Semantic-guided Recoloring for Compact Image Vectorization

Min Lu[1] Yuanfeng He[1] Anthony Chen[2] Zheng Gu[1]
Zhida Sun[1] Zhenyu Wang[1] Daniel Cohen-Or[3] Hui Huang[1]

[1] Shenzhen University [2] Peking University [3] Tel Aviv University

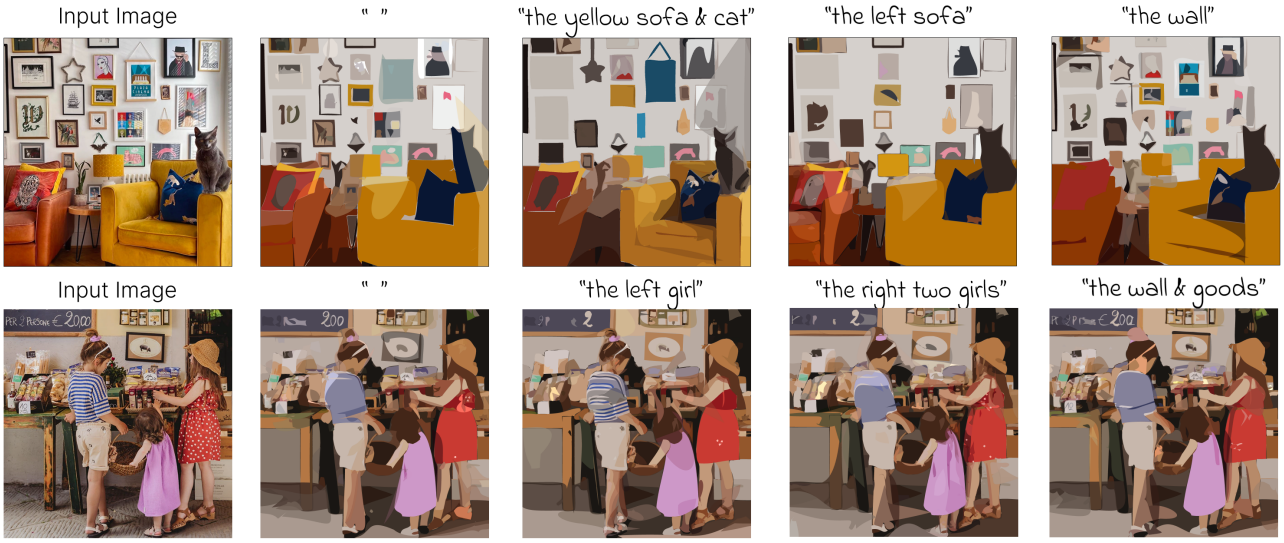

Figure 1. With user specified semantic guidance (e.g., "the wall" and "the yellow sofa & cat"), our approach allocates a fixed budget of vector primitives (128 for the first row examples and 512 for the second row) to better reconstruct semantically salient regions, while still providing a faithful but more compact representation of the remaining areas.

Abstract

We present a semantic-guided vectorization method that transforms an initial set of fine, over-segmented masks into a compact and coherent collection of vector regions. Our key contribution is a recoloring-based clustering framework that merges neighboring masks by optimizing a shared color assignment in image space. Semantic saliency, either user-specified or automatically estimated, modulates this optimization: regions of high semantic importance are encouraged to preserve distinct colors, while less important areas are driven toward color similarity and thus merge. This process progressively consolidates redundant masks, reducing clutter and concentrating vector primitives where they matter most. Combined with a final differentiable rendering refinement, the method produces SVG representations that balance visual fidelity with reduced complexity and improved editability. Experiments show that our approach yields more compact vectorizations and better allocation of primitives compared to pixel-driven or segmentation-driven baselines.

1. Introduction

Vector graphics offer a compact, resolution-independent, and easily editable representation of images, making them widely used across design and graphics workflows [15, 47]. Yet automatically converting complex images into vector form remains a challenging task. Existing approaches typically fall into two categories. Pixel-driven clustering methods rely on low-level similarities such as color and texture [17, 29], while segmentation-driven methods convert the mask outputs of foundation models such as SAM [23] into vector shapes [40, 51]. Both lines of work tend to produce large collections of small regions that do not reflect the semantic importance of different parts of the scene, often oversampling highly textured areas while under-representing content that is visually or conceptually central.

This imbalance arises because neither pixel clustering

nor segmentation masks provides guidance on which regions of an image should receive more or fewer vector primitives. SAM-based methods [40, 51], for example, generate highly detailed but unstructured mask sets, producing a dense '*mask soup*' that lacks any prioritization. Consequently, vector primitives are distributed uniformly regardless of relevance, which limits the interpretability and editability of the resulting representation.

In this work, we propose a semantic-guided vectorization method that incorporates high-level cues into the simplification process by steering the allocation of vector primitives with a text-conditioned semantic saliency map. Instead of relying solely on low-level appearance, the method uses a relevance map derived from image–text alignment to highlight regions mentioned in a short prompt (e.g., 'the yellow sofa & cat') (see Figure 1), producing a spatial importance signal that reflects semantic intent rather than just visual contrast. This map guides the system to preserve detail where the prompt indicates meaningful content, while allowing redundant areas to merge more aggressively, leading to vectorizations that better respect semantic priorities.

Our key idea is to recast mask clustering as a recoloring optimization in image space. Instead of grouping masks through hand-designed distance metrics, we assign colors to them and optimize these assignments so that adjacent regions with similar appearance and low semantic weight naturally converge toward shared colors, while high-salience regions are encouraged to maintain distinct ones. The text-conditioned semantic saliency map plays a central role here: it penalizes color changes in semantically important areas and promotes aggressive smoothing in less relevant ones. This unified, differentiable mechanism integrates spatial coherence, visual similarity, and semantic guidance, leading to a controlled consolidation of redundant masks and a more deliberate allocation of vector primitives, ultimately improving compactness and editability.

We demonstrate that this semantic-guided clustering, combined with a standard differentiable-rendering refinement stage, produces vectorizations that balance visual fidelity with reduced structural complexity. Compared to pixel-based and segmentation-based baselines, our method yields more compact representations and allocates vector primitives more effectively according to semantic priorities.

2. Related Work

2.1. Image Vectorization

Image vectorization, the process of converting raster images into geometric primitives, has been a long-standing problem [11, 39]. Traditional methodologies are primarily built upon fitting primitives to low-level image features such as edges, colors, and textures. A dominant early strategy was contour tracing, exemplified by algorithms like Potrace [36] and AutoTrace [41]. These methods first perform color quantization on the input and then trace the boundaries of these quantized regions, fitting Bézier curves to the resulting paths. A parallel class of region-based methods sought to improve contour regularity and noise reduction by first partitioning the image. This includes techniques such as mean-shift clustering [8], SLIC superpixels [1], and graph-based segmentation [14].

Other approaches have focused on partitioning the image into non-overlapping 2D patches, such as triangles [3, 9], or refining decompositions with curved boundaries and gradients [38, 42, 46]. More sophisticated fitting techniques, like diffusion curves [30], represent images via color boundaries that guide a diffusion process, enabling smooth-shaded results [43, 49]. While effective for specific domains like clip art [10, 13] or line drawings [12, 45], these traditional methods remain fundamentally limited to pixel-level similarity. This results in visually accurate yet semantically agnostic representations that often overpopulate highly textured regions while underrepresenting meaningful content, thereby limiting interpretability and editability.

The advent of deep learning, particularly the development of differentiable rasterizers like DiffVG [25], has enabled a new class of end-to-end vectorization methods. These approaches can be broadly divided into optimization-based and generative models. Optimization-based methods, such as LIVE [34], leverage differentiable rendering to iteratively optimize the parameters of vector primitives by backpropagating a raster-space loss. This optimization approach has been extended by works like SGLIVE [50], which incorporates gradient-aware segmentation to support radial gradients, and Hirschorn et al. [17], who introduce a top-down iterative process that adds primitives based on pixel clustering and prunes them based on an importance score to ensure compactness. In contrast, generative models frame the task as a translation problem. Early examples, such as SVG-VAE [28], DeepSVG [4], and Im2Vec [35] employ autoencoder or recurrent architectures to predict a sequence of SVG commands. More recent generative works have explored sophisticated architectures; Chen et al. [7] leverage a transformer model to assemble vectorizations from a predefined pool of simple primitives, while SuperSVG [19] trains a model to predict vectors from a superpixel-based image decomposition in a coarse-to-fine manner. Other novel frameworks include NeuralSVG [31], which uses an implicit neural representation, akin to NeRFs, to encode a layered SVG into a small MLP optimized via score distillation, LayerTracer [37], which employs a diffusion transformer to learn the sequential process of layered design, generating construction blueprints that are then vectorized, and T2V-NPR [47], which introduces a neural path representation learned via a dual-branch VAE and uses text-conditioned score distillation to generate SVGs directly

from prompts.

A separate, hybrid pipeline has also emerged that leverages the powerful segmentation capabilities of large-scale foundation models. Methods like SAMVG [51] and LIVSS [40] first employ the Segment Anything Model (SAM) [23] to decompose an image into a large collection of masks. These masks are then traced and converted into vector regions. While this approach successfully inherits object-level boundaries from the segmentation model, the resulting output is often a "mask soup"—a flat collection of vector paths that lacks structural hierarchy and semantic coherence. In contrast, our method integrates semantic saliency directly into the vectorization process, adaptively allocating vector density according to semantic importance to produce compact and meaningful SVG representations.

2.2. Clustering for Image Smplification

Clustering-based simplification merges visually similar pixels or regions to reduce structural complexity and achieve compact representations. Classical methods such as k-means, mean-shift [8], and SLIC superpixels [1] rely on color and spatial proximity, while hierarchical segmentation approaches refine boundaries via contrast and contour cues [2, 14]. Although effective for compact representations, these techniques remain appearance-driven and often ignore semantic consistency. Palette-based recoloring and abstraction methods [5, 22] similarly simplify color distributions but lack structural or semantic control.

Recent works incorporate high-level features into clustering, such as semantic superpixels [27, 44] and CLIP-based grouping [33], aligning regions with object semantics yet remaining detached from vector representation. Building on these insights, we formulate vector simplification as a semantics-aware clustering problem solved via iterative recoloring, merging adjacent shapes with shared semantic attributes to achieve compact and semantically coherent vector outputs.

2.3. Image Saliency Prediction

Saliency prediction aims to estimate the spatial distribution of visually or semantically important regions in an image. Early models relied on low-level contrast features such as intensity, color, and orientation [16, 20], while deep approaches like DeepGaze and DSS exploit high-level semantics from CNN or transformer backbones to model human attention [18, 21, 24]. More recent works leverage vision–language embeddings and foundation models to capture semantic relevance beyond human fixation, producing maps that highlight conceptually meaningful objects [27, 33, 48]. In our framework, such saliency maps provide semantic priors that guide adaptive vector allocation, ensuring that vector density concentrates on semantically salient regions while redundant areas are simplified.

Figure 2. **Segmentation results by SAM.** The issue of over-segmented masks results in unstructured challenges for vectorization processes.

3. Preliminary

Optimize-based Vectorization. The overall vectorization pipeline in this work builds upon the paradigm of optimize-based image vectorization via differentiable rendering [26]. Let $\mathbf{S} = \{s_1, s_2, \ldots, s_N\}$ denote the initial set of vector primitives (e.g., closed shapes of Bézier curves). The rendering function $\mathcal{R}$ maps these primitives to a raster image $\mathbf{I}_{\text{vec}} = \mathcal{R}(\mathbf{S})$. The optimization objective is to minimize a loss $\mathcal{L}$ between the rendered vector image $\mathbf{I}_{\text{vec}}$ and the target raster image $\mathbf{I}_{\text{target}}$:

$$\mathbf{S}^* = \arg\min_{\mathbf{S}} \mathcal{L}\left(\mathcal{R}(\mathbf{S}), \mathbf{I}_{\text{target}}\right) \tag{1}$$

where $\mathbf{S}^*$ represents the optimized set of vector primitives. The gradient-based optimization is enabled by the differentiable nature of $\mathcal{R}$, allowing backward propagation of the loss to update the parameters of $\mathbf{S}$.

Primitive Initialization. The vectorization process begins by initializing shape primitives $\mathbf{S} = \{s_1, s_2, \ldots, s_N\}$ from the input image $\mathbf{I}$. Existing approaches typically follow one of two strategies: (1) visual clustering: methods like LIVE [29], O&R [17], SuperSVG [19] cluster pixels on low-level features using algorithms such as K-means, DBSCAN, SuperPixel, or (2) semantic segmentation: methods leverage foundation models like SAM [23] to generate segmentation masks [40, 51]. However, SAM produces a large and unstructured set of masks. This over-segmentation often results in a fragmented collection of masks that are visually disjointed and semantically incoherent, which necessitates aggressive post-processing. As shown in Figure 2, coherent objects like "ground" and "bench" are divided into multiple segments due to texture variations caused by shadows, lighting changes, and other environmental factors. Our method addresses this limitation through semantic-guided clustering, which generates compact, structurally coherent primitives.

Mask Clustering Problem. Given an initial set of N over-segmented masks $\mathcal{M} = \{m_1, m_2, \ldots, m_N\}$, the problem is defined as finding a consolidated set of K coherent regions

$\mathcal{C} = \{c_1, c_2, \ldots, c_K\}$, where $K \ll N$, subject to three fundamental constraints:

- **Spatial Coherence**: Masks should be merged only if they are spatially adjacent or overlapping.
- **Semantic Consistency**: Masks belonging to similar semantic entities should be grouped together.
- **Visual Similarity**: Masks with similar appearance characteristics should be clustered.

4. Semantic-guided Clustering

Unlike traditional clustering methods that rely on distance metrics to partition a discrete set of masks, our *semantic-guided clustering* approach reformulates clustering as a continuous optimization problem of coloring in image space. While conventional cluster methods (e.g., DBSCAN, K-means) operate on pre-computed mask features using pairwise distance calculations, our method achieves mask grouping through color assignment, where adjacent regions with semantic affinity are merged by sharing colors. This recoloring strategy naturally integrates spatial adjacency, visual similarity, and semantic coherence in a unified optimization framework, producing regions that correspond to meaningful visual groups rather than just statistically similar clusters.

As shown in Figure 3, our pipeline consists of three stages: (1) **Semantic-aware Coloring**: We start by randomly initializing a color palette and a color assignment matrix. Guided by a semantic saliency image (either user-specified or automatically extracted from CLIP [32]), we optimize both the palette and the assignment matrix so that regions with high semantic importance are vibrantly colored to closely match their original appearance in the input image, while low-importance regions are assigned colors that are intentionally kept very similar to one another, reducing unnecessary variation across neighboring masks. (2) **Local Color Smoothing**: In this stage, we freeze the color palette learned in the first stage and adjust the color assignments to enforce spatial coherence among adjacent masks, from which adjacent masks with shared colors are detected and merged. Through iterative N_r rounds of semantic coloring and local smoothing, the method progressively merges regions with similar colors while preserving semantically critical boundaries. In each round, a certain number of masks is reduced, resulting in a hierarchical, progressively coarser segmentation. (3) **Visual Optimization**: This final stage aggregates merged masks and adds additional detailed primitives, optimizing them to minimize the MSE loss between the rasterized vector image and input image via differentiable rendering.

4.1. Semantic Saliency Image Generation

Semantic guidance is introduced through a semantic importance map, which can be obtained in increasingly automated levels of user interaction. At the most explicit level, users may directly draw on the image to indicate which regions should receive higher semantic weight, giving them full, fine-grained control over the guidance signal. A lighter form of interaction is to provide a short text prompt, such as "butterfly's wings", from which we derive a semantic relevance map using CLIP-based image–text alignment [32]. To convert CLIP predictions into spatial relevance A, we adopt the gradient-weighted attention rollout method of [6], where relevance is propagated through the transformer layers via $A \leftarrow A + \text{Cam} \times A$, with Cam given by the element-wise product of attention probabilities and their gradients. We extract the relevance from the `[CLS]` token to the patch embeddings to produce a pixel-level saliency map indicating which regions contribute most to the text–image matching score. When no prompt is provided, our method takes a random Gaussian distribution without specific semantic emphasis.

4.2. Stage I: Semantic-aware Coloring

The objective of this first stage is to perform a global simplification of the scene's appearance by learning an optimal, compact color palette and an initial assignment of colors to masks, while preserving semantic fidelity. Let an input image be decomposed into a set of N fine-grained segmentation masks $\mathcal{M} = \{\text{mask}_1, \text{mask}_2, \ldots, \text{mask}_N\}$. We define a color palette P of K colors, where $P = \{\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_K\}$ and each $\mathbf{c}_j$ is a vector in RGB space. Our goal is to find the optimal palette P^* and a set of soft assignment weights $W = \{w_{ij}\}$ that minimize a composite loss function. The loss function $\mathcal{L}_s$ for this stage is a weighted sum of three components:

Weighted MSE Loss This ensures the recoloring process respects semantic importance. Given a semantic saliency image A that highlights semantically salient regions, this loss penalizes large color differences in areas of high importance.

$$\mathcal{L}_{\text{wmse}} = \frac{1}{|\Omega|} \sum_{p \in \Omega} A(p) \cdot \left\| \mathbf{I}^{\text{rec}}(p) - \mathbf{I}^{\text{orig}}(p) \right\|^2, \quad (2)$$

where $\mathbf{I}^{\text{orig}}$ is the original image, $\mathbf{I}^{\text{rec}}$ is the recolored image, $A(p)$ is the semantic importance weight at pixel p, and Ω denotes the image spatial domain with $|\Omega| = H \times W$ representing the total number of pixels.

Weighted Total Variation Loss This enforces spatially adaptive smoothness in the recolored image $\mathbf{I}^{\text{rec}}$ using a smoothness weight map $\mathbf{W}$, where $\mathbf{W}(p) = 1 - A(p)$ and $A(p) \in [0, 1]$ is the semantic importance at pixel p. The loss penalizes color variations between adjacent pixels, with stronger smoothing applied to low-importance regions:

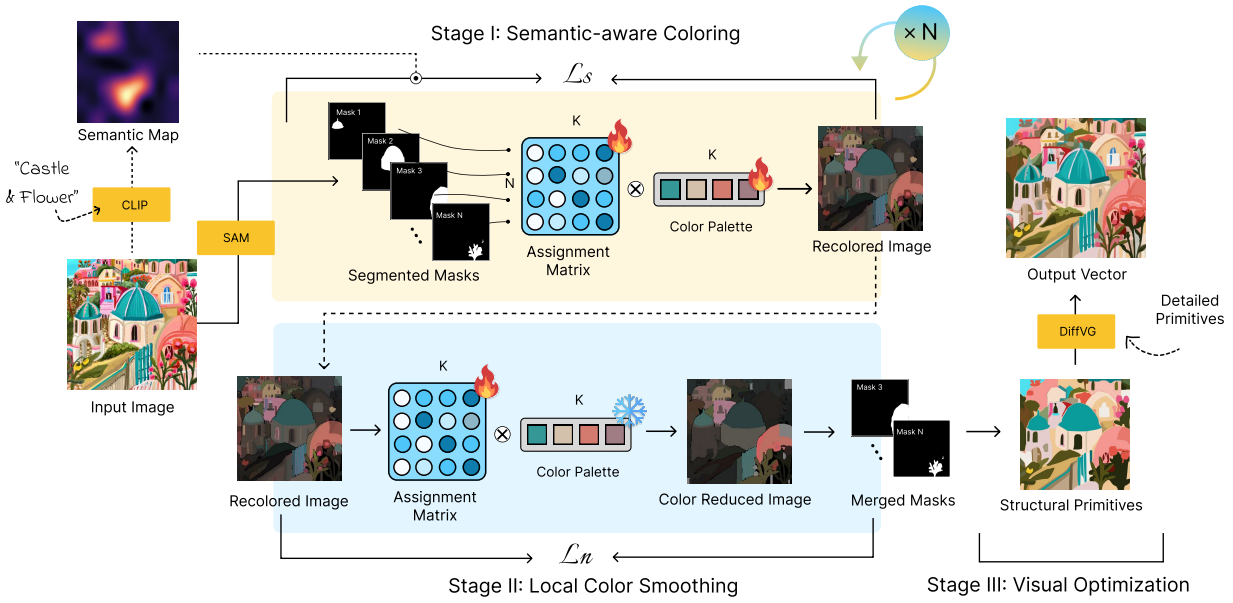

Figure 3. **Semantic-guided clustering via mask coloring**. Guided by a semantic importance map, our method begins with **Stage I**, where mask colors are optimized to produce a semantically meaningful global palette. The resulting palette is then frozen, and **Stage II** refines the local color assignments of individual masks to suppress noise and enforce spatial consistency. As neighboring masks converge toward similar colors, they naturally merge into coherent structural primitives. Stages I and II are alternated for several rounds, progressively stabilizing these structural groupings. Finally, **Stage III** introduces finer-grained primitives on top of the established structure and optimizes them to complete the hierarchical vectorization process.

$$\mathcal{L}_{\text{WTV}} = \frac{1}{|\mathcal{E}|} \sum_{(p,q)\in\mathcal{E}} \frac{\mathbf{W}(p) + \mathbf{W}(q)}{2} \cdot \left\| \mathbf{I}^{\text{rec}}(p) - \mathbf{I}^{\text{rec}}(q) \right\|^2 \tag{3}$$

Here $\mathcal{E}$ is the set of adjacent pixel pairs in the 4-connected grid, and $\mathbf{W}(p) \in [0, 1]$ indicates the smoothing strength (higher values for low semantic importance regions).

Assignment Entropy Loss This loss encourages confident color assignments by minimizing the entropy of the assignment probability distribution across the image.

$$\mathcal{L}_{\text{ent}} = -\frac{1}{|\Omega|} \sum_{p\in\Omega} \sum_{j=1}^{K} P_j(p) \log P_j(p) \tag{4}$$

where $P_j(p)$ is the probability of pixel p being assigned to palette color c_j, and Ω denotes the image spatial domain.

The total loss for Stage 1 is formulated as follows:

$$\mathcal{L}_s = \lambda_{\text{wmse}}\mathcal{L}_{\text{wmse}} + \lambda_{\text{WTV}}\mathcal{L}_{\text{WTV}} + \lambda_{\text{ent}}\mathcal{L}_{\text{ent}} \tag{5}$$

where λ_{wmse}, λ_{WTV} and λ_{ent} stand for the weight for each loss component. Both the palette P and the assignment weights W are optimized at this stage.

4.3. Stage II: Local Color Smoothing

In the second stage, we fix the optimized color palette P^* from Stage I and optimize only the assignment weights W to enforce local spatial smoothness. The loss function $\mathcal{L}_n$ for this stage is a weighted sum of two components:

Adjacency-based Merging Loss In Stage I, the semantic saliency is optimized and visually encoded by colors. The goal of this stage is to enforce color consistency between adjacent masks while preserving diversity among non-adjacent regions. Therefore, a loss is proposed to penalize the color variation among masks and their adjacent masks, defined as follows:

$$\mathcal{L}_{\text{adj}} = \underbrace{\|\mathbf{C}' - \mathbf{C}^*\|_2^2}_{\text{color matching}} + \lambda_c \frac{1}{N} \sum_{i=1}^{N} \sum_{j\in\mathcal{A}(i)} \left\| \mathbf{c}_i - \mathbf{c}_j \right\|_2^2 - \lambda_d \frac{1}{N} \sum_{i=1}^{N} \sum_{k\notin\mathcal{A}(i)} \left\| \mathbf{c}_i - \mathbf{c}_k \right\|_2^2$$

Here, $\mathbf{c}_i \in \mathbb{R}^3$ denotes the predicted RGB color assigned to region i, and $\mathbf{C}' = [\mathbf{c}_1, \ldots, \mathbf{c}_N]^\top$ collects the predicted

colors for all N regions, while $\mathbf{C}^* = [\mathbf{c}_1^*, \ldots, \mathbf{c}_N^*]^\top$ contains their corresponding target colors. The set $\mathcal{A}(i)$ represents the indices of regions that are spatially adjacent to region i, whereas $k \notin \mathcal{A}(i)$ enumerates non-adjacent regions excluding i itself. The hyperparameters λ_c and λ_d balance the contribution of the adjacency-consistency and non-adjacency-diversity terms, respectively. The first term enforces color fidelity with respect to the target colors, the second encourages adjacent regions to adopt similar colors and thus merge into coherent groups, and the final term pushes non-adjacent regions to remain distinct, preventing degenerate over-merging.

Assignment Entropy Loss The same assignment-entropy loss from Stage 1 is retained to maintain a simple color representation.

The total loss for Stage 2 is:

$$\mathcal{L}_n = \lambda_{\text{adj}} \mathcal{L}_{\text{adj}} + \lambda_{\text{ent}} \mathcal{L}_{\text{ent}} \tag{6}$$

Only the assignment weights W are optimized in this stage.

4.4. Stage III: Visual Optimization

The merged masks from Stages I and II form structural vector primitives that serve as semantic and structural containers. These are then augmented together with randomly initialized primitives to enhance visual details, such as textures and shading (i.e., detailed primitives). Using the semantic importance image as input, detailed primitives are assigned with higher probability in high-saliency regions and lower probability in low-saliency areas. The ratio of structural primitives and detailed primitives *structural-to-detailed ratio* is set to 40% structural primitives and 60% detailed by default. After detailed primitives are initialized as small circular areas with a 5-pixel radius, they are optimized via differentiable rendering, minimizing the image-space MSE loss between the rasterized vector graphic and the original image.

5. Experiments

In this section, we first introduce the details on our experimental implementation in Section 5.1, then we provide comparision results with previous methods in Section 5.2. Finally, we conduct ablation studies in Section 5.3.

5.1. Implementation Details

We implemented this method using PyTorch with the Adam optimizer. The learning rates are set to 0.01 for both color palette and assignment logits. For all examples in this work, 20 iterative rounds of Stage I and Stage II are performed, with an average of 4.83 masks reduced per iteration. Within each round, 1000 iterations and 125 iterations are performed within Stage I and Stage II, respectively. The color palette is assigned to the original number of segmented masks. Masks are segmented using the SAM model at checkpoint `sam_vit_h_4b8939.pth`. We used NVIDIA GeForce RTX 4090 with 24 GB of memory.

5.2. Comparison Results

We evaluate our method on 100 complex scene images spanning natural landscapes, indoor scenes, and static compositions. Comparisons are made against five image vectorization methods: DiffVG [26], LIVE [29], O&R [17], SGLIVE [50], and LIVSS [40]. All methods use fixed primitive counts of 64, 128, and 256 for fair comparison. Our method maintains a $40\%/60\%$ ratio of structural to detailed primitives, consistent with LIVSS. Semantic guidance is disabled by using white noise input for the vanilla version of our approach.

Visual Quality Figure 4 demonstrates that our method constructs more delicate visual details in vector representation compared to five baseline methods. For instance, in the second example, our method reconstructs the details of the 'flowers' and 'gentleman', whereas these details are notably less defined in other methods. Figure 5 compares MSE, LPIPS, and PSNR metrics across the testing dataset. Our method achieves lower MSE and LPIPS scores than all five baseline methods. DiffVG achieves superior MSE by directly optimizing this pixel-level objective; however, it produces visually fragmented results with heavily overlapping primitives, as shown in Figure 4.

Table 1. Comparison of the average VeC and standard deviation of the 100 testing images. Best result marked in bold.

VeC (%)	DiffVG	LIVE	O&R	SGLIVE	LIVSS	Ours
Avg. ↑	26.3	45.0	36.1	41.6	47.3	**51.0**
Std. ↓	27.9	**22.1**	30.8	24.8	28.3	26.7

Structural Quality To evaluate the spatial alignment between vector primitives and semantic object regions, we follow previous work [40] and use the metric Vector Compactness (VeC), which quantifies the proportion of vectors that are well-contained (over 85% area overlap) within a given semantic mask relative to all vectors overlapping that region. Table 1 reports that our method achieves superior vector compactness than the five baselines.

5.3. Ablation Studies

We conduct ablation experiments to evaluate the effectiveness of key components in our pipeline. We first compare our design of the recoloring-as-clustering mechanism with the standard clustering methods (i.e., K-means and DBSCAN), and then we ablate the effect of semantic guidance. Finally, we analyze the impact of the structure-to-detailed ratio on the final output.

Figure 4. **Qualitative comparison of visual fidelity**: compared to the five baseline methods, our approach better captures intricate image details (e.g., the texture of 'vase' in the second row example), meanwhile maintains a good overall structure (e.g., the integrity of 'gentleman' in the third row example). All examples are vectorized with 128 primitives.

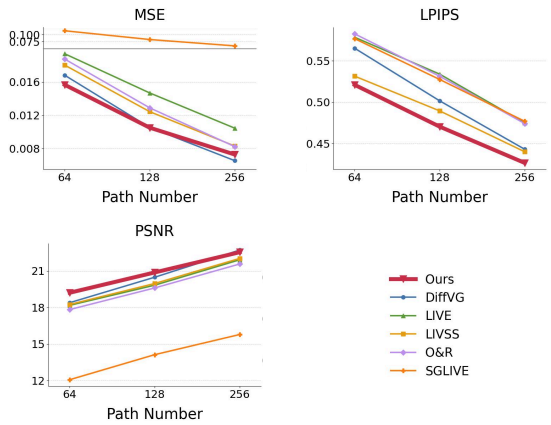

Figure 5. **Quantitative visual quality comparison**: our method outperforms the five baseline methods in all three metrics, MSE, PSNR, and LPIPS. DiffVG achieves strong MSE performance through dedicated optimization of this pixel-level objective, but it fails to maintain balanced perceptual quality in LPIPS.

Ablation on Clustering We ablate the choice of clustering method during the iterative mask merging process. Figure 6 shows the comparison of our method with the standard clustering methods, K-means and DBSCAN. Although results from K-Means are visually more similar to the input image, they exhibit severe artifacts on shapes, limiting interpretability and editability. Yet, DBSCAN fails to produce coherent structural shapes. In contrast, our results are not only visually delicate but also structurally coherent and highly editable.

Ablation on Semantic Guidance We examine the impact of *semantic image* on guiding the distribution of structural vector primitives, compared to an ablated version that takes a random Gaussian distribution image without certain semantic specification (i.e., without semantic guidance). As shown in Figure 7, semantic guidance like 'truck' in the first example merges background primitives (e.g., 'house'), focusing attention on relevant regions. Similarly, 'flowers' guidance in the second example redirects primitives toward floral areas while merging irrelevant masks like 'fruit' and 'bottles'.

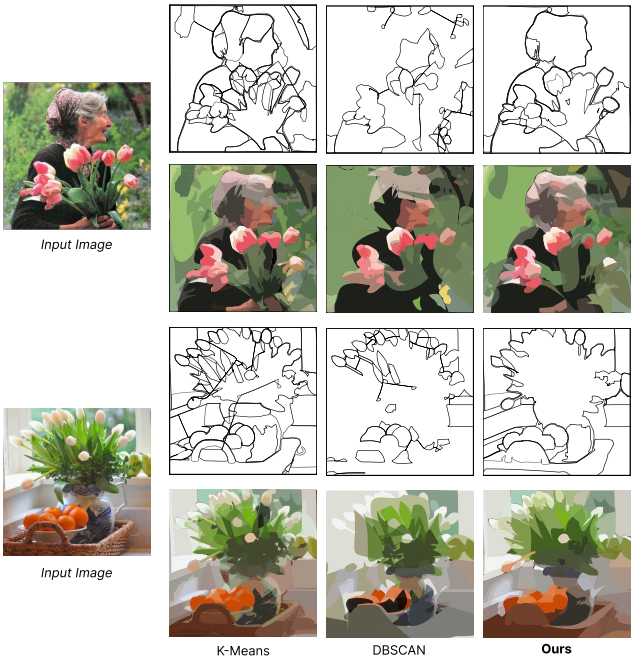

Figure 6. **Ablation on clustering method**: for each example, images in the first and second rows represent the structural shapes and visual results of each method, correspondingly. Both examples are vectorized with 64 vector primitives.

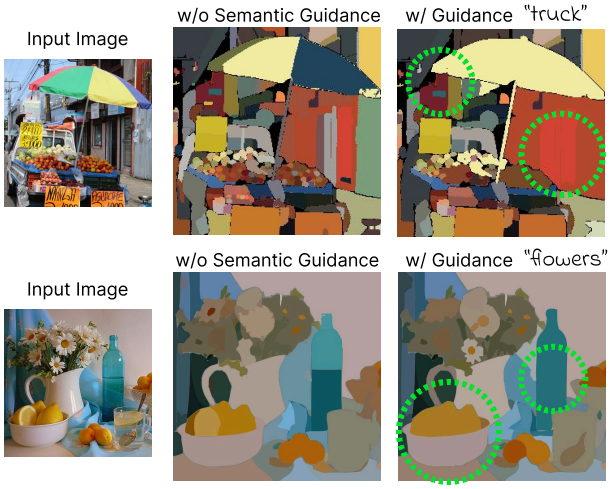

Figure 7. **Ablation on semantic guidance**: When conditioned on text prompts, our method effectively merges vector primitives outside regions of interest (highlighted by green circles). For clarity, only structural primitives are visualized in these examples.

Ablation on Structural-to-Detailed Ratio We investigate how the structural-to-detailed primitive ratio influences vectorization quality in Stage III. Five ratio levels (30%, 35%, 40%, 45%, and 50% structural primitives) are evaluated across three path counts (64, 128, and 256 total paths).

Figure 8 illustrates how MSE varies with different ratio levels. The results demonstrate that this ratio balances structural representation against visual quality. As the structural ratio increases, visual quality degrades due to fewer detailed primitives. Based on our analysis, we set the ratio to 40%, which represents the optimal trade-off point.

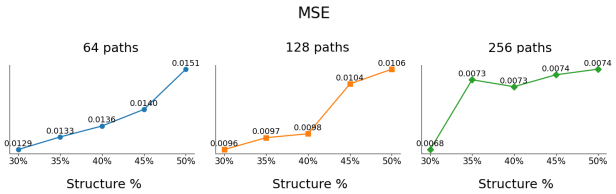

Figure 8. **Ablation on structural-to-detailed ratio**: testing five ratios over three primitive conditions, this ratio trades off structural integrity against visual quality.

6. Conclusions, Limitations and Future Work

Across this work, we introduced a semantic guided recoloring framework for consolidating large sets of over segmented masks into a more compact vector representation. Instead of imposing an explicit hierarchy or object level structure, the method focuses on guiding the merging process through a text conditioned semantic saliency map. This relevance signal helps preserve distinctions in regions associated with meaningful content, while redundant areas naturally converge. The result is a vectorization that places primitives where they are most useful, reducing fragmentation and improving editability.

Although the approach leads to more coherent groupings, it does not yet produce a fully structured or layered decomposition. Some regions remain partially fragmented, and the representation does not recover a clear ordering of shapes. The outcome is therefore semantically informed rather than fully semantic, yet the improvement in coherence and the reduction of redundancy demonstrate the practical value of semantic guided recoloring.

A central idea in our framework is to begin with a rich pool of masks and guide their consolidation through differentiable recoloring instead of discarding information at the start. This formulation allows the semantic relevance map to influence which shapes fuse and which remain distinct. Only after this consolidation stage do we refine the representation through differentiable rendering to recover visual detail.

This direction suggests further opportunities for integrating semantic cues into vectorization workflows. Future research may explore stronger grouping signals, clearer organizational principles, and more expressive consolidation strategies, potentially enabling higher level editing operations that respond directly to semantic intent.

References

[1] Radhakrishna Achanta et al. Slic superpixels compared to state-of-the-art superpixel methods. *IEEE TPAMI*, 2012. 2, 3

[2] Pablo Arbelaez, Michael Maire, Charless Fowlkes, and Jitendra Malik. Contour detection and hierarchical image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(5):898–916, 2011. 3

[3] Sebastiano Battiato, Giovanni Gallo, and Giuseppe Messina. Svg rendering of real images using data dependent triangulation. In *Proceedings of the 20th Spring Conference on Computer Graphics*, page 185–192, New York, NY, USA, 2004. Association for Computing Machinery. 2

[4] Axel Carlier et al. Deepsvg: A hierarchical generative network for vector graphics animation. In *SIGGRAPH Asia*, 2020. 2

[5] Huiwen Chang, Kalyan Sunkavalli, Johannes Kopf, and Sing Bing Kang. Palette-based photo recoloring. *ACM TOG*, 2015. 3

[6] Hila Chefer, Shir Gur, and Lior Wolf. Generic attention-model explainability for interpreting bi-modal and encoder-decoder transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 397–406, 2021. 4

[7] Ye Chen, Bingbing Ni, Xuanhong Chen, and Zhangli Hu. Editable image geometric abstraction via neural primitive assembly. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 23457–23466, 2023. 2

[8] Dorin Comaniciu and Peter Meer. Mean shift: A robust approach toward feature space analysis. *IEEE TPAMI*, 2002. 2, 3

[9] Laurent Demaret, Nira Dyn, and Armin Iske. Image compression by linear splines over adaptive triangulations. *Signal Processing*, 86(7):1604–1616, 2006. 2

[10] Edoardo Alberto Dominici, Nico Schertler, Jonathan Griffin, Shayan Hoshyari, Leonid Sigal, and Alla Sheffer. Polyfit: perception-aligned vectorization of raster clip-art via intermediate polygonal fitting. *ACM Trans. Graph.*, 39(4), 2020. 2

[11] Maria Dziuba, Ivan Jarsky, Valeria Efimova, and Andrey Filchenkov. Image vectorization: a review, 2023. 2

[12] Jean-Dominique Favreau, Florent Lafarge, and Adrien Bousseau. Fidelity vs. simplicity: a global approach to line drawing vectorization. *ACM Trans. Graph.*, 35(4), 2016. 2

[13] Jean-Dominique Favreau, Florent Lafarge, and Adrien Bousseau. Photo2clipart: image abstraction and vectorization using layered linear gradients. *ACM Trans. Graph.*, 36 (6), 2017. 2

[14] Pedro F Felzenszwalb and Daniel P Huttenlocher. Efficient graph-based image segmentation. In *IJCV*, 2004. 2, 3

[15] Jon Ferraiolo, Fujisawa Jun, and Dean Jackson. *Scalable vector graphics (SVG) 1.0 specification*. iuniverse Bloomington, 2000. 1

[16] Jonathan Harel, Christof Koch, and Pietro Perona. Graph-based visual saliency. In *NIPS*, 2006. 3

[17] 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*, pages 2148–2156, 2024. 1, 2, 3, 6

[18] Qibin Hou et al. Deeply supervised salient object detection with short connections. In *CVPR*, 2017. 3

[19] Teng Hu, Ran Yi, Baihong Qian, Jiangning Zhang, Paul L Rosin, and Yu-Kun Lai. Supersvg: Superpixel-based scalable vector graphics synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24892–24901, 2024. 2, 3

[20] Laurent Itti, Christof Koch, and Ernst Niebur. A model of saliency-based visual attention for rapid scene analysis. *IEEE TPAMI*, 1998. 3

[21] Hao Jiang et al. Saliency detection via depth-induced saliency estimation and refinement. In *CVPR*, 2021. 3

[22] Jaemin Kim et al. Deep recolorization for art and design. In *ICCV*, 2019. 3

[23] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4015–4026, 2023. 1, 3

[24] S. S. S. Kruthiventi et al. Deepfix: A fully convolutional neural network for predicting human eye fixations. In *IEEE TIP*, 2017. 3

[25] Tzu-Mao Li, Bálint Lukács, et al. Differentiable vector graphics rasterization for editing and learning. In *SIGGRAPH*, 2020. 2

[26] Tzu-Mao Li, Michal Lukáč, Michaël Gharbi, and Jonathan Ragan-Kelley. Differentiable vector graphics rasterization for editing and learning. *ACM Trans. Graph.*, 39(6), 2020. 3, 6

[27] Zhenyu Li et al. Semantic-aware region clustering for image abstraction. In *CVPR*, 2023. 3

[28] Raphael Gontijo Lopes, David Ha, Douglas Eck, and Jonathon Shlens. A learned representation for scalable vector graphics, 2019. 2

[29] 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 16293–16302, 2022. 1, 3, 6

[30] Alexandrina Orzan, Adrien Bousseau, Holger Winnemöller, Pascal Barla, Joëlle Thollot, and David Salesin. Diffusion curves: a vector representation for smooth-shaded images. *ACM Transactions on Graphics (TOG)*, 27(3):1–8, 2008. 2

[31] Sagi Polaczek, Yuval Alaluf, Elad Richardson, Yael Vinker, and Daniel Cohen-Or. Neuralsvg: An implicit representation for text-to-vector generation, 2025. 2

[32] 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. 4

[33] Alec Radford et al. Learning transferable visual models from natural language supervision. In *ICML*, 2021. 3

[34] Pradyumna Reddy et al. Live: Towards high-fidelity vector graphics with a learned differentiable rasterizer. In *ICCV*, 2021. 2

[35] Byungjun Ryu et al. Im2vec: Synthesizing vector graphics without vector supervision. In *CVPR*, 2023. 2

[36] Peter Selinger. Potrace: a polygon-based tracing algorithm. 2003. 2

[37] Yiren Song, Danze Chen, and Mike Zheng Shou. Layertracer: Cognitive-aligned layered svg synthesis via diffusion transformer, 2025. 2

[38] Jian Sun, Lin Liang, Fang Wen, and Heung-Yeung Shum. Image vectorization using optimized gradient meshes. *ACM Trans. Graph.*, 26(3):11–es, 2007. 2

[39] Xingze Tian and Tobias Günther. A survey of smooth vector graphics: Recent advances in representation, creation, rasterization, and image vectorization. *IEEE Transactions on Visualization and Computer Graphics*, 30(3):1652–1671, 2024. 2

[40] Zhenyu Wang, Jianxi Huang, Zhida Sun, Yuanhao Gong, Daniel Cohen-Or, and Min Lu. Layered image vectorization via semantic simplification. *arXiv preprint arXiv:2406.05404*, 2024. 1, 2, 3, 6

[41] Martin Weber. Autotrace—converting bitmap to vector graphics. http://autotrace.sourceforge.net/. 2

[42] Tian Xia, Binbin Liao, and Yizhou Yu. Patch-based image vectorization with automatic curvilinear feature alignment. *ACM Transactions on Graphics (TOG)*, 28(5):1–10, 2009. 2

[43] Guofu Xie, Xin Sun, Xin Tong, and Derek Nowrouzezahrai. Hierarchical diffusion curves for accurate automatic image vectorization. *ACM Trans. Graph.*, 33(6), 2014. 2

[44] Hongyang Xu et al. Semantic superpixels via deep embedding. In *ECCV*, 2022. 3

[45] Chuan Yan, Yong Li, Deepali Aneja, Matthew Fisher, Edgar Simo-Serra, and Yotam Gingold. Deep sketch vectorization via implicit surface extraction. *ACM Transactions on Graphics (TOG)*, 43(4):1–13, 2024. 2

[46] Ming Yang, Hongyang Chao, Chi Zhang, Jun Guo, Lu Yuan, and Jian Sun. Effective clipart image vectorization through direct optimization of bezigons. *IEEE Transactions on Visualization and Computer Graphics*, 22(2):1063–1075, 2016. 2

[47] Peiying Zhang, Nanxuan Zhao, and Jing Liao. Text-to-vector generation with neural path representation, 2024. 1, 2

[48] Till Zhang et al. Clipseg: Image segmentation using text and vision. In *CVPR*, 2023. 3

[49] Shuang Zhao, Frédo Durand, and Changxi Zheng. Inverse diffusion curves using shape optimization. *IEEE Transactions on Visualization and Computer Graphics*, 24(7):2153–2166, 2018. 2

[50] Hengyu Zhou, Hui Zhang, and Bin Wang. Segmentation-guided layer-wise image vectorization with gradient fills. *arXiv preprint arXiv:2408.15741*, 2024. 2, 6

[51] Haokun Zhu, Juang Ian Chong, Teng Hu, Ran Yi, Yu-Kun Lai, and Paul L Rosin. Samvg: A multi-stage image vectorization model with the segment-anything model. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4350–4354. IEEE, 2024. 1, 2, 3