TOPOCONTOUR: PERSISTENT-HOMOLOGY-GATED COUNTERFACTUAL PATH DELETION FOR FIXED-BUDGET FILLED SVG REPAIR

Anonymous authors
Paper under double-blind review

ABSTRACT

Repairing a filled SVG requires mapping a raster-space defect to the vector path that caused it. We introduce **TopoContour**, a topology-aware framework that converts unmatched PH events into path proposals and tests them through counterfactual Bézier path deletion under a fixed geometry budget. Across synthetic data and two real filled-SVG domains, TopoContour consistently improves topology repair and path responsibility relative to residual-driven alternatives. On Tabler Filled and Bootstrap Filled, the topology-repair rate improves over residual-centered gating by 24.7 and 27.5 percentage points, respectively. Compared with exhaustive ungated search, TopoContour evaluates 47.3% and 48.0% fewer candidates while retaining 95.8% and 96.5% of its topology-repair performance. Boundary F1 remains comparable. The same pattern holds for asymmetric blobs and rounded-square Bézier contours, with topology-repair rates between 0.894 and 0.952. Increasing the path budget from 4 to 16 raises the reduction in candidate evaluations to 86.2% on Bootstrap and 93.1% on Tabler, with counterfactual repair-decision speedups of 12.5× and 13.7×. The results show that PH events can serve as effective, transferable action proposals for efficient path-level vector repair.

1 INTRODUCTION

Repairing a filled SVG is different from correcting a raster image. A raster discrepancy points to pixels, while an SVG discrepancy also has a structural cause: several closed Bézier contours jointly determine the visible boundary and interior topology. When an unintended contour creates a hole, a useful repair must identify the responsible path and edit the existing vector representation without disturbing unrelated geometry.

Differentiable rasterizers and learned vectorizers make it possible to optimize or predict complete vector graphics from image-space supervision (Li et al., 2020; Carlier et al., 2020; Reddy et al., 2021; Ma et al., 2022; Jain et al., 2023). Local SVG repair asks a more specific question. Given a target raster, a current rendering, and a fixed-budget path bank, which path should be changed, and how should that choice be verified? Pixel residuals offer a useful spatial cue, but a large residual may be caused by antialiasing, a displaced boundary, or a broad exterior contour. These signals do not by themselves identify the path responsible for a topological event.

We introduce **TopoContour**, a PH-gated method for counterfactual path deletion. An unmatched hole event in the target and current raster pair supplies the location and scale of a candidate gate. Each compatible closed Bézier path is deleted temporarily and rendered. The final decision is then based on the topological and local reconstruction effects of that intervention. In this way, PH proposes a small action space and counterfactual rendering verifies the selected path.

The fixed-budget setting makes this distinction measurable. Ungated counterfactual search evaluates every active path and provides an exhaustive reference under the same action score. Across synthetic data, Tabler Filled SVGs, and Bootstrap Filled SVGs, matched events improve responsibility and topology repair over residual-centered and shuffled-event gates. They also retain a similar repair-quality operating point while reducing candidate evaluations by about 48% with four paths and by up to 93.1% with sixteen paths.

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.

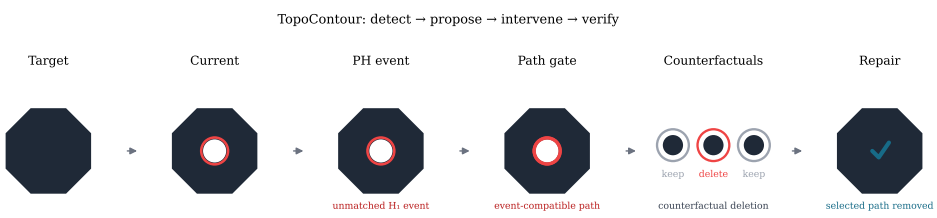


Figure 1: The repair sequence: an unmatched hole event is detected, compatible paths are proposed, candidate deletions are rendered, and the best valid repair is selected or rejected.

This study formulates PH-gated counterfactual Bézier deletion as a path-level repair problem and introduces a vector-native protocol with exact labels for the injected path. The experiments connect event localization with responsibility, repair quality, and candidate cost. They cover elliptical, asymmetric-blob, and rounded-square closed-contour corruptions, as well as a zero-retuning transfer from Tabler Filled to Bootstrap Filled.

2 Related Work

Differentiable vector graphics support gradient-based optimization of path geometry and raster-to-vector systems that predict layered or hierarchical representations (Li et al., 2020; Loper & Black, 2014; Kato et al., 2018; Liu et al., 2019; Laine et al., 2020; Carlier et al., 2020; Reddy et al., 2021; Ma et al., 2022; Jain et al., 2023). Sequential vector representations have also been used for learned sketch generation (Ha & Eck, 2018). This line of work mainly addresses construction or global reconstruction. TopoContour instead starts from an existing fixed-budget path bank and studies a local structural intervention whose effect can be inspected directly.

Persistent homology (PH) summarizes connected components and cavities across a filtration (Edelsbrunner et al., 2002; Edelsbrunner & Harer, 2010; Carlsson, 2009; Otter et al., 2017). Topology-aware losses and differentiable topological layers use such summaries to guide segmentation, representation learning, or reconstruction (Hu et al., 2019; Gabrielsson et al., 2020; Clough et al., 2022; Hofer et al., 2017; Moor et al., 2020; Carriere et al., 2020). The role of PH here is different. An unmatched event supplies a localized proposal, after which rendering tests the effect of changing a particular path.

Cubic Bézier curves are compact primitives for editable vector geometry. Their control points support local shape refinement, but the rasterized result of several contours depends on their joint fill semantics. Geometric proximity and residual overlap can identify a region for editing, yet neither one establishes which path is responsible for a topological event. The counterfactual rendering step addresses this question by evaluating the result of deleting each proposed path.

The stability of persistence diagrams motivates persistent events as summaries of image-space structure (Cohen-Steiner et al., 2007). Counterfactual and attribution methods likewise study how interventions or feature changes explain a prediction (Pearl, 2009; Ribeiro et al., 2016; Lundberg & Lee, 2017; Sundararajan et al., 2017; Koh & Liang, 2017; Miller, 2019). Exhaustive leave-one-out evaluation can measure every deletion, but it also invokes the renderer for paths that have little relation to the event. The proposed procedure keeps the same direct verification for plausible paths and uses event support to reduce the number of tested actions.

3 Problem and Method

3.1 Fixed-budget repair

We represent an SVG by a path bank $\mathcal{P} = \{(P_i, z_i)\}_{i=1}^{B}$, where each P_i is a closed contour of S cubic Bézier segments and $z_i \in \{0, 1\}$ records whether the path is present. The main repair experiments use $B = 4$ and $S = 4$; the efficiency study keeps the same interface while varying $B \in \{4, 8, 16\}$. Even-odd composition renders the bank as an occupancy field $R(\mathcal{P})$. An action

either deletes one active path or abstains. Since deletion changes only a presence state, the geometry budget and all retained control points stay fixed.

Reference-guided repair receives a target raster $I^\star$, a current path bank, and its rendering I. The current rendering contains a hole that is absent from the target. The injected path, event center, and corruption metadata are hidden from the method and used only for evaluation. A deletion is responsible when it removes the injected topological event under the common evaluator.

A spurious-hole corruption is a one-dimensional homology event that appears in the current rendering but has no counterpart in the target. In the vector-native protocol, inserting a closed Bézier contour into an unused slot creates this event without changing the remaining paths. Deleting path i gives

$$\mathcal{P}^{(-i)} = \{(P_j, z_j \mathbf{1}[j \neq i])\}_{j=1}^{B}.$$

The inserted slot therefore provides an exact responsibility label, while the geometry budget, control-point count, and unaffected paths remain unchanged. The construction also separates inference from evaluation. The method receives the target and current rasters, the control points, and the path-presence states, but not the inserted slot or event metadata.

3.2 Event proposal and counterfactual verification

We compute cubical persistence representations for the target and current occupancy fields and match their hole events. An unmatched current event is written as $e = (q_e, r_e, \pi_e)$, with center, support scale, and persistence. Dense polyline sampling then gives the distance $d_i(e)$ from each active path to the event center. The event-conditioned candidate set is

$$\mathcal{C}(e) = \{i : z_i = 1,\ d_i(e) \leq \lambda r_e\},$$

where $\lambda = 3.0$ is selected on the Tabler development split and then fixed. This gate removes paths that are unlikely to cause the event, but it does not make the final decision.

For every $i \in \mathcal{C}(e)$, we render the counterfactual $I^{(-i)} = R(\mathcal{P}^{(-i)})$. A lexicographic utility first prioritizes removal of the unmatched event, followed by local reconstruction gain, minimal change outside the event support, and deterministic tie breaking. The selected action is

$$i^\star = \arg\max_{i \in \mathcal{C}(e)} U(I^\star, I, I^{(-i)}, e),$$

The method abstains when no gated deletion provides a positive topological or local benefit. Ungated search applies the same utility to all active paths, which isolates the effect of the proposal set. Residual-centered and shuffled-event gates keep the counterfactual procedure unchanged and replace only the matched event cue.

3.3 Soft Even-Odd composition

Let $m_i(x) \in [0, 1]$ be the soft occupancy induced by path geometry. The differentiable even-odd composition is

$$E(x) = \frac{1 - \prod_{i=1}^{B}(1 - 2 z_i m_i(x))}{2}.$$

For binary occupancies, this expression equals the standard even-odd rule. Commutativity makes it invariant to path order, and its polynomial form is differentiable wherever the rasterizer is differentiable. Appendix A.10 gives the arguments and the numerical consistency check. The representation preserves filled-SVG semantics and can support later local geometry refinement; the experiments here focus on the discrete deletion action.

As an implementation check, all 24 permutations of a four-contour configuration produce a maximum numerical discrepancy of 1.11×10^{-16}. Relative to CairoSVG, binary IoU ranges from 0.9890

Table 1: Evaluation domains and their roles. All rows use fixed-budget closed Bézier paths; complete secondary measurements are given in Appendix A.

Domain	Cases	Construction	Role
Controlled synthetic	12	Programmatic scenes with known events and path roles	Mechanism validation
Tabler raster-derived	646 pairs; 128 test	SVG-derived spurious holes with fixed-budget initialization	Held-out repair
Tabler vector-native	88 icons; 365 cases	One closed contour inserted into an unused slot	Main causal evaluation
Bootstrap vector-native	63 icons; 218 cases	Same insertion protocol on Bootstrap Filled v1.13.1	Zero-retuning transfer
Shape and budget studies	Tabler plus Bootstrap	Blob and rounded-square corruptions; $B = 4, 8, 16$	Shape generalization and efficiency

to 0.9965 over raster resolutions from 128 to 512 pixels, and the common evaluator reports identical topology. These checks establish renderer consistency independently of the responsibility experiments.

3.4 Candidate validity and complexity

Deleting a path changes only its presence bit. The retained curves keep their original control points, closure, and segment count. After each action, the evaluator checks budget validity, finite control points, path closure, self-intersection status, and raster validity. With P active paths, ungated counterfactual search performs P candidate evaluations, whereas the event-conditioned method evaluates $K = |\mathcal{C}(e)|$. Candidate rendering is the repeated operation in the decision process, so the ratio K/P measures the computational effect of the proposal.

4 Experimental Setup

The experiments mirror the sequence of the proposed decision. We test localization of an unmatched PH event, path responsibility, repair quality, and search cost. The real-domain studies then examine the same event-to-path relation under a held-out corruption protocol, a second Filled-SVG collection, non-elliptical closed contours, and larger path banks.

4.1 Protocol and datasets

All main evaluations use 128×128 rasters, four closed paths with four cubic segments per path, and even-odd filling. The held-out Tabler split selects λ on development cases only. Bootstrap uses the frozen value and the same evaluator without retuning. In the vector-native protocol, one closed contour is inserted into an unused slot. The slot supplies an exact responsibility label while the remaining path bank is unchanged. Table 1 summarizes the resulting datasets.

The raster-derived Tabler collection contains 646 eligible pairs, split into 454 training, 64 development, and 128 test cases. The vector-native benchmarks contain 365 Tabler cases from 88 icons and 218 Bootstrap cases from 63 icons. After construction, every method is denied the corruption slot, event center, and event scale. At decision time, the target raster is the only reference information available; no method receives the injected path label or corruption coordinates.

4.2 Controls and metrics

The controls change either the proposal signal or the search breadth. Exact max-residual uses the largest target/current residual and exact single-path renderings. Residual-centered gating substitutes a residual centroid for the PH center, while shuffled-event gating breaks event-to-instance correspondence and keeps the same gate interface. Ungated search evaluates every active path with the

Table 2: Compared methods and control structure. All methods use the same fixed path budget and candidate utility unless stated otherwise; no method receives corruption metadata.

Method	Proposal signal	Search and action
TopoContour	Matched PH event	Event-gated; shared counterfactual utility
Residual-centered gating	Residual center	Event-gated; shared counterfactual utility
Shuffled-event gating	Mismatched PH event	Event-gated; shared counterfactual utility
Ungated counterfactual search	None	All active paths; shared counterfactual utility
Global leave-one-out reference	None	All active paths; global target IoU

Table 3: Controlled mechanism evaluation.

Stage	Cue or selector	Result
Event localization	PH event	**12 / 12**; 0.00 px mean error
	Residual centroid	9 / 12; 2.06 px mean error
	Residual peak	0 / 12; 8.55 px mean error
Responsibility	Event-gated fixed budget	**12 / 12**
	Nearest active path	8 / 12
	Random assignment	4 / 12
	Residual overlap	0 / 12
Repair	PH plus gated responsibility	**12 / 12**; IoU 1.000, Boundary F1 1.000
	PH plus random responsibility	4 / 12; IoU 0.746
	Pixel-only selector	0 / 12; IoU 0.234

shared utility. The global leave-one-out reference also evaluates every deletion, but ranks actions by global target IoU. Table 2 summarizes these comparisons.

All methods receive the same target raster, current raster, Bézier control points, path-presence mask, renderer, action interface, and evaluator. Identity leaves the current path bank unchanged. Nearest active path chooses the path closest to the inferred event, and random active provides a selector sanity check. Exact max-residual is the appearance-driven baseline in the held-out protocol. Ungated search shares the counterfactual utility of TopoContour but tests every active path, whereas the global leave-one-out reference uses global target IoU. This design separates the proposal signal from the scoring rule and search breadth.

The primary metrics are topology-repair rate, responsibility accuracy, and mean candidate evaluations. Final raster IoU, Boundary F1, abstention, and catastrophic-edit rate provide complementary measures; a catastrophic edit has $\Delta\mathrm{IoU} < -0.10$. The evaluator also checks closure, budget validity, and self-intersection. Where reported, paired confidence intervals use 20,000 bootstrap resamples.

5 Results

5.1 The event-to-action mechanism

The controlled cases separate localization, responsibility, and repair. The matched PH event finds the known hole center in all 12 cases. The residual centroid succeeds in 9 cases, and the residual peak succeeds in none. Given the matched event, event-gated attribution identifies the responsible path in all 12 cases, with IoU and Boundary F1 equal to 1.000. Nearest-path, random, and residual-overlap selectors perform less well, indicating that the result depends on both the event cue and the path-level intervention. The two topology evaluators agree across all 270 combinations of resolution, threshold, and controlled structure. Table 3 gives the stage-wise results.

Table 4: Corruption-metadata-free repair on the held-out Tabler split.

Method	Topology	Resp. accuracy	Final IoU	ΔIoU	Boundary F1	Abstain.
Identity	0.1094	N/A	0.7369	0.0000	0.1405	1.0000
Nearest active path	0.6328	0.5703	0.7268	-0.0101	0.1340	0.0391
Exact max-residual	0.4375	0.0000	0.0780	-0.6589	0.0577	0.0000
TopoContour	**0.7031**	**0.6328**	**0.7414**	**+0.0045**	**0.1418**	0.1719
Global leave-one-out reference	0.6719	0.7734	0.7496	+0.0126	0.1464	0.0859
Random active	0.5547	0.2344	0.4778	-0.2591	0.0951	0.0000

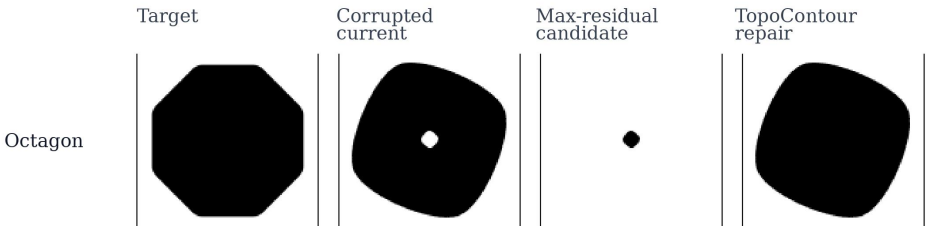


Figure 2: Representative comparison of exact max-residual selection and TopoContour on the Tabler diagnostic set. The third panel shows the candidate selected by exact max-residual, and the fourth panel shows the verified repair produced by TopoContour.

5.2 Held-out Tabler repair

The held-out Tabler test hides the corruption slot and event metadata after gate selection. From an initial IoU of 0.7369, TopoContour reaches a topology-repair rate of 0.7031, responsibility accuracy of 0.6328, final IoU of 0.7414, and Boundary F1 of 0.1418. Its topology-repair rate exceeds exact max-residual by 0.2656 and identity by 0.5938, while final IoU changes by +0.0045. The global leave-one-out reference reaches final IoU 0.7496 and a topology-repair rate of 0.6719, placing the event-driven decision close to the target-aware reference on the structural metric. Table 4 reports the results.

The comparison with exact max-residual separates structural correction from residual magnitude. Its topology-repair rate is 0.4375 and its final IoU is 0.0780, since the selected path often removes a dominant foreground contour. TopoContour abstains on 17.19% of cases and avoids catastrophic edits while preserving the starting silhouette. The global leave-one-out reference remains a useful target-aware reference, but it evaluates every active path and is not available as a corruption-metadata-free proposal rule.

Figure 2 shows a representative repair in which the max-residual candidate changes a large foreground contour, while the event-gated counterfactual selects the contour whose deletion removes the unmatched hole. The panel illustrates the aggregate action study rather than providing a separate quantitative evaluation.

5.3 Vector-native repair across SVG domains

The vector-native benchmark makes path responsibility explicit by inserting one closed contour per case. On Tabler, TopoContour repairs 0.9288 of 365 cases and identifies the injected path in 0.7507. Residual-centered gating reaches 0.6822 and 0.2877, respectively. Shuffling the event lowers the two values to 0.7644 and 0.5068, showing that candidate restriction alone does not explain the gain. Without parameter retuning, the Bootstrap benchmark gives TopoContour a topology-repair rate of 0.8945 and responsibility accuracy of 0.8349, compared with 0.6193 and 0.1743 for residual-centered gating. The aggregate results appear in Table 5.

Table 5: Vector-native results on Tabler Filled and Bootstrap Filled.

Domain	Method	Topology	Responsibility	Boundary F1	Candidates	Catastrophic
Tabler	Exact max-residual	0.5589	0.0000	0.0491	2.97	1.0000
	Residual-centered gating	0.6822	0.2877	0.1076	1.34	0.3863
	Shuffled-event gating	0.7644	0.5068	0.1344	1.39	0.0740
	TopoContour	**0.9288**	**0.7507**	**0.1440**	**1.53**	**0.0110**
	Ungated counterfactual search	0.9699	0.9479	0.1494	2.90	0.0055
Bootstrap	Exact max-residual	0.6972	0.0000	0.0668	2.77	0.9450
	Residual-centered gating	0.6193	0.1743	0.1028	1.10	0.4358
	Shuffled-event gating	0.6560	0.3807	0.1228	1.02	0.2798
	TopoContour	**0.8945**	**0.8349**	**0.1371**	**1.34**	**0.0413**
	Ungated counterfactual search	0.9266	0.8991	0.1407	2.59	0.0183

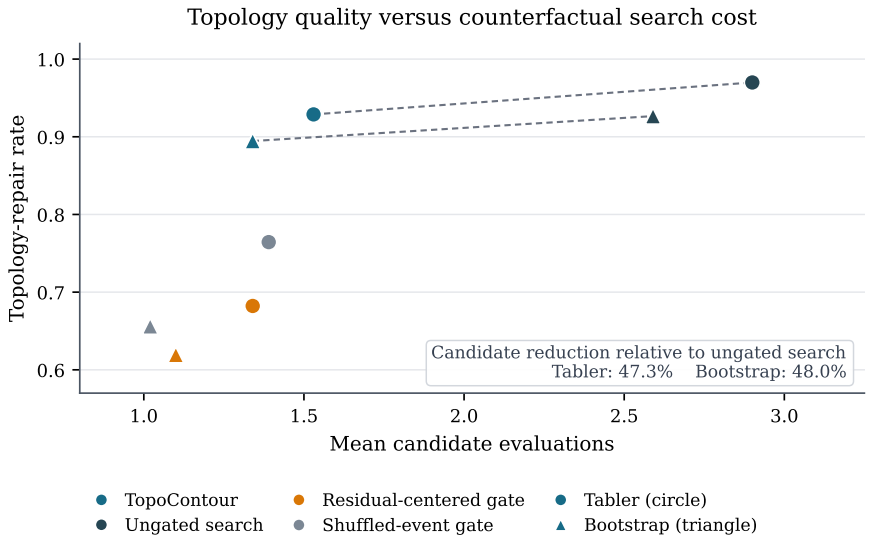


Figure 3: Topology quality versus counterfactual search cost under the four-path budget. Marker shape denotes domain and color denotes proposal method.

The comparison with ungated search measures the cost of restricting the proposal set. On Tabler, ungated search reaches topology repair 0.9699 and responsibility accuracy 0.9479. TopoContour gives Boundary F1 of 0.1440 versus 0.1494 and reduces the mean candidate count from 2.90 to 1.53. On Bootstrap, the corresponding topology-repair rates are 0.9266 and 0.8945, Boundary F1 values are 0.1407 and 0.1371, and candidate counts are 2.59 and 1.34. The shared utility therefore keeps the two methods at a similar quality operating point while the event proposal removes unrelated actions.

5.4 Quality and efficiency trade-off and shape transfer

With four paths, TopoContour evaluates 47.3% fewer candidates than ungated search on Tabler and 48.0% fewer on Bootstrap. It retains 95.8% and 96.5% of the corresponding topology-repair rates, while the Boundary F1 differences are only 0.0054 and 0.0036. Figure 3 summarizes this operating point. The event-to-path relation also extends beyond elliptical corruptions. Across asymmetric blobs and rounded squares in both domains, topology-repair rates range from 0.8936 to 0.9524 and responsibility accuracy ranges from 0.7609 to 0.8182. Table 6 reports the shape-wise results, and Appendix A gives the complete paired metrics.

Table 6: Generalization to non-elliptical closed Bézier corruptions. Candidate reduction is measured relative to ungated search; full paired comparisons are given in the appendix.

Domain	Shape	Cases	Topology	Responsibility	Reduction
Tabler	Asymmetric blob	63	**0.9524**	**0.8095**	**61.8%**
Tabler	Rounded square	66	**0.9394**	**0.8182**	**56.9%**
Bootstrap	Asymmetric blob	46	**0.9130**	**0.7609**	**48.8%**
Bootstrap	Rounded square	47	**0.8936**	**0.7872**	**47.2%**

Table 7: Candidate evaluation and repair-decision scaling. K_{T} and P_{U} are the mean candidate counts for TopoContour and ungated search; time reports median milliseconds as TopoContour/ungated.

Domain	B	K_{T}	P_{U}	Red.	Time (ms), T/U	Speedup
Tabler	4	1.05	3.30	68.2%	8.7 / 33.0	3.8×
Tabler	8	1.05	7.30	85.6%	9.6 / 69.8	7.3×
Tabler	16	1.05	15.30	**93.1%**	11.7 / 160.1	**13.7**×
Bootstrap	4	1.42	2.84	50.0%	8.7 / 24.5	2.8×
Bootstrap	8	1.63	6.84	76.2%	9.7 / 61.5	6.3×
Bootstrap	16	2.05	14.84	**86.2%**	12.0 / 150.1	**12.5**×

After aligning cases for which all methods produce an output, icon-cluster bootstrap intervals for the topology-repair advantage of TopoContour over residual-centered gating remain positive for both shapes in both domains. The intervals for the comparison with shuffled-event gating are also positive. These paired results indicate that the event-to-path relation is not tied to the ellipse template. Table 6 reports the corresponding effect sizes and candidate reductions.

5.5 Event scale and clearance

The Tabler benchmark varies event scale and clearance. At high clearance, topology-repair rates are 1.0000, 0.9398, and 0.9398 for large, medium, and small events; at low clearance, they are 0.9524, 0.8605, and 0.8302. Mean candidate counts range from 1.23 to 2.14. Appendix A gives the stratified measurements.

5.6 Scaling with the path bank

At a 16-path budget, TopoContour eliminates 93.1% of candidate evaluations on Tabler and 86.2% on Bootstrap, with decision-time speedups of 13.7× and 12.5×. Gated search stays near one or two candidates as the bank grows, whereas ungated search grows with the number of active paths. Table 7 gives the timing measurements.

6 Discussion

The ablations identify the source of the gain. Residual-centered gating keeps counterfactual verification but replaces event structure with appearance magnitude. Shuffling keeps a similar budget while breaking event-to-icon correspondence. Their lower rates show that a compact search is useful when tied to the observed event; ungated search remains the quality reference, whereas TopoContour reaches a comparable structural decision at lower cost.

The two real-domain protocols support this interpretation. Raster-derived Tabler hides the corruption slot, whereas vector-native benchmarks attach an exact label to the injected path under the same deletion interface. In both domains, matched events improve structural repair and path responsibility, while shuffling removes the gain at a comparable candidate scale. The result points to event-to-path correspondence rather than a particular supervision source.

Because the action changes one presence bit, selected edits leave retained geometry unchanged.

Efficiency follows from this separation. Counterfactual rendering verifies semantics, while event localization avoids unrelated paths. From four to sixteen slots, ungated candidates grow from roughly three to fifteen, but the gated count stays near one or two. The resulting speedup reflects the proposal mechanism rather than a weaker score or a smaller rendering problem.

PH therefore has a concrete algorithmic role: it proposes paths, counterfactual intervention tests them, and rendering verifies the result. Persistence stability motivates persistent events as scalar-field summaries (Cohen-Steiner et al., 2007), but persistence alone does not guarantee topological equivalence. Appendix A.10 states the elementary binary consistency, permutation invariance, and differentiability properties of Soft Even-Odd composition.

7 Conclusion

TopoContour turns fixed-budget filled-SVG repair into path responsibility: a matched PH event proposes candidate contours, and counterfactual deletion verifies the intervention. Controlled and real-domain experiments, including zero-retuning Bootstrap transfer and non-elliptical corruptions, show consistent event-to-path behavior. Candidate evaluations fall by up to 93.1% and decision time by up to 13.7× while repair quality remains close to exhaustive search.

AI use statement

Generative AI tools were used to assist with language polishing. The authors take responsibility for the final manuscript.

References

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*, 2020.

Gunnar Carlsson. Topology and data. *Bulletin of the American Mathematical Society*, 46(2): 255–308, 2009. doi: 10.1090/S0273-0979-09-01249-X. URL https://www.ams.org/journals/bull/2009-46-02/S0273-0979-09-01249-X/.

Mathieu Carriere, Frederic Chazal, Yuichi Ike, Theo Lacombe, Martin Royer, and Yuhei Umeda. Perslay: A neural network layer for persistence diagrams and new graph topological signatures. In *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108, pp. 2786–2796. PMLR, 2020. URL https://proceedings.mlr.press/v108/carriere20a.html.

James R. Clough, Nicholas Byrne, Ilkay Oksuz, Veronika A. Zimmer, Julia A. Schnabel, and Andrew P. King. A topological loss function for deep-learning based image segmentation using persistent homology. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(12): 8766–8779, 2022.

David Cohen-Steiner, Herbert Edelsbrunner, and John Harer. Stability of persistence diagrams. *Discrete and Computational Geometry*, 37:103–120, 2007.

Herbert Edelsbrunner and John Harer. *Computational Topology: An Introduction*. American Mathematical Society, 2010.

Herbert Edelsbrunner, David Letscher, and Afra Zomorodian. Topological persistence and simplification. *Discrete and Computational Geometry*, 28:511–533, 2002.

Rickard Brüel Gabrielsson, Bradley J. Nelson, Anjan Dwaraknath, and Primoz Skraba. A topology layer for machine learning. In *Proceedings of the International Conference on Artificial Intelligence and Statistics*, 2020.

David Ha and Douglas Eck. A neural representation of sketch drawings. In *International Conference on Learning Representations*, 2018. URL https://iclr.cc/virtual/2018/poster/293.

Christoph Hofer, Roland Kwitt, Marc Niethammer, and Andreas Uhl. Deep learning with topological signatures. In *Advances in Neural Information Processing Systems*, volume 30, 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/hash/883e881bb4d22a7add958f2d6b052c9f-Abstract.html.

Xiaoling Hu, Fuxin Li, Dimitris Samaras, and Chao Chen. Topology-preserving deep image segmentation. In *Advances in Neural Information Processing Systems*, 2019.

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*, 2023.

Hiroharu Kato, Yoshitaka Ushiku, and Tatsuya Harada. Neural 3d mesh renderer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3907–3916, 2018. doi: 10.1109/CVPR.2018.00411. URL https://openaccess.thecvf.com/content_cvpr_2018/html/Kato_Neural_3D_Mesh_CVPR_2018_paper.html.

Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70, pp. 1885–1894. PMLR, 2017. URL https://proceedings.mlr.press/v70/koh17a.html.

Samuli Laine, Janne Hellsten, Tero Karras, Yeongho Seol, Jaakko Lehtinen, and Timo Aila. Modular primitives for high-performance differentiable rendering. *ACM Transactions on Graphics*, 39 (6):1–14, 2020. doi: 10.1145/3414685.3417861. URL https://dl.acm.org/doi/10.1145/3414685.3417861.

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

Shichen Liu, Tianye Li, Weikai Chen, and Hao Li. Soft rasterizer: A differentiable renderer for image-based 3d reasoning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 7708–7717, 2019. doi: 10.1109/ICCV.2019.00780. URL https://openaccess.thecvf.com/content_ICCV_2019/html/Liu_Soft_Rasterizer_A_Differentiable_Renderer_for_Image-Based_3D_Reasoning_ICCV_2019_paper.html.

Matthew M. Loper and Michael J. Black. OpenDR: An approximate differentiable renderer. In *Computer Vision – ECCV 2014*, volume 8695, pp. 154–169. Springer International Publishing, 2014. doi: 10.1007/978-3-319-10584-0_11. URL https://is.mpg.de/ics/publications/loper-eccv-2014.

Scott M. Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems*, volume 30, 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/hash/8a20a8621978632d76c43dfd28b67767-Abstract.html.

Xu Ma, Yuqian Zhou, Xingqian Xu, Bin Sun, Valerii Filev, Nikita Orlov, Yun Fu, and Humphrey Shi. Towards layer-wise image vectorization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

Tim Miller. Explanation in artificial intelligence: Insights from the social sciences. *Artificial Intelligence*, 267:1–38, 2019. doi: 10.1016/j.artint.2018.07.007. URL https://www.sciencedirect.com/science/article/pii/S0004370218305988.

Michael Moor, Max Horn, Bastian Rieck, and Karsten Borgwardt. Topological autoencoders. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119, pp. 7045–7054. PMLR, 2020. URL https://proceedings.mlr.press/v119/moor20a.html.

Nina Otter, Mason A. Porter, Ulrike Tillmann, Peter Grindrod, and Heather A. Harrington. A roadmap for the computation of persistent homology. *EPJ Data Science*, 6(1):17, 2017. doi: 10.1140/epjds/s13688-017-0109-5. URL https://link.springer.com/article/10.1140/epjds/s13688-017-0109-5.

Judea Pearl. Causal inference in statistics: An overview. *Statistics Surveys*, 3: 96–146, 2009. doi: 10.1214/09-SS057. URL https://projecteuclid.org/journals/statistics-surveys/volume-3/issue-none/Causal-inference-in-statistics-An-overview/10.1214/09-SS057.full.

Pradyumna Reddy, Michael Gharbi, Michal Lukáč, and Niloy J. Mitra. Im2Vec: Synthesizing vector graphics without vector supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Why should i trust you? explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1135–1144, 2016. doi: 10.1145/2939672.2939778. URL https://dl.acm.org/doi/10.1145/2939672.2939778.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70, pp. 3319–3328. PMLR, 2017. URL https://proceedings.mlr.press/v70/sundararajan17a.html.

A Additional Experimental Details

A.1 Path and corruption construction

We evaluate a cubic Bézier segment with control points c_0, c_1, c_2, c_3 as

$$b(t) = (1-t)^3 c_0 + 3(1-t)^2 t c_1 + 3(1-t)t^2 c_2 + t^3 c_3, \qquad t \in [0,1].$$

Adjacent segments share terminal anchors, which makes every active path closed. We construct the current rendering by inserting a closed contour into an unused slot under even-odd filling. The target and current path banks differ in one presence state, while the control points of every pre-existing path stay unchanged. The method does not receive the inserted slot, event center, or event scale. A deletion is counterfactually responsible when it removes the unmatched hole event and satisfies the common repair evaluator.

A.2 Protocol constants

All main real-icon evaluations use 128×128 rasters. The repair-quality protocol uses four paths with four cubic Bézier segments per path and even-odd fill semantics. The raster-derived protocol uses curvature-based contour initialization. We select the gate multiplier on the Tabler development split and fix it at 3.0 for all subsequent evaluations. The held-out paired test uses bootstrap seed 20260805 and 20,000 resamples.

A.3 Geometry checks

After each action, the evaluator checks that the path-bank dimensions remain fixed, every active path is closed, all control-point values are finite, and the rendering is valid. Self-intersection is measured independently of path closure. Since deletion changes only a presence bit, every retained path keeps its original control points.

A.4 Baseline implementation details

All comparisons use the same target raster, current raster, path bank, renderer, action interface, and utility unless an ablation explicitly changes one of these factors. Exact max-residual uses the largest target/current residual and exact single-path renderings. Residual-centered gating substitutes a residual centroid for the matched event center, while shuffled-event gating takes an event descriptor from another case and keeps the event-gated interface. Ungated counterfactual search evaluates every active path with the shared utility. The global leave-one-out reference also evaluates every deletion, but ranks actions by global target IoU. Identity, nearest-active-path, random assignment, and residual-overlap selectors appear only in the controlled or held-out diagnostics.

The event-conditioned utility uses a lexicographic order: removal of the unmatched event, event-local reconstruction gain, minimal change outside the event support, and deterministic tie breaking. The gate multiplier is selected on the Tabler development split only. Test-time decisions do not use the injected responsibility label, event center, or clean vector graphic.

A.5 PH event extraction and gate

We convert the target and current occupancy fields to cubical persistence representations. Hole events are matched between the two filtrations. A current event without a target counterpart is represented by its spatial center q_e, support scale r_e, and persistence π_e. For an active path, dense polyline sampling gives

$$d_i(e) = \min_{x \in P_i} \|x - q_e\|_2.$$

The gate retains path i when $d_i(e) \le \lambda r_e$. The event center need not coincide with the pixel of maximum residual. The event center summarizes a cavity across a filtration, while the residual peak

describes an appearance discrepancy. This distinction motivates the residual-centered and shuffled-event controls.

A.6 Non-elliptic corruption details

The asymmetric blob uses four radial knots at unequal distances from the center and is converted into a closed contour with four cubic segments. The rounded square uses four corner knots under the same segment budget. For each shape, we evaluate three combinations of scale and clearance. Table 8 gives the complete quality metrics for TopoContour and complements Table 6.

Table 8: Complete metrics for TopoContour on non-elliptical corruptions.

Domain	Shape	Cases	Final IoU	ΔIoU	Boundary F1	Catastrophic
Tabler	Asymmetric blob	63	0.7560	+0.0128	0.1629	0.0000
Tabler	Rounded square	66	0.7445	+0.0170	0.1635	0.0000
Bootstrap	Asymmetric blob	46	0.6703	-0.0541	0.1496	0.0870
Bootstrap	Rounded square	47	0.6947	-0.0170	0.1488	0.0426

For paired inference, we align cases by domain, shape, icon, radius, and clearance and use icons as bootstrap clusters. With 20,000 resamples, the 95% confidence intervals for the topology-repair difference between TopoContour and residual-centered gating are [0.300, 0.650] for Bootstrap blobs, [0.325, 0.625] for Bootstrap rounded squares, [0.339, 0.649] for Tabler blobs, and [0.344, 0.667] for Tabler rounded squares. The corresponding intervals for the difference between TopoContour and shuffled-event gating are [0.117, 0.367], [0.125, 0.383], [0.103, 0.299], and [0.122, 0.322].

A.7 Event scale and clearance strata

The Tabler vector-native benchmark covers six combinations of event scale and clearance. Under high clearance, topology-repair rates for large, medium, and small events are 1.0000, 0.9398, and 0.9398; under low clearance, they are 0.9524, 0.8605, and 0.8302. Some Bootstrap strata contain fewer cases. Their topology-repair rates range from 0.5455 to 0.9643, with the lowest value occurring among the 11 large, low-clearance cases. Figure 4 shows the sample counts.

A.8 Repair-decision timing protocol

The scaling study adds identical path pairs that cancel under even-odd composition, expanding a four-slot path bank to 8 and 16 paths without changing the rendering. TopoContour and ungated search share the target raster, current raster, inferred PH event, renderer, and action score. Each method runs three times per case, and we report the median for each condition. Timing begins immediately before candidate selection and ends when the method returns the selected counterfactual action; data loading and one-time preprocessing are excluded. Each budget uses 20 Tabler repair cases and 19 Bootstrap repair cases.

Icon-level paired bootstrap analysis shows that candidate reduction increases with the path budget. At $B = 16$, the 95% confidence intervals are [92.4%, 93.6%] on Tabler and [75.8%, 91.8%] on Bootstrap. The corresponding ratios of aggregate median decision times are 13.7× and 12.5×.

A.9 Extended interpretation

The shuffled-event ablation preserves the distribution of event descriptors and a similar candidate budget, but removes the correspondence between the event and the current geometry. The resulting decrease in responsibility and topology repair indicates that a small candidate set is useful when it is tied to the actual defect. The fixed-budget action makes the structural effect explicit: the method changes one presence bit instead of adding contours or reallocating control points across the icon.

The experiments move from controlled validation to held-out repair, vector-native responsibility labels, zero-retuning transfer, shape variation, and path-bank scaling. A learned event-to-path compatibility model could amortize the proposal stage for larger path banks while retaining counterfactual

rendering as supervision or final verification. The same event-to-action interface could later support insertion, splitting, or local control-point updates once matching event-action pairs are available.

Residuals identify where two renderings differ, but a path-level edit also requires evidence about the effect of changing a vector object. Here, PH supplies a localized candidate set and counterfactual rendering tests whether deleting a proposed path resolves the event. The shuffled-event ablation keeps a comparable gate interface while breaking event-to-instance correspondence, so the comparison measures the matched event rather than only the benefit of a smaller search space.

The fixed path and control-point budget separates structural selection from representation growth. The repaired output does not gain capacity through additional contours or a global redistribution of control points. Only one presence bit changes, while unrelated geometry remains available for inspection. This keeps the action transparent for editable assets and allows the quality and efficiency comparison to use the same representation across domains.

At larger budgets, exhaustive verification remains semantically useful but scales with the number of active paths. The timing study expands the path bank with pairs that cancel under even-odd composition, so the rendering stays unchanged while the number of unrelated counterfactuals grows. The resulting speedup measures the computational value of event localization without changing image complexity.

A.10 Representation properties

For binary path occupancies, Soft Even-Odd composition reduces to the usual parity rule because each factor is either zero or one. The product is commutative, so the rendered occupancy is invariant to a permutation of the path bank. For a differentiable rasterizer, the composition is a polynomial in the path occupancies and is differentiable wherever the rasterizer is differentiable. These facts justify using a shared, unordered path bank for counterfactual deletion. They do not imply equality of Betti numbers or homotopy equivalence from a zero scalar topology loss.

For binary consistency, each factor in the product is (+1) for zero occupancy and (-1) for unit occupancy. The product is therefore (-1) exactly when an odd number of active paths cover the point, which gives (E(x)=1). Permutation invariance follows directly from commutativity of multiplication. For continuous masks, the expression is polynomial in each mask value; composing it with a differentiable rasterizer propagates gradients to the control points wherever that rasterizer is differentiable. The persistence stability theorem bounds changes in persistence diagrams by changes in the underlying scalar field and motivates the event representation, but the theorem is used here as a stability motivation rather than as a guarantee of homotopy equivalence.

A.11 Additional visual diagnostics

The appendix provides the stratified event-scale measurements and additional positive repair examples that complement the aggregate results in the main text. These figures make the operating conditions and the selected edits easier to inspect without interrupting the main argument.

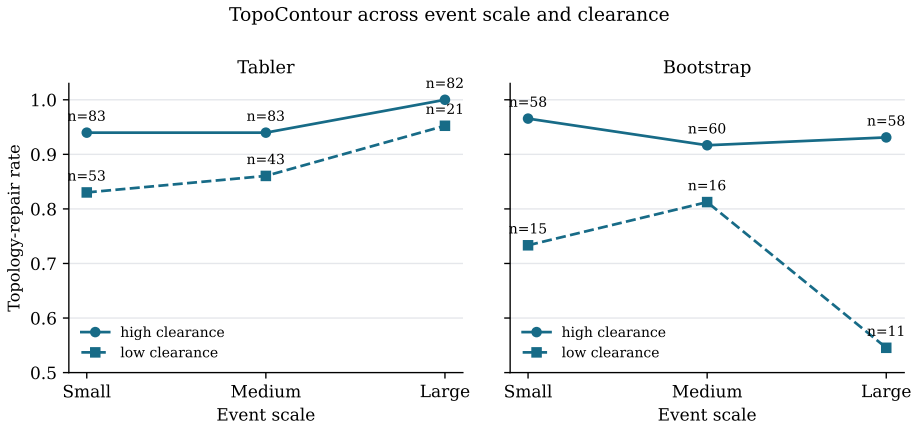


Figure 4: Topology-repair rate by event scale and clearance. Sample counts are shown because vector-native eligibility produces unequal strata.

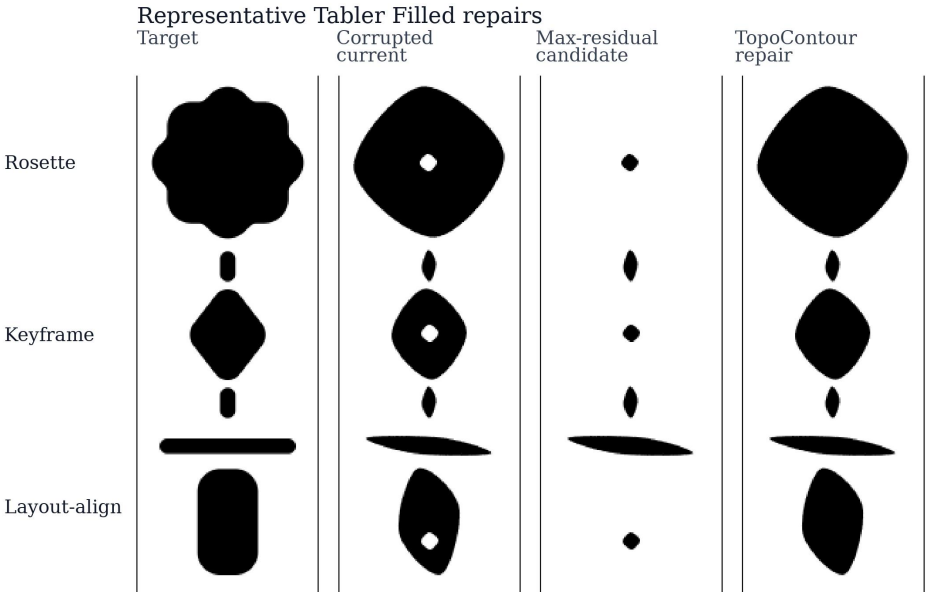


Figure 5: Representative repairs from the Tabler diagnostic set. Each row shows the target, the corrupted rendering, the exact max-residual candidate, and the result after TopoContour verification.