CONNECTED BUT IRRELEVANT: SEPARATOR NON-INTERFERENCE IN NEURAL GRAPH ALGORITHMS

Anonymous authors
Paper under double-blind review

ABSTRACT

Attach a subgraph to a graph through a single bridge. Every original minimum-spanning-tree edge label and shortest-path distance is provably unchanged, yet the extension changes the input and the algorithm's execution trace. Do neural graph algorithms respect this independence? We introduce SepBench, a paired evaluation of separator non-interference with oracle-certified answer-preserving and answer-changing extensions. A reproduced graph transformer trained on 16-node graphs violates the contract on 53.3–65.9% of exactly solved 64-node bases when extended by 64 nodes, across three seeds. Violations also occur when the composite graph is no larger than the training size, although at much lower rates, and in a message-passing model without global attention. Inference-time interventions implicate cross-block information flow: block-membership masks retain paths through shared articulation states. Executing blocks with private articulation copies removes these paths. Both tested transformers then exhibit zero all-pair violations, with base accuracy and sensitivity essentially unchanged; message passing retains numerical exceptions. In a separate matched training comparison, augmentation and consistency regularisation reduce violations while lowering response to answer-changing edits from 97.6% to 68.9%, mostly through augmentation alone. These results show why invariance must be evaluated jointly with sensitivity: a lower violation rate can reflect reduced responsiveness to changes that matter.

1 INTRODUCTION

Classical graph algorithms specify which parts of an input can affect an answer. Minimum-spanning-tree membership depends on cycles within an edge's biconnected block (Kruskal, 1956); shortest-path distances in a region separated from the source by one vertex depend on the outside through that vertex's distance (Dijkstra, 1959). These dependencies hold independently of the input distribution. Neural algorithmic reasoning (Veličković & Blundell, 2021; Veličković et al., 2022) trains models to imitate algorithmic outputs, but high aggregate accuracy does not establish that a model respects the same dependencies. A small number of changed edge labels can leave F1 high while exposing dependence on information that cannot affect the answer.

We test one such independence with a single-bridge graft (Figure 1a). Attach a connected subgraph H to a base graph G through one new edge. No cycle can cross that bridge, so every MST label inside G is unchanged. Positive weights likewise prevent a shorter path between old vertices through H. The extension changes the token count, reachable set and execution trace, while preserving the original answer. This gives a precise behavioural contract: predictions on the old graph should remain unchanged. SepBench pairs this test with an oracle-certified control attached through two edges, where an old answer does change. The control checks whether low violation rates coexist with responsiveness to relevant structure.

The distinction matters even for accurate models. We reproduce the released 16M-parameter graph transformer of Stoll et al. (2025) within its reported size-generalisation band. At 0.9815 structured macro-F1 on 64-node bases, it changes a protected decision on 62.1% of the bases it had solved *exactly* after a 64-node graft. The composite is eight times the training size; violations persist at

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.

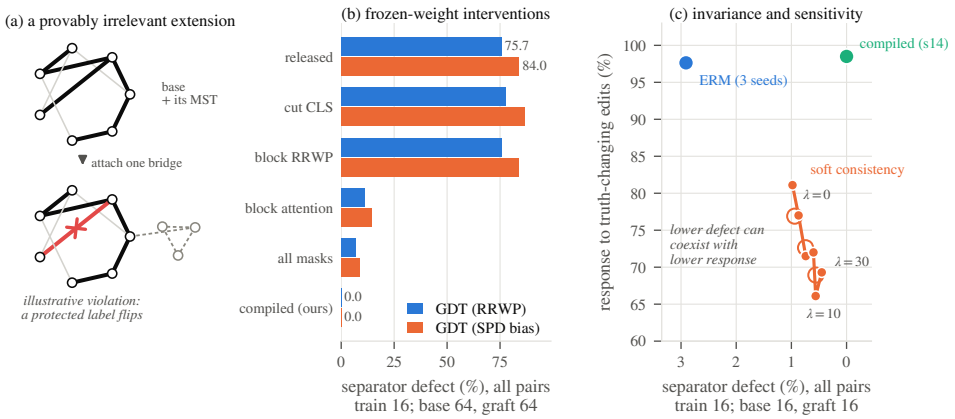

Figure 1: **(a)** A single-bridge extension preserves every original MST label; the crossed edge illustrates a violation. **(b)** Inference-time interventions on two transformers (64+64 nodes, seed 14): block attention reduces violations, partitioning removes them. **(c)** Invariance and sensitivity at 16+16 nodes: the consistency sweep (seed 14; rings are three-seed means) lowers defect and response together. Both panels use all-pair rates.

much lower rates when the composite is no larger than a training instance. A second transformer and a message-passing network also violate the contract, the latter at 6.1% on 16-node bases despite having no explicit global communication channel.

Inference-time interventions on frozen weights connect these violations to cross-block information flow. Removing the global CLS token does not consistently improve the result. Restricting attention to within-block token pairs removes most violations at $n = 64$, but the tested masks retain paths through articulation vertices shared by two blocks. We therefore execute blocks with private copies of these vertices, using only the task's bounded portal state between blocks. Both transformers exhibit zero all-pair violations under this construction, with utility essentially unchanged in the tested settings. Message passing reaches zero on exactly solved bases but retains numerical exceptions on all pairs.

Training for the invariance provides a complementary test. Answer-preserving augmentation and a consistency penalty in the style of Bevilacqua et al. (2023) reduce defects five-fold in a matched training comparison, while response to answer-changing edits falls by twenty-nine percentage points. Most of that sensitivity loss occurs with augmentation alone, before adding the penalty, and base-distribution accuracy remains nearly constant (Table 2). Thus, measuring violations alone would miss a substantial loss of responsiveness. This possibility is known (Arjovsky et al., 2019; Tramèr et al., 2020); certifying both kinds of intervention makes its empirical cost measurable.

Contributions.

1. **A certified evaluation and failure case.** SepBench jointly measures non-interference and sensitivity, exposing violations in three architectures despite high base accuracy (§§2–3).
2. **A structural diagnosis and repair.** Frozen-weight interventions distinguish global-channel ablations from blockwise execution, which removes shared articulation states while retaining utility in the tested transformers (§4, §4.1).
3. **A measured invariance–sensitivity trade-off.** A controlled comparison separates augmentation from regularisation and shows why both answer-preserving and answer-changing tests are needed (§5); shortest paths provide a second task (§6).

2 SEPARATOR NON-INTERFERENCE AND SEPBENCH

Write $G = (V, E, w)$ for an undirected graph with distinct positive edge weights. Our two tasks are minimum spanning forest, whose output is a label $y_e \in \{0, 1\}$ per edge, and single-source shortest path from a fixed root, whose output is a predecessor π_v per vertex.

Definition 1 (Admissible extension)**.** *$G' = (V \cup V_H,\ E \cup E_H \cup \{b\})$ is an* admissible extension *of G if $H = (V_H, E_H)$ is connected and vertex-disjoint from G, $b = \{a, h\}$ is a single new edge with $a \in V$, $h \in V_H$, and the weights of G' are positive and distinct. We call b the bridge, a the attachment vertex, and C the resulting contract.*

Everything we protect follows from one classical fact: b is a cut edge of G', so no cycle of G' contains a vertex of both V and V_H. The blocks (biconnected components) of G' are therefore exactly the blocks of G, those of H, and $\{b\}$ itself—equivalently, the connected components of the graphic matroid, computed by depth-first search in linear time (Tarjan, 1972).

Proposition 1 (Protected scope)**.** *For every admissible extension G' of G: (i) for every $e \in E$, e lies in the minimum spanning forest of G' if and only if it lies in that of G; (ii) if the SSSP root lies in V, then $d_{G'}(v) = d_G(v)$ for every $v \in V$ and the set of shortest paths inside G is unchanged.*

Part (i) is the cycle property of Kruskal (1956) applied blockwise; part (ii) holds because b is the only edge between the two sides, so a path that leaves V traverses it twice, which cannot shorten anything under positive weights. Predecessors need one further step: distinct weights make the spanning forest unique but not the shortest path ($1{+}4 = 2{+}3$), and Appendix A gives the tie-break along with every proof. We call the labels named in Proposition 1 the *protected outputs*, and a model *C-noninterfering* when its predictions on the protected outputs are identical on G and on G'—the term is information-flow security's (Goguen & Meseguer, 1982).

The contract is parameter-free: it concerns equality of discrete predictions, with no tolerance or distributional assumption. This does not require bitwise equality of internal states, whose drift we report separately. At $n = 16$ the released generator produces 2.48 natural bridges per graph on average. Separator structure therefore occurs in training, although this alone does not make the grafted graphs distribution-matched.

Certified input pairs. Each SepBench cell contains answer-preserving pairs (G, G'_{keep}) and a separately sampled set of answer-changing pairs (G, G'_{chg}). Bases and grafts use the backbone's released graph generator. The former extension attaches through one bridge; the latter uses two edges, allowing a cycle through old and new material. Draws for the control are rejected unless an independent oracle confirms that at least one old label changes. The MST control uses cheap portal edges to obtain such changes more often (Appendix D). The two pair sets share sizes and generator settings, not individual base graphs; each set is held fixed across model arms. We write n for base size and use an equally sized graft unless both sizes are specified. Thus, a cell at $n = 64$ evaluates a 128-node composite.

Violation and sensitivity metrics. Let $f(G)$ denote the decoded predictions on the protected indices $P(G)$, and let $E_f(G)$ be the event that all these predictions are correct. For answer-preserving pairs, define

$$D_{\text{all}}(f) = \Pr[f(G'_{\text{keep}})_{P(G)} \neq f(G)],$$
$$D_{\text{exact}}(f) = \Pr[f(G'_{\text{keep}})_{P(G)} \neq f(G) \mid E_f(G)].$$

The conditional rate isolates corruption of a base answer that was entirely correct. Its denominator depends on the model: the primary GDT cell has $1{,}522$ exactly solved bases and $D_{\text{exact}} = 62.09\%$, compared with $D_{\text{all}} = 75.70\%$. We therefore use all-pair rates for cross-arm comparisons and also report paired tests on the common exactly solved subset in §4. A low conditional rate alone need not imply low violation on the full sample.

For MST answer-changing pairs, the oracle identifies the edges whose true labels change. The *should-change* rate records a predicted change on at least one of those edges; correctness requires all their new labels to be right. An unrelated flip does not count as a response, and cross-arm comparisons use all control pairs. SSSP instead measures predecessor response and correctness

Table 1: The main result, three training seeds; 64 nodes attached to a 64-node base. "masks" is the strongest mask-only arm. Every column that places arms side by side uses the all-pair denominator, as §2 requires; "exact bases" conditions on the model having solved the base entirely, so that denominator moves with the arm.

	all pairs %			exact bases			base macro-F1		should-chg %	
seed	rel.	masks	comp.	rel. defects	%	comp. %	rel.	comp.	rel.	comp.
14	75.70	6.98	**0.00**	945/1522	62.09	**0.00**	0.9815	0.9817	73.2	72.8
15	77.04	5.71	**0.00**	899/1365	65.86	**0.00**	0.9809	0.9810	70.9	70.8
16	70.50	7.59	**0.00**	423/794	53.27	**0.00**	0.9775	0.9774	69.6	69.7

over vertices whose oracle *distance* changes, conditional on an exactly solved base (§6). We keep this task-specific distinction explicit.

Accuracy, non-interference and sensitivity answer separate questions: a constant predictor satisfies the invariance, a responsive one can still be wrong, and neither test identifies the algorithm implemented (Appendix A.4).

Evaluation protocol. MST labels come from an independent union-find Kruskal implementation, and SSSP labels from Dijkstra; neither uses model predictions or generator bookkeeping. MST scores are decoded as a maximum-score spanning forest with deterministic tie-breaking. Intervals over pairs are Wilson score (Wilson, 1927); per-edge rates use a bootstrap that resamples graphs. We verify checkpoint loading and repeated-evaluation consistency before computing paired metrics. The primary decision rule was fixed before training: a conditional defect of at least 5%, with a graph-level 95% interval lower bound above 2% and at least 500 exactly solved bases, supports the method claim. Smaller effects and insufficient samples are reported separately. This is an inclusion and interpretation rule for the experiment, not a tolerance in the mathematical contract. Appendices C–E give the full rule, numerical regime and reproducibility checks.

3 Trained models violate the contract

Backbones. Our primary backbone is the released 16-layer, 384-dimensional, 16-head graph transformer of Stoll et al. (2025) with RRWP positional encodings (Ma et al., 2023), trained on the authors' own MST task (10^6 graphs at $n = 16$, tested at $n = 64$). We verified our reproduction against the published size-OOD score before running any paired evaluation: 0.9634 against a reported 0.9604 ± 0.0091. The second backbone replaces RRWP with a shortest-path-distance attention bias in the style of Ying et al. (2021), trained identically. The third is a message-passing network (Xu et al., 2019; Hu et al., 2020), trained identically again, and it is structurally unlike the other two: no explicit global token, no all-pairs attention, no global positional encoding, and a receptive field equal to its layer count. We report it at n=16; at n=64 it solves only 183 of 10,000 bases exactly, so the frozen rule returns STOP_INSUFFICIENT_EXACT_BASES; we withhold its conditional estimate under that rule (§8).

Table 1 gives the headline. Across three training seeds, the released model changes a provably-unchanged edge label on 53.3–65.9% of the instances it had solved exactly. The maximum change in its logit margin on a protected edge reaches 32.7, showing that output-score drift can be substantial as well. The second architecture is worse, at $1{,}013$ of the $1{,}338$ bases it solved exactly (Table S7). The third violates the contract on 6.06% of its exactly-solved bases at n=16 (Table S8); it is a smaller number on a structurally different model, and §4 is where it does the most work.

Size controls. Every cell above extends the instance, so size extrapolation may amplify violations. Graph-level rates also grow with the number of protected edges even when per-edge rates are similar. Matching training to base size gives similar per-edge rates at $n = 16$ and $n = 64$; evaluating the $n = 16$ model on $n = 64$ bases instead raises that rate more than fivefold (Appendix G). Graph-level rates alone cannot distinguish these effects.

To test whether violations require composites larger than a training instance, we hold the *composite* at 16 vertices and vary the base/graft split (Table S5). The model still changes protected answers on

0.26–0.45% of the bases it solves exactly, with every interval clear of zero. The compiled arm is 0 there, at a maximum drift of exactly 0.0, bitwise.

A third point lies between. Train the same architecture at 64 and the same 128-node composite is twice its training size rather than eight times: the released model changes a protected label on 16.08% of the $9{,}155$ bases it solves exactly, six times the primary cell's supply, at 0.9987 in-distribution. Every step of §4 repeats there—cutting the CLS token moves the defect to 15.81%, masking attention to 1.47%, and partitioning to 0.00% on each arm's exactly solved subset. Together, these controls show that exceeding the training size is not necessary for a violation, while the largest rates occur under substantial size extrapolation. The 16-node composites still differ structurally from training graphs and are paired with smaller bases; they do not eliminate all distribution shift. Applied to these cells, the frozen rule returns STOP_METHOD_CLAIM, its threshold having been set for the primary cell: they support existence of the effect at smaller sizes, not the headline magnitude. The model trained at 64 clears the rule.

4 Diagnosing and enforcing non-interference

We intervene on each global channel at inference time, one at a time and then together, on the same weights.

Global-channel ablations have mixed effects. The backbone has an explicit global token that every token writes to and reads from. We pre-specified this token as a candidate source of interference. Because each arm solves different bases, we compare on the pairs all six solved and count repairs against breaks (Table S4, Holm-corrected over its 29 tests). Cutting the CLS token is significant in every cell that has one—the message-passing backbone does not—and *in no consistent direction*: $86/34$ repaired-to-broken on one GDT seed, $65/105$ on another. Localising the positional encoding barely moves them at $n{=}64$, significant in none of four cells after correction, the closest being 8 against 1, and is significantly worse at the training size ($52/144$). These results do not support removing CLS or localising the encoding as a reliable repair. They do not exclude either channel from carrying interference: an ablation can remove both useful and harmful dependencies, with effects that depend on the remaining paths.

Restricting cross-block attention has a larger effect. Restricting attention to within-block token pairs takes GDT from 75.70% to 11.07% of all pairs and SPD from 83.99% to 14.18%, with little change in accuracy ($0.9815 \to 0.9816$, $0.9806 \to 0.9807$). On the shared subset it repairs between 181 and 603 instances against at most 10 broken, one direction in every $n{=}64$ cell. This supports cross-block attention as a major route in these cells. It does not remove all computational paths between the blocks, because their articulation token remains shared.

The size-controlled cells sharpen that. There the same mask makes the defect *worse*—2.02% against a released 0.87% on the $8{+}8$ split, in the same direction on all three splits, and paired at the training size it breaks 205 instances against 136 repaired ($p < 10^{-3}$). Thus, the same structural intervention can improve or worsen violation rates across sizes. This is consistent with competing effects of removing cross-block dependence and perturbing a trained computation; the experiments do not separately identify their magnitudes. The mask is therefore a useful diagnostic, but it does not enforce the contract.

Block-membership masks retain shared-state paths. Applying all three masks at once leaves 6.98% (GDT) and 8.89% (SPD) all-pair violations. The tested mask allows attention between tokens sharing a block. An articulation vertex belongs to two blocks, so its state can receive information from one and transmit it to the other in subsequent layers (Figure 2). Such a mask does not structurally guarantee non-interference for arbitrary weights. The disjoint-union control supports this explanation: the mask-based construction has zero observed violations when no articulation token is shared, but fails on bridge grafts. This argument concerns masks that preserve those within-block connections, rather than all symmetric masks.

An asymmetric mask escapes the argument: forbid protected-side tokens from reading graft-side ones while allowing the reverse, and every attention path from the graft to a protected output is cut without duplicating a token. We ran it on the model trained at 64 and observe a 0.40% exact-

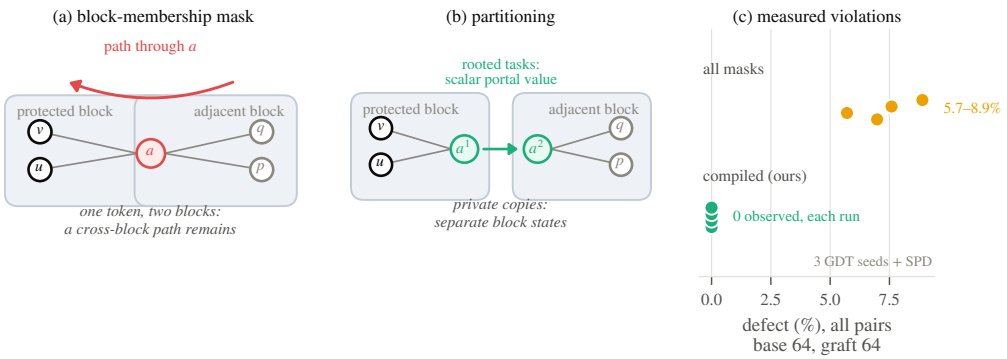

Figure 2: Shared states under block-membership masking. **(a)** Allowing all within-block attention pairs leaves a path through the shared articulation token a. The schematic shows two adjacent blocks. **(b)** Partitioning gives each block a private copy; rooted tasks pass only a scalar portal value. **(c)** Three GDT seeds and SPD at n=64: all three masks together still leave 5.7–8.9% of all pairs with violations; partitioning has zero observed violations in every run.

base defect, with maximum drift 14.5. Cutting attention paths alone leaves another possible route: positional encodings are still computed on the whole composite, so a protected token's input can already depend on the graft. The diagnostic also uses information unavailable to an unrooted MST predictor: which side of the bridge was designated as protected when constructing the pair. A rooted task does supply an orientation; §6 uses the source to order blocks.

Local message passing also violates the contract. The masks above suppress global channels in a model that has them. The message-passing backbone of §3 has none to suppress, and violates the contract anyway, on 6.06% of the bases it solves exactly at n=16. Cutting a token it does not have and localising an encoding it does not use each repair a single instance (Table S4); only partitioning reaches 0.00%, repairing 525 and breaking none. The bridge is a real edge over which the network sends messages, so locality alone does not enforce task-specific independence. With no edge between the parts, the same model is exactly 0.00% over 9,090 bases at maximum drift 3.3×10^{-5} in fp32. Violations also persist under the graft in fp32: 6.21% for the released arm, with no observed exact-base violations under partitioning. Thus, bf16 rounding alone does not explain the effect.

4.1 Blockwise execution

The task's own decomposition gives a sufficient construction for isolating protected computations.

Proposition 2 (Task decomposition)**.** *Minimum spanning forest and source-rooted shortest path each admit an exact factorization over the block-cut decomposition with a bounded portal state: for MST only block membership is required, and for SSSP one scalar per articulation vertex, its distance from the root.*

The proof is in Appendix A. This factorization preserves the information needed by the target task; it does not imply that a trained network retains its accuracy after its inputs are partitioned. Nor does the contract force every predictor to factorize in this way (Appendix A.4).

Construction. An iterative Tarjan pass first extracts biconnected edge blocks. Each block becomes an independent graph with local vertex indices and private copies of its articulation vertices. We then recompute its input features and positional encodings and score it with the same frozen network, batching independent blocks. Finally, scores are scattered back by edge identity and decoded as a spanning forest. Bridges are set symbolically because every spanning forest contains them; the network predicts all remaining edge scores. For rooted tasks, blocks are ordered from the source and a scalar predicted portal value is composed with local outputs (Appendix B).

For MST in exact deterministic arithmetic, an admissible graft leaves each old block's inputs and scores unchanged. The structured decoder also factorizes over these blocks, so old edge labels

Table 2: Consistency training at base size $n = 16$ with 16-node grafts. Training rows are three-seed means; brackets give the defect range. Defect and response comparisons use all pairs; exact-base rates have arm-specific denominators. Compiled uses the primary seed-14 checkpoint. The $n = 64$ columns distinguish structured decoding from the released raw protocol.

		defect %		should-chg	max		$n = 64$ F1	
method	λ	all pairs	exact base	%	drift	base F1	struct.	raw
ERM (matched batch)	–	2.91 [2.82, 2.97]	2.27	97.6	30.1	0.9984	0.9785	0.9615
soft consistency	0	0.94 [0.83, 1.01]	0.52	76.9	14.8	0.9987	0.9517	0.9381
soft consistency	1	0.74 [0.68, 0.81]	0.43	72.6	10.9	0.9985	0.9445	0.9318
soft consistency	10	0.56 [0.55, 0.56]	0.33	68.9	6.3	0.9985	0.9266	0.8413
compiled (ours)	–	**0.00**	**0.00**	**98.5**	0.0	0.9985	0.9821	–

remain unchanged. The construction does not guarantee correctness within a block: the compiled primary cell scores 0.9817 structured macro-F1, rather than solving MST exactly. Only bridge membership is supplied symbolically. On nontrivial blocks, membership still depends on learned scores, so partitioning does not replace the predictor with an exact MST solver.

Measured utility and numerical behaviour. We observe zero exact-base violations for every tested seed and architecture, and zero all-pair violations for both transformers (Table 1). Protected score drift is at most 0.125 in the reported cells; finite-precision execution therefore need not produce identical scores even when labels agree. The compiled MPNN has all-pair defects of 0.01% at $n = 16$ and 1.48% at $n = 64$, where its conditional estimate is excluded. Its scatter operations also show order-dependent numerical variation (Appendix E). Across the four $n = 64$ transformer cells, base F1 changes by 0.0001 to 0.0002 in either direction. The GDT response to answer-changing edits moves by at most 0.4 percentage points across seeds. These are empirical utility results, distinct from the structural guarantee, and show no sensitivity loss comparable to the training baseline below. Partitioning is also faster than dense attention from $n = 32$ on, host-side decomposition included (Appendix F).

5 Consistency training and sensitivity

Can training for the invariance achieve the same behaviour? We implement a consistency baseline in the style of Bevilacqua et al. (2023): a penalty of weight λ on the discrepancy between predictions on G and G'_{keep}, over exactly the extension family used in evaluation. Both the base and grafted graph also enter the task loss, so $\lambda = 0$ is an augmentation-only control. Concretely, the task term averages cross-entropy on the two views, and the consistency term is the symmetric KL divergence between their class probabilities on aligned original edges. It is averaged over those edges; new graft edges enter the task loss but not the consistency term. The objective thus directly penalises the evaluated answer-preserving discrepancy without using answer-changing control pairs in training.

Table 2 separates the augmentation and penalty effects. Each training condition uses three seeds; ERM and consistency training are matched on batch size and optimiser steps. Their defect ranges do not overlap between successive conditions. The compiled row is a reference using the primary seed-14 checkpoint from §3, rather than a recompilation of the matched ERM runs.

Augmentation. The defect falls $2.91\% \to 0.94\%$ between ERM and λ=0—a factor of 3.1, the largest movement in the table—but λ=0 switches the consistency term off entirely. That belongs to the augmentation: seeing each base beside a grafted copy, both entering the task loss. Response to truth-changing edits falls $97.6\% \to 76.9\%$ across the same gap, at zero penalty weight. On exactly-solved bases the same gap reads $2.27\% \to 0.52\%$, a factor of 4.3. The different reduction illustrates why we compare all-pair rates: the models solve different subsets of bases.

Consistency penalty. Sweeping λ over $0 \to 1 \to 10$ reduces the defect 0.94% to 0.56%, a factor of 1.7, and reduces response 76.9% to 68.9%. Maximum output drift falls $2.3\times$ and size generalisation degrades. Three more weights on seed 14 give a six-point trade-off curve (Figure 1c): the

all-pair defect falls monotonically across all six, while response turns back up at the largest weight. Base-distribution accuracy stays nearly constant throughout (0.9976–0.9988), so this accuracy measure does not reveal the trade-off. Together, augmentation and regularisation reduce defect about five-fold and sensitivity by twenty-nine percentage points; none of the tested weights enforces exact invariance.

Size generalisation degrades too, and more under the backbone's released protocol, the one comparable to its published score, than under SepBench's structured decode, which repairs much of the raw damage: from 0.9615 to 0.8413 read raw and from 0.9785 to 0.9266 read structurally.

The joint reduction in violations, response and drift is consistent with reduced responsiveness, rather than evidence that the model has learned to ignore only irrelevant changes. This is an empirical result for the tested training objective and perturbation family, not an unavoidable cost of every invariance objective. The compiled reference combines zero observed violations with high response, showing that the observed trade-off is not imposed by the task itself. Without the should-change column, the same training experiment would appear to improve invariance while preserving utility.

6 Extension to shortest paths

The contract is not specific to spanning trees. Single-source shortest path over the min-plus semiring admits the same decomposition, with the certificate reduced to one scalar per portal: the root-to-articulation distance. Table S2 evaluates a global and a blockwise predictor, *trained separately* with the same architecture and budget. This differs from the frozen-weight MST interventions in §4; it tests whether the decomposition supports learning on a second task. A global model at high in-distribution parent accuracy changes a provably-unchanged predecessor on 6.5% and 6.6% of exactly-solved bases; the compiled model is exactly zero on all three seeds, on all pairs as well. Among answer-changing control pairs whose base it solves exactly, it responds on 93.6–98.0% and predicts the correct composite-graph predecessors at all distance-affected vertices on 86.8–94.7%. Both percentages use the same exact-base denominator; the latter is not correctness conditional on responding.

The SSSP response is scored over every distance-affected vertex, a proxy for sensitivity over the affected region (Appendix C). The decomposition also supports max-min, but our widest-path experiment was not informative (§8).

7 Related work

Invariance and sensitivity. Causal-invariance objectives for answer-preserving augmentation include Bevilacqua et al. (2023), following Mitrovic et al. (2021); Rodionov & Prokhorenkova (2023) study hint-free invariance regularisation. The general formulation of Arjovsky et al. (2019) already identifies the null representation as a degenerate invariant solution. Behavioural testing pairs invariance with directional expectations (Ribeiro et al., 2020), and invariance to relevant edits is a known adversarial failure (Tramèr et al., 2020). We instantiate this distinction with an exact task oracle: a theorem certifies the answer-preserving family, and oracle rejection certifies each answer-changing control. Our consistency baseline measures the resulting trade-off in neural graph algorithms; it is not a new regulariser.

Provably correct neural execution. Rodionov & Prokhorenkova (2025) restrict node and edge states to a finite set and prove the learned algorithm correct on any input; Qharabagh et al. (2026) learn local instructions and prove exact execution with high probability under bounded degree and finite precision. These results certify task correctness. Our contract instead restricts which input changes may affect an output, even when that output is wrong. The guarantees are complementary: an exactly correct executor satisfies the contract, while the diagnostic is useful for high-accuracy models that remain imperfect.

Biconnectivity in graph learning. Blocks, cut vertices and SPQR trees are established tools: Zhang et al. (2023) separate GNN classes by biconnectivity, Dimitrov et al. (2023) decompose planar graphs, and Chen et al. (2026) restrict higher-order interactions to biconnected components.

Our decomposition uses the same classical structure and a standard Tarjan pass. The distinction is its purpose: we use independently executed blocks to enforce an input-dependence restriction, and evaluate that restriction alongside task accuracy and sensitivity.

Structured attention masks. Structural restrictions on graph attention form an existing family, including Li et al. (2025). Our tested symmetric block-membership mask leaves attention paths through shared articulation tokens. This explains why that mask does not enforce the contract for arbitrary weights; it is not an impossibility result for structured masks in general. An asymmetric diagnostic removes those paths using the known protected side, but retains a small measured residual (§4).

Graph transformers and certified robustness. Our backbones follow Ying et al. (2021); Rampášek et al. (2022); Ma et al. (2023), and our primary model is the released GDT of Stoll et al. (2025). Certified robustness to graph modification (Li & Wang, 2025) bounds worst-case output change under a perturbation budget. Our contract concerns exact invariance under a restricted family whose extension size is unbounded.

8 LIMITATIONS

Scope. We study undirected graphs, positive distinct weights and single-bridge extensions. A 2-connected base is one block and receives no internal decomposition; a bridge graft is still isolated in separate blocks. The computational benefit depends on the block structure. Multi-vertex separators would require richer portal states and are not evaluated.

Widest path. Max-min admits the same task decomposition, and the compiled predictor has 0 of 403 non-attachment defects at 0.0 value drift. However, local predictive accuracy and the answer-changing control were too weak to support an informative learned comparison. In particular, its shortcut control used the shortest-path oracle and did not certify max-min answer changes. We therefore make no empirical generalisation claim for this task.

Excluded cells. The pre-specified minimum of 500 exactly-solved bases excludes the conditional estimate for the message-passing backbone at $n = 64$: it solves 183 of $10{,}000$ bases. We report its $n = 16$ result on $9{,}110$ exact bases and its all-pair rates, to which that threshold does not apply (§4.1). At $n = 128$ no model solves a base exactly. The SSSP global baseline covers two of three seeds; seed 15 diverged twice.

Numerics. The deterministic construction gives an exact structural guarantee, but its implementation uses bf16 autocast and can introduce numerical differences. We report discrete violations and continuous drift separately. Compiled protected drift is ≤ 0.125 in the included cells and 0.47 including the excluded message-passing cell. Zero observed discrete violations therefore do not imply bitwise equality; the message-passing all-pair exceptions are reported in §4.1.

Generality. The benchmark relies on an exact task oracle and a certified answer-preserving edit family. Our results do not establish how often comparable contracts exist in tasks without this structure, or whether the measured training trade-off persists across other objectives and perturbation families.

9 CONCLUSION

Single-bridge extensions expose a dependence that aggregate accuracy does not test: accurate neural graph predictors can change answers that are provably unchanged. Frozen-weight interventions identify cross-block attention as a major route in the tested transformers, while blockwise execution enforces the MST contract structurally and removes their observed discrete violations. Consistency training reduces violations alongside response to relevant edits, with much of both changes already arising from augmentation. Evaluating accuracy, non-interference and answer-changing sensitivity together reveals these distinct behaviours.

AI USE STATEMENT

This work is ours: the research idea, the conceptual framework, the experimental design, the decision rules, the survey of related work, and the text of this paper. Generative AI was used in a supporting role throughout, always to a specification we set, and we report that against the task list in ICLR's AI policy.

Of the tasks requiring disclosure, it assisted us by implementing methods to our specification, auditing that implementation and repairing logic errors, criticising our experimental design and analysis after we had fixed them, writing up proofs we had worked out, and commenting on results we then interpreted. It was not used to develop the conceptual framework, to propose or refine the hypothesis, to generate data, to clean or reformat data, for qualitative analysis, or for translation. Among the tasks whose disclosure is recommended, it likewise assisted to our direction: implementing and debugging experiment code, helping with the figures and tables, and copy-editing the manuscript.

We have reviewed all AI-assisted work. No experimental result passed through it: every number comes from a result file the evaluation scripts wrote, generated into the tables and figures by script and re-derived from the data by a second one, so a stale sentence fails the build (Appendix D). Every citation was checked against the publisher record and dropped where we could not verify it. The claims, the pre-specified decision rule, the scope and exclusion decisions, and the judgement of what to report—including results against our own hypotheses—are ours, and we take responsibility for this paper.

REPRODUCIBILITY STATEMENT

The frozen decision rule, the provenance mechanism, and the loader-enforced reproducibility check that caught a silently mis-loaded checkpoint are described in Appendix D; backbone configuration, hyperparameters, the numerical regime and the compute budget are in Appendix C; proofs of every formal statement, with their domain conditions stated rather than assumed away, are in Appendix A.

The code is available at `https://anonymous.4open.science/r/sepnoninf-92B1/`: the benchmark generators and their oracle, the separator compiler, the backbone adapter with each global channel separately ablatable, and the evaluation entry points. No dataset is needed—every graph comes from a seeded deterministic generator, so fixing the seed fixes the data.

REFERENCES

Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*, 2019.

Beatrice Bevilacqua, Kyriacos Nikiforou, Borja Ibarz, Ioana Bica, Michela Paganini, Charles Blundell, Jovana Mitrovic, and Petar Veličković. Neural algorithmic reasoning with causal regularisation. In *International Conference on Machine Learning (ICML)*, 2023.

Rongqin Chen, Fan Mo, Pak Lon Ip, Shenghui Zhang, Dan Wu, Ye Li, and Leong Hou U. Connectivity-guided sparsification of 2-FWL GNNs: Preserving full expressivity with improved efficiency. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2026. The Co-Sparsify method. arXiv:2511.12838.

Edsger W. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1: 269–271, 1959.

Radoslav Dimitrov, Zeyang Zhao, Ralph Abboud, and İsmail İlkan Ceylan. PlanE: Representation learning over planar graphs. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Joseph A. Goguen and José Meseguer. Security policies and security models. In *IEEE Symposium on Security and Privacy*, 1982.

Weihua Hu, Bowen Liu, Joseph Gomes, Marinka Zitnik, Percy Liang, Vijay Pande, and Jure Leskovec. Strategies for pre-training graph neural networks. In *International Conference on Learning Representations (ICLR)*, 2020.

Joseph B. Kruskal. On the shortest spanning subtree of a graph and the traveling salesman problem. *Proceedings of the American Mathematical Society*, 7(1):48–50, 1956.

Chenyang Li, Jinsong Chen, John E. Hopcroft, and Kun He. DAM-GT: Dual positional encoding-based attention masking graph transformer for node classification. *arXiv preprint arXiv:2505.17660*, 2025.

Jiate Li and Binghui Wang. AGNNCert: Defending graph neural networks against arbitrary perturbations with deterministic certification. In *USENIX Security Symposium*, 2025. arXiv:2502.00765.

Liheng Ma, Chen Lin, Derek Lim, Adriana Romero-Soriano, Puneet K. Dokania, Mark Coates, Philip Torr, and Ser-Nam Lim. Graph inductive biases in transformers without message passing. In *International Conference on Machine Learning (ICML)*, 2023.

Jovana Mitrovic, Brian McWilliams, Jacob Walker, Lars Buesing, and Charles Blundell. Representation learning via invariant causal mechanisms. In *International Conference on Learning Representations (ICLR)*, 2021.

Muhammad Fetrat Qharabagh, Artur Back de Luca, George Giapitzakis, and Kimon Fountoulakis. Learning to execute graph algorithms exactly with graph neural networks. *arXiv preprint arXiv:2601.23207*, 2026.

Ladislav Rampášek, Michael Galkin, Vijay Prakash Dwivedi, Anh Tuan Luu, Guy Wolf, and Dominique Beaini. Recipe for a general, powerful, scalable graph transformer. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. Beyond accuracy: Behavioral testing of NLP models with CheckList. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2020.

Gleb Rodionov and Liudmila Prokhorenkova. Neural algorithmic reasoning without intermediate supervision. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Gleb Rodionov and Liudmila Prokhorenkova. Discrete neural algorithmic reasoning. In *International Conference on Machine Learning (ICML)*, 2025. arXiv:2402.11628.

Timo Stoll, Luis Müller, and Christopher Morris. Generalizable insights for graph transformers in theory and practice. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. Spotlight. arXiv:2511.08028.

Robert Tarjan. Depth-first search and linear graph algorithms. *SIAM Journal on Computing*, 1(2): 146–160, 1972.

Florian Tramèr, Jens Behrmann, Nicholas Carlini, Nicolas Papernot, and Jörn-Henrik Jacobsen. Fundamental tradeoffs between invariance and sensitivity to adversarial perturbations. In *International Conference on Machine Learning (ICML)*, 2020.

Petar Veličković and Charles Blundell. Neural algorithmic reasoning. *Patterns*, 2(7), 2021.

Petar Veličković, Adrià Puigdomènech Badia, David Budden, Razvan Pascanu, Andrea Banino, Misha Dashevskiy, Raia Hadsell, and Charles Blundell. The CLRS algorithmic reasoning benchmark. In *International Conference on Machine Learning (ICML)*, 2022.

Edwin B. Wilson. Probable inference, the law of succession, and statistical inference. *Journal of the American Statistical Association*, 22(158):209–212, 1927.

Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *International Conference on Learning Representations (ICLR)*, 2019.

Chengxuan Ying, Tianle Cai, Shengjie Luo, Shuxin Zheng, Guolin Ke, Di He, Yanming Shen, and Tie-Yan Liu. Do transformers really perform badly for graph representation? In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

Bohang Zhang, Shengjie Luo, Liwei Wang, and Di He. Rethinking the expressive power of GNNs via graph biconnectivity. In *International Conference on Learning Representations (ICLR)*, 2023.