HARNESSPAINTER: GROUNDING VISUAL REASONING FOR IN-CONTEXT STROKE ORGANIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Stroke-based rendering (SBR) represents images through interpretable brush primitives, enabling editable and executable painting programs beyond pixel grids. However, existing SBR methods primarily optimize for final image fidelity, while the organization of individual strokes remains largely unexplored. As a result, generated paintings may achieve high visual similarity but still contain inconsistent stroke ordering, fragmented contours, and redundant primitives. In this work, we introduce **HarnessPainter**, a closed-loop framework that formulates stroke organization as an executable program refinement problem. Our key insight is to decouple high-level visual reasoning from continuous geometric manipulation: a vision-language model analyzes structural imperfections and proposes group-level editing strategies, while differentiable SBR operators perform precise geometric updates. An execution harness further verifies each modification through rendering-based constraints, ensuring that semantic decisions are translated into reliable stroke transformations. HarnessPainter progressively reorganizes stroke programs through three stages: semantic ordering that establishes coarse-to-fine painting structure, flow-guided alignment that organizes stroke trajectories according to visual contours, and contour consolidation that merges redundant fragments into coherent curves. Extensive experiments demonstrate that our framework produces more structured and compact stroke representations while preserving competitive image fidelity, achieving improved stroke coherence and process-level quality over existing SBR approaches.

1 INTRODUCTION

Stroke-based rendering (SBR) represents visual content through explicit graphical primitives, providing an interpretable interface between image appearance and executable painting programs. Unlike raster images that only describe the final appearance, stroke programs additionally encode how visual structures are decomposed and executed. A meaningful painting program should therefore not only reproduce the image faithfully, but also preserve the organization behind mark placement: broad regions should emerge before fine details, strokes should follow underlying visual structures, and continuous contours should be represented by coherent trajectories rather than fragmented pieces.

Recent advances in differentiable rendering and learning-based stroke prediction have significantly improved stroke generation and optimization efficiency. Differentiable rasterization enables continuous refinement of stroke parameters through image losses, while neural painting methods provide efficient stroke prediction and semantic guidance. However, these approaches primarily focus on matching the final rendered image. A critical gap remains: a visually accurate painting does not necessarily correspond to a well-organized stroke program. Direct optimization against pixel-level objectives can produce disordered execution, where detailed strokes appear before structural foundations, neighboring strokes follow inconsistent directions, and continuous boundaries are approximated by excessive short primitives.

This limitation originates from the mismatch between semantic organization and geometric optimization. Organizing a painting requires discrete decisions, such as determining which regions should appear first, which strokes belong to the same visual structure, and which fragmented primitives should be consolidated. In contrast, geometric refinement requires continuous optimization

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.

over parameters such as curve control points, widths, and positions. Pure numerical optimization can accurately reduce local image errors but lacks the semantic understanding required to determine whether multiple strokes represent a shared structure. Conversely, high-level reasoning models can describe structural intentions but are not suitable for directly manipulating continuous geometric parameters.

To bridge this gap, we introduce **HarnessPainter**, a closed-loop framework that formulates stroke organization as executable program refinement. Instead of allowing a reasoning model to directly modify stroke geometry, we establish a separation between semantic planning and numerical execution. A vision-language model observes the current painting state and proposes high-level editing decisions, while specialized differentiable SBR tools perform the corresponding geometric operations. Each modification is executed within an explicit verification harness, where candidate edits are rendered and accepted only when they satisfy structural and fidelity constraints. This design grounds abstract visual judgments into bounded and executable stroke transformations.

HarnessPainter reorganizes stroke programs through three progressive stages. First, **semantic stroke ordering** establishes a coarse-to-fine execution hierarchy, allowing foundational structures to emerge before surface details. Second, **structure-guided stroke alignment** improves local consistency by aligning stroke trajectories with coherent visual flow fields and structural contours. Third, **contour-preserving stroke consolidation** reduces fragmented primitives by replacing compatible stroke groups with fewer continuous curves while preserving rendering fidelity.

The resulting framework provides an explicit interface between visual reasoning and parametric graphics manipulation. Rather than replacing differentiable optimization with a learned generator, HarnessPainter augments existing SBR pipelines with semantic inspection, controlled editing, and verifiable execution. Experiments demonstrate that the proposed framework improves stroke organization, reduces redundant primitives, and produces more coherent painting processes while maintaining competitive image fidelity.

Our contributions are summarized as follows:

- We introduce a closed-loop stroke program refinement framework that connects vision-language reasoning with differentiable SBR optimization through executable editing operations and verification.
- We propose a progressive organization pipeline that resolves stroke disorder from semantic ordering to structural alignment and contour consolidation.
- We develop an auditable execution mechanism that verifies stroke modifications through rendering feedback and structural constraints, enabling reliable refinement of complex painting programs.

2 Related Work

Optimization-based rendering and abstraction. Early painterly rendering uses multiscale, image-aligned curves (Hertzmann, 1998). Differentiable rasterization enables geometry fitting (Li et al., 2020); Stylized Neural Painting uses a learned renderer and optimal transport (Zou et al., 2021), while Im2Oil controls fineness through adaptive sampling (Tong et al., 2022). Recent dual-parameter stroke planning targets structural coherence (Liu et al., 2026). Semantic priors also guide explicit curves: CLIPasso and CLIPascene control recognizable abstraction (Vinker et al., 2022; 2023), and VectorFusion and DiffSketcher transfer diffusion priors to vector representations (Jain et al., 2023; Xing et al., 2023). Thus, semantic curve optimization and compactness are established capabilities. Our focus is their integration with stage-constrained group editing and contextual replay of an ordered painting program.

Learned painters and process generation. SPIRAL synthesizes graphics programs through adversarial reinforcement learning (Ganin et al., 2018), and Learning to Paint uses model-based reinforcement learning (Huang et al., 2019). Paint Transformer predicts stroke sets (Liu et al., 2021), Compositional Neural Painter selects painting regions (Hu et al., 2023), and MambaPainter predicts strokes in a single step (Sawada & Katsurai, 2024). Intelli-Paint already introduces progressive layering and semantic attention (Singh et al., 2022), while multi-granularity semantic alignment organizes artist-like painting hierarchically (Hu et al., 2024b). Semantic progression alone therefore

does not distinguish our design. We study subsequent group-level alignment and consolidation under a shared verification interface. In the image domain, Painting Many Pasts models alternative painting histories (Zhao et al., 2020); Inverse Painting, AnimatePainter, and Loomis Painter synthesize intermediate states (Chen et al., 2024; Hu et al., 2025; Pobitzer et al., 2025). We use generated states as targets for an explicit program, requiring separate evaluation of perceptual progression and executable stroke properties.

Tool-augmented visual reasoning. Visual Programming composes vision tools through executable programs (Gupta & Kembhavi, 2023), Visual Sketchpad supplies visual reasoning artifacts (Hu et al., 2024a), and ReAct interleaves reasoning and actions (Yao et al., 2023). Tool composition is therefore an established pattern. HarnessPainter specializes it to order-sensitive stroke edits through stable IDs, protected context, replacement lineage, and rendered acceptance gates. Comparisons with fixed-rule control and one-pass planning test whether this specialization adds value beyond the numerical tools.

3 METHOD

Given a reference image, HarnessPainter organizes its reconstruction through three stages: Semantic Stroke Ordering, Structure-Guided Stroke Alignment, and Contour-Preserving Stroke Consolidation (Figure 1).

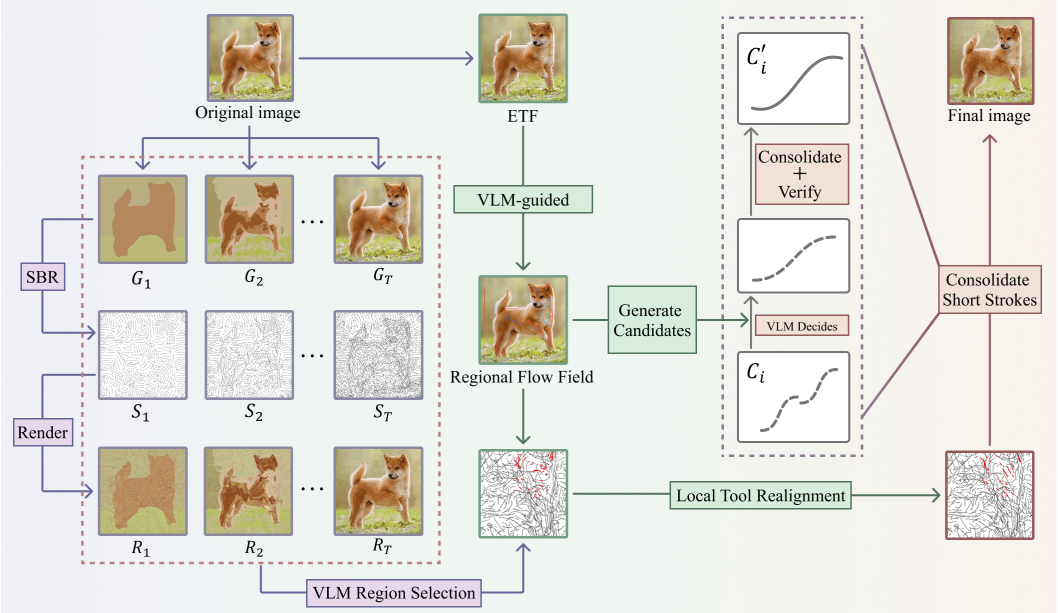


Figure 1: **HarnessPainter overview.** From the reference, VLM guidance and staged fitting produce the stroke program, followed by flow-guided alignment and consolidation.

3.1 OVERVIEW

Given a reference image $I^\star$, the framework produces an accepted ordered stroke sequence $\mathcal{P} = (s_1, \ldots, s_N)$. The three stages determine when strokes are added, how neighboring strokes follow the local shape, and which compatible fragments are represented by fewer strokes. The same sequence supports both layered rendering and the geometric edits described below.

Preliminaries: Stroke representation and rendering. We represent each stroke s_i with three control points $\mathbf{p}_{i0}, \mathbf{p}_{i1}, \mathbf{p}_{i2} \in \mathbb{R}^2$, a constant half-width w_i, an RGB color $\mathbf{c}_i \in [0,1]^3$, an opacity $o_i \in [0,1]$, and a rendering depth z_i. The first and last control points specify the endpoints, while

the middle point controls the curve. Its quadratic Bézier centerline is

$$\mathbf{b}_i(u) = (1-u)^2\mathbf{p}_{i0} + 2u(1-u)\mathbf{p}_{i1} + u^2\mathbf{p}_{i2}, \qquad u \in [0,1]. \tag{1}$$

The ordered sequence records drawing order, and stroke IDs link each stroke to its region and candidate groups. Rendering uses a sigmoid coverage profile and depth-based compositing; the full definition is given in Appendix A.1.1.

Shared plan and processing flow. A shared flow plan is prepared before stage fitting and remains fixed throughout subsequent processing. Its regional guidance defines the active drawing areas for the six fitting stages. These stages produce the fitted stroke sequence $\mathcal{P}_{\text{fit}}$ while retaining the parameters of previously accepted strokes. Alignment then processes this sequence across selected regions, and consolidation receives the accepted alignment result. Alignment preserves stroke count, drawing order, depth, IDs, and appearance; consolidation replaces only groups that pass its geometric and image-quality checks.

VLM guidance and local execution. The vision–language model (VLM) provides regional guidance, protected polygons, flow guide curves, and candidate selections. Local tools convert these choices into stage targets, masks, flow fields, curve updates, replacement strokes, and rendered checks. This division connects image-level decisions to continuous geometry while keeping the acceptance rules explicit. Appendix A.1.5 summarizes the observations and decisions used at each step.

3.2 Semantic Stroke Ordering

Semantic stroke ordering builds the representation from broad color regions to finer variations, so the accepted drawing develops in a coarse-to-fine order.

Stage targets. Local tools first apply panoptic segmentation to the reference image and then quantize colors within each region using a palette. Early targets represent the image with broad color blocks. Subsequent targets progressively recover the reference colors and finer variations. The sixth target is $I^\star$, evaluated within that stage's active drawing region. The VLM supplies regional planning, while segmentation and palette quantization construct the target images used for fitting.

Progressive fitting. Each stage adds a new batch of strokes and fits it toward the corresponding target. The accepted strokes from earlier stages retain their parameters, providing a stable base for the new batch. Stage regions may overlap, so later strokes can add detail over earlier color blocks through the depth-based compositing described in Section 3.1. The six stages reach a total of 4000 strokes; Appendix A.1.2 lists their individual budgets and internal identifiers.

Drawing regions. A local compiler converts the hierarchical region plan into active masks that guide where each stage adds strokes. Each stroke carries a region assignment, and its creation stage records when it enters the ordered sequence. The VLM also supplies protected polygons, which restrict the strokes eligible for later alignment and consolidation. After the six fitting stages, the fitted stroke sequence provides the drawing order and image representation for the two geometric editing stages.

3.3 Structure-Guided Stroke Alignment

Building on the fitted stroke sequence, alignment organizes neighboring strokes along the local shape. It adjusts their orientation, curvature, and position while preserving stroke count, drawing order, and appearance. The operation uses the shared flow plan prepared before fitting, so every task follows the same directional guidance.

Shared flow guidance. A flow direction describes the orientation along which a stroke should extend at a given location. Edge tangent flow (ETF) computed from $I^\star$ provides image tangents and their confidence (Kang et al., 2007; 2009). The VLM observes the image and ETF and specifies regional guide curves at three scales: structure, form, and texture. Local tools combine nearby guide tangents with weights reflecting distance and directional agreement, producing a field $\mathcal{F}$. Locations

with support at least 0.08 provide reliable guidance for alignment and consolidation. Tangents represent unoriented lines: directions separated by π have the same orientation. The tools express this equivalence through double angles when combining directions, as detailed in Appendix A.1.3. For unit tangents $\mathbf{u}$ and $\mathbf{v}$, their angular difference is

$$\theta(\mathbf{u}, \mathbf{v}) = \arccos\left(|\mathbf{u}^\top \mathbf{v}|\right), \qquad \theta \in [0, \pi/2]. \tag{2}$$

This angle treats both signs of a tangent consistently.

Regional selection. The VLM observes the reference image, current rendering, flow visualization, and regional statistics. It selects regions and provides an alignment strength, a neighborhood radius around an anchor, a priority, and preferred stroke IDs. Local tools use these choices to assemble tasks, assigning each selected stroke to one task. Neighboring strokes provide context for evaluating local consistency, while the task's selected strokes receive the geometric update.

Geometric adjustment. For each task, local tools compute a support-weighted common direction. Regions with substantial changes in flow are divided into locally consistent segments, allowing each group to follow the underlying shape. Each stroke is then rotated about the midpoint of its endpoint chord to match its segment's direction. In nearly straight segments, the middle control point moves toward the chord; in curved segments, it moves toward a streamline obtained by integrating the flow. Finally, a normal translation brings stroke midpoints toward the segment's median streamline. Bounds on rotation, control-point displacement, translation, and arc length keep these changes local; Appendix A.1.3 gives the values.

Task acceptance. A task is retained when neighboring directions improve or already satisfy the angular tolerance, agreement with the flow is preserved or improved, the straight-segment condition is satisfied, and the control points undergo a meaningful update. The straight-segment condition measures the fraction of middle control points close to their chords. These geometric checks are followed by the common image-quality checks below. Accepted tasks update the stroke sequence; other tasks retain their preceding geometry.

3.4 Contour-Preserving Stroke Consolidation

Consolidation builds on the aligned stroke sequence by combining compatible fragments along a shared flow. Local tools identify candidate groups and construct replacement curves, while the VLM selects among the remaining groups after straightforward pairs have been processed. Both paths use the same spatial and image-quality checks.

Candidate construction. The tools register two types of groups. *Longitudinal chains* connect strokes end to end along the flow, using nearby predecessors and successors. *Tubular groups* collect strokes lying side by side or separated by larger gaps within a narrow band. Flow agreement and spatial distances determine which strokes can be grouped; tubular groups also use color compatibility and crossing checks. A high-confidence subset satisfies joint conditions on path support, flow agreement, relative gap, and color similarity. Appendix A.1.4 specifies the thresholds and their applicability.

Local pair consolidation. High-confidence longitudinal groups containing exactly two strokes are processed locally in decreasing order of path alignment. The tools select groups with disjoint source strokes and attempt to replace each pair with one stroke. Successful replacements remove their source strokes from the remaining candidate groups. This first pass handles pairs with strong geometric support before the model selects additional groups.

Model-selected consolidation. The VLM receives representative groups from the remaining registered candidates. It can choose merge or keep for an individual group and specify a target stroke count and flow strength. It can also provide a region-level rule for eligible groups. Individual decisions take precedence, so an explicit keep decision preserves that group's source strokes. Region-level rules apply to high-confidence candidates that have no individual decision. Appendix A.1.4 describes representative selection and the available regional rules.

Replacement curves. For an authorized group, the tools construct fewer curves using the fixed flow field. A coverage corridor defines the local area in which a replacement can connect the source fragments while following their existing shape. The replacement must bridge the relevant gap and remain within this permitted area. A short appearance fit adjusts color and width near the source values for at most six steps. Candidates satisfying the corridor and image-quality checks replace their source groups, and the remaining groups keep their original strokes.

Common image-quality checks. Alignment and consolidation share two PSNR conditions, both measured against $I^\star$. Let $\mathcal{P}$ be the accepted stroke sequence immediately before an edit and $\mathcal{P}'$ its candidate. The fixed fitting result $\mathcal{P}_{\text{fit}}$ provides the global reference, while $\mathcal{P}$ provides the local reference. Writing Q_{global} and Q_{local} for full-image and edited-region PSNR, respectively, we require

$$Q_{\text{global}}(\mathcal{P}') \geq Q_{\text{global}}(\mathcal{P}_{\text{fit}}) - 1.0\,\text{dB}, \tag{3}$$

$$Q_{\text{local}}(\mathcal{P}') \geq Q_{\text{local}}(\mathcal{P}) - 2.5\,\text{dB}. \tag{4}$$

The local comparison uses the same region for the parent and candidate. The fixed global reference bounds cumulative degradation, and the local reference limits the impact of each edit. A task or merge group is committed after its geometric conditions and both PSNR conditions pass. Otherwise, its preceding state is retained, together with all earlier accepted edits. Geometry determines the proposed organization, and these image-quality checks determine which changes enter the final accepted stroke sequence.

4 Experiments

4.1 Setup

Aim. We evaluate whether the exported stroke program is ordered in time and coherent in space. Broad structure should precede detail, strokes belonging to one form should share a direction, and continuous contours should be represented by coherent curves. SBR quality has conventionally been measured by reconstruction metrics such as PSNR, which quantify agreement with the reference image. While pixel-level fidelity is certainly one objective of SBR, stroke organization, artistic character, and drawing order are also important objectives. The comparisons below examine HarnessPainter on these dimensions, and the corresponding renders verify that an organized program still depicts its reference.

Schedule. Semantic stroke ordering proceeds through six stages. The programs stored after these stages contain 480, 880, 1440, 2720, 3680, and 4000 strokes, respectively. Strokes accepted in an earlier stage retain their parameters, and the final stage is fitted to the reference. Alignment and consolidation are then applied once to this 4000-stroke program.

Acceptance. An edit is retained when the depicted object, its occlusion, and its principal edges remain visible. Full-image PSNR may decrease by at most $1.0\,\text{dB}$ relative to the staged program, and PSNR within the affected region by at most $2.5\,\text{dB}$. Proposals exceeding either bound are rejected. These bounds exclude broken edits; they are not the quantity under comparison. Stroke organization is inspected on the centerlines drawn at a uniform width.

4.2 Comparison with unguided SBR

Figure 2 compares the program after ordering, alignment, and consolidation with unguided SBR fit independently to the same references.

Our rendering recovers the object, its overall contours, and its edge details. The centerlines further distinguish scale: long strokes establish the background and the major forms, while shorter strokes supply the detail supported by those forms. Unguided SBR, optimized primarily for reconstruction, accounts for the residual with many short fragments. The resulting decomposition attains a high PSNR from a modest budget, yet the strokes collapse into short, thick marks with soft boundaries, and contours are accumulated rather than drawn. At matched PSNR, our rendering preserves clearer forms and sharper edges than unguided SBR, suggesting that pixel-level alignment is only

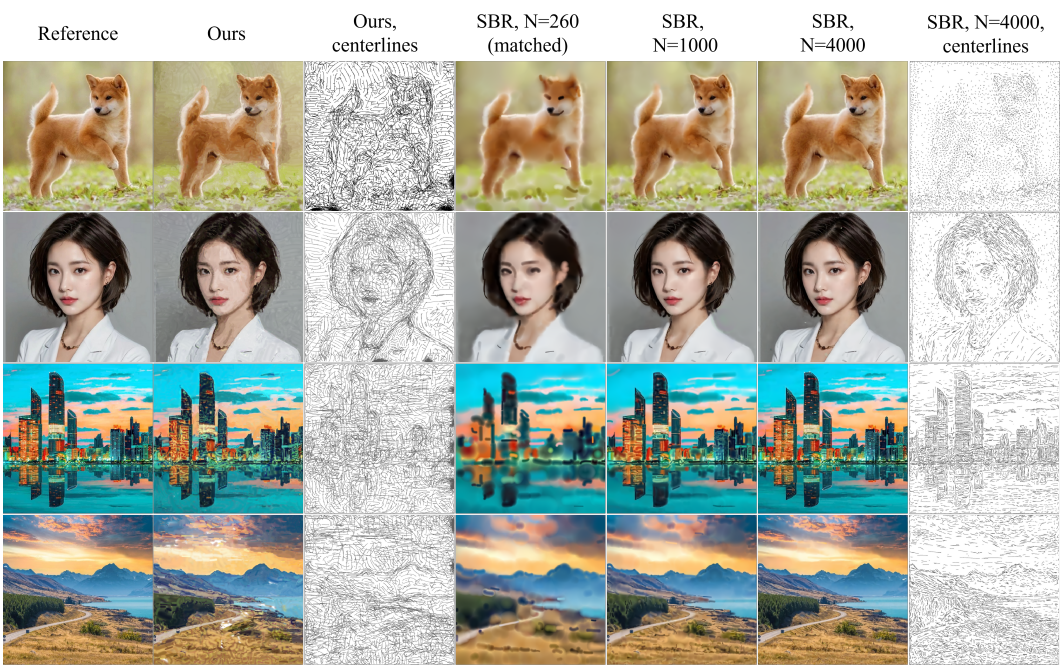


Figure 2: **Comparison with unguided SBR.** The renders remain close to the reference. Our centerlines preserve extended strokes over the background and major forms, whereas the unguided centerlines tend to degenerate into stipples.

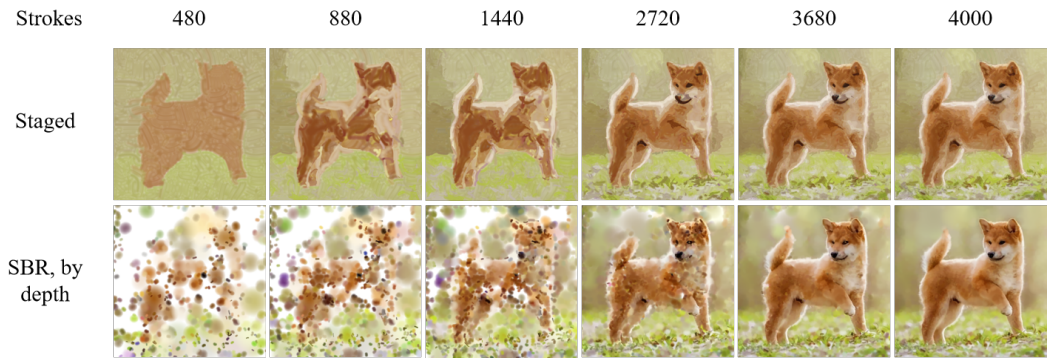


Figure 3: **Semantic stroke ordering.** Staged fitting develops the subject from a blocked-in color mass into a finished form. At equal stroke counts, an unguided depth replay begins as scattered stipples and resolves the body only at a late stage.

one objective of SBR rather than the entire task. Section 4.5 provides a detailed discussion of the advantages of our method in terms of structure and stroke character.

4.3 Semantic stroke ordering

Semantic ordering establishes drawing order and coarse-to-fine structure within a single program. In Figure 3, the staged sequence follows a painterly progression: broad color masses first, then the dog's body, facial features and shadows, and finally surface detail. At matched stroke counts, the unguided prefixes leave a larger portion of the canvas empty. Background regions that should remain broad color fields break into stippled strokes, and the sequence fails to develop a layered structure from coarse forms to fine marks.

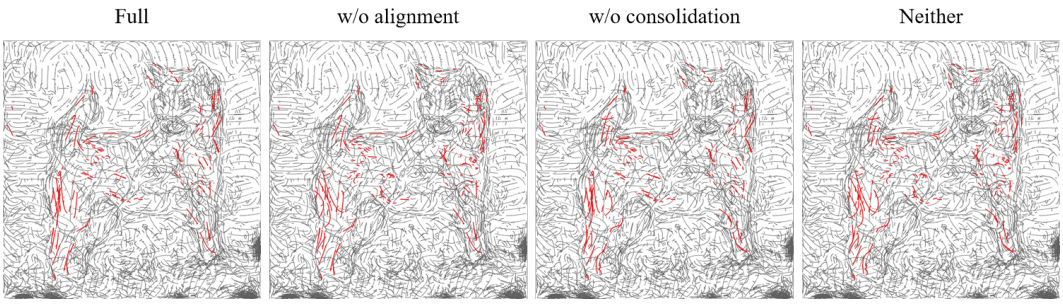


Figure 4: **Ablation of alignment and consolidation.** Red marks strokes modified by the enabled edit. Applying alignment and consolidation together yields a regular arrangement.

4.4 Ablations

Figure 4 isolates the effects of alignment and consolidation on one staged program. Without alignment, many strokes are irregularly oriented or curve beyond their form. Without consolidation, short fragments remain, and the stroke budget is consumed by marks that could have been represented by single strokes. Applied together, the two edits produce a more regular arrangement and expend the limited budget more effectively. The staged program attains $28.414\,\mathrm{dB}$ against the reference. After alignment and high-confidence consolidation, PSNR is $27.751\,\mathrm{dB}$, a decrease of $0.663\,\mathrm{dB}$. The loss is small relative to the gain in arrangement, indicating that artistic organization and reconstruction fidelity can coexist in the same program.

4.5 Artistic character

Coarse-to-Fine Stroke Hierarchy. Figure 5 compares the two programs on the dog image at equal stroke counts. Our early centerlines are long strokes that carry the background and the body; later stages add shorter strokes for the face and coat while preserving the reconstructive contribution of small marks. At the same counts, unguided SBR remains stippled, and its early prefixes barely cover the canvas. The same sequence can guide manual painting: coarse masses are presented first, and detail strokes are introduced only after the underlying form is in place.

Budget and Pictorial Mode. Figure 6 varies stroke count to control how realistically this organization is rendered. At 4000 strokes, our representation is less realistic than unguided SBR at the same budget. Stippled strokes cover residual error efficiently, and our coarse strokes are occluded by the finer strokes subsequently painted over them, so part of the early budget does not appear on the final canvas. Increasing the budget to 12000 strokes restores high reconstruction quality while the marks remain legible, producing a realistic rendering of the same program. At 4000 strokes, the organization remains explicit and the stroke character is stronger, producing an abstract rendering. The budget therefore selects the pictorial mode without discarding the hierarchy.

5 Limitations

HarnessPainter depends on the visual priors of its planner and generator: an incorrect region assignment, stage dependency, or directional prior can produce an orderly sequence that misrepresents the scene. Rendered fidelity gates constrain image changes but cannot certify semantic correctness or artistic intent. The stroke vocabulary also limits expressiveness: constant-width, solid-color quadratic curves cannot capture arbitrary inflections, textured marks, or pigment interactions, while conservative contour chaining excludes closed loops and ambiguous junctions. Model calls, repeated fitting, and rejected proposals incur costs whose practical trade-offs remain to be measured. Finally, digital rollback does not undo physical paint deposition; robotic execution would require observation-based replanning and feasible corrections, as in FRIDA (Schaldenbrand et al., 2023). Future work should examine uncertainty-aware planning, richer stroke primitives, and broader artist evaluation. Generated process visualizations should be distinguished from recordings of an artist's actual work.

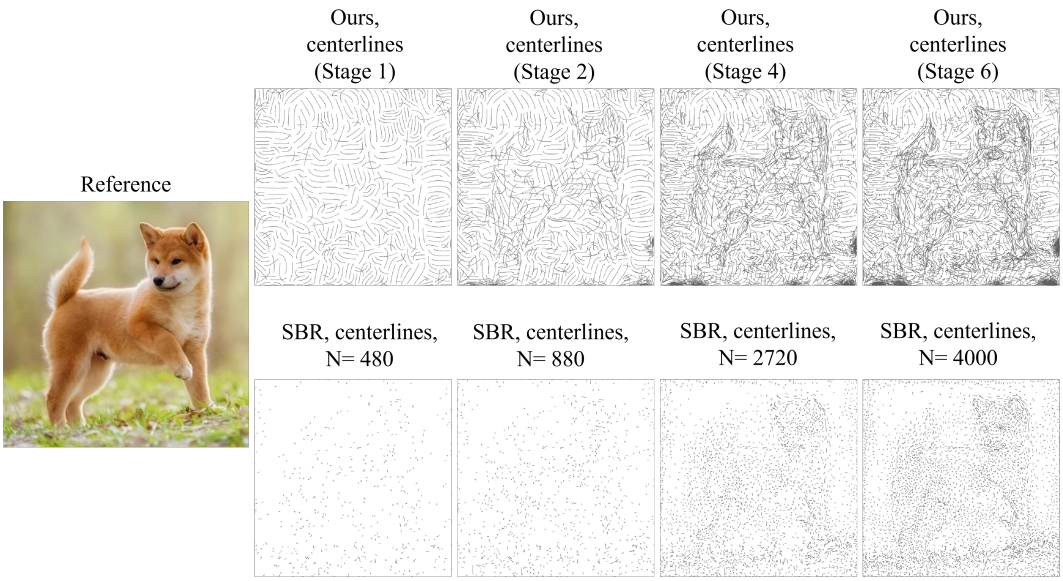


Figure 5: **Coarse-to-fine stroke hierarchy.** At equal stroke counts, our centerlines move from long strokes over the ground and body to shorter strokes on the face and coat. The unguided prefixes remain stipples.

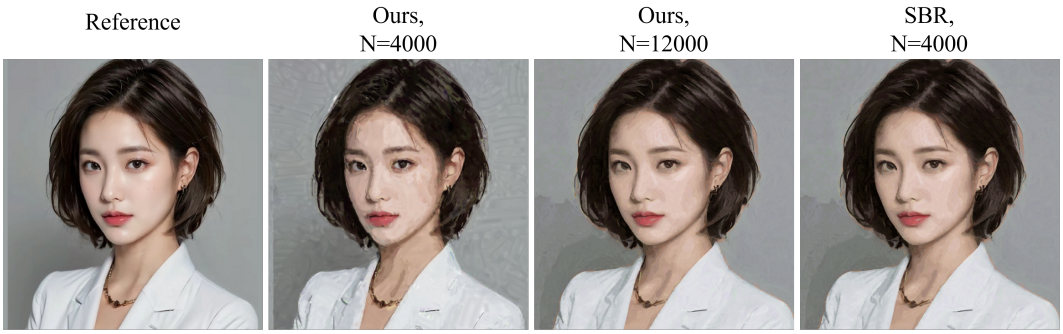


Figure 6: **Budget, fidelity, and stroke character.** The lower budget keeps the organized program visibly painterly. A larger budget improves reconstruction while the strokes remain legible. Unguided SBR at the lower budget is smoother and closer to the reference.

6 Conclusion

We have presented HarnessPainter, a framework for refining the semantic order, spatial arrangement, and contour continuity of painting strokes. Its central design connects group-level VLM decisions to numerical fitting and rendered acceptance, with stage constraints and stroke lineage preserving the context of each edit. The formulation separates an intended change from its executable consequence. Experiments show that the proposed framework improves stroke organization, reduces fragmented primitives, and produces more coherent painting processes while maintaining competitive image fidelity. The staged fitting develops subjects from broad color masses to fine details, flow-guided alignment harmonizes stroke directions with regional structure, and contour consolidation replaces fragmented marks with coherent curves. These results support the effectiveness of the proposed design.

REFERENCES

Bowei Chen, Yifan Wang, Brian Curless, Ira Kemelmacher-Shlizerman, and Steven M. Seitz. Inverse painting: Reconstructing the painting process. In *SIGGRAPH Asia 2024 Conference Papers*, pp. 1–11, 2024. doi: 10.1145/3680528.3687574. URL https://inversepainting.github.io/.

Yaroslav Ganin, Tejas Kulkarni, Igor Babuschkin, S. M. Ali Eslami, and Oriol Vinyals. Synthesizing programs for images using reinforced adversarial learning. In Jennifer Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 1666–1675. PMLR, 10–15 Jul 2018. URL https://proceedings.mlr.press/v80/ganin18a.html.

Tanmay Gupta and Aniruddha Kembhavi. Visual programming: Compositional visual reasoning without training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14953–14962, 2023. URL https://openaccess.thecvf.com/content/CVPR2023/html/Gupta_Visual_Programming_Compositional_Visual_Reasoning_Without_Training_CVPR_2023_paper.html.

Aaron Hertzmann. Painterly rendering with curved brush strokes of multiple sizes. In *Proceedings of SIGGRAPH*, pp. 453–460, 1998. doi: 10.1145/280814.280951. URL https://mrl.cs.nyu.edu/publications/painterly98/.

Junjie Hu, Shuyong Gao, Qianyu Guo, Yan Wang, Qishan Wang, Yuang Feng, and Wenqiang Zhang. AnimatePainter: A self-supervised rendering framework for reconstructing painting process. *arXiv preprint arXiv:2503.17029*, 2025. URL https://arxiv.org/abs/2503.17029.

Teng Hu, Ran Yi, Haokun Zhu, Liang Liu, Jinlong Peng, Yabiao Wang, Chengjie Wang, and Lizhuang Ma. Stroke-based neural painting and stylization with dynamically predicted painting region. In *Proceedings of the 31st ACM International Conference on Multimedia*, MM '23, pp. 7470–7480. ACM, October 2023. URL https://doi.org/10.1145/3581783.3611766.

Yushi Hu, Weijia Shi, Xingyu Fu, Dan Roth, Mari Ostendorf, Luke Zettlemoyer, Noah A. Smith, and Ranjay Krishna. Visual sketchpad: Sketching as a visual chain of thought for multimodal language models. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 139348–139379. Curran Associates, Inc., 2024a. doi: 10.52202/079017-4423. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/fb82011040977c7712409fbdb5456647-Abstract-Conference.html.

Zhangli Hu, Ye Chen, Zhongyin Zhao, Jinfan Liu, Bilian Ke, and Bingbing Ni. Towards artist-like painting agents with multi-granularity semantic alignment. In *Proceedings of the 32nd ACM International Conference on Multimedia*, MM '24, pp. 10191–10199, New York, NY, USA, 2024b. Association for Computing Machinery. ISBN 9798400706868. URL https://doi.org/10.1145/3664647.3681245.

Zhewei Huang, Wen Heng, and Shuchang Zhou. Learning to paint with model-based deep reinforcement learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 8709–8718, 2019. URL https://openaccess.thecvf.com/content_ICCV_2019/html/Huang_Learning_to_Paint_With_Model-Based_Deep_Reinforcement_Learning_ICCV_2019_paper.html.

Ajay Jain, Amber Xie, and Pieter Abbeel. VectorFusion: Text-to-SVG by abstracting pixel-based diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1911–1920, June 2023. URL https://openaccess.thecvf.com/content/CVPR2023/html/Jain_VectorFusion_Text-to-SVG_by_Abstracting_Pixel-Based_Diffusion_Models_CVPR_2023_paper.html.

Henry Kang, Seungyong Lee, and Charles K. Chui. Coherent line drawing. In *Proceedings of the 5th International Symposium on Non-Photorealistic Animation and Rendering*, NPAR '07, pp. 43–50, New York, NY, USA, 2007. Association for Computing Machinery. ISBN 9781595936240. URL https://doi.org/10.1145/1274871.1274878.

Henry Kang, Seungyong Lee, and Charles K. Chui. Flow-based image abstraction. *IEEE Transactions on Visualization and Computer Graphics*, 15(1):62–76, 2009. URL https://doi.org/10.1109/TVCG.2008.81.

Tzu-Mao Li, Michal Lukáč, Michaël Gharbi, and Jonathan Ragan-Kelley. Differentiable vector graphics rasterization for editing and learning. *ACM Trans. Graph. (Proc. SIGGRAPH Asia)*, 39 (6):193:1–193:15, 2020. URL https://cseweb.ucsd.edu/~tzli/diffvg/.

Jinfan Liu, Wuze Zhang, Zhangli Hu, Zhehan Zhao, Ye Chen, and Bingbing Ni. Differentiable stroke planning with dual parameterization for efficient and high-fidelity painting creation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 26721–26730, June 2026. URL https://openaccess.thecvf.com/content/CVPR2026/html/Liu_Differentiable_Stroke_Planning_with_Dual_Parameterization_for_Efficient_and_High-Fidelity_CVPR_2026_paper.html.

Songhua Liu, Tianwei Lin, Dongliang He, Fu Li, Ruifeng Deng, Xin Li, Errui Ding, and Hao Wang. Paint transformer: Feed forward neural painting with stroke prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 6598–6607, 2021. URL https://openaccess.thecvf.com/content/ICCV2021/html/Liu_Paint_Transformer_Feed_Forward_Neural_Painting_With_Stroke_Prediction_ICCV_2021_paper.html.

Markus Pobitzer, Chang Liu, Chenyi Zhuang, Teng Long, Bin Ren, and Nicu Sebe. Loomis painter: Reconstructing the painting process. *arXiv preprint arXiv:2511.17344*, 2025. URL https://arxiv.org/abs/2511.17344.

Tomoya Sawada and Marie Katsurai. MambaPainter: Neural stroke-based rendering in a single step. In *SIGGRAPH Asia 2024 Posters*, SA Posters '24, pp. 1:1–1:2, New York, NY, USA, 2024. Association for Computing Machinery. ISBN 9798400711381. URL https://doi.org/10.1145/3681756.3697906.

Peter Schaldenbrand, James McCann, and Jean Oh. FRIDA: A collaborative robot painter with a differentiable, Real2Sim2Real planning environment. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 11712–11718. IEEE, May 2023. URL https://doi.org/10.1109/ICRA48891.2023.10160702.

Jaskirat Singh, Cameron Smith, Jose Echevarria, and Liang Zheng. Intelli-Paint: Towards developing more human-intelligible painting agents. In *Proceedings of ECCV*, pp. 685–701, 2022. URL https://1jsingh.github.io/intelli-paint.

Zhengyan Tong, Xiaohang Wang, Shengchao Yuan, Xuanhong Chen, Junjie Wang, and Xiangzhong Fang. Im2Oil: Stroke-based oil painting rendering with linearly controllable fineness via adaptive sampling. In *Proceedings of the 30th ACM International Conference on Multimedia*, MM '22, pp. 1035–1046, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450392037. URL https://doi.org/10.1145/3503161.3547759.

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 Trans. Graph.*, 41(4):86:1–86:11, jul 2022. ISSN 0730-0301. URL https://doi.org/10.1145/3528223.3530068.

Yael Vinker, Yuval Alaluf, Daniel Cohen-Or, and Ariel Shamir. CLIPascene: Scene sketching with different types and levels of abstraction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4146–4156, October 2023. URL https://openaccess.thecvf.com/content/ICCV2023/html/Vinker_CLIPascene_Scene_Sketching_with_Different_Types_and_Levels_of_Abstraction_ICCV_2023_paper.html.

XiMing Xing, Chuang Wang, Haitao Zhou, Jing Zhang, Qian Yu, and Dong Xu. DiffSketcher: Text guided vector sketch synthesis through latent diffusion models. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 15869–15889. Curran Associates, Inc., 2023. doi: 10.52202/075280-0698.

URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/333e67fc4728f147d31608db3ca78e09-Abstract-Conference.html.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. ReAct: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=WE_vluYUL-X.

Amy Zhao, Guha Balakrishnan, Kathleen M. Lewis, Frédo Durand, John V. Guttag, and Adrian V. Dalca. Painting many pasts: Synthesizing time lapse videos of paintings. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8435–8445, 2020. URL https://xamyzhao.github.io/timecraft/.

Zhengxia Zou, Tianyang Shi, Shuang Qiu, Yi Yuan, and Zhenwei Shi. Stylized neural painting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15689–15698, 2021. URL https://jiupinjia.github.io/neuralpainter/.

A SUPPLEMENTARY MATERIAL

Section A.1 describes stroke rendering, stage budgets, geometric editing, and acceptance conditions, followed by the analytic distance calculation for quadratic Bézier curves. Section A.2 contains the additional evaluation protocols.

A.1 MORE IMPLEMENTATION DETAILS

A.1.1 STROKE RENDERING

Spatial coverage and opacity. For pixel center $\mathbf{x}$, let $d_i(\mathbf{x}) = \min_{u\in[0,1]} \|\mathbf{x} - \mathbf{b}_i(u)\|_2$ be its distance to stroke i. The renderer uses the half-width w_i from Section 3.1 and a sigmoid $\sigma(v) = 1/(1+e^{-v})$ to compute spatial coverage:

$$k_i(\mathbf{x}) = \max\left\{0, \frac{\sigma((w_i - d_i(\mathbf{x}))/(\tau w_i)) - 1/2}{\sigma(1/\tau) - 1/2}\right\}, \qquad \tau = 0.3. \tag{5}$$

The normalized profile equals one on the centerline and reaches zero at the boundary $d_i = w_i$. The source alpha is $\alpha_i(\mathbf{x}) = o_i k_i(\mathbf{x})$, where o_i is the stroke opacity. Each stroke contributes once to a pixel through its maximum spatial coverage. For this distance-based profile, that maximum is attained at the closest point on the curve.

Depth-based compositing. Let $i_1, \ldots, i_N$ enumerate strokes in increasing depth, from the frontmost stroke to the backmost. Starting with accumulated color $\mathbf{C}_0 = \mathbf{0}$ and alpha $A_0 = 0$, the renderer applies

$$\begin{aligned} \mathbf{C}_k(\mathbf{x}) &= \mathbf{C}_{k-1}(\mathbf{x}) + \mathbf{c}_{i_k}\alpha_{i_k}(\mathbf{x})\big(1 - A_{k-1}(\mathbf{x})\big), \\ A_k(\mathbf{x}) &= A_{k-1}(\mathbf{x}) + \alpha_{i_k}(\mathbf{x})\big(1 - A_{k-1}(\mathbf{x})\big), \\ \mathcal{R}(\mathcal{P})(\mathbf{x}) &= \mathbf{C}_N(\mathbf{x}) + \mathbf{1}\big(1 - A_N(\mathbf{x})\big). \end{aligned} \tag{6}$$

These front-to-back alpha updates allocate the remaining uncovered fraction to each successive stroke and then to the white background. During fitting, larger drawing indices receive smaller depths, allowing later strokes to cover earlier layers.

A.1.2 STAGE TARGETS AND FITTING SCHEDULE

Panoptic segmentation supplies image regions, and palette quantization within each region creates progressively detailed targets. A local compiler turns the hierarchical region plan into active drawing masks. The final target uses the reference image within its active region. Table 1 gives the six-stage schedule; the execution-stage column is used throughout the main text.

Table 1: **Stage fitting schedule.** Each stage fits newly added strokes while retaining the parameters of all previously accepted strokes.

Execution stage	Internal ID	Added strokes	Cumulative strokes
1	1	480	480
2	2	400	880
3	3	560	1440
4	5	1280	2720
5	7	960	3680
6	8	320	4000

Stage masks may overlap, supporting successive layers of color and detail. Each stroke belongs to one region, while its stage records its creation pass. Protected polygons supplied by the VLM restrict eligibility for alignment and consolidation. Both editing stages run after all six fitting stages are complete and use the flow plan prepared before fitting.

A.1.3 REGIONAL ALIGNMENT

Unoriented flow representation. Guide tangents use the double-angle representation $\phi(\vartheta) = (\cos 2\vartheta, \sin 2\vartheta)$, which assigns the same vector to angles separated by π. Let $\bar{\phi}(\mathbf{x})$ be the support-weighted mean of nearby guide tangents in this representation. At supported locations, the corresponding orientation and unit tangent field are

$$\vartheta_F(\mathbf{x}) = \tfrac{1}{2}\,\mathrm{atan2}(\bar{\phi}_2(\mathbf{x}), \bar{\phi}_1(\mathbf{x})), \qquad \mathcal{F}(\mathbf{x}) = (\cos\vartheta_F(\mathbf{x}), \sin\vartheta_F(\mathbf{x})). \tag{7}$$

This representation provides a common coordinate system for averaging and interpolating unoriented directions. Distance to the guides and agreement among their tangents determine the directional support used by the local tools.

Task construction. A region selection specifies its ID, alignment strength, anchor neighborhood radius, priority, and preferred stroke IDs. The radius lies between 4 and 128 pixels. The local tools assign selected strokes to disjoint tasks. Same-region neighbors within 12 pixels provide additional context for the angular acceptance check and retain their geometry when they are outside the task. Flow locations with support below 0.08 are excluded from alignment and consolidation.

Geometric operations. For a task, the tools compute a support-weighted common direction and split the flow into segments when its turn exceeds $25°$. Each stroke rotates about its chord midpoint toward the segment direction, with a maximum rotation of $90°$. The computed rotation is applied directly. For a segment whose flow turn is below $8°$, the middle control point moves toward the chord. For larger turns, it moves toward the streamline obtained by integrating the field. Its displacement is bounded by $\min(1.5\widetilde{W}, 0.5L_i)$, where L_i is the stroke's arc length and $\widetilde{W}$ is the median stroke-width value used by the editing tools. A translation along the flow normal brings each midpoint toward the segment's median streamline, with displacement at most $\widetilde{W}$. Tangential position is preserved during this translation. The accepted arc length remains between 85% and 115% of its preceding value.

The width values W_i and $\widetilde{W}$ below follow the editing specification's width convention. Their conversion to the renderer's half-width w_i remains to be confirmed. This distinction keeps the supplied geometric thresholds separate from the renderer notation.

Geometric acceptance. All four conditions below must hold for an alignment task:

1. The neighboring-stroke angle, evaluated with the same-region context, decreases by at least $5°$, or its final value is below $15°$.
2. Error relative to the fixed flow field is preserved or reduced.
3. In nearly straight segments, the fraction of middle control points near their chords increases. A point is near its chord when its distance is at most $0.15W_i$. A fraction already equal to one satisfies this condition.
4. The mean displacement of the control points is at least 2 pixels.

The task then passes through the PSNR checks in Section 3.4. Stroke count, drawing order, depth, IDs, width, color, and opacity remain fixed throughout alignment.

A.1.4 CANDIDATE CONSTRUCTION AND CONSOLIDATION

Longitudinal chains. These groups connect strokes end to end. The line connecting their centers must have mean alignment with the flow of at least 0.72 and minimum alignment of at least 0.45. The gap is at most $\max(2\,\mathrm{px}, 0.5\widetilde{W})$. Links use the nearest predecessor and successor along the flow, preserving the intervening stroke order along the path. A path is split when support falls below 0.08 or the connection clearly turns back on itself.

Tubular groups. These groups cover side-by-side strokes and larger longitudinal gaps in a narrow band. Lateral distance is at most $\max(4\,\mathrm{px}, 2W)$, and the gap along the flow is at most $\max(8\,\mathrm{px}, 3W)$, with W denoting the width used for the candidate. Each stroke has alignment of at least 0.72 with the band axis, and source color distance is at most 0.15. Connections use the nearest neighbor on either side; crossing strokes belong to separate groups.

High-confidence candidates. The high-confidence subset satisfies four conditions jointly: path support is at least 0.12, mean alignment along the path is at least 0.78, the gap relative to stroke width is at most 0.5, and source color distance is at most 0.15. This subset determines eligibility for local pair consolidation and for region-level merge rules.

Local pair pass. The local pass processes high-confidence longitudinal candidates containing exactly two strokes. Tubular groups, fragments created by splitting backtracking paths, and candidates assembled from multiple windows proceed to the remaining-candidate stage. Eligible pairs are sorted by decreasing path alignment. Selection assigns each stroke to at most one pair, skipping a later pair that shares a stroke with an already selected pair. Each selected pair proposes one replacement stroke with flow strength 0.65.

Remaining candidates and model decisions. After the local pass, successfully used source strokes are removed from subsequent candidates. The remaining registry contains tubular groups, chains of three or more strokes, pairs below the high-confidence threshold, and high-confidence pairs skipped during selection whose source strokes are still present. The tools select at most 64 representatives according to alignment and priority. For each displayed candidate, the VLM can specify merge or keep, a target count, and a flow strength.

The model can also assign one of three rules to a region: turn regional merging off, reduce eligible groups to approximately half their source count, or reduce longitudinal groups to approximately one third. The two reduction rules use flow strengths 0.65 and 0.80, respectively. A regional rule applies to high-confidence candidates in that region that have no individual decision. An individual keep or merge decision takes precedence over the regional rule. Candidates receiving no merge authorization retain their source strokes.

Curve construction and checks. Both the local and model-selected paths use the frozen flow field to generate replacement geometry. A coverage corridor accommodates the connection between source fragments and bounds the replacement near the original shape. The tools first check the corridor, then apply the common PSNR conditions to candidates passing that check. Appearance fitting adjusts color and width near the source values for at most six steps. Each accepted group replaces its source strokes; a group that fails a check retains its preceding geometry and appearance.

A.1.5 Model Decisions and Local Execution

Table 2 summarizes the VLM's information and decisions. Local tools maintain the current ordered stroke sequence, stroke IDs, regional information, fixed flow plan, and candidate registry needed to resolve these selections. Written rationales explain selections, while the numeric fields determine the requested operation.

Table 2: **Model guidance and local execution.** The VLM supplies regional and candidate decisions. Local tools compute geometry and apply acceptance checks.

Step	Model information and decision	Local operation
Shared planning	Reference image and ETF; regional guide curves at structure, form, and texture scales, drawing regions, and protected polygons.	Construct directional guidance and support; compile stage masks; build targets from segmentation and palettes.
Alignment	Reference, current render, flow view, and regional statistics; region IDs, strength, radius, priority, and preferred strokes.	Assemble tasks, rotate and reshape strokes, translate along the normal, and check task acceptance.
Consolidation	Representatives from the remaining registry; individual merge/keep decisions, target counts, strengths, and regional rules.	Resolve selections, construct replacement curves, fit appearance, and check corridor and PSNR conditions.

Acceptance and retained state. The six-stage result supplies the fixed global PSNR reference. Each current accepted stroke sequence supplies the local reference for its next task or merge group. The local reference and candidate are evaluated over the same edited region against the corresponding reference-image region. Geometry determines the proposed change; Equations 3 and 4 deter-

mine whether its rendered result remains within the image-quality limits. Passing candidates update the current stroke sequence. Each failed task or group restores its own preceding state, preserving earlier accepted changes. The output after consolidation is the complete accepted stroke sequence.

A.1.6 Analytic Distance to a Quadratic Bézier Curve

Distance query. The splatting kernel in Equation 5 requires the closest distance from pixel center $\mathbf{x}$ to a stroke. Dropping the stroke index, write

$$\mathbf{U} = \mathbf{p}_0 - 2\mathbf{p}_1 + \mathbf{p}_2, \quad \mathbf{V} = \mathbf{p}_1 - \mathbf{p}_0, \quad \mathbf{W} = \mathbf{p}_0 - \mathbf{x}, \qquad \mathbf{b}(u) - \mathbf{x} = \mathbf{U}u^2 + 2\mathbf{V}u + \mathbf{W}. \tag{8}$$

The query minimizes $f(u) = \|\mathbf{b}(u) - \mathbf{x}\|_2^2$ over $u \in [0, 1]$. Its minimum occurs at an endpoint or an interior stationary point. Differentiating and dividing by four gives the cubic

$$Au^3 + Bu^2 + Cu + D = 0, \qquad \begin{cases} A = \mathbf{U}^\top \mathbf{U}, \\ B = 3\mathbf{U}^\top \mathbf{V}, \\ C = 2\mathbf{V}^\top \mathbf{V} + \mathbf{U}^\top \mathbf{W}, \\ D = \mathbf{V}^\top \mathbf{W}. \end{cases} \tag{9}$$

Here D denotes the constant coefficient of this cubic. All coordinates and stroke half-widths use the same pixel units; any numerical rescaling is undone before evaluating the coverage kernel.

Closed-form real roots. For $A > 0$, substitute $u = y - B/(3A)$ to obtain $y^3 + py + q = 0$, with

$$p = \frac{3AC - B^2}{3A^2}, \qquad q = \frac{2B^3 - 9ABC + 27A^2D}{27A^3}, \qquad \Delta = \left(\frac{q}{2}\right)^2 + \left(\frac{p}{3}\right)^3. \tag{10}$$

When $\Delta > 0$, the unique real root is

$$y = \mathrm{cbrt}\left(-q/2 + \sqrt{\Delta}\right) + \mathrm{cbrt}\left(-q/2 - \sqrt{\Delta}\right), \tag{11}$$

where $\mathrm{cbrt}(v) = \mathrm{sign}(v)|v|^{1/3}$ is the real cube root. When $\Delta < 0$, all three real roots are obtained from

$$y_k = 2\sqrt{-p/3}\cos\left(\frac{\phi + 2\pi k}{3}\right), \quad \phi = \arccos\left(\mathrm{clip}\left(\frac{-q/2}{\sqrt{-(p/3)^3}}, -1, 1\right)\right), \quad k = 0, 1, 2. \tag{12}$$

For $\Delta = 0$, let $v = \mathrm{cbrt}(-q/2)$; the distinct roots are $2v$ and $-v$. This also handles the triple root $p = q = 0$ without dividing by zero. Each root is transformed back to u before interval filtering.

Candidate selection and degenerate curves. Collect all real stationary roots in $[0, 1]$ together with both endpoints:

$$\mathcal{U} = \{0, 1\} \cup \{u \in [0, 1] : Au^3 + Bu^2 + Cu + D = 0\}, \qquad u^\star = \arg\min_{u \in \mathcal{U}} f(u), \qquad d = \sqrt{f(u^\star)}. \tag{13}$$

Testing every candidate is necessary because a cubic root can correspond to a local maximum. Exact ties use the smaller parameter for reproducibility. If $\mathbf{U} = \mathbf{0}$, the curve is the segment from $\mathbf{p}_0$ to $\mathbf{p}_2$. With $\mathbf{e} = \mathbf{p}_2 - \mathbf{p}_0$, its closest parameter is

$$u^\star = \mathrm{clip}\left(\frac{(\mathbf{x} - \mathbf{p}_0)^\top \mathbf{e}}{\mathbf{e}^\top \mathbf{e}}, 0, 1\right). \tag{14}$$

If $\mathbf{e} = \mathbf{0}$ as well, the distance is simply $\|\mathbf{x} - \mathbf{p}_0\|_2$. Coincident endpoints alone do not imply a point: a nonzero $\mathbf{U}$ still requires the cubic solver.

Numerical implementation. The exact branches above require finite-precision safeguards. Discriminant tests use a scale-aware tolerance proportional to machine precision and $\max(1, |p|^3, q^2)$. Nearly repeated roots are deduplicated, and roots just outside $[0, 1]$ are clamped only within a specified interval tolerance. Each retained stationary candidate is checked using its polynomial residual. Ill-conditioned queries fall back to interval root isolation: roots of the derivative $3Au^2 + 2Bu + C$

partition $[0, 1]$ into monotone intervals, sign-changing intervals are bisected, and critical points with near-zero cubic residual are retained to capture repeated roots.

For a nearly linear curve, replacing it by its endpoint segment is an approximation, not an exact degeneracy. The uniform positional difference is at most $\|\mathbf{U}\|_2/4$, which also bounds the distance error. A segment shortcut is therefore permitted only when this bound is below a configured geometric tolerance; otherwise the cubic path is retained. These checks prevent a small leading coefficient or cancellation in Cardano's formula from silently selecting an incorrect closest point.

Differentiation and rasterization. Once the minimizing candidate $u^\star$ is selected, its index is held fixed for differentiation. At a unique nondegenerate minimum with $d > 0$, the envelope derivative is

$$\frac{\partial d}{\partial \mathbf{p}_j} = \omega_j(u^\star)\frac{\mathbf{b}(u^\star) - \mathbf{x}}{d}, \qquad (\omega_0, \omega_1, \omega_2) = ((1-u)^2,\, 2u(1-u),\, u^2). \tag{15}$$

Interior stationarity removes the derivative through $u^\star$; an endpoint minimum has a fixed parameter. Tied minima and zero distance require an explicit subgradient convention. A numerical implementation may clamp the backward denominator while retaining the unmodified forward distance; that surrogate and its threshold must be recorded. Rasterization evaluates only pixels in the control-point bounding box expanded by w_i, since a Bézier curve lies within its control-point convex hull and coverage vanishes outside its half-width. Per-pixel queries can be parallelized while retaining the prescribed compositing order.

A.2 Additional Experiments

Status. The following are planned extensions of Section 4; no measurements or qualitative outcomes are yet available. They test the harness and renderer separately before assessing their interaction.

In-context demonstrations and feedback. Vary the number of demonstrations over $\{0, 1, 2, 4, 8\}$ using nested development-only example sets. Compare successful examples alone with success-and-rejection trajectories at matched context budgets. Separately remove rendered feedback, diagnostic text, or rationales while preserving action schemas, numerical tools, and fitting budgets. Report edit yield, held-out process quality, fidelity, model usage, and total latency. Repeated example selections separate the effect of example count from the choice of particular demonstrations.

Verification and failure recovery. Compare full verification with variants removing the cumulative gate, protected-ring checks, or semantic checkpoints. Controlled invalid proposals exercise stale IDs, excessive displacements, boundary crossings, noncontiguous merges, and timeouts. Record false acceptance and rejection rates against independently inspected cases, as well as whether rollback restores geometry, ordering, caches, and lineage. These tests assess enforcement of specified conditions, not universal semantic correctness. All failures remain in the quality and cost summaries.

Distance accuracy and gradient checks. Evaluate ordinary curves, nearly straight curves, point degeneracies, coincident endpoints with nonzero curvature, repeated cubic roots, and queries near stroke boundaries. Compare analytic distances against a high-precision root-isolation reference and a dense polyline approximation with increasing sample count. Report median and worst-case distance error in pixels, fallback frequency, and raster differences at fixed edge softness. Central finite differences check gradients only at unique, nondegenerate minima; tied minima and clipping boundaries are reported separately. This avoids interpreting a nondifferentiable query as a smooth-gradient test failure.

Quality–cost sensitivity. Sweep image resolution, stroke budget, edge softness, and contour compression ratio while keeping all other settings fixed. Measure distance evaluation, compositing, backward differentiation, numerical fitting, model calls, and verification separately, alongside total elapsed time. GPU timings require synchronization and disclosed warm-up; remote-call latency and rejected work remain part of the end-to-end budget. Equal-time and equal-stroke comparisons are reported separately so that a faster solver is not credited with gains caused solely by additional optimization.

Generalization and qualitative failures. Report results by portraits, still lifes, landscapes, and stylized paintings rather than only as a pooled average. Prespecified examples should include texture-heavy regions, weak boundaries, crossing contours, closed loops, incorrect semantic priors, and intentionally discontinuous marks. For each, display the reference, stage targets, rendered checkpoints, flow overlays, merge lineage, and any rejected proposal. Image-level paired confidence intervals accompany quantitative comparisons; artist judgments independently assess whether improved internal scores correspond to plausible painting processes.