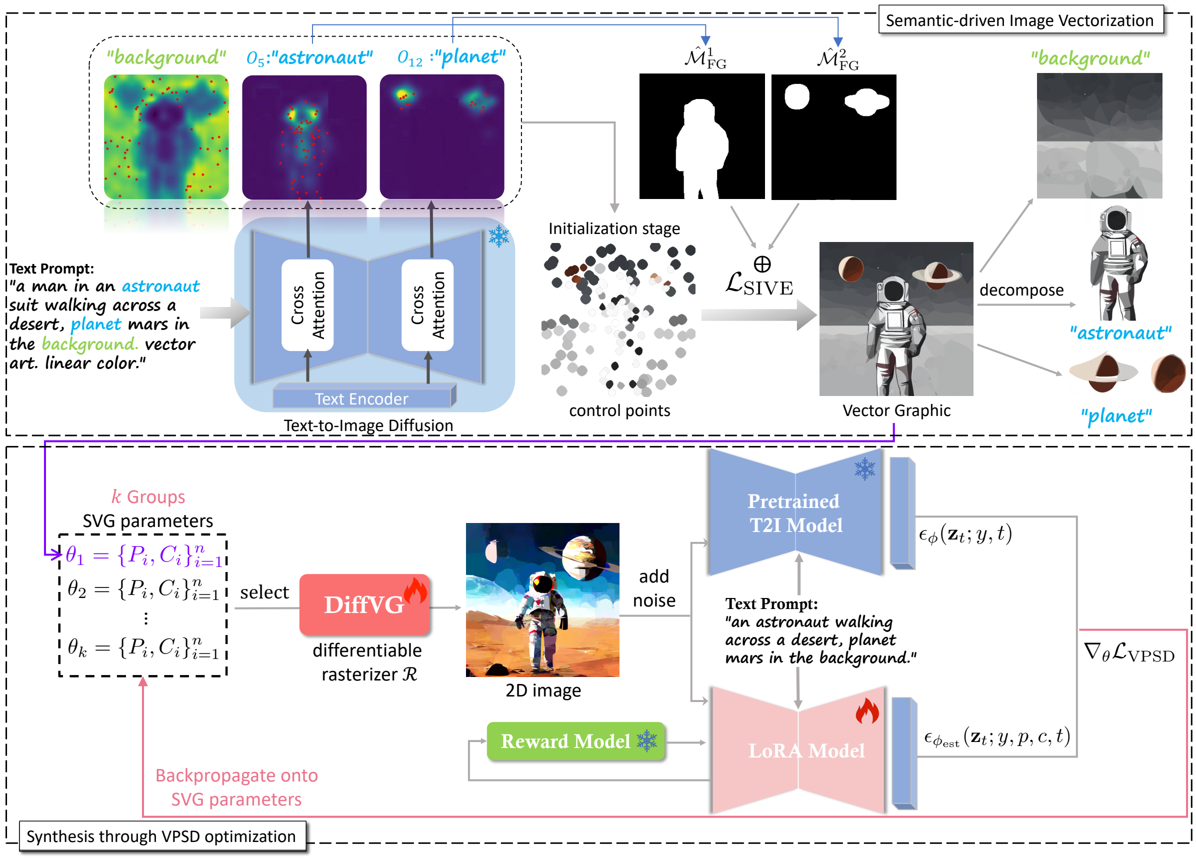

Figure 2. Overview of SVGDreamer. The method consists of two parts: semantic-driven image vectorization (SIVE) and SVG synthesis through VPSD optimization. The result obtained from SIVE can be used as input of VPSD for further refinement.

3.1.2 Semantic-aware Optimization

In this stage, we utilize an attention-based mask loss to separately optimize the objects in the foreground and background. This ensures that control points remain within their respective regions, aiding in object decomposition. Namely, the hierarchy only exists within the designated object and does not get mixed up with other objects. This strategy fuels the permutations and combinations between objects that form different vector graphics, and enhances the editability of the objects themselves.

Specifically, we convert the attention map obtained during the initialization stage into reusable masks $\hat{\mathcal{M}} = \{\{\hat{\mathcal{M}}_{\text{FG}}\}_{o=1}^{O}, \hat{\mathcal{M}}_{\text{BG}}\}$, O foregrounds and one background mask in total. We do this by setting the attention score to 1 if it is greater than the threshold value, and to 0 otherwise.

$$\mathcal{L}_{\text{SIVE}} = \sum_{i}^{O} \left(\hat{\mathcal{M}}_i \odot I - \hat{\mathcal{M}}_i \odot \mathbf{x} \right)^2 \tag{2}$$

where I is the target image, $\hat{\mathcal{M}}$ is mask, $\mathbf{x} = \mathcal{R}(\theta)$ is the rendering.

3.2. Vectorized Particle-based Score Distillation

While vectorizing a rasterized diffusion sample is lossy, recent techniques [12, 48] have identified the SDS loss [22] as beneficial for our task of generating vector graphics. To synthesize a vector image that matches a given text prompt y, they directly optimize the parameters $\theta = \{P_i, C_i\}_{i=1}^n$ of a differentiable rasterizer $\mathcal{R}(\theta)$ via SDS loss. At each iteration, the differentiable rasterizer is used to render a raster image $\mathbf{x} = \mathcal{R}(\theta)$, which is augmented to obtain a $\mathbf{x}_a$. Then, the pretrained latent diffusion model (LDM) ϵ_ϕ uses a VAE encoder [3] to encode $\mathbf{x}_a$ into a latent representation $\mathbf{z} = \mathcal{E}(\mathbf{x_a})$, where $\mathbf{z} \in \mathbb{R}^{(H/f)\times(W/f)\times 4}$ and f is the encoder downsample factor. Finally, the gradient of SDS is estimated by,

$$\nabla_\theta \mathcal{L}_{\text{SDS}}(\phi, \mathbf{x} = \mathcal{R}(\theta)) \triangleq \mathbb{E}_{t,\epsilon,a}\left[w(t)(\epsilon_\phi(\mathbf{z}_t; y, t) - \epsilon) \frac{\partial \mathbf{z}}{\partial \mathbf{x}_a} \frac{\partial \mathbf{x}_a}{\partial \theta} \right] \tag{3}$$

where $w(t)$ is the weighting function. And noised to form $\mathbf{z}_t = \alpha_t \mathbf{x}_a + \sigma_t \epsilon$.

Unfortunately, SDS-based methods often suffer from issues such as shape over-smoothing, color over-saturation, limited diversity in results, and slow convergence in synthe-

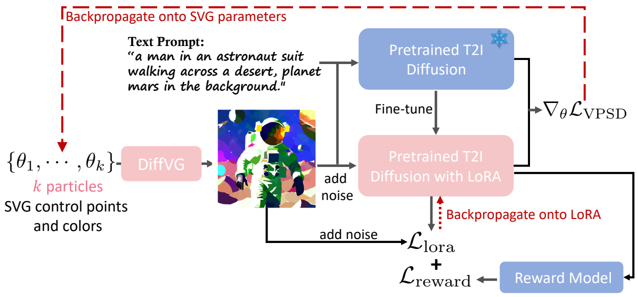

Figure 3. The process of VPSD.

sis results [11, 12, 22, 48]. Inspired by the principled variational score distillation framework [45], we propose vectorized particle-based score distillation (VPSD) to address the aforementioned issues. Instead of modeling SVGs as a set of control points and corresponding colors like SDS, we model SVGs as the distributions of control points and colors respectively. In principle, given a text prompt y, there exists a probabilistic distribution μ of all possible vector shapes representations. Under a vector representation parameterized by θ, such a distribution can be modeled as a probabilistic density $\mu(\theta|y)$. Compared with SDS that optimizes for the single θ, VPSD optimizes for the whole distribution μ, from which we can sample θ. Motivated by previous particle-based variational inference methods, we maintain k groups of vector parameters $\{\theta\}_{i=1}^{k}$ as particles to estimate the distribution μ, and $\theta(i)$ will be sampled from the optimal distribution μ^* if the optimization converges. This optimization can be realized through two score functions: one that approximates the optimal distribution with a noisy real image, and one that represents the current distribution with a noisy rendered image. The score function of noisy real images can be approximated by the pretrained diffusion model [26] $\epsilon_\phi(\mathbf{z}_t; y, t)$. The score function of noisy rendered images is estimated by another noise prediction network $\epsilon_{\phi_{\text{est}}}(\mathbf{z}_t; y, p, c, t)$, which is trained on the rendered images by $\{\theta\}_{i=1}^{k}$. The gradient of VPSD can be formed as,

$$\begin{aligned}&\nabla_\theta \mathcal{L}_{\text{VPSD}}(\phi, \phi_{\text{est}}, \mathbf{x} = \mathcal{R}(\theta)) \triangleq \\ &\mathbb{E}_{t,\epsilon,p,c}\left[w(t)(\epsilon_\phi(\mathbf{z}_t; y, t) - \epsilon_{\phi_{\text{est}}}(\mathbf{z}_t; y, p, c, t))\frac{\partial \mathbf{z}}{\partial \theta}\right]\end{aligned} \tag{4}$$

where p and c in $\epsilon_{\phi_{\text{est}}}$ indicate control point variables and color variables, the weighting function $w(t)$ is a hyperparameter. And $t \sim \mathcal{U}(0.05, 0.95)$.

In practice, as suggested by [45], we parameterize ϵ_ϕ using a LoRA (Low-rank adaptation [10]) of the pretrained diffusion model. The rendered image not only serves to calculate the VPSD gradient but also gets updated by LoRA,

$$\mathcal{L}_{\text{lora}} = \mathbb{E}_{t,\epsilon,p,c}\left\|\epsilon_{\phi_{\text{est}}}(\mathbf{z}_t; y, p, c, t) - \epsilon\right\|_2^2 \tag{5}$$

where ϵ is the Gaussian noise. Only the parameters of the LoRA model will be updated, while the parameters of other diffusion models will remain unchanged to minimize computational complexity.

In [45], only randomly selected particles update the LoRA network in each iteration. However, this approach neglects the learning progression of vector particles, which are used to represent the optimal SVG distributions. Furthermore, these networks typically require numerous iterations to approximate the theoretical optimal distribution, resulting in slow convergence. In VPSD, we introduce a Reward Feedback Learning method, as Fig. 3 illustrates. This method leverages a pre-trained reward model [49] to assign reward scores to samples collected from LoRA model. Then LoRA model subsequently updates from these reweighted samples,

$$\mathcal{L}_{\text{reward}} = \lambda \mathbb{E}_y\left[\psi(r(y, g_{\phi_{\text{est}}}(y)))\right] \tag{6}$$

where $g_{\phi_{\text{est}}}(y)$ denotes the generated image of μ model with parameters ϕ_{est} corresponding to prompt y, and r represents the pretrained reward model [49], ψ represents reward-to-loss map function implemented by ReLU, and $\lambda = 1e-3$. We used the DDIM [32] to rapidly sample k samples during the early iteration stage. This method saves 2 times the iteration step for VPSD convergence and improves the aesthetic score of the SVG by filtering out samples with low reward values in LoRA.

Our final VPSD objective is then defined by the weighted average of the three terms,

$$\min_\theta \nabla_\theta \mathcal{L}_{\text{VPSD}} + \mathcal{L}_{\text{lora}} + \lambda_{\text{r}} \mathcal{L}_{\text{reward}} \tag{7}$$

where λ_{r} indicates reward feedback strength.

3.3. Vector Representation Primitives

In addition to text prompts, SVGDreamer provides a variety of vector representations for style control. These vector representations are achieved by limiting primitive types and their parameters. Users can control the art style generated by SVGDreamer by modifying the input text or by constraining the set of primitives and parameters. We explore six settings: 1) **Iconography** is the most common SVG style, consists of several paths and their fill colors. This style allows for a wide range of compositions while maintaining a minimalistic expression. We utilize closed form Bézier curves with trainable control points and fill colors. 2) **Sketch** is a way to convey information with minimal expression. We use open form Bézier curves with trainable control points and opacity. 3) **Pixel Art** is a popular video-game inspired style, frequently used for character and background art. We use square SVG polygons with fill colors. 4) **Low-Poly** is to consciously cut and pile up a certain number of simple geometric shapes according to the modeling laws of objects. We use square SVG polygons with trainable control points and fill colors. 5) **Painting** is a means of approximating the painter's painting style in vector graphics. We

Figure 4. Qualitative comparison of different methods. Note that DiffSketcher was originally designed for vector sketch generation; therefore, we re-implemented it to generate RGB vector images.

use open form Bézier curves with trainable control points, stroke colors and stroke widths. 6) **Ink and Wash Painting** is a traditional Chinese art form that utilizes varying concentrations of black ink. We use open form Bézier curves with trainable control points, opacity, and stroke widths.

4. Experiments

4.1. Qualitative Evaluation

Figure 4 presents a qualitative comparison between SVGDreamer and existing text-to-SVG methods. Compared to CLIPDraw [4], SVGDreamer synthesizes SVGs with higher fidelity and detail. We also compare our work with SDS-based methods [12, 48], emphasizing our ability to address issues such as shape over-smoothing and color oversaturation. As shown in the fifth column, SIVE achieves semantic decoupling but cannot overcome the inherently smooth nature of SDS. As observed in the last two columns, our approach demonstrates superior detail compared to the SDS-based approach, regardless of whether the model was optimized from scratch or through the entire process. Consequently, this leads to a higher aesthetic score.

4.2. Quantitative Evaluation

To demonstrate the effectiveness of our proposed method, we conducted comprehensive experiments to evaluate the model across various aspects, including Fréchet Inception Distance (FID) [6], Peak Signal-to-Noise Ratio (PSNR) [9], CLIPScore [23], BLIPScore [13], Aesthetic score [29] and Human Performance Score [47] (HPS). Table 1 presents a comparison of our approach with the most representative text-to-SVG methods, including CLIPDraw [4], VectorFusion [12], and DiffSketcher [48]. We conducted a quantitative evaluation of the six styles identified in Sec. 3.3, with each style comprising 10 unique prompts and 50 synthesized SVGs per prompt. For diversity evaluation of vector graphics and fill color saturation, we used SD sampling results as a Ground Truth (GT) and calculated FID and PSNR metrics respectively. The quantitative analysis in the first two columns indicates that our method surpasses other methods in terms of FID and PSNR. This suggests that our method offers a greater range of diversity compared to SDS-based synthesis [12, 48]. To assess the consistency between the generated SVGs and the provided text prompts, we used both CLIPScore and BLIPScore. To measure the perceptual quality of synthetic vector images, we measure aesthetic scores using the LAION aesthetic classifier [29]. Besides, we use HPS to evaluate our approach from a human aesthetic perspective.

4.3. Ablation Study

4.3.1 SIVE v.s. LIVE [17]

LIVE [17] offers a comprehensive image vectorization process that optimizes the vector graph in a hierarchical, layerwise fashion. However, as Fig. 6 illustrates, LIVE struggles to accurately capture and distinguish between various subjects within an image, which can result in the same paths being superimposed on different visual subjects. When tasked with representing complex vector graphics requiring a greater number of paths, LIVE tends to superimpose path hierarchies across different objects, complicating the SVG representation and making it difficult to edit. The resulting SVGs often contain complex and redundant shapes that can be inconvenient for further editing.

In contrast, SIVE is capable of generating succinct SVG forms with semantic-driven structures that align more closely with human perception. SIVE efficiently assigns paths to objects, enabling object-level vectorization.

4.3.2 VPSD v.s. LSDS [11, 12] v.s. ASDS [48]

The development of text-to-SVG [12, 48] was inspired by DreamFusion [22], but the resulting vector graphics have limited quality and exhibit a similar over-smoothness as the DreamFusion reconstructed 3D models. The main distinction between ASDS and LSDS lies in the augmentation of the input data. As demonstrated in Table 1 and Fig. 4, our approach demonstrates superior performance compared to

Table 1. Quantitative comparison of different methods.

Method / Metric	FID [6]↓	PSNR [9]↑	CLIPScore [23]↑	BLIPScore [13]↑	Aesthetic [29]↑	HPS [47]↑
CLIPDraw [4]	160.64	8.35	0.2486	0.3933	3.9803	0.2347
VectorFusion (scratch) [12]	119.55	6.33	0.2298	0.3803	4.5165	0.2334
VectorFusion [12]	100.68	8.01	0.2720	0.4291	4.9845	0.2450
DiffSketcher(RGB) [48]	118.70	6.75	0.2402	0.4185	4.1562	0.2423
SVGDreamer (from scratch)	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**

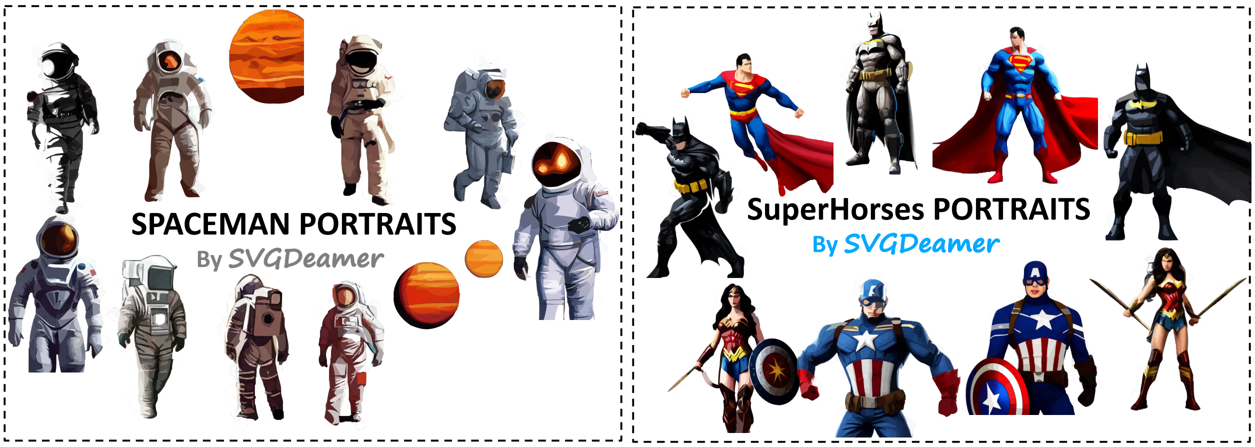

Figure 5. Examples of vector assets created by our SVGDreamer.

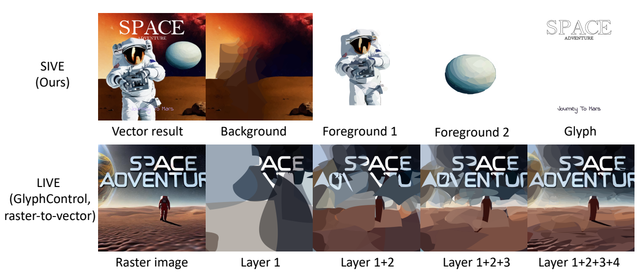

Figure 6. Comparison of LIVE vectorization with SIVE. In the first row, "Foreground 1" and "Foreground 2" refer to *Astronaut* and *Plants*, respectively. Glyphs have been added manually and were not produced by our method. In the LIVE setup, we follow the protocol outlined in VectorFusion [12], which represents a vector image with 128 paths distributed across four layers, with 32 paths in each layer.

the SDS-based approach in terms of FID. This indicates that our method is able to maintain a higher level of diversity without being affected by mode-seeking disruptions. Additionally, our approach achieves a higher PSNR compared to the SDS-based approach, suggesting that our method avoids the issue of supersaturation caused by averaging colors.

4.4. Applications of SVGDreamer

Our proposed tool, SVGDreamer, is capable of generating vector graphics with exceptional editability. Therefore, it can be utilized to create vector graphic assets for poster and logo design. As shown in Fig. 5, all graphic elements in the two poster examples are generated by our SVGDreamer. Designers can easily recombine these elements with glyph to create unique posters. Additional examples of posters and logo designs can be found in Supplementary.

5. Conclusion

In this work, we have introduced SVGDreamer, an innovative model for text-guided vector graphics synthesis. SVGDreamer incorporates two crucial technical designs: Semantic-Driven Image Vectorization (SIVE) and Vectorized Particle-Based Score Distillation (VPSD). These empower our model to generate vector graphics with high editability, superior visual quality, and notable diversity. SVGDreamer is expected to significantly advance the application of text-to-SVG models in the design field.

Limitations. The editability of our method, which depends on the text-to-image (T2I) model used, is currently limited. However, future advancements in T2I diffusion models could enhance the decomposition capabilities of our approach, thereby extending its editability. Moreover, exploring ways to automatically determine the number of control points at the SIVE object level is valuable.

Acknowledgement. This work is supported by the CCF-Baidu Open Fund Project and Young Elite Scientists Sponsorship Program by CAST.

References

[1] Alexandre Carlier, Martin Danelljan, Alexandre Alahi, and Radu Timofte. Deepsvg: A hierarchical generative network for vector graphics animation. *Advances in Neural Information Processing Systems (NIPS)*, 33:16351–16361, 2020. 2

[2] Jingye Chen, Yupan Huang, Tengchao Lv, Lei Cui, Qifeng Chen, and Furu Wei. Textdiffuser: Diffusion models as text painters. *arXiv preprint arXiv:2305.10855*, 2023. 1

[3] Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (NIPS)*, pages 12873–12883, 2021. 5

[4] Kevin Frans, Lisa Soros, and Olaf Witkowski. CLIPDraw: Exploring text-to-drawing synthesis through language-image encoders. In *Advances in Neural Information Processing Systems (NIPS)*, 2022. 1, 2, 7, 8

[5] David Ha and Douglas Eck. A neural representation of sketch drawings. In *International Conference on Learning Representations (ICLR)*, 2018. 2

[6] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems (NIPS)*, 30, 2017. 7, 8

[7] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022. 2, 5

[8] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems (NIPS)*, pages 6840–6851, 2020. 2

[9] Alain Horé and Djemel Ziou. Image quality metrics: Psnr vs. ssim. In *2010 20th International Conference on Pattern Recognition*, pages 2366–2369, 2010. 7, 8

[10] Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations (ICLR)*, 2022. 2, 6, 7, 8

[11] Shir Iluz, Yael Vinker, Amir Hertz, Daniel Berio, Daniel Cohen-Or, and Ariel Shamir. Word-as-image for semantic typography. *ACM Transactions on Graphics (TOG)*, 42(4), 2023. 2, 6, 7

[12] 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 (CVPR)*, 2023. 1, 2, 4, 5, 6, 7, 8

[13] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International Conference on Machine Learning (ICML)*, pages 12888–12900. PMLR, 2022. 7, 8

[14] Tzu-Mao Li, Michal Lukáč, Gharbi Michaël, and Jonathan Ragan-Kelley. Differentiable vector graphics rasterization for editing and learning. *ACM Transactions on Graphics (TOG)*, 39(6):193:1–193:15, 2020. 1, 2, 4

[15] 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 (CVPR)*, pages 300–309, 2023. 4

[16] Raphael Gontijo Lopes, David Ha, Douglas Eck, and Jonathon Shlens. A learned representation for scalable vector graphics. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019. 2

[17] 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 (CVPR)*, pages 16314–16323, 2022. 2, 4, 7

[18] 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. 4

[19] Piotr Mirowski, Dylan Banarse, Mateusz Malinowski, Simon Osindero, and Chrisantha Fernando. Clip-clop: Clip-guided collage and photomontage. *arXiv preprint arXiv:2205.03146*, 2022. 1, 2

[20] Alexander Quinn 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. In *Proceedings of the 39th International Conference on Machine Learning (ICML)*, pages 16784–16804, 2022. 1, 2

[21] Xingang Pan, Bo Dai, Ziwei Liu, Chen Change Loy, and Ping Luo. Do 2d {gan}s know 3d shape? unsupervised 3d shape reconstruction from 2d image {gan}s. In *International Conference on Learning Representations (ICLR)*, 2021. 4

[22] Ben Poole, Ajay Jain, Jonathan T. Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. In *The Eleventh International Conference on Learning Representations (ICLR)*, 2023. 2, 4, 5, 6, 7

[23] 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 (ICML)*, pages 8748–8763. PMLR, 2021. 1, 2, 7, 8

[24] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022. 1, 2

[25] Pradyumna Reddy, Michael Gharbi, Michal Lukac, and Niloy J Mitra. Im2vec: Synthesizing vector graphics without vector supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7342–7351, 2021. 2

[26] 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 (CVPR), pages 10684–10695, 2022. 1, 2, 4, 6

[27] 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. In *Advances in Neural Information Processing Systems (NIPS)*, pages 36479–36494, 2022. 1, 2, 4

[28] Peter Schaldenbrand, Zhixuan Liu, and Jean Oh. Styleclipdraw: Coupling content and style in text-to-drawing synthesis. *arXiv preprint arXiv:2111.03133*, 2022. 1, 2

[29] Christoph Schuhmann. Improved aesthetic predictor. https://github.com/christophschuhmann/improved-aesthetic-predictor, 2022. 7, 8

[30] I-Chao Shen and Bing-Yu Chen. Clipgen: A deep generative model for clipart vectorization and synthesis. *IEEE Transactions on Visualization and Computer Graphics*, 28 (12):4211–4224, 2022. 2

[31] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 2256–2265, 2015. 2

[32] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations (ICLR)*, 2021. 6

[33] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. In *Advances in Neural Information Processing Systems (NIPS)*, 2019. 2

[34] Yiren Song and Yuxuan Zhang. Clipfont: Text guided vector wordart generation. In *33rd British Machine Vision Conference 2022, BMVC 2022, London, UK, November 21-24, 2022*, 2022. 1

[35] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations (ICLR)*, 2021. 2

[36] Yiren Song, Xuning Shao, Kang Chen, Weidong Zhang, Zhongliang Jing, and Minzhe Li. Clipvg: Text-guided image manipulation using differentiable vector graphics. In *Proceedings of the Conference on Artificial Intelligence (AAAI)*, 2023. 2

[37] StabilityAI. If by deepfloyd lab at stabilityai. https://github.com/deep-floyd/IF, 2023. 1, 2

[38] Hao Su, Xuefeng Liu, Jianwei Niu, Jiahe Cui, Ji Wan, Xinghao Wu, and Nana Wang. Marvel: Raster gray-level manga vectorization via primitive-wise deep reinforcement learning. *IEEE Transactions on Circuits and Systems for Video Technology (T-CSVT)*, 2023. 2

[39] Yingtao Tian and David Ha. Modern evolution strategies for creativity: Fitting concrete images and abstract concepts. In *Artificial Intelligence in Music, Sound, Art and Design*, pages 275–291. Springer, 2022. 2

[40] Yael Vinker, Ehsan Pajouheshgar, Jessica Y Bo, Roman Christian Bachmann, Amit Haim Bermano, Daniel Cohen-Or, Amir Zamir, and Ariel Shamir. Clipasso: Semantically-aware object sketching. *ACM Transactions on Graphics (TOG)*, 41(4):1–11, 2022. 1, 2

[41] Yael Vinker, Yuval Alaluf, Daniel Cohen-Or, and Ariel Shamir. Clipascene: Scene sketching with different types and levels of abstraction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4146–4156, 2023. 1

[42] Haochen Wang, Xiaodan Du, Jiahao Li, Raymond A. Yeh, and Greg Shakhnarovich. Score jacobian chaining: Lifting pretrained 2d diffusion models for 3d generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12619–12629, 2023. 4

[43] Yizhi Wang and Zhouhui Lian. Deepvecfont: Synthesizing high-quality vector fonts via dual-modality learning. *ACM Transactions on Graphics (TOG)*, 40(6), 2021. 2

[44] Yizhi Wang, Gu Pu, Wenhan Luo, Pengfei Wang, Yexin ans Xiong, Hongwen Kang, Zhonghao Wang, and Zhouhui Lian. Aesthetic text logo synthesis via content-aware layout inferring. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 2

[45] 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. *arXiv preprint arXiv:2305.16213*, 2023. 4, 6

[46] Ronghuan Wu, Wanchao Su, Kede Ma, and Jing Liao. Iconshop: Text-based vector icon synthesis with autoregressive transformers. *arXiv preprint arXiv:2304.14400*, 2023. 2

[47] Xiaoshi Wu, Keqiang Sun, Feng Zhu, Rui Zhao, and Hongsheng Li. Human preference score: Better aligning text-to-image models with human preference. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2096–2105, 2023. 7, 8

[48] Ximing Xing, Chuang Wang, Haitao Zhou, Jing Zhang, Qian Yu, and Dong Xu. Diffsketcher: Text guided vector sketch synthesis through latent diffusion models. In *Advances in Neural Information Processing Systems (NIPS)*, 2023. 1, 2, 4, 5, 6, 7, 8

[49] Jiazheng Xu, Xiao Liu, Yuchen Wu, Yuxuan Tong, Qinkai Li, Ming Ding, Jie Tang, and Yuxiao Dong. Imagereward: Learning and evaluating human preferences for text-to-image generation, 2023. 2, 6, 4, 8

[50] Yukang Yang, Dongnan Gui, Yuhui Yuan, Haisong Ding, Han Hu, and Kai Chen. Glyphcontrol: Glyph conditional control for visual text generation. 2023. 1