DIFFERENTIABLE ONE LINE DRAWING

Anonymous authors
Paper under double-blind review

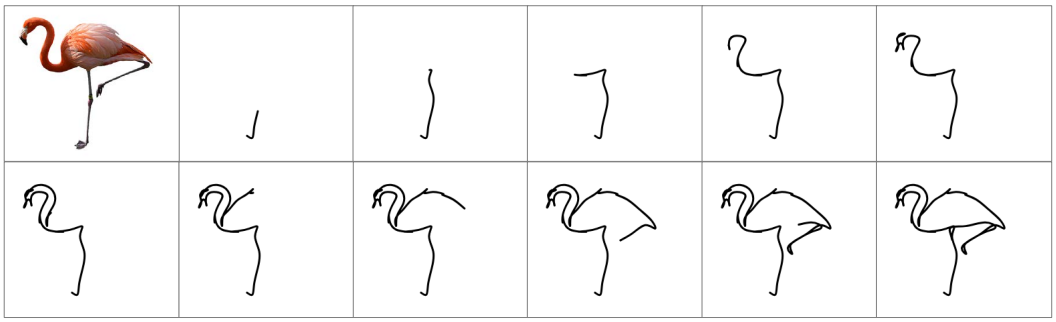

Figure 1: **A trajectory to watch and imitate.** The reference flamingo appears at the upper left. Subsequent panels, read left to right across the top row and then the bottom row, progressively reveal one generated curve, from the leg through the body, neck, and head to internal detail. These are drawing-replay stages, not optimization iterations. The supplemental HTML provides dynamic examples.

ABSTRACT

Image-based one-line drawing seeks to turn a reference into a single stroke that preserves the subject's structure and provides a drawing process that users can inspect and imitate. Existing image generators can suggest the appearance of one-line art without providing a continuous path, while vector synthesis can change the reference geometry or route the stroke without following the subject's form. We propose a contour-guided differentiable framework that couples structural fidelity with locally organized drawing order. It first constructs a path along sampled foreground-boundary points, inserting attention-selected interior points near their closest boundary points. It then refines a cubic spline through image-feature matching and geometric regularization; optional width optimization varies line thickness along the stroke. An attachment-length bound explains the locality of the initialization. Progressive replay exposes the drawing order, as illustrated in the supplemental HTML. Our aim is to turn the visual economy associated with skilled one-line drawing into a process that non-experts can inspect, imitate, and experience.

1 INTRODUCTION

One-line drawing depicts a subject with a single uninterrupted stroke (See Fig. 1). Its economy makes it useful for illustration, product-design sketches, and advertising, while the act of drawing offers an accessible way to explore visual abstraction. The cat, flower, glass, and figure sketches in Fig. 2 illustrate how a few connected contours and details can convey recognizable forms. Producing such drawings requires deciding both which structures to retain and how to connect them. Image-based generation can make this process available to non-experts as a trajectory to watch and imitate, and to designers and illustrators as an editable draft.

Our goal is to generate both a recognizable drawing and an explicit route for constructing it. A vector curve records where the stroke travels and in which order, enabling geometric editing and progressive replay. In Fig. 1, the flamingo emerges as one stroke develops from the leg through the body, neck, and head before adding internal detail. This example highlights two requirements:

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.



Figure 2: **Artistic motivation.** Artist-drawn examples illustrate how a single economical trajectory can convey recognizable forms. We aim to preserve this balance between structural abstraction and continuous curve organization in image-conditioned one-line drawing

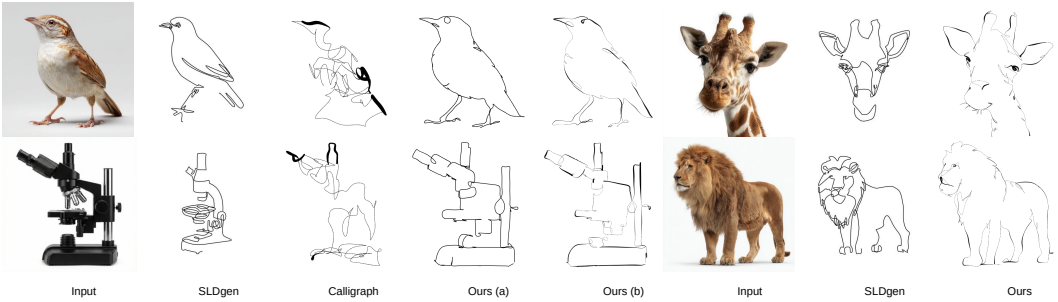

Figure 3: **Image-conditioned one-line abstractions.** Left: bird and microscope inputs with SLDgen, Calligraph, and two outputs of our method, labeled (a) and (b). Right: giraffe and lion inputs with SLDgen and our outputs. Compare the silhouette, pose, and spatial arrangement of parts with each input: semantic recognizability alone does not establish structural correspondence. These static examples illustrate the goal of retaining reference structure while producing a coherent curve; they do not isolate the effect of text guidance or establish traversal quality. Our drawing progression is illustrated in Figure 1 and the supplemental HTML.

structural fidelity, preserving the reference's silhouette, proportions, and arrangement of parts; and *form-following progression*, organizing the route around those structures with smooth transitions and local visits to details. These requirements can compete: including more detail can create long detours or dense loops, whereas simplifying the route can omit distinctive features. We seek a useful balance for a drawing that can be inspected as a process as well as a finished image.

Raster generation addresses the appearance of this task but does not directly specify its construction. The GPT-image and Nano Banana configurations examined by Magne et al. (2026) can produce convincing one-line-style images, yet pixel-space generation does not enforce a single connected stroke. Recovering a path requires vectorization and, when marks are disconnected, additional connections that can introduce travel and overdraw. For a drawing demonstration, the output therefore needs an explicit continuous route in addition to the final image.

SLDgen (Magne et al., 2026) and Calligraph (Berio et al., 2025b) make substantial progress by optimizing continuous vector curves. The remaining routing challenge is how that curve proceeds through the subject. SLDgen initializes a spline using a traveling-salesperson-problem (TSP) tour over sampled points; Calligraph demonstrates single-stroke abstraction using Voronoi sampling and TSP initialization. A TSP tour specifies a visit order and can be replayed, but minimizing travel among samples does not explicitly preserve the sequence of points encountered along the object's boundary. A route may therefore connect nearby locations without developing the outline and its details in a coherent progression. The missing ingredient is a shape-based organization of the route, rather than continuity or replay alone.

Preserving the reference structure is a separate challenge, even with substantial computation. SLDgen (Magne et al., 2026) combines text-conditioned diffusion with image-derived depth, while the evaluated Calligraph (Berio et al., 2025b) configuration uses text and style-image guidance. These conditions support expressive abstraction without enforcing the reference's precise geometry. Fig. 3 shows the resulting distinction: the SLDgen microscope alters the eyepiece and base geometry, and the Calligraph example loses much of the instrument's recognizable arrangement; the SLDgen giraffe also changes the head proportions. Figure 7 shows departures in neck shape and body structure for the evaluated vector baselines. Such departures matter for the recognizable abstractions motivated by Fig. 2 and the reference-based drawing process illustrated in Figure 1.

We address both requirements with a contour-guided construction followed by image-based refinement. First, we sample the foreground boundary in its tracing order; these boundary points serve as *contour anchors*. We also sample points inside the object using image-attention scores to select salient details; these are the *interior samples*. Inserting each interior sample after its nearest contour anchor creates local visits to detail within the boundary sequence. The resulting ordered points initialize a single cubic B-spline. This construction makes the shape scaffold explicit before optimization. An upper bound on the initial path length relates added travel to the distances between interior samples and their anchors, explaining why local attachment can avoid large detours.

During refinement, direct matching to frozen image features ties the rendered curve to the input, while geometric penalties regulate bending, control spacing, and nearby segments. The point sequence remains fixed as its positions move, retaining the initialized traversal order. To extend the draft's visual expression, an optional continuous width field changes how strongly different portions of the line appear: thicker portions carry more visual weight, while thinner portions make a lighter mark. This adds control over line thickness without introducing additional strokes. The resulting path can be progressively revealed, paused, and inspected in the supplemental HTML. Our experiments evaluate image correspondence and trajectory properties; whether this supports easier imitation requires a user study.

Our contributions are threefold. First, we introduce a contour-local insertion rule that combines boundary order with attention-selected interior detail, together with a length bound whose attachment term is minimized by nearest-anchor assignment. Second, we develop a drawing pipeline that uses this ordered initialization for image-guided curve refinement and optional continuous width optimization, connecting reference-based abstraction to an explicit trajectory for replay. Third, paired experiments on inputs identify the contribution of local assignment to both path economy and image alignment, and separate regularization and width analyses characterize the trade-offs in the resulting drawings.

2 Related Work

2.1 Continuous-Line Ordering and Sketch Generation

Continuous-line drawing jointly involves abstraction and path ordering. Early TSP-art methods connect image-dependent samples into a single tour (Bosch & Herman, 2004; Kaplan & Bosch, 2005), while graph-based approaches support continuous-line abstraction and hierarchical detail (Wong & Takahashi, 2011; 2013a;b). Recent SLDgen instead optimizes a single spline with semantic guidance (Magne et al., 2026). These methods highlight that ordering is an integral part of continuous-line representation.

Sequential sketch models generate drawings as ordered strokes, including SketchRNN (Ha & Eck, 2018), CoSE (Aksan et al., 2020), BézierSketch (Das et al., 2020), and DoodleFormer (Bhunia et al., 2022). However, stroke order does not imply a globally connected trajectory. Unlike multi-stroke sketch abstraction methods, our problem requires both semantic abstraction and a globally ordered centerline, making stroke connectivity part of the representation rather than a post-processing constraint.

2.2 Differentiable Vector Graphics and Image Abstraction

Classical B-splines provide compact curve representations with local control and smoothness (de Boor, 2001), and have been extensively studied for geometric curve fitting (Wang et al., 2006). Differentiable vector rasterization further enables image-space objectives to directly optimize vector geometry (Li et al., 2020). Building on this paradigm, CLIPasso performs semantic object abstraction with sparse vector strokes (Vinker et al., 2022), while VectorFusion and DiffSketcher introduce diffusion priors for vector graphics and sketch synthesis (Jain et al., 2023; Xing et al., 2023). SwiftSketch further learns image-to-vector sketch generation from constructed image–sketch pairs (Arar et al., 2025).

Related SVG generation methods learn vector representations directly. DeepSVG models hierarchical SVG structures (Carlier et al., 2020), while IconShop sequentializes SVG commands for text-guided vector icon synthesis (Wu et al., 2023). These approaches primarily address vector con-

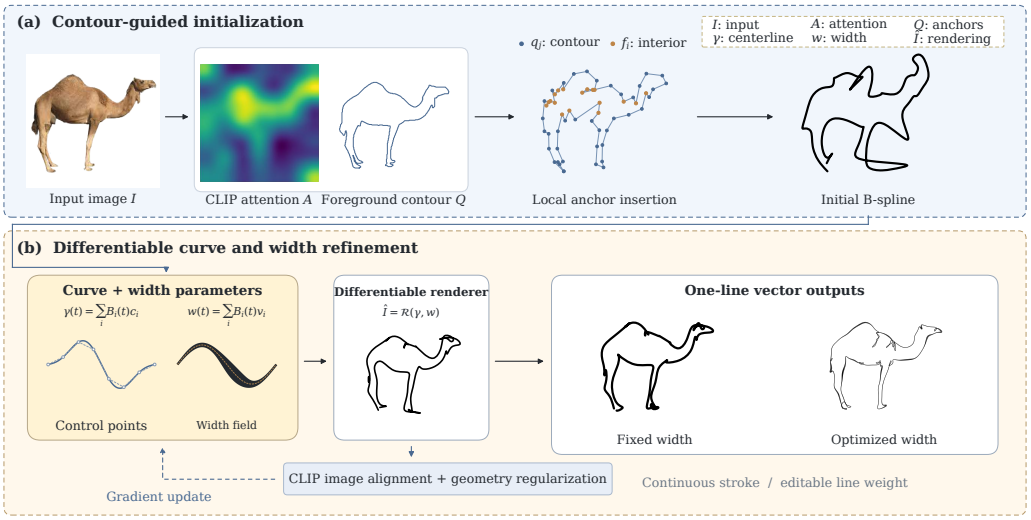

Figure 4: **Pipeline of our method.** A foreground contour and CLIP attention construct an ordered trajectory through local anchor insertion. The resulting B-spline and optional width field are refined through differentiable rendering with image alignment and geometric regularization.

tent generation rather than the topology of a single drawing trajectory. In our setting, all retained structures must instead be organized along one continuous centerline while preserving the geometry of a specific reference image.

2.3 Long Curves and Expressive Marks

Long parametric curves provide a natural representation for one-line graphics but require balancing image fidelity, smoothness, and trajectory complexity. Berio et al. integrate long smoothing B-splines with differentiable graphics for image abstraction and variable-width marks (Berio et al., 2025b), while Tojo et al. optimize continuous curves for fabricable 3D wire art (Tojo et al., 2024). Related work on robot drawing further studies executable trajectories and movement organization (Berio et al., 2025a).

Our spline representation and Bézier conversion therefore rely on standard spline properties rather than introducing a new curve primitive (de Boor, 2001). Our contribution instead couples contour-local initialization with image-conditioned refinement and an explicit traversal order. Since geometric simplicity does not necessarily imply perceptual or artistic quality, we distinguish measurable trajectory properties from artist preference (Gryaditskaya et al., 2019).

3 Method

3.1 Overview and Representation

Figure 4 summarizes our framework. Given an input image $I \in [0,1]^{H\times W\times 3}$, we construct a single continuous centerline $\gamma : [0,1] \to \mathbb{R}^2$ and, optionally, a positive width field $w : [0,1] \to \mathbb{R}_+$. The parameter t specifies traversal order rather than physical time or arc length. A differentiable renderer $\mathcal{R}$ produces $\hat{I} = \mathcal{R}(\gamma, w)$, which is optimized against the input together with geometric regularization. The traversal order is fixed after initialization, while curve geometry and, when enabled, line width are optimized.

3.2 Contour-Guided Initialization

For images with complex backgrounds, we optionally apply a foreground segmentation model (e.g., U^2-Net (Qin et al., 2020) or SAM (Kirillov et al., 2023)) and extract the longest contour from the resulting mask; this preprocessing is omitted when the foreground is already clean. Sampling the contour in tracing order gives $Q = (q_0, \ldots, q_{N_c-1})$, where $q_j \in \mathbb{R}^2$ denotes a contour anchor. The

control budget is

$$N = \max(Ks, 30) + 3, \qquad N_f = \lfloor \rho N \rfloor, \qquad N_c = N - N_f, \tag{1}$$

where K is the requested complexity budget, s is the segments-per-stroke option, and $\rho \in [0, 1]$ controls the fraction assigned to interior evidence. In our main configuration, $K = 200$, $s = 1$, and $\rho = 0.4$, yielding $N = 203$ controls.

Let A denote the CLIP saliency map and M_{in} the eroded foreground mask. We draw N_f interior samples $F = \{f_i\}_{i=1}^{N_f}$ according to

$$p(x) = \frac{[A(x)M_{\text{in}}(x)]^2}{\sum_y [A(y)M_{\text{in}}(y)]^2}. \tag{2}$$

Additional sampling details and fallback rules are given in Appendix A.

Each interior sample is attached to its nearest contour anchor,

$$a(i) = \arg \min_{0 \le j < N_c} \|f_i - q_j\|_2. \tag{3}$$

Let $F_j = \{f_i : a(i) = j\}$ denote the samples assigned to q_j, sorted by their distance to the anchor. We form the open ordered sequence

$$R = (q_0, F_0, q_1, F_1, \ldots, q_{N_c-1}, F_{N_c-1}), \tag{4}$$

which directly defines a global traversal while keeping interior excursions local to the contour.

3.3 Locality of Anchor Insertion

The nearest-anchor construction admits a simple bound explaining its locality.

Proposition 1 (Attachment bound)**.** *Let $L(R)$ denote the control-polygon length of Eq. equation 4, and let*

$$L_Q = \sum_{j=0}^{N_c-2} \|q_{j+1} - q_j\|_2.$$

For any assignment a and ordering within its groups,

$$L(R) \le L_Q + 2 \sum_{i=1}^{N_f} \|f_i - q_{a(i)}\|_2. \tag{5}$$

Nearest-anchor assignment minimizes the right-hand side over assignments to the fixed anchors.

Proof. For a group $(f_1, \ldots, f_m)$ attached to q_j, let $r_k = \|f_k - q_j\|_2$. By the triangle inequality, each internal edge is at most $r_k + r_{k+1}$, while the transition to q_{j+1} is at most $r_m + \|q_{j+1} - q_j\|_2$. Summing over all groups gives Eq. equation 5. Since each attachment term is independent, assigning f_i to its nearest anchor minimizes the bound. □

The bound concerns initialization rather than the optimized trajectory: it does not claim that the resulting route is globally shortest. Assignment requires $O(N_f N_c)$ distance comparisons, followed by sorting within each anchor group.

3.4 Spline and Width Parameterization

The ordered sequence R initializes N cubic B-spline control points $\{c_i\}_{i=0}^{N-1}$. Let $B_i(t)$ denote the normalized nonnegative cubic B-spline basis functions. We parameterize the centerline and width as

$$\gamma(t) = \sum_{i=0}^{N-1} B_i(t) c_i, \qquad w(t) = \sum_{i=0}^{N-1} B_i(t) v_i, \tag{6}$$

where v_i is the width coefficient associated with c_i. A fixed-width run replaces $w(t)$ with a scalar; otherwise, v_i is optimized and clamped to $[0.1, 3.0]$ in renderer units.

The local support and partition-of-unity properties of cubic B-splines provide smooth local control over geometry and width (de Boor, 2001). Fixed-width output is converted to connected cubic Bézier segments, whereas variable-width output samples the spline and exports consecutive ribbon polygons. Exact endpoint and export conventions are provided in Appendix A.

3.5 DIFFERENTIABLE CURVE OPTIMIZATION

We optimize the rendered curve against the input using a frozen CLIP RN101 encoder (Radford et al., 2021). Let T_a denote the a-th paired image transformation, ϕ_ℓ the feature map at layer ℓ, and ϕ_g the global embedding. Over one original and four augmented views, the image objective is

$$\mathcal{L}_{\text{img}} = 100\,\mathbb{E}_a\Bigg[\sum_{\ell=1}^{5}\alpha_\ell\,\text{MSE}\Big(\phi_\ell(T_a\hat{I}),\phi_\ell(T_aI)\Big) + 0.1\left(1-\cos\Big(\phi_g(T_a\hat{I}),\phi_g(T_aI)\Big)\right)\Bigg], \tag{7}$$

where $\alpha = (0, 0, 1, 1, 0)$.

For geometric regularization, define normalized controls $\bar{c}_i = (c_i^x/W, c_i^y/H)$, the number of control-polygon segments $S = N - 1$, their midpoints $\bar{m}_i = (\bar{c}_i + \bar{c}_{i+1})/2$, and unit directions

$$\bar{d}_i = \frac{\bar{c}_{i+1} - \bar{c}_i}{\|\bar{c}_{i+1} - \bar{c}_i\|_2 + 10^{-6}}.$$

We use

$$\mathcal{L}_b = \text{mean}_{1\le i<N-1}\,\|\Delta^2\bar{c}_i\|_2^2 + \max_{1\le i<N-1}\|\Delta^2\bar{c}_i\|_2^2, \tag{8}$$

$$\mathcal{L}_s = \text{Var}_{0\le i<S}\left(\|c_{i+1} - c_i\|_2\right), \tag{9}$$

$$\mathcal{L}_p = \frac{1}{S^2}\sum_{\substack{0\le i,j<S\\ |i-j|>1}}\frac{(\bar{d}_i^\top\bar{d}_j)^2 + 0.5}{\|\bar{m}_i - \bar{m}_j\|_2^2 + 0.02^2}, \tag{10}$$

where $\Delta^2\bar{c}_i = \bar{c}_{i-1} - 2\bar{c}_i + \bar{c}_{i+1}$. Here $\mathcal{L}_b$ suppresses abrupt bending, $\mathcal{L}_s$ regularizes control-point spacing, and $\mathcal{L}_p$ discourages nearby nonadjacent segments, particularly those with parallel or antiparallel orientations.

The final objective is

$$\mathcal{L} = \mathcal{L}_{\text{img}} + 0.5\mathcal{L}_b + 0.01\mathcal{L}_s + 0.01\mathcal{L}_p. \tag{11}$$

We optimize control positions $\{c_i\}$ with Adam at learning rate 1.0; when variable width is enabled, $\{v_i\}$ is jointly optimized at learning rate 0.01.

4 EXPERIMENTS

We evaluate our method from four aspects: trajectory construction, geometric regularization, structural correspondence with existing single-line methods, and computational cost. We further demonstrate variable-width rendering and progressive replay as extensions of the same ordered centerline representation.

4.1 EXPERIMENTAL SETUP

Our controlled ablation uses 51 input images and five configurations, resulting in 255 runs. Unless otherwise specified, all configurations use 203 control points, $\rho = 0.4$, fixed line width, a 512×512 canvas, and 300 optimization iterations. Each regularizer ablation removes only the corresponding loss term, while the random-anchor variant replaces nearest-anchor assignment using the same sampled contour and interior evidence. Complete configurations are provided in Appendix B.1.

We evaluate trajectory geometry using length normalized by the canvas diagonal and cumulative absolute turning angle. Image correspondence is measured primarily with Image CLIP (Radford et al., 2021). For broader benchmark comparison, we additionally report Text CLIP, DINOv2 similarity (Oquab et al., 2024), aesthetic score, and FID. Trajectory measurements and image-based metrics are detailed in Appendices B.2 and C.1, respectively.

Figure 5 provides an overall comparison with representative image-generation, sketch-vectorization, and single-line drawing methods. Our method achieves strong image-level and semantic correspondence while satisfying the strict single-trajectory representation. Complete numerical benchmark results are reported in Appendix D.1.

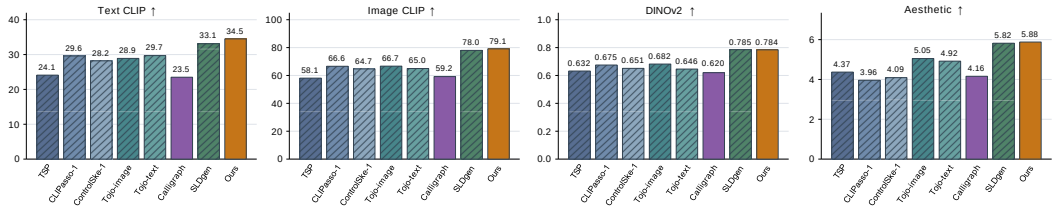

Figure 5: **Benchmark comparison across representative methods.** Text CLIP, Image CLIP, DINOv2, and aesthetic scores are shown for representative raster, multi-stroke, connected, and single-line methods. Our method maintains strong semantic and image correspondence while producing an explicit single continuous vector trajectory.

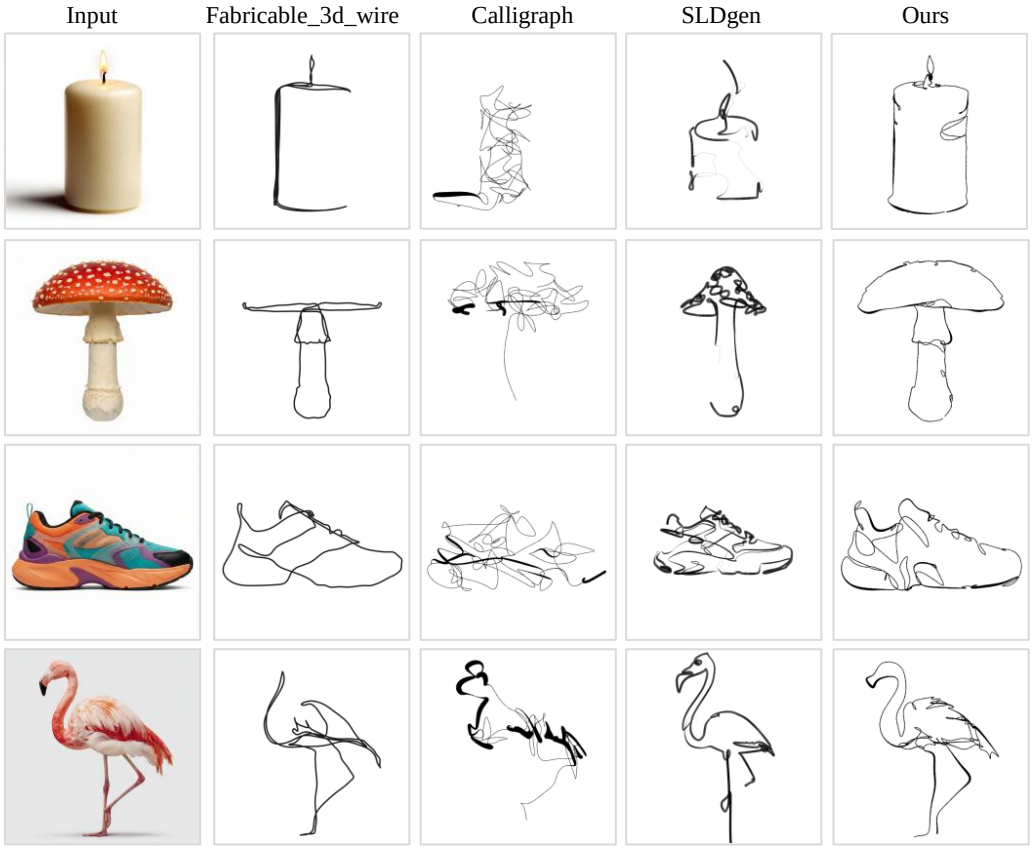

Figure 6: **Qualitative comparison with single-line vector baselines.** From left to right: input, Fabricable 3D Wire (Tojo et al., 2024), Calligraph (Berio et al., 2025b), SLDgen (Magne et al., 2026), and our method. Across diverse objects, our method better preserves the input silhouette and part structure while maintaining a coherent single continuous trajectory.

As shown in Fig. 6, the compared methods can generate recognizable single-line abstractions, but differ substantially in their preservation of reference geometry. Our results more consistently retain the global silhouette, proportions, and major part structure across objects such as the candle, mushroom, shoe, and flamingo. Additional qualitative comparisons are provided in Appendix D.2.

4.2 Trajectory Construction and Ablation

Effect of local anchor insertion. We first evaluate the proposed contour-local assignment before optimization. The mean initial polygon length is 10.51 with nearest anchors, compared with 53.98 for random assignment. The corresponding bounds from Proposition 1 are 20.08 and 67.42, respectively. After differentiable refinement, nearest-anchor assignment reduces normalized trajectory length from 20.61 to 5.26 and increases Image CLIP from 64.88 to 79.10, a gain of 14.22 points (Ta-

Table 1: Controlled ablation on 51 inputs. Length is normalized by the canvas diagonal, and Turn denotes cumulative absolute turning in radians. The last column reports the Image CLIP difference between the full model and each variant with a 95% paired bootstrap interval.

Variant	Length ↓	Turn (rad) ↓	Image CLIP ↑	Δ Image CLIP [95% interval]
Random anchor assignment	20.61	417.59	64.88	14.22 [12.12, 16.28]
Without bending $\mathcal{L}_b$	5.21	299.66	77.05	2.05 [1.30, 2.75]
Without proximity $\mathcal{L}_p$	5.20	297.99	77.54	1.56 [0.97, 2.09]
Without spacing $\mathcal{L}_s$	6.34	315.82	76.02	3.08 [2.19, 3.95]
Full method	5.26	**296.11**	**79.10**	—

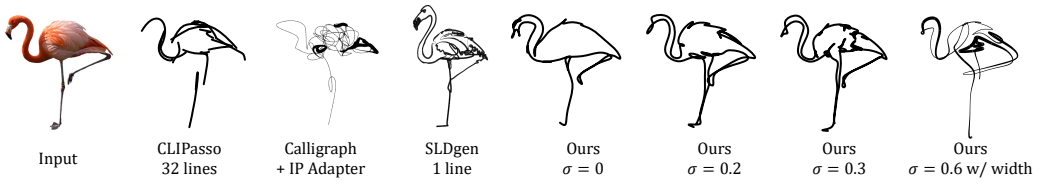

Figure 7: **Structural comparison and detail allocation.** From left: input, CLIPasso, Calligraph, SLDgen, and outputs of our method with different interior allocations and line-width settings.

ble 1). These results show that local insertion substantially reduces unnecessary long-range traversal while preserving strong image correspondence.

Effect of geometric regularization. Among the three geometric terms, the spacing loss has the clearest effect on trajectory economy: removing $\mathcal{L}_s$ increases normalized length from 5.26 to 6.34 and turning from 296.11 to 315.82, while reducing Image CLIP by 3.08 points. Removing the bending or proximity terms produces smaller changes in trajectory geometry and image correspondence. Together, the regularizers encourage a compact and smooth trajectory while retaining the initialized drawing order. Additional metric results for all component variants are provided in Appendix C.2. Figures 7–9 further show that contour-guided initialization provides a coherent global structure that is progressively refined toward salient object details.

4.3 Comparison with Vector Baselines

We compare with Calligraph (Berio et al., 2025b), SLDgen (Magne et al., 2026), and 3D wire art (Tojo et al., 2024) using 50 paired image–output examples. All vector outputs are rendered on a white 512×512 canvas and evaluated against their corresponding input images. Evaluation details for this paired setting are provided in Appendix B.3.

As shown in Table 2, our method obtains the highest Image CLIP, spatial DINO, and edge F1 scores, indicating stronger preservation of semantic content, spatial structure, and reference contours. The aesthetic score is also close to the highest baseline result. Together with Fig. 6, these results demonstrate that the proposed lightweight structural optimization preserves reference-specific geometry effectively without relying on a heavy generative prior. This paired evaluation uses the corresponding paired vector outputs, whereas Appendix D.1 reports the separately saved benchmark collection.

4.4 Line Width and Progressive Replay

Our representation also supports continuous line-width optimization without introducing additional strokes. Different portions of the same centerline can therefore receive different visual weights while preserving a single ordered trajectory. Quantitative comparison of the fixed- and optimized-width configurations is provided in Appendix B.4, and the corresponding export conventions are detailed in Appendix A.2.

Because the centerline is globally ordered, the SVG can be progressively revealed according to its curve parameter. The accompanying HTML demonstration visualizes this sequence, allowing the



Figure 8: One-line drawing results of our method.

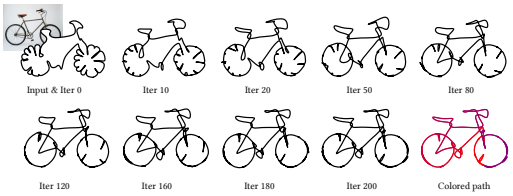

Figure 9: Optimization progression of a one-line bicycle sketch.

Table 2: Comparison on 50 paired inputs. Bold indicates the highest value in each column.

Method	Image CLIP ↑	Spatial DINO ↑	Edge F1 ↑	Aesthetic ↑
Calligraph (Berio et al., 2025b)	58.06	0.235	53.35	4.61
SLDgen (Magne et al., 2026)	75.74	0.414	35.96	5.04
3D wire art (Tojo et al., 2024)	65.08	0.326	39.91	**5.30**
Ours	**78.83**	**0.529**	**55.54**	5.21

generated result to be inspected both as a final vector drawing and as an ordered drawing process. Further export and replay details are provided in Appendix B.5.

4.5 Computational Cost

We compare end-to-end computational cost on the same camel input using an RTX 5090 GPU, with each method retaining its native optimization schedule. As shown in Table 3, our method completes the example in 112.95 s with 6.65 GiB peak additional GPU memory, compared with 348.86 s/14.66 GiB for Calligraph (Berio et al., 2025b) and 2,029.90 s/26.06 GiB for SLDgen (Magne et al., 2026). This demonstrates the computational advantage of our lightweight structure-guided optimization. The complete measurement protocol is provided in Appendix B.6.

Table 3: Single-case computational cost under the native schedule of each method.

Method	Updates	Time (s) ↓	GPU (GiB) ↓
Calligraph (Berio et al., 2025b)	300	348.86	14.66
SLDgen (Magne et al., 2026)	4,001	2,029.90	26.06
Ours	200	**112.95**	**6.65**

4.6 Discussion and Limitations

Our results show that effective one-line drawing does not require heavy generative priors or prolonged optimization. By combining a simple contour-guided initialization with frozen image features and differentiable spline refinement, our method achieves stronger structural correspondence than SLDgen (Magne et al., 2026) and Calligraph (Berio et al., 2025b), while requiring substantially less computation. This suggests that explicitly organizing the trajectory around the reference structure is particularly effective for image-conditioned one-line drawing. The main limitation is the dependence on reliable structural evidence: disconnected parts, very thin structures, and dense internal details may still introduce redundant turns or self-intersections. More adaptive topology handling and part-aware trajectory organization provide promising directions for further improvement.

5 Conclusion

We presented a lightweight differentiable framework for image-conditioned one-line drawing. Contour anchors and attention-selected interior evidence first establish a globally ordered trajectory, which is then refined as a single B-spline through image alignment and geometric regularization. Unlike approaches relying on diffusion guidance or heavier style priors, our method requires no model training and produces results with stronger structural correspondence using substantially less computation. Experiments demonstrate more economical trajectories and competitive semantic fidelity, while the explicit centerline naturally supports variable width, vector editing, and progressive replay. These results highlight the effectiveness of treating trajectory organization itself as a central component of one-line drawing generation.

REPRODUCIBILITY STATEMENT

The method, objective, and experimental configurations are specified in Sections 3 and 4, with implementation and evaluation details provided in Appendices A–C. No task-specific model training is performed. The source code and associated implementation resources will be publicly released upon completion of the review process.

AI USE STATEMENT

In this work, we used generative AI tools for language polishing and for assisting in the development of the project webpage and HTML-based interactive demonstrations.

REFERENCES

Emre Aksan, Thomas Deselaers, Andrea Tagliasacchi, and Otmar Hilliges. CoSE: Compositional stroke embeddings. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

Ellie Arar, Yarden Frenkel, Daniel Cohen-Or, Ariel Shamir, and Yael Vinker. SwiftSketch: A diffusion model for image-to-vector sketch generation. In *ACM SIGGRAPH 2025 Conference Papers*, pp. 1–12, 2025.

Daniel Berio, Guillaume Clivaz, Michael Stroh, Oliver Deussen, Réjean Plamondon, Sylvain Calinon, and Frederic Fol Leymarie. Image-driven robot drawing with rapid lognormal movements. In *2025 34th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*, pp. 2126–2132. IEEE, 2025a. doi: 10.1109/RO-MAN63969.2025.11217913.

Daniel Berio, Michael Stroh, Sylvain Calinon, Frederic Fol Leymarie, Oliver Deussen, and Ariel Shamir. Neural image abstraction using long smoothing B-splines. *ACM Transactions on Graphics*, 44(6):225:1–225:11, 2025b. doi: 10.1145/3763345.

Ankan Kumar Bhunia, Salman Khan, Hisham Cholakkal, Rao Muhammad Anwer, Fahad Shahbaz Khan, Jorma Laaksonen, and Michael Felsberg. Doodleformer: Creative sketch drawing with transformers. In *Computer Vision – ECCV 2022*, pp. 338–355. Springer, 2022. doi: 10.1007/978-3-031-19790-1_21.

Robert Bosch and Adrianne Herman. Continuous line drawings via the traveling salesman problem. *Operations research letters*, 32(4):302–303, 2004.

Alexandre Carlier, Martin Danelljan, Alexandre Alahi, and Radu Timofte. DeepSVG: A hierarchical generative network for vector graphics animation. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

Ayan Das, Yongxin Yang, Timothy M. Hospedales, Tao Xiang, and Yi-Zhe Song. Béziersketch: A generative model for scalable vector sketches. In *Computer Vision – ECCV 2020*, pp. 632–647. Springer, 2020. doi: 10.1007/978-3-030-58574-7_38.

Carl de Boor. *A Practical Guide to Splines*. Springer, 2001. Revised edition.

Yulia Gryaditskaya, Mark Sypesteyn, Jan Willem Hoftijzer, Sylvia C. Pont, Frédo Durand, and Adrien Bousseau. Opensketch: a richly-annotated dataset of product design sketches. *ACM Trans. Graph.*, 38:232:1–232:16, 2019.

David Ha and Douglas Eck. A neural representation of sketch drawings. In *International Conference on Learning Representations*, 2018.

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. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

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*, pp. 1911–1920, 2023.

Craig S. Kaplan and Robert Bosch. Tsp art. In *Renaissance Banff: Mathematics, Music, Art, Culture*, pp. 301–308, 2005.

Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollar, and Ross Girshick. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4015–4026, 2023.

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

Tanguy Magne, Alexandre Binninger, Ruben Wiersma, and Olga Sorkine-Hornung. Single line drawing generation via semantics-driven optimization. *Computer Graphics Forum*, pp. e70502, 2026. doi: 10.1111/cgf.70502.

Maxime Oquab, Timothee Darcet, Theo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024. URL https://arxiv.org/abs/2304.07193.

Xuebin Qin, Zichen Zhang, Chao Huang, Masood Dehghan, Osmar Zaiane, and Martin Jagersand. U^2-net: Going deeper with nested u-structure for salient object detection. *Pattern Recognition*, 106:107404, 2020.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Proceedings of the International Conference on Machine Learning*, volume 139, pp. 8748–8763, 2021.

Kenji Tojo, Ariel Shamir, Bernd Bickel, and Nobuyuki Umetani. Fabricable 3d wire art. In *ACM SIGGRAPH 2024 Conference Papers*, 2024. doi: 10.1145/3641519.3657453.

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.

Wenping Wang, Helmut Pottmann, and Yang Liu. Fitting B-spline curves to point clouds by curvature-based squared distance minimization. *ACM Transactions on Graphics*, 25(2):214–238, 2006. doi: 10.1145/1138450.1138453.

Fernando J Wong and Shigeo Takahashi. A graph-based approach to continuous line illustrations with variable levels of detail. *Computer Graphics Forum*, 30(7):1931–1939, 2011.

Fernando J Wong and Shigeo Takahashi. Abstracting images into continuous-line artistic styles. *The Visual Computer*, 29(6):729–738, 2013a.

Fernando J Wong and Shigeo Takahashi. Hierarchical design of continuous line illustrations. In *International Conference on Computer Graphics Theory and Applications*, volume 2, pp. 131–138. SCITEPRESS, 2013b.

Ronghuan Wu, Wanchao Su, Kede Ma, and Jing Liao. IconShop: Text-guided vector icon synthesis with autoregressive transformers. *ACM Transactions on Graphics*, 42(6), 2023. doi: 10.1145/3618364.

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*, volume 36, pp. 15869–15889, 2023.

A Implementation and Checkpoint Conventions

A.1 Foreground Preprocessing and Initialization

Foreground segmentation is optional in our pipeline. For images with a clean foreground, the input is processed directly. For complex backgrounds, we optionally apply a foreground segmentation model, such as U^2-Net (Qin et al., 2020) or SAM (Kirillov et al., 2023), before contour extraction. We extract the longest discrete foreground contour and sample contour anchors using uniformly spaced contour-array indices rather than exact arc-length resampling.

The saliency encoder is CLIP ViT-B/32, distinct from the RN101 encoder used during optimization. The attention map is aligned with the foreground mask, which is eroded three times using a 5×5 structuring element to obtain the interior region. The saved configuration uses a saliency temperature of 3.0 and sampling without replacement. A numerical 10^{-8} is added during probability normalization, and unmasked attention is used when the masked attention mass falls below 10^{-5}. For short contours, all available contour anchors are retained and unused anchor slots are reassigned to interior samples.

Inserted interior samples receive independent Gaussian jitter with standard deviation 10^{-4} in normalized image coordinates before being scaled to the canvas. Neither the original contour nor the sampled interior landmarks is imposed as a hard constraint during subsequent optimization.

Interior samples assigned to the same contour anchor are ordered by increasing distance to that anchor. The resulting trajectory is open: the first contour anchor is not appended at the end, and an interior excursion does not explicitly return to its anchor before proceeding to the next contour group. Moving an inserted sample by ϵ_i changes the control-polygon length by at most $2\|\epsilon_i\|_2$. Therefore, Proposition 1 remains valid after adding $2\sum_i \|\epsilon_i\|_2$ to its right-hand side. Nearest-anchor assignment requires $O(N_f N_c)$ distance evaluations, followed by sorting within each anchor group.

A.2 Spline Parameterization and Vector Export

Spline locality and stability. For the normalized nonnegative B-spline basis, $\sum_i B_i(t) = 1$. Consequently, bounded width coefficients produce a width field within the same bounds. For perturbations δv_i and δc_i of the width coefficients and control points, respectively,

$$|\delta w(t)| \le \|\delta v\|_\infty, \qquad \|\delta\gamma(t)\|_2 \le \max_i \|\delta c_i\|_2. \tag{12}$$

Only coefficients whose basis support contains t affect the curve at that parameter value. Cubic B-splines with simple interior knots are C^2 over the interior of the valid parameter domain (de Boor, 2001).

Fixed-width conversion. With padded controls p_i, the Bézier controls for each consecutive four-control window are

$$\begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \end{bmatrix} = \frac{1}{6} \begin{bmatrix} 1 & 4 & 1 & 0 \\ 0 & 4 & 2 & 0 \\ 0 & 2 & 4 & 0 \\ 0 & 1 & 4 & 1 \end{bmatrix} \begin{bmatrix} p_i \\ p_{i+1} \\ p_{i+2} \\ p_{i+3} \end{bmatrix}. \tag{13}$$

Repeating both endpoint controls twice creates $N + 1$ windows; therefore, the requested budget K should not be interpreted as the exact number of exported Bézier spans. Adjacent windows share their endpoints. This conversion is a standard linear mapping and is also used in long-spline graphics formulations (Berio et al., 2025b).

Variable-width export. Following the normalized uniform-basis evaluation used in SLD-gen (Magne et al., 2026), the variable-width branch samples the unpadded controls and width coefficients at 1,000 parameter values between the fourth and fourth-from-last uniform knots. Consecutive centerline samples define a polygonal ribbon. Estimated unit normals n_k produce cross sections

$$p_k - w_k n_k, \qquad p_k + w_k n_k, \tag{14}$$

so the exported full thickness is $2w_k$, nominally within $[0.2, 6.0]$ pixels. Degenerate tangents use an epsilon-stabilized normal.

The fixed-width SVG instead uses a scalar `stroke-width` attribute, which is 1.0 in the saved fixed-width collection. Consequently, equal numerical width values in the two rendering branches do not imply identical exported thickness. No artist-derived width supervision or explicit width-smoothness loss is used; width continuity is induced by the shared cubic B-spline basis.

A.3 Optimization and Checkpoint Selection

Augmentation and optimization. Four paired views use random perspective transformation with distortion scale 0.5 and probability 1, followed by a 224×224 random resized crop with area scale 0.8 and aspect ratio 1. The unaugmented view is also included. CLIP normalization is applied to every view. Learning-rate scheduling is disabled. The geometry terms are computed from the control polygon, with unbiased sample variance for the spacing term and ordered nonadjacent segment pairs for the proximity term.

Checkpoint selection. The controlled reruns execute exactly 300 updates with early stopping disabled. Checkpoint selection is performed every ten iterations using the shared CLIP-based evaluation criterion. The evaluation score is computed before an optimizer update, whereas the corresponding SVG is serialized after that update; therefore, the saved SVG represents the subsequent optimizer state. This convention is shared across all controlled variants, and all reported trajectory measurements are computed directly from the saved SVG outputs.

B Evaluation Protocol and Additional Results

B.1 Experimental Configurations

The controlled experiment contains 51 input images and five variants, yielding 255 completed runs. All controlled variants use the same input images, seed 0, a 512×512 canvas, and identical sampled contour and interior evidence. Each ablation modifies only the indicated component. The fixed-/variable-width collections and runtime test use the configurations summarized at right.

Table 4: Experiment-specific configurations.

	Controlled ablations	Width collections	Runtime case
Requested budget K	200	200	100
Controls N	203	203	103
Interior ratio ρ	0.4	0.4	0.6
Updates	300	300	200
Width mode	Fixed	Fixed / learned	Learned

B.2 Trajectory Measurements

For each selected SVG, every cubic Bézier segment is sampled at 48 uniform parameter subdivisions. Trajectory length is computed as the sum of successive chord lengths normalized by the canvas diagonal. Cumulative turn is the sum of absolute turning angles between successive nonzero chords.

The discretization is numerically stable for the evaluated outputs: increasing the sampling density from 24 to 48 subdivisions changes trajectory length by at most 0.027% and cumulative turn by at most 0.086% across the 255 controlled outputs. Confidence intervals for paired Image CLIP differences use 10,000 paired bootstrap resamples over the 51 inputs.

B.3 Paired Vector Evaluation

The paired vector comparison uses all 50 image–output correspondences from the SLDgen evaluation collection. All vector outputs are rasterized on a white 512×512 canvas and compared with their corresponding reference inputs. Image CLIP evaluates global semantic correspondence, while spatial DINO and edge F1 characterize spatial and contour correspondence, respectively. The aesthetic predictor provides an additional learned image-quality measure.

Table 5: Comparison of the saved fixed- and optimized-width configurations.

Configuration	Text CLIP ↑	Image CLIP ↑	DINOv2 ↑	Aesthetic ↑	FID ↓
Fixed width	34.5	79.10	0.784	5.88	179.6
Optimized width	32.39	78.90	0.801	5.10	233.3

The paired comparison contains Calligraph (Berio et al., 2025b), SLDgen (Magne et al., 2026), the image-conditioned 3D wire-art baseline (Tojo et al., 2024), and our method. The numerical results are reported in Table 2 in the main paper.

B.4 Fixed- and Variable-Width Analysis

The optimized-width configuration obtains higher DINOv2 similarity (0.801 versus 0.784), whereas the fixed-width configuration obtains higher Text CLIP, Image CLIP, and aesthetic scores and a lower FID. Both configurations retain the same single-centerline formulation; width optimization modifies the visible line emphasis without introducing additional drawing paths.

B.5 Vector Export and Progressive Replay

Fixed-width outputs are stored as one connected sequence of cubic Bézier segments. Variable-width outputs consist of consecutive filled ribbon polygons sampled from the same ordered centerline and retain progressive-reveal metadata. The two SVG representations therefore correspond to the same single-centerline formulation rather than different numbers of drawing paths.

Progressive replay reveals the trajectory according to its curve parameter. The replay timing is used for visualization rather than as a measurement of physical hand motion. The supplemental HTML provides interactive visualization of the ordered SVG trajectories, their drawing progression, and additional high-resolution results.

B.6 Runtime Measurement Details

Runtime is measured using a serialized fresh-process execution on the same 248×248 RGB camel input, internally rendered at 512^2, on an RTX 5090 GPU with 31.8 GiB memory. The measured interval includes local model loading, preprocessing, initialization, optimization, and final output generation, while first-time model downloads are excluded. Calligraph additionally receives its required style image, and SLDgen uses the caption "a single line drawing of a camel."

The native optimization schedules are retained: 300 updates for Calligraph, 4,001 for SLDgen, and 200 for our runtime configuration. GPU telemetry is sampled every 0.2 s. The estimated GPU energy consumptions are 14.43 Wh, 89.63 Wh, and 2.62 Wh for Calligraph, SLDgen, and our method, respectively.

C Metric Definitions and Complete Ablations

C.1 Image and Distributional Metrics

Text CLIP and Image CLIP use `openai/clip-vit-large-patch14` with the retained TorchMetrics scoring pipeline, with cosine similarity scaled by 100. Text prompts are reconstructed from input filenames using the recorded single-line-drawing template. DINOv2 (Oquab et al., 2024) uses `facebook/dinov2-base`; all last-hidden-state tokens are averaged, and cosine similarity is mapped to $(1 + \cos)/2$. The aesthetic score uses the SLDgen evaluation implementation of Aesthetic Predictor V2.5 with `google/siglip-so400m-patch14-384` in bfloat16.

Saved SVG animations are frozen at their final visible state and rasterized by CairoSVG at 512×512 over white. FID (Heusel et al., 2017) is computed using `pytorch-fid` 0.3.0 with the FID-specific

Table 6: Supplementary five-metric results for the evaluated component configurations (seed 0). "Anchor" denotes nearest-contour-anchor assignment; its absence denotes random assignment using the same sampled evidence.

Variant	Anchor	$\mathcal{L}_b$	$\mathcal{L}_p$	$\mathcal{L}_s$	Text CLIP ↑	Image CLIP ↑	DINOv2 ↑	Aesthetic ↑	FID ↓
w/o $\mathcal{L}_b$	✓	✗	✓	✓	32.3	77.0	0.785	4.88	201.6
w/o $\mathcal{L}_p$	✓	✓	✗	✓	32.5	77.5	0.786	4.91	191.2
w/o $\mathcal{L}_s$	✓	✓	✓	✗	31.7	76.0	0.787	4.86	209.0
Random anchor assignment	✗	✓	✓	✓	27.0	64.9	0.631	4.49	273.4
Full method	✓	✓	✓	✓	34.5	79.1	0.788	5.88	179.6

Table 7: Extended image-based benchmark. The first four scores of the inherited baselines are reported by SLDgen (Magne et al., 2026); Calligraph and Ours are evaluated locally. FID is recomputed against the same reference collection. Vector and one-line indicate the output representation.

Method	Vector	One line	Text CLIP ↑	Image CLIP ↑	DINOv2 ↑	Aesthetic ↑	FID ↓
ChatGPT (gpt-image-1.5)	✗	✗	32.5	83.2	0.859	5.52	216.6
Nano Banana (gemini-3-pro-image-preview)	✗	✗	32.1	83.5	0.871	5.50	250.7
Flux.2 [max] (flux-2-max)	✗	✗	32.5	79.6	0.765	5.82	197.2
CLIPasso (multi-stroke) (Vinker et al., 2022)	✓	✗	32.3	80.0	0.824	4.51	299.1
ControlSketch (multi-stroke) (Arar et al., 2025)	✓	✗	31.7	79.6	0.817	4.52	270.3
ChatGPT connected	✓	✓	33.1	79.3	0.827	5.78	219.2
Nano Banana connected	✓	✓	32.8	80.7	0.830	5.56	239.4
Flux.2 [max] connected	✓	✓	33.1	76.8	0.759	6.13	190.8
CLIPasso connected (Vinker et al., 2022)	✓	✓	32.1	73.7	0.755	4.76	252.7
ControlSketch connected (Arar et al., 2025)	✓	✓	32.2	76.2	0.767	4.84	228.7
Kaplan and Bosch (1000 points) (Kaplan & Bosch, 2005)	✓	✓	24.1	58.1	0.632	4.37	383.2
CLIPasso (single stroke) (Vinker et al., 2022)	✓	✓	29.6	66.6	0.675	3.96	286.2
ControlSketch (single stroke) (Arar et al., 2025)	✓	✓	28.2	64.7	0.651	4.09	302.4
Tojo et al. (image) (Tojo et al., 2024)	✓	✓	28.9	66.7	0.682	5.05	160.2
Tojo et al. (text) (Tojo et al., 2024)	✓	✓	29.7	65.0	0.646	4.92	147.4
Calligraph (Berio et al., 2025b)	✓	✓	23.5	59.2	0.620	4.16	196.6
SLDgen (Magne et al., 2026)	✓	✓	33.1	78.0	0.785	5.82	176.4
Ours	✓	✓	34.5	79.1	0.784	5.88	179.6

Inception-v3 pool-3 features of dimension 2,048:

$$\mathrm{FID} = \|\mu_r - \mu_g\|_2^2 + \mathrm{Tr}\left(\Sigma_r + \Sigma_g - 2(\Sigma_r\Sigma_g)^{1/2}\right). \tag{15}$$

Here (μ_r, Σ_r) and (μ_g, Σ_g) denote the empirical feature statistics of the reference and generated collections, respectively.

The same 56-image artist-reference collection, `OneLineRefData`, is used when recomputing FID. Reference images receive EXIF orientation correction, alpha compositing over white, and RGB conversion before the implementation's internal 299×299 resize and normalization. Cached reference statistics are verified and reused across methods.

C.2 Complete Component Ablations

D Additional Benchmark and Qualitative Comparisons

D.1 External Benchmark Context

Table 7 records the numerical values visualized in Figure 5. The first four metric columns for the inherited baseline configurations are transcribed from SLDgen (Magne et al., 2026). Their FIDs are recomputed from 50 supplied supplemental PNGs per method against the common `OneLineRefData` reference collection. All five Calligraph metrics are obtained from the regenerated 50-output evaluation, while the Ours row uses the saved 50-output fixed-width collection. Representation flags for inherited methods follow the source categorization. ControlSketch is listed separately from the learned SwiftSketch model described in the same work (Arar et al., 2025).

D.2 Additional Qualitative Comparisons

Building on SLDgen (Magne et al., 2026), we provide additional qualitative comparisons in Figures 10 and 11. Across these examples, our method maintains a single continuous trajectory while preserving the major silhouette and spatial arrangement of object parts. Raster-generation baselines often retain richer local detail, whereas the evaluated single-line vector baselines exhibit larger structural deviations in several examples. The supplementary HTML provides high-resolution vector outputs, drawing trajectories, running parameters, and additional examples.

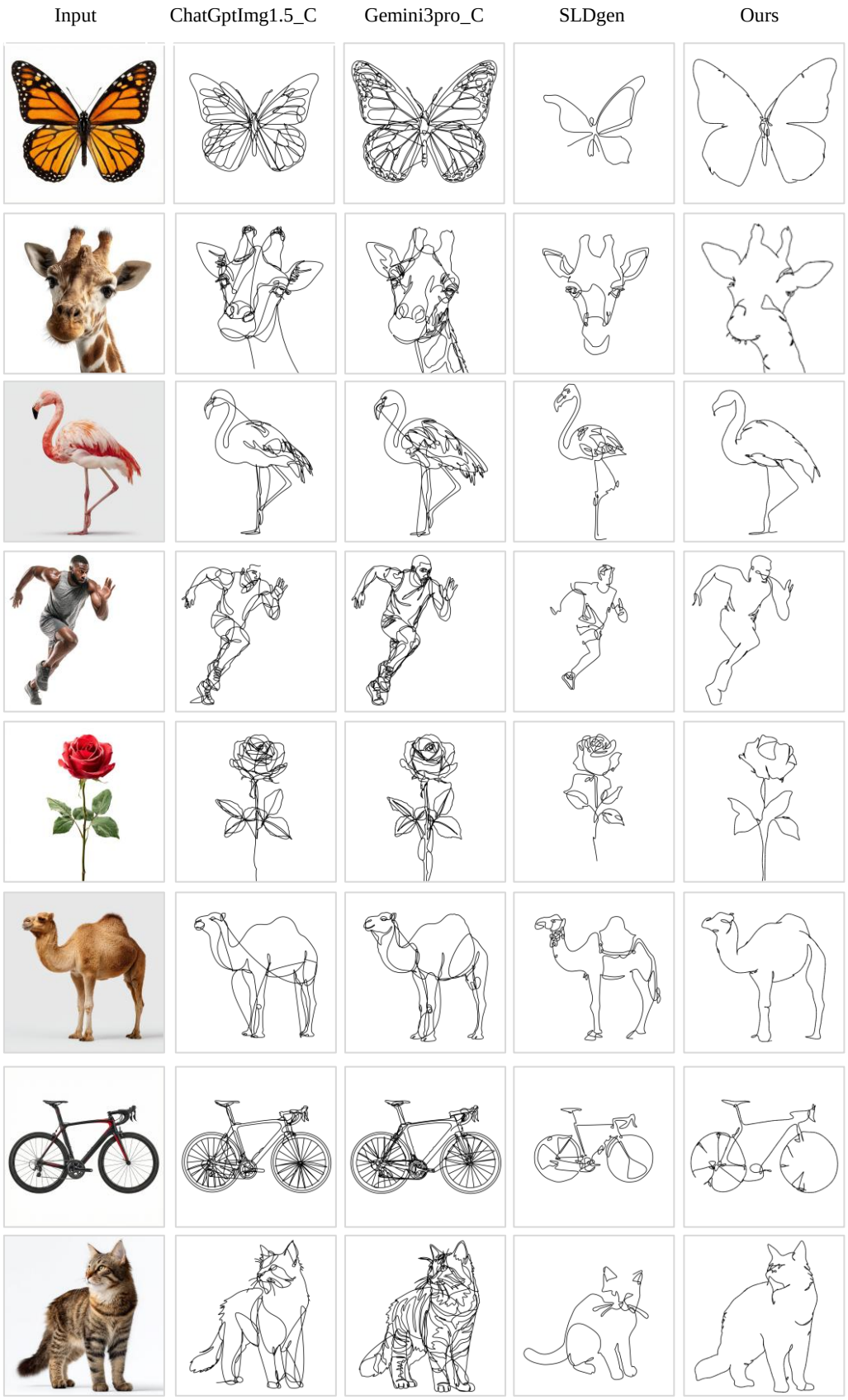

Figure 10: Qualitative comparison with ChatGPT-Image-1.5-Connected, Gemini3Pro-Connected, SLDgen, and ours. Our method maintains a continuous trajectory while preserving the major structure of the reference. Additional high-resolution vector trajectories are provided in the supplementary HTML.

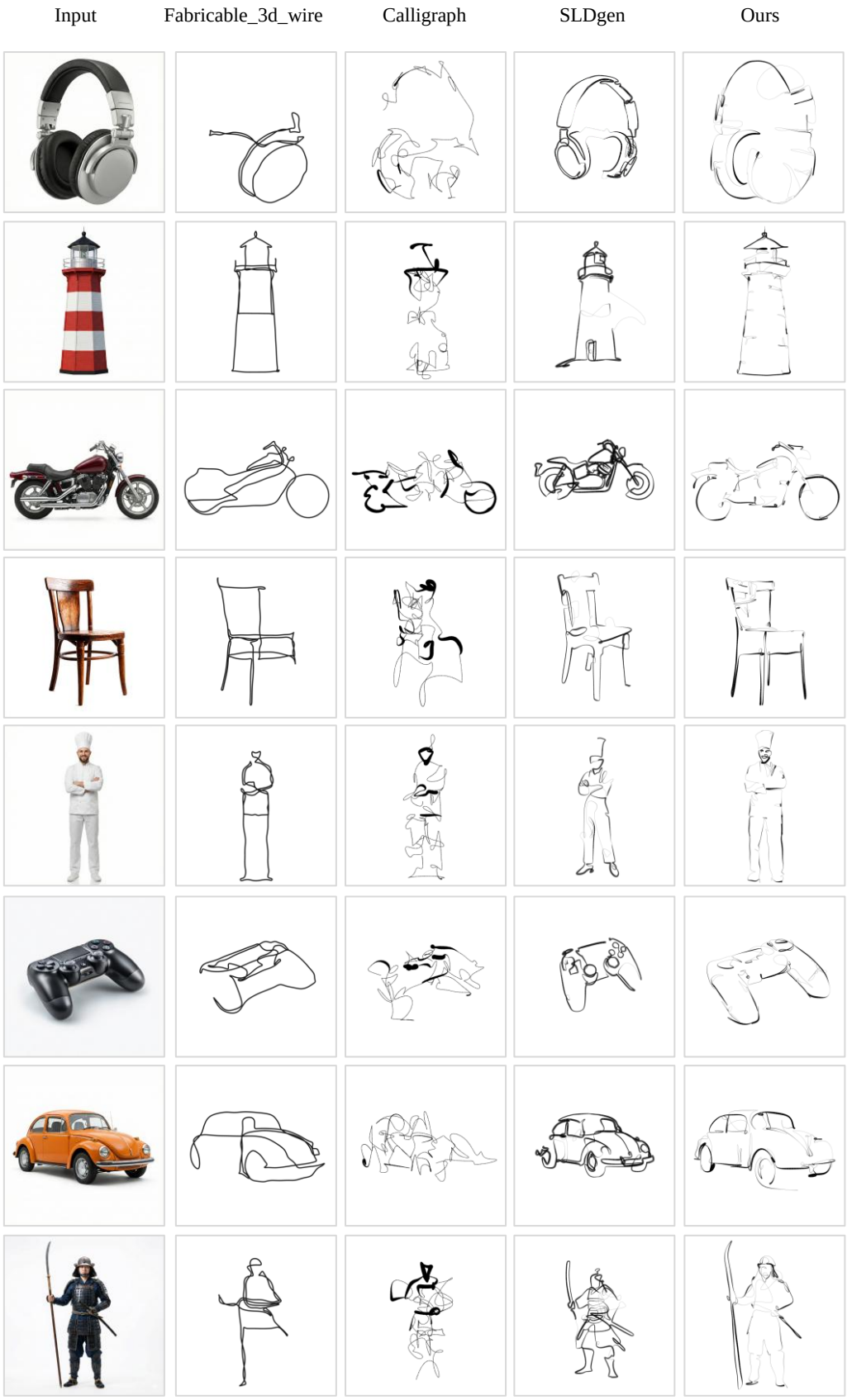

Figure 11: Qualitative comparison with Fabricable 3D Wire Art, Calligraph, SLDgen, and ours. Our method maintains a coherent single trajectory while preserving the major object structure. Additional high-resolution vector trajectories are provided in the supplementary HTML.