VICOB: PERCEPTUAL-TO-SYMBOLIC BINDING WITH SELECTIVE REBINDING

Anonymous authors
Paper under double-blind review

ABSTRACT

Multimodal editing systems must translate perceptual intent into exact symbolic references that downstream editors can execute. Even when a model identifies the intended object, it may emit an invalid or valid-but-wrong reference. Correcting such errors requires not only recovering a plausible alternative, but also deciding *whether the existing binding should be overwritten*.

We introduce VICOB, a selective-rebinding interface that separates recovery from overwrite authorization. ViCoB retrieves an operation-compatible alternative from executable source, but replaces the incumbent only when inference-visible evidence supports intervention. Under fixed retrieval, where all policies receive the same recovered alternative, Always-rerank repairs more errors than ViCoB (230 vs. 188) but causes substantially more regressions (85 vs. 13), yielding lower exact binding (63.8% vs. 66.7%). Across 1,750 cases and three frozen multimodal backbones, ViCoB improves exact binding from 49.1% to 64.4%.

Downstream and transfer experiments further show that corrected bindings can improve executable editing and that selective rebinding extends across SVG and DOM. These results support treating overwrite authorization as a distinct, measurable decision between multimodal perception and symbolic execution.

1 INTRODUCTION

Visual editing requires bridging a mismatch between perceptual intent and executable state. Users refer to objects through appearance, text, layout, or relations, whereas downstream editors act on exact symbolic references. An instruction such as "make the lower blue label bold" must ultimately bind to one SVG text node, and "connect the two highlighted boxes" to an ordered endpoint pair. A multimodal model may understand the intended target yet emit an absent identifier, a visually similar sibling, or a valid but incorrect executable reference. Once emitted, this reference becomes the symbolic anchor consumed by the editor.

Correcting such errors involves two distinct decisions: *recovery*—which executable alternative is best supported—and *authorization*—whether that alternative should overwrite the incumbent. The second decision is not resolved by candidate ranking alone. A recovered alternative may repair an incorrect incumbent, but it may also displace one that was already correct. Executable validity is likewise insufficient: a valid reference can still denote the wrong object. Rebinding is therefore a retain-or-replace intervention with an asymmetric repair–regression tradeoff.

We introduce VICOB, a deterministic selective-rebinding interface that separates recovery from overwrite authorization. ViCoB projects operation-compatible candidates from executable source and ranks a top alternative, but adopts that alternative only when inference-visible evidence supports intervention. This evidence includes executable validity, identifier regime, type compatibility, and candidate scores derived from the current source and rendering. Otherwise, ViCoB preserves the canonicalized incumbent. The method uses no gold targets, patches, or execution outcomes at inference.

The key experiment isolates authorization from recovery. In a case-disjoint fixed-retrieval replay, all policies receive the same candidate inventory, scores, VLM preference, and retrieved alternative; they differ only in whether they retain or replace the incumbent. Always-rerank repairs more errors than ViCoB (230 vs. 188), but causes substantially more regressions (85 vs. 13), yielding lower

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.

exact binding (63.8% vs. 66.7%). Thus, recovering more errors does not imply a better overwrite policy: the value of a proposed correction depends also on the risk of replacing an incumbent that may already be correct.

Supporting experiments characterize this decision and its consequences. Inference-visible evidence predicts incumbent reliability and overwrite benefit before intervention, while learned authorization policies exhibit the same underlying repair–regression tradeoff. Across the full 1,750-case evaluation with three frozen multimodal backbones, ViCoB improves exact binding from 49.1% to 64.4%. Downstream experiments show that improved bindings translate into better editing when the editor can reliably consume the selected reference, and transfer experiments test the same selective-rebinding principle across SVG and DOM, with an exploratory WebMMU evaluation on real-web code editing.

Contributions. We make three contributions. **First**, we formulate executable binding correction as a retain-or-replace intervention, separating candidate recovery from overwrite authorization under asymmetric repair–regression risk. **Second**, we introduce a fixed-retrieval evaluation that isolates authorization by holding the recovered alternative and its evidence fixed, and show that aggressive reranking can repair more errors yet produce worse final binding because of regressions. **Third**, we instantiate selective rebinding with a deterministic, inference-visible policy and evaluate its diagnostics, downstream consumption, and transfer across executable representations.

2 Related Work

Visual grounding and referring expressions. Referring-expression comprehension and open-vocabulary grounding map language descriptions to visual regions, masks, or semantic objects, as in MDETR, GLIP, and Grounding DINO (Kamath et al., 2021; Li et al., 2022; Liu et al., 2024). Related work on GUI grounding extends this problem to interactive interfaces, where language instructions must be associated with actionable screen elements (Qian et al., 2024; Lei et al., 2025; Yang et al., 2025). These methods primarily ask which visual element a language expression denotes. Executable editing introduces a subsequent interface: the perceived referent must be bound to the exact symbolic object consumed by the editor. ViCoB studies this perceptual-to-symbolic boundary given a frozen multimodal prediction, rather than proposing a new visual grounding model.

Structured generation and executable representations. Constrained decoding and structured generation enforce syntactic or schema requirements through incremental parsing, grammar constraints, or guided generation (Scholak et al., 2021; Ugare et al., 2025; Willard & Louf, 2023). Such constraints can ensure that an executable argument is well formed or belongs to an admissible space, but validity does not determine denotation: multiple valid identifiers may correspond to different executable objects. ViCoB therefore treats executable source not only as an output constraint but also as evidence for binding perceptual intent to symbolic identity. Canonical identifiers, types, text, labels, relations, and other source-visible structure define candidate compatibility and provide evidence for retaining or revising an existing binding.

Reranking, verification, and selective prediction. Rerankers and verifiers compare alternative predictions, while selective prediction and abstention study when a model should act versus withhold a prediction, and learning-to-defer extends this decision to delegation (Geifman & El-Yaniv, 2017; Madras et al., 2018). ViCoB instead starts from an already executable commitment. Its action is neither to emit the top-ranked candidate nor to abstain: it selects between preserving the incumbent program reference and replacing it with a retrieved reference. This makes the decision counterfactual and asymmetric—the same replacement can repair a wrong incumbent or destroy a correct one. Candidate-ranking quality is therefore insufficient to evaluate the policy; the relevant outcomes are repairs, regressions, coverage, and final exact binding under shared retrieved evidence. The deterministic rule is a transparent intervention specification, not a claim that learned predictors are unnecessary: it makes explicit which inference-visible evidence and cost-sensitive action boundary are being tested alongside learned alternatives.

Visual agents and executable interfaces. Visual and GUI agents connect interface perception to actions such as clicking, typing, navigation, and tool execution (Baechler et al., 2024; Hong et al., 2024; You et al., 2024; Lu et al., 2024; Cheng et al., 2024). Recent GUI-grounding work further

emphasizes accurate localization of actionable interface elements as a prerequisite for reliable agents (Qian et al., 2024; Lei et al., 2025; Yang et al., 2025). Agent benchmarks evaluate these capabilities in longer web and desktop trajectories (Deng et al., 2023; Zhou et al., 2024; Koh et al., 2024; Drouin et al., 2024; Xie et al., 2024), while tool-use work studies the generation of executable calls and arguments (Schick et al., 2023; Li et al., 2023). ViCoB isolates a narrower boundary shared by these systems: converting a perceptually specified target into an exact executable reference and deciding when that symbolic binding should be revised. ExBind provides the controlled executable-binding setting (Wang et al., 2026), and SVGEditBench V2 provides source SVG cases used in the SEB conversion (Nishina & Matsui, 2025). Our SVG, DOM, and exploratory real-web evaluations test this binding interface across executable representations rather than claiming general long-horizon agent competence.

3 METHOD

3.1 EXECUTABLE BINDING CORRECTION

We adopt the executable-reference binding setting of ExBind (Wang et al., 2026). An instance contains a rendering I, editable program C, instruction x, operation o, and a frozen model prediction b_{raw}. The target $b^\star$ is the exact executable reference compatible with o; for an edge operation, it is an ordered endpoint pair (e_s, e_t). We call b_{raw} the *incumbent binding*. Exact canonical equality is required because a valid but incorrect identifier directs the editor to the wrong symbolic object.

ViCoB separates recovering a top alternative from authorizing that alternative to modify executable state. Given the operation-compatible candidate set $E_o(C)$ and evidence score S, target recovery proposes the top-scoring alternative

$$e_1 = \arg \max_{e \in E_o(C)} S(e). \tag{1}$$

A selective rebinding policy $A(z) \in 0, 1$ therefore determines whether the retrieved alternative is authorized to replace the incumbent:

$$\hat{b} = \begin{cases} e_1, & A(z) = 1, \\ \mathrm{canon}(b_{\mathrm{raw}}), & A(z) = 0, \end{cases} \tag{2}$$

where z contains only inference-visible pre-decision evidence. The deterministic policy exposes a fixed, inference-visible action boundary, allowing candidate ranking and the consequence of replacement to be evaluated separately.

3.2 INTERFACE CONTRACT

ViCoB may use only $(I, C, x, o, b_{\mathrm{raw}})$. Gold references, gold patches, and execution outcomes are used only for offline evaluation. Candidate attributes are parsed from executable source, and experimental perturbations modify only inference-visible source or rendering fields. No gold outcome is available to either target recovery or overwrite authorization.

Figure 1 summarizes this inference boundary: executable source exposes evidence for both recovery and overwrite authorization, whereas gold references and execution outcomes remain evaluation-only.

3.3 TARGET RECOVERY FROM EXECUTABLE EVIDENCE

Candidate projection. ViCoB deterministically parses the editable program into canonical identifiers with primitive type, normalized text and labels, bounding box, relations, and existing graph structure. Operation compatibility filters these objects into $E_o(C) = e_{i\,i=1}^{n}$. This defines the executable candidates that may realize the intended referent.

Evidence score. For each $e \in E_o(C)$, ViCoB constructs inference-visible evidence

$$\mathcal{E}(e \mid x, I, C, o, b_{\mathrm{raw}}), \tag{3}$$

including text and label correspondence, frozen VLM preference, type and operation compatibility, applicable geometry or relations, and consistency with the incumbent.

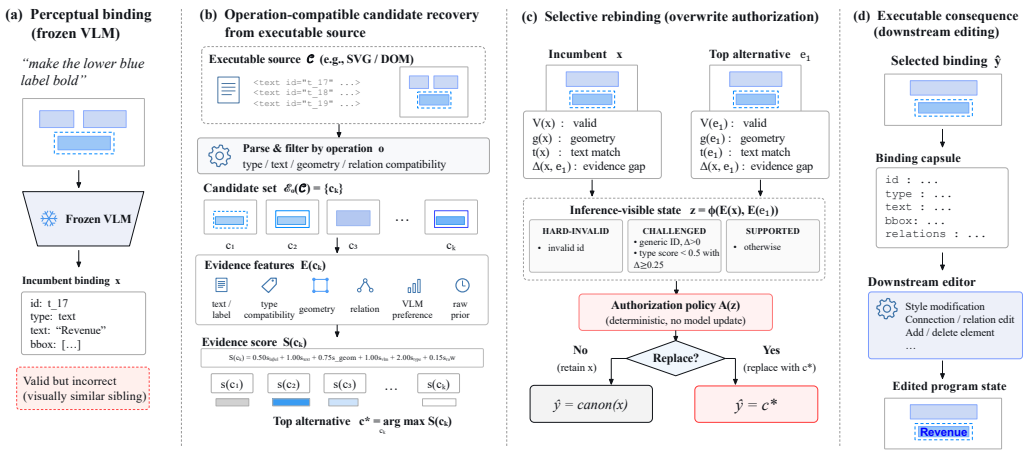


Figure 1: **ViCoB separates candidate recovery from overwrite authorization.** A frozen VLM produces an incumbent symbolic binding x, which may be executable yet incorrect (a). From the executable source, ViCoB recovers an operation-compatible top alternative e_1 using inference-visible evidence (b). A separate authorization policy then decides whether to retain x or replace it with e_1 (c), and the selected binding $\hat{y}$ is exposed to the downstream editor as a binding capsule (d). This decomposition allows overwrite authorization to be evaluated independently under fixed retrieval, including its repair–regression tradeoff.

For a single-target operation, the fixed score is

$$S(e) = 0.50s_{\text{label}}(x,e) + 1.00s_{\text{text}}(x,e) + 0.75s_{\text{geom}}(x,e,I) + 1.00s_{\text{vlm}}(x,e,I) + 2.00s_{\text{type}}(o,e) + 0.15s_{\text{raw}}(b_{\text{raw}},e). \quad (4)$$

Text and label terms measure referential correspondence; geometry captures explicit spatial cues; type enforces compatibility with the requested operation; the VLM term is a frozen candidate preference; and the Raw term favors the incumbent in close comparisons. The same historically frozen configuration is used across all primary Stage-A suites and backbones; its provenance is given in Appendix C.1.

For connection operations, ViCoB jointly selects an ordered pair:

$$(e_s, e_t) = \arg\max_{e_i, e_j} \left[S_s(e_i) + S_t(e_j) + w_p g(e_i, e_j) - 3\mathbb{1}[e_i = e_j] - 2\mathbb{1}[\text{existing}(e_i, e_j)]\right], \quad (5)$$

where g is the deterministic pair-geometry bonus and $w_p = 1.5$ when enabled. Type-incompatible candidates are removed before scoring. Single-target operations use a stable argmax, and set-valued style requests use deterministic top-k selection after the same compatibility filter.

3.4 Incumbent Assessment and Overwrite Authorization

Target recovery proposes e_1, but candidate rank alone does not determine whether replacing b_{raw} is beneficial. ViCoB therefore assesses an executable evidence state for the incumbent before authorizing replacement. Let $V(b_{\text{raw}})$ denote executable validity, $g = \mathbb{1}[\text{generic}(b_{\text{raw}})]$ the generic-identifier indicator, t_{raw} the incumbent type score, and $\Delta = S(e_1) - S(b_{\text{raw}})$ for valid incumbent candidates. These inference-visible variables define three pre-decision states:

$$\text{HARD-INVALID}: \quad \neg V(b_{\text{raw}}), \quad (6)$$

$$\text{CHALLENGED}: \quad V(b_{\text{raw}}) \wedge \left[(g = 1 \wedge \Delta > 0) \vee (t_{\text{raw}} < 0.5 \wedge \Delta \geq 0.25)\right], \quad (7)$$

$$\text{SUPPORTED}: \quad V(b_{\text{raw}}) \wedge \neg\text{CHALLENGED}. \quad (8)$$

These states are assigned entirely from inference-visible evidence, before gold correctness or the consequence of intervention is observed. Validity asks whether the incumbent can be executed; type evidence asks whether its primitive is compatible with the requested operation; identifier regime and score margin ask whether source and perceptual evidence support a recovered alternative strongly

enough to challenge it. The states therefore characterize whether the incumbent is invalid, challenged by available executable evidence, or sufficiently supported to retain.

The reported ViCoB authorization policy uses the fixed correction set

$$\begin{aligned} \mathcal{C}_{\text{inv}} &= \{\neg V(b_{\text{raw}})\}, \\ \mathcal{C}_{\text{gen}} &= \{V(b_{\text{raw}}),\, g = 1,\, \Delta > \tau_g\}, \\ \mathcal{C}_{\text{margin}} &= \{V(b_{\text{raw}}),\, \neg(g = 1 \wedge \Delta > \tau_g),\, t_{\text{raw}} < \tau_t,\, \Delta \geq \tau_r\}, \\ \mathcal{C} &= \mathcal{C}_{\text{inv}} \cup \mathcal{C}_{\text{gen}} \cup \mathcal{C}_{\text{margin}}, \end{aligned} \tag{9}$$

with $\tau_g = 0$, $\tau_t = 0.5$, and $\tau_r = 0.25$. The authorization decision in Eq. 2 is therefore

$$A(z) = \mathbb{1}[z \in \mathcal{C}]. \tag{10}$$

Every instance outside $\mathcal{C}$ returns $\hat{b} = \text{canon}(b_{\text{raw}})$ exactly. Thus, retrieval of a higher-scoring alternative does not automatically trigger replacement: the alternative is adopted only within the declared correction set. This fallback preserves a supported incumbent but does not assume that every authorized replacement is correct. For endpoint pairs, the reported configuration retains every valid Raw pair and rebinds only invalid pairs. The score weights and gate constants are historically frozen, hand-specified ordinal choices; they are not theoretically derived or fitted on held-out overwrite outcomes (Appendix C.1).

Frozen rebind-confidence diagnostic. For the overwrite diagnostic only, we also report the existing candidate-confidence scalar $\kappa(e_1) = \min_r \left[S_r(e_{1,r}) - \max_{e \neq e_{1,r}} S_r(e)\right]$, where r indexes emitted binding roles. Thus, κ is the minimum selected-vs-runner-up score margin for a single target or endpoint pair. It is computed from the same frozen candidate scores already used by ViCoB; it does not retrain, rerank, or modify the reported authorization policy. We use κ only to test whether inference-visible confidence ranks the observed consequence of overwriting the incumbent, not to estimate a calibrated causal damage probability.

Candidate-prompt setting. When the frozen model selects directly from an executable candidate list, ViCoB preserves a valid object choice and applies the same authorization policy to group and endpoint requests.

3.5 Binding capsules for downstream execution

Stage-B evaluates whether a downstream editor can consume the selected executable binding. ViCoB serializes inference-visible binding information into a compact capsule containing the canonical identifier, primitive type, normalized text, bounding box, and relation tags. All experimental arms use the same image, instruction, executable candidate inventory, model checkpoint, decoding procedure, executor, and renderer. *Raw* receives the inventory and instruction directly; *Typed* additionally receives type, endpoint-order, and patch-schema constraints; and *ViCoB* receives the Typed context plus the frozen predicted binding capsule.

Stage-B therefore evaluates the layered path executable binding $\rightarrow$ binding consumption / patch decoding $\rightarrow$ execution, separating binding quality from the downstream editor's ability to convert that binding into a valid patch.

4 Experiments

4.1 Evaluation protocol

Our main experiment asks whether overwrite authorization matters independently of target recovery. We isolate this decision through a fixed-retrieval replay: every policy receives the same frozen candidate inventory, candidate scores, VLM preference, and retrieved top alternative, and differs only in whether it retains or replaces the incumbent. Supporting experiments first establish that executable evidence can recover useful alternatives and that overwrite risk is observable before intervention, then evaluate downstream consumption and transfer across executable representations. Target-recovery decomposition, parameter sensitivity, complete learned-policy operating points, cost-sensitive utilities, and detailed transfer results are reported in the appendix.

Table 1: **Executable binding across primary, transfer, and downstream settings.** Primary rates pool three frozen backbones over 1,750 underlying cases; all differences are computed from unrounded values.

Setting	Scope	Cases	Raw	ViCoB	Δ	Rep.	Regr.	Valid (%)
Primary Stage-A	*Frozen ViCoB configuration*							
EVG-Bench	semantic	200	73.7	**81.8**	+8.2	50	0	–
EVG-Perturbed	semantic	800	64.9	**77.3**	+12.4	330	30	–
Novel-SVG	semantic	50	54.0	**54.7**	+0.7	2	0	–
SEB-hard	generic	100	24.3	**44.3**	+20.0	70	10	–
SEB-random	generic	100	24.7	**53.3**	+28.7	95	10	–
SEB-perturbed	generic	500	23.3	**43.9**	+20.6	355	50	–
All primary	mixed	**1,750**	**49.1**	**64.4**	**+15.3**	**902**	**100**	–
Transfer	*Separately scoped adapters*							
GLM-4.6V SVG	backbone	450	41.6	**52.9**	+11.3	52	1	100.0
SVGEditBench V2	source-disjoint	100	16.0	**20.0**	+4.0	4	0	100.0
Novel-SVG adapted	schema	300×3	57.9	**67.8**	+9.9	89	0	100.0
DOM	representation	500×2	74.6	**99.2**	+24.6	246	0	100.0
WebMMU root	exploratory	95	27.4	**33.7**	+6.3	15	9	100.0
Downstream consumption	*960 matched cases/editor; frozen execution-validity guard*							
Qwen2.5-VL-3B	editor	960	58.8	**64.9**	+6.1	111	52	89.0
Qwen2.5-VL-7B	editor	960	79.6	**84.8**	+5.2	65	15	99.9
Qwen3-VL-4B	editor	960	85.1	**86.5**	+1.4	24	11	100.0

The primary Stage-A evaluation contains 1,750 underlying cases evaluated with three frozen multimodal backbones, yielding 5,250 paired model–case observations. For fixed-retrieval evaluation, cases are split by `dataset::base_sample_id` into 598 calibration and 152 held-out test cases (4,203 and 1,047 model–case observations). Thresholds and learned authorization policies use calibration cases only, with features restricted to inference-visible pre-decision evidence; gold bindings and overwrite outcomes are never available at inference. Paired uncertainty intervals resample underlying case clusters, and intervention precision uses Wilson intervals. Dataset construction, freezing, and evaluation formulas are detailed in Appendices B, C.1, and C.4.

Recipe provenance. The primary ViCoB configuration is a historically frozen, hand-specified ordinal recipe shared across all primary suites and backbones, with no split-specific or model-specific overrides. Its coefficients were not fitted to the reported primary labels. Appendix C.1 records the development and freeze boundary (Table 13) and reports coarse post-freeze sensitivity replays (Table 14). Transfer settings use representation-specific schema/interface adaptations, while risk-gate margins and learned authorization policies use held-out calibration only and do not modify the primary recipe.

4.2 Recovering executable bindings

We first establish that executable source provides useful evidence for recovering symbolic references. On the primary Stage-A matrix, ViCoB improves exact binding from 49.1% to 64.4% (+15.3 points), with 902 repairs and 100 regressions (Table 1). An executable-input verifier reaches 58.9% while requiring an additional 0.73-second VLM call per model–case observation; further recovery controls are reported in Appendix C.3 (Table 16).

The suite-level pattern is shown in Figure 3(a), with backbone detail in Appendix Figure 6(a). Recovery alone does not isolate the overwrite decision, which we test under fixed retrieval in Section 4.4.

4.3 DIAGNOSTIC: IS OVERWRITE RISK OBSERVABLE BEFORE INTERVENTION?

Table 2 tests whether overwrite authorization has measurable pre-decision structure. State assignment uses only frozen validity, Raw identifier presence, candidate scores, type evidence, identifier regime, and artifact availability; gold correctness and overwrite outcome are joined only after assignment. Raw is wrong in 94.3% of HARD-INVALID states and 75.8% of CHALLENGED states, whereas it is correct in 76.3% of SUPPORTED states. Thus, inference-visible evidence contains information about whether preserving the incumbent is plausible.

We next ask whether the consequence of replacing the incumbent is predictable before intervention. The existing rebind-confidence scalar is the minimum selected-vs-runner-up candidate-score margin across emitted binding roles. It is not a learned damage estimator and does not modify the reported ViCoB policy. On the 503 held-out model–case observations for which the frozen top alternative differs from Raw, it predicts a beneficial overwrite—a repair rather than a regression or no-benefit replacement—with AUROC 0.661 and AUPRC 0.591. This exceeds the top-minus-Raw score margin (0.494 AUROC) and frozen VLM margin (0.443).

Table 2: Pre-overwrite diagnostic evidence. All quantities use frozen inference-visible states and candidate scores before correctness is joined. A beneficial overwrite is a repair rather than a regression or no-benefit replacement.

A. Incumbent reliability

State	N	Raw reliability [95% CI]
Hard-invalid	1,155	94.3 wrong [92.1,96.3]
Challenged	824	75.8 wrong [72.2,79.3]
Supported	2,515	76.3 correct [74.1,78.4]

B. Beneficial overwrite ($N = 503$)

Signal	AUROC	AUPRC
Rebind confidence	**.661**	**.591**
Top-minus-Raw score	.494	.478
VLM margin	.443	.441

C. Consequence by confidence bin

Bin	N	Rep.	Regr.	Rep./Regr. (%)
$(-\infty, 0]$	308	103	67	33.4/21.8
$(0, .05]$	16	6	2	37.5/12.5
$(.05, .15]$	32	14	7	43.8/21.9
$(.15, .40]$	56	47	3	**83.9**/5.4
$(.40, \infty)$	91	60	6	65.9/6.6

Overwrite confidence provides predictive signal but is insufficient as a standalone authorization rule: repair–regression outcomes vary non-monotonically with confidence and across frozen backbones. We therefore treat it as inference-visible evidence within the authorization decision. Detailed diagnostics and learned predictors appear in Appendix Table 19 and Appendix D.2.

4.4 CONTROLLED INTERVENTION: DOES SELECTIVE AUTHORIZATION HELP?

We isolate overwrite authorization by holding recovery fixed: every policy receives the same candidate inventory, scores, VLM preference, and retrieved alternative, and differs only in whether it retains or replaces the incumbent.

On the held-out split, ViCoB reaches 66.7% exact binding with 188 repairs and 13 regressions, versus 63.8% and 230/85 for Always-rerank (Table 3; Figure 2(a)). More repairs therefore need not yield a better intervention. Across 21 development-defined runner-up-margin thresholds, exact binding rises with coverage and never falls below Raw; this scalar sweep does not reproduce ViCoB's multifeature decision (Figure 2(b); Appendix Figure 5).

The full matrix shows the same tradeoff when recovery and authorization are evaluated jointly (Table 3). The held-out comparison isolates authorization; its +2.9-point ViCoB–Always difference has a paired 95% interval of $[-0.5, +6.5]$ points, including zero.

Valid-but-wrong challenge. We isolate 325 executable-valid but wrong incumbents (Table 3). With correct incumbents removed by conditioning, regression risk disappears: Always-rerank repairs 135 cases versus ViCoB's 93. This expected reversal locates the value of selectivity in protecting incumbents whose correctness is unknown; executable validity alone does not guarantee it.

Learned and cost-sensitive authorization. Learned gates use only inference-visible features and case-disjoint calibration data, with operating points selected on calibration rather than matched coverage. The strongest discriminator, a random forest, reaches AUROC 0.887 and AUPRC 0.661; its zero-calibration-regression operating point reaches $60.97 \pm 0.53\%$ exact binding with 115.4 ± 5.5

Table 3: **Selective rebinding under fixed retrieval.** Retain-or-replace policies are compared on the full primary matrix, a fixed-retrieval held-out split, and the valid-but-wrong challenge. Fixed-retrieval panels share candidate evidence and the retrieved alternative; policies differ only in action. N denotes model–case observations; learned rows are five-seed means, and differences are computed from unrounded values.

Policy	Exact (%)	Δ Raw	Rep.	Regr.	Net	Cov. (%)	Prec. (%)
Full primary matrix ($N = 5{,}250$)	*Target recovery and authorization jointly evaluated*						
Raw	49.1	—	0	0	0	0.0	—
Always-rerank	63.1	+14.0	**1,087**	352	735	—	—
ViCoB	**64.4**	**+15.3**	902	**100**	**802**	—	—
Fixed retrieval, held-out ($N = 1{,}047$)	*Same candidate evidence and retrieved alternative*						
Raw	50.0	—	0	0	0	0.0	—
Always-rerank	63.8	+13.8	**230**	85	145	48.0	73.0
Validity-only	59.0	+9.0	95	0	95	19.0	100.0
Logistic gate (5-seed)	59.89	+9.93	104.0	0.0	104.0	15.66	100.0
MLP gate (5-seed)	58.40	+8.44	88.4	0.0	88.4	11.10	100.0
Random forest (5-seed)	60.97	+11.02	115.4	0.0	115.4	15.45	100.0
ViCoB gate	**66.7**	**+16.7**	188	13	**175**	32.8	93.5
Valid-but-wrong challenge ($N = 325$)	*All incumbents are executable-valid but incorrect*						
Raw	0.00	—	0	0	0	0.00	—
Validity-only	0.00	0.00	0	0	0	0.00	—
Always-rerank	**41.54**	+41.54	**135**	0	**135**	67.38	100.0
ViCoB gate	28.62	+28.62	93	0	93	40.31	100.0

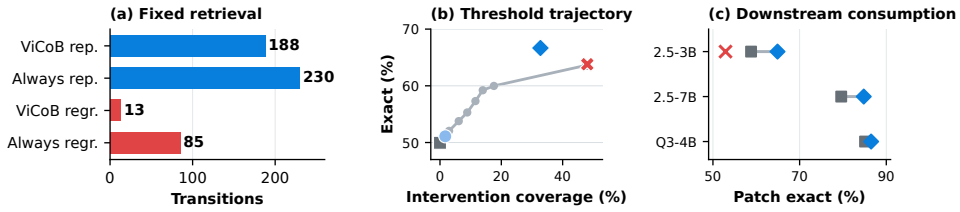


Figure 2: **From rebinding to downstream editing.** (a) With candidate evidence and the retrieved alternative fixed on 1,047 observations, Always-rerank yields 230 repairs/85 regressions versus ViCoB's 188/13; blue bars show repairs and red bars regressions. (b) Exact binding along 21 development-defined runner-up-margin thresholds rises with coverage; default ViCoB is a separate blue diamond. (c) Guarded patch exactness on 960 matched cases per editor (Raw gray squares; ViCoB blue diamonds); the red cross marks unguarded 3B capsule consumption. Improved binding can help editing, but does not guarantee it. The paired 95% interval for ViCoB–Always exact difference includes zero.

repairs and no observed test regressions. Thus, overwrite benefit is predictable, but prediction alone does not determine the intervention operating point. We therefore do not interpret this comparison as evidence that deterministic authorization is intrinsically superior to learned authorization; the policies instantiate different operating points on the same repair–regression tradeoff.

Appendix D.2 reports the complete learned-policy results and cost-sensitive operating points; neither retunes ViCoB.

4.5 Downstream Binding Consumption

Under the execution-validity guard, patch exactness improves for all three editors (Table 1; Figure 2(c)). Figure 3(b) shows why binding gains alone do not guarantee execution: the unguarded 3B capsule reduces wrong-binding failures but increases schema-invalid patches.

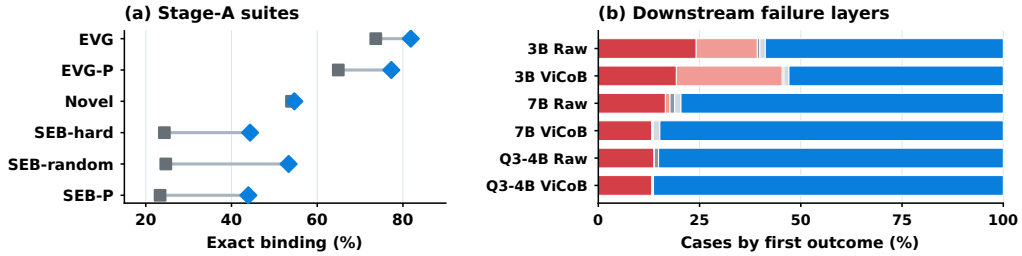


Figure 3: **Recovery gains and downstream failure layers.** (a) Exact binding on the primary Stage-A suites (Raw gray squares; ViCoB blue diamonds). (b) First observed outcome on 960 matched cases per editor: dark red is wrong binding, light red is schema-invalid patch, gray is another incorrect outcome, and blue is exact success. This panel uses the unguarded frozen-capsule arm, not the guarded patch exactness in Table 1; for 3B, fewer wrong bindings coexist with more schema failures.

In the unguarded 3B arm, patch exactness falls to 52.9%, below Raw. Binding quality determines the executable anchor; binding consumption determines whether the editor can turn it into a valid patch.

Matched transition tables, guard ablations, failure-layer decompositions, and editor-specific breakdowns are reported in Appendix Tables 4, 30, and 31.

4.6 Transfer across executable representations

Finally, we test whether selective rebinding extends beyond the primary SVG setting. Results remain positive across additional model and representation settings, including GLM-4.6V, source-disjoint SVGEditBench V2, adapted Novel-SVG, and DOM (Table 1), although operation-level effects are heterogeneous (Appendix Table 9).

Transfer depends on the executable evidence exposed by each representation. On Novel-SVG, the historical internal slice improves by only +0.7 points, whereas an adapted 300-case replay improves by +9.9 after exposing endpoint, duplicate-label, and replacement-name evidence. On DOM, exact binding improves from 74.6% to 99.2%, with 246 repairs and no regressions. The near-saturated DOM result is representation-specific and should not be interpreted as evidence that perceptual-to-symbolic ambiguity is generally solved. Together, these results support transfer of the recovery–authorization principle across SVG and DOM, while showing that its concrete realization and difficulty remain representation-specific.

Exploratory real-web boundary check. We further evaluate a WebMMU Code Editing component-root conversion (Awal et al., 2025). Raw, Always-rerank, and ViCoB reach 27.4%, 32.6%, and 33.7% exact binding, respectively. Always-rerank produces 18 repairs and 13 regressions, compared with 15 and 9 for ViCoB. This exploratory result reproduces the repair–regression tradeoff on real-web code, but we do not treat it as evidence of broad real-web generalization.

5 Conclusion

ViCoB treats executable binding correction as two decisions: recover a plausible alternative and authorize its replacement of the incumbent. Fixed-retrieval experiments hold the alternative and its evidence constant, showing that Always-rerank repairs more errors but also causes more regressions. Selective authorization therefore improves final binding by choosing an explicit repair–regression operating point, rather than replacing every recoverable incumbent.

Inference-visible evidence makes overwrite benefit measurable, while learned and cost-sensitive controls occupy the same intervention tradeoff. Corrected bindings can improve downstream editing when consumed reliably, and the recovery–authorization principle transfers across SVG and DOM with representation-specific evidence. The evaluation focuses on single-step decisions, with WebMMU as a real-web boundary check; within this scope, candidate recovery alone is insufficient because overwriting an existing symbolic commitment is a separate decision with direct executable consequences.

AI USE STATEMENT

Generative AI tools were used to assist with language editing and manuscript organization. The authors take full responsibility for the final manuscript and its contents.

REPRODUCIBILITY STATEMENT

We document the benchmark composition, data splits, evaluation protocol, configuration provenance, statistical procedures, and representation-specific adapters in the appendices. The primary ViCoB configuration is frozen and shared across the primary evaluation suites and multimodal backbones, while learned authorization policies and risk-gate margins use case-disjoint calibration data. We additionally report sensitivity analyses, policy operating points, repair–regression statistics, and detailed transfer results. The anonymized code, frozen configurations, and evaluation artifacts needed to reproduce the reported experiments are available at https://anonymous.4open.science/r/vicob-76B6.

REFERENCES

Rabiul Awal, Mahsa Massoud, Aarash Feizi, Zichao Li, Suyuchen Wang, Christopher Pal, Aishwarya Agrawal, David Vazquez, Siva Reddy, Juan A. Rodriguez, Perouz Taslakian, Spandana Gella, and Sai Rajeswar. WebMMU: A benchmark for multimodal multilingual website understanding and code generation. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 25118–25145. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.emnlp-main.1276. URL https://aclanthology.org/2025.emnlp-main.1276/.

Gilles Baechler, Srinivas Sunkara, Maria Wang, Fedir Zubach, Hassan Mansoor, Vincent Etter, Victor Carbune, Jason Lin, Jindong Chen, and Abhanshu Sharma. ScreenAI: A vision-language model for UI and infographics understanding. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*, pp. 3058–3068, 2024. doi: 10.24963/ijcai.2024/339.

Kanzhi Cheng, Qiushi Sun, Yougang Chu, Fangzhi Xu, Li YanTao, Jianbing Zhang, and Zhiyong Wu. Seeclick: Harnessing GUI grounding for advanced visual GUI agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 9313–9332. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.acl-long.505.

Xiang Deng, Yu Gu, Boyuan Zheng, Shijie Chen, Sam Stevens, Boshi Wang, Huan Sun, and Yu Su. Mind2Web: Towards a generalist agent for the web. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Alexandre Drouin, Maxime Gasse, Massimo Caccia, Issam H. Laradji, Manuel Del Verme, Tom Marty, Léo Boisvert, Megh Thakkar, Quentin Cappart, David Vazquez, Nicolas Chapados, and Alexandre Lacoste. WorkArena: How capable are web agents at solving common knowledge work tasks? In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 11642–11662. PMLR, 2024.

Yonatan Geifman and Ran El-Yaniv. Selective classification for deep neural networks. In *Advances in Neural Information Processing Systems*, volume 30, pp. 4878–4887, 2017.

Wenyi Hong, Weihan Wang, Qingsong Lv, Jiazheng Xu, Wenmeng Yu, Junhui Ji, Yan Wang, Zihan Wang, Yuxiao Dong, Ming Ding, and Jie Tang. CogAgent: A visual language model for GUI agents. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14281–14290, 2024.

Aishwarya Kamath, Mannat Singh, Yann LeCun, Gabriel Synnaeve, Ishan Misra, and Nicolas Carion. MDETR: Modulated detection for end-to-end multi-modal understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 1780–1790, 2021.

Jing Yu Koh, Robert Lo, Lawrence Jang, Vikram Duvvur, Ming Lim, Po-Yu Huang, Graham Neubig, Shuyan Zhou, Russ Salakhutdinov, and Daniel Fried. VisualWebArena: Evaluating

multimodal agents on realistic visual web tasks. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 881–905. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.acl-long.50.

Weixian Lei, Difei Gao, and Mike Zheng Shou. Grounding multimodal large language model in GUI world. In *The Thirteenth International Conference on Learning Representations*, 2025.

Liunian Harold Li, Pengchuan Zhang, Haotian Zhang, Jianwei Yang, Chunyuan Li, Yiwu Zhong, Lijuan Wang, Lu Yuan, Lei Zhang, Jenq-Neng Hwang, Kai-Wei Chang, and Jianfeng Gao. Grounded language-image pre-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10965–10975, 2022. doi: 10.1109/CVPR52688.2022.01069.

Minghao Li, Yingxiu Zhao, Bowen Yu, Feifan Song, Hangyu Li, Haiyang Yu, Zhoujun Li, Fei Huang, and Yongbin Li. API-bank: A comprehensive benchmark for tool-augmented LLMs. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 3102–3116. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.emnlp-main.187.

Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, and Lei Zhang. Grounding DINO: Marrying DINO with grounded pre-training for open-set object detection. In *Computer Vision – ECCV 2024*, volume 15105 of *Lecture Notes in Computer Science*, pp. 38–55. Springer Nature Switzerland, 2024. doi: 10.1007/978-3-031-72970-6_3.

Yadong Lu, Jianwei Yang, Yelong Shen, and Ahmed Awadallah. OmniParser for pure vision based GUI agent. *arXiv preprint arXiv:2408.00203*, 2024.

David Madras, Toniann Pitassi, and Richard Zemel. Predict responsibly: Improving fairness and accuracy by learning to defer. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

Kunato Nishina and Yusuke Matsui. SVGEditBench V2: A benchmark for instruction-based SVG editing. *IEICE Technical Report*, 124(445):28–40, 2025. PRMU2024-52.

Yijun Qian, Yujie Lu, Alexander Hauptmann, and Oriana Riva. Visual grounding for user interfaces. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 6: Industry Track)*, pp. 97–107, Mexico City, Mexico, 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.naacl-industry.9.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. In *Advances in Neural Information Processing Systems*, volume 36, 2023. doi: 10.52202/075280-2997.

Torsten Scholak, Nathan Schucher, and Dzmitry Bahdanau. PICARD: Parsing incrementally for constrained auto-regressive decoding from language models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 9895–9901. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.emnlp-main.779.

Shubham Ugare, Tarun Suresh, Hangoo Kang, Sasa Misailovic, and Gagandeep Singh. Syncode: LLM generation with grammar augmentation. *Transactions on Machine Learning Research*, 2025.

Ziqian Wang, Yuxiao Cheng, Tingxiong Xiao, and Jinli Suo. ExBind: A controlled diagnostic benchmark for visual-to-executable correspondence. *arXiv preprint arXiv:2609.01344*, 2026. URL https://arxiv.org/abs/2609.01344.

Brandon T. Willard and Rémi Louf. Efficient guided generation for large language models. *arXiv preprint arXiv:2307.09702*, 2023.

Tianbao Xie, Danyang Zhang, Jixuan Chen, Xiaochuan Li, Siheng Zhao, Ruisheng Cao, Toh Jing Hua, Zhoujun Cheng, Dongchan Shin, Fangyu Lei, Yitao Liu, Yiheng Xu, Shuyan Zhou, Silvio Savarese, Caiming Xiong, Victor Zhong, and Tao Yu. OSWorld: Benchmarking multimodal agents for open-ended tasks in real computer environments. In *Advances in Neural Information Processing Systems*, 2024.

Table 4: Downstream binding consumption moved from the main paper. Patch exactness and transition accounting are reported for the 960-case matched holdout under the frozen execution-validity guard; differences are computed from unrounded values.

Editor	Cases	Raw	ViCoB	Δ	Rep.	Regr.	Valid (%)
Qwen2.5-VL-3B	960	58.8	**64.9**	+6.1	111	52	89.0
Qwen2.5-VL-7B	960	79.6	**84.8**	+5.2	65	15	99.9
Qwen3-VL-4B	960	85.1	**86.5**	+1.4	24	11	100.0

Table 5: Evidence and robustness controls moved from the main paper. Component ablations use the primary Stage-A matrix; held-out gate and learned-ranker controls test intervention and retrieval separately.

Panel	Condition	n/protocol	Exact (%)	Δ	R/R or note
A. Held-out intervention control					
Gate	Risk gate	80/20 splits	54.8	+5.4 vs. Raw	57/0; 8.8% cov.
B. Evidence ablation					
Ablation	Raw frozen output	1,750	49.1	-15.3	no interface
Ablation	Full ViCoB	1,750	**64.4**	—	reported
Ablation	No text-symbolic	1,750	36.4	-28.0	text removed
Ablation	No type constraint	1,750	62.0	-2.4	compatibility removed
Ablation	No unary geometry	1,750	**64.4**	0.0	geometry removed
Ablation	No Raw-preserving gate	1,750	63.1	-1.3	always rerank
Ablation	VLM preference only	1,750	30.7	-33.6	symbolic removed
C. Learned retrieval control					
Ranker	Deterministic ViCoB	1,191 groups	**71.8**	—	fixed score
Ranker	Random forest	1,191 groups	67.5	-4.3	five-seed gap 3.3

Yuhao Yang, Yue Wang, Dongxu Li, Ziyang Luo, Bei Chen, Chao Huang, and Junnan Li. Aria-UI: Visual grounding for GUI instructions. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 22418–22433, Vienna, Austria, 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.findings-acl.1152.

Keen You, Haotian Zhang, Eldon Schoop, Floris Weers, Amanda Swearngin, Jeffrey Nichols, Yinfei Yang, and Zhe Gan. Ferret-UI: Grounded mobile UI understanding with multimodal LLMs. In *Computer Vision – ECCV 2024*, volume 15122 of *Lecture Notes in Computer Science*, pp. 240–255. Springer Nature Switzerland, 2024. doi: 10.1007/978-3-031-73039-9_14.

Shuyan Zhou, Frank F. Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Tianyue Ou, Yonatan Bisk, Daniel Fried, Uri Alon, and Graham Neubig. Webarena: A realistic web environment for building autonomous agents. In *International Conference on Learning Representations*, 2024.

A Supplementary result tables retained from main-paper compression

Table 6 records the frozen recoverability funnel; it characterizes candidate availability rather than an overwrite decision.

Figure 4 visualizes the observable recovery-to-authorization path in this decomposition.

B Evaluation protocol and reproducibility

Case construction. The Stage-A portion of the legacy ExBind-Bench inventory used here contains clean EVG-Bench, EVG-Perturbed, novel-SVG, and Symbolic Evidence Benchmark (SEB)

Table 6: Frozen target recoverability moved from the main paper. Rates use the single-reference exports ($N = 3{,}669$); this availability analysis does not decide whether a retrieved candidate should overwrite Raw.

Metric	Rate (%)	95% CI	Gap vs. Final
Raw exact	47.4	[45.1, 49.7]	−18.4
Retrieved R@1	64.0	[61.9, 66.1]	−1.8
Retrieved R@3	82.2	[80.3, 84.0]	+16.4
Retrieved R@5	86.3	[84.5, 88.0]	+20.5
Final ViCoB	**65.8**	[63.5, 68.1]	—

Table 7: Observed recovery–authorization error decomposition on the frozen Stage-A ledger ($N = 5{,}250$). Categories are mutually exclusive. "Unavailable" records missing ranking-export fields and is not interpreted as an empty candidate set.

Outcome category	Count	Interpretation
Candidate evidence unavailable in export	335	Candidate projection cannot be audited from the frozen ranking artifact.
Retrieval miss	576	Gold is not recovered within the stored top-5 ranking.
Post-retrieval resolution error	249	A replacement occurs but does not yield the gold binding.
Authorization miss	330	Retrieval evidence is available, but the incumbent is retained.
Successful repair	902	Raw-wrong incumbent becomes exactly correct.
Safe retention	2,478	Raw-correct incumbent remains correct.
Harmful authorization	100	Raw-correct incumbent becomes wrong.

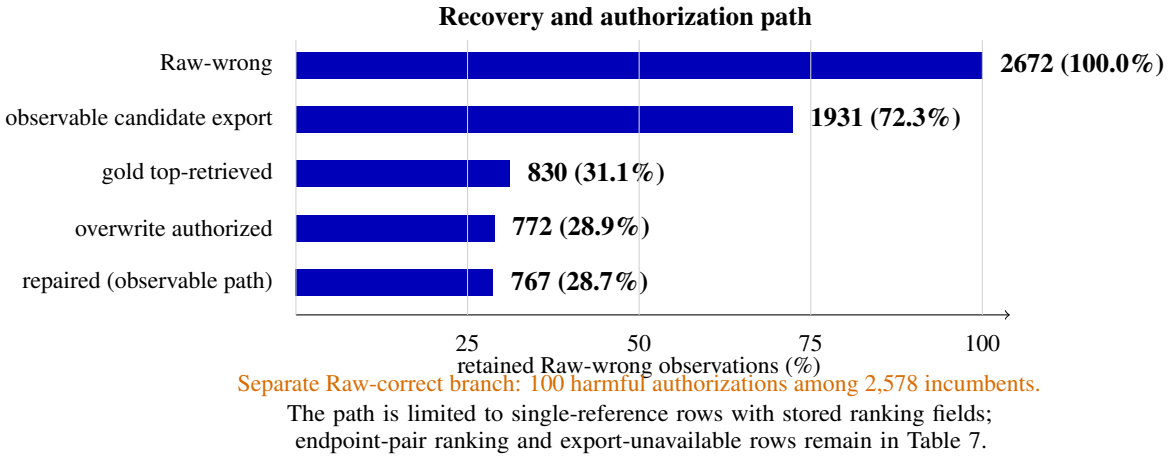


Figure 4: **Where executable binding correction fails.** The observable path is restricted to single-reference rows with stored ranking fields and accounts for 767 of the 902 total repairs; endpoint-pair and export-unavailable rows remain in Table 7. Counts and percentages use the Raw-wrong denominator, and the separate branch measures harmful authorization from Raw-correct incumbents.

Table 8: Primary Stage-A suite composition and stressors. These rows preserve the suite sizes used by the compact main-paper result.

Suite	Cases	Stressor or scope
EVG-Bench	200	clean executable templates
EVG-Perturbed	800	identifier and visual-label perturbations
Novel-SVG (internal)	50	held-out template families
SEB-hard	100	ambiguous identifiers
SEB-random	100	randomized evidence
SEB-perturbed	500	perturbation stress tests
All	1,750	primary Stage-A matrix

Table 9: Frozen EVG perturbation lineage (three-backbone average, exact binding accuracy, %). Each condition contains the same 200 source cases; the 800-case EVG-P release consists of these four conditions. Differences are computed from unrounded values; anchor mismatch is deliberately absent from this aggregate.

Treatment	Raw	ViCoB	Δ
Clean source rendering	73.7	81.8	+8.2
Similar identifier	67.2	79.8	+12.7
Symbolic-identifier shuffle	63.2	76.5	+13.3
Visual-label shuffle	52.7	70.3	+17.7

cases (Wang et al., 2026). EVG and Novel-SVG are synthetic executable templates (the latter uses held-out template families); SEB is an EVG-compatible conversion of SVGEditBench V2 source cases (Nishina & Matsui, 2025), evaluated with the binding metric defined here. Each case stores the rendered image, instruction, executable SVG, candidate inventory, operation, and gold canonical reference solely for offline scoring. EVG-Perturbed includes clean, similar-identifier, symbolic-identifier-shuffle, and visual-label-shuffle treatments; anchor mismatch is outside the frozen aggregate. SEB separately stresses ambiguous identifiers, randomized evidence, and perturbations. Legacy anchor-mismatch conditions are retained as separate reference rows. Table 8 gives the suite counts and stressors, while Table 9 makes the perturbation lineage explicit.

For completeness, the schema-adapted Novel-SVG replay in Appendix 26 is a separate 300-case expansion across held-out template families (900 model–case observations). Its source contract contains duplicate visible labels, natural-language endpoint instructions using "toward", and replacement names that expose a unique executable-ID suffix; the adapter therefore enables the endpoint parser, duplicate-label ordered-pair gate, and explicit replacement-name rule described in Appendix F. The raw model outputs, candidate inventories, and source metadata are held fixed across arms. This expansion is an interface-compatibility analysis and is not added to the primary Stage-A matrix or used to choose the canonical Stage-A coefficients.

Stage-B eligibility. The Stage-B protocol uses EVG cases that contain a non-empty executable gold patch and a canonical `after.svg`: 200 clean and 800 perturbed cases. The first 20 lexicographic cases from each source split are prompt-development data, leaving a frozen 180+780=960-case holdout for each checkpoint. Novel-SVG lacks canonical `after.svg` for this contract, and SEB examples have no non-empty executable gold patch; they define Stage-A binding coverage and fall outside the Stage-B contract.

Pairing and leakage prevention. We keep model outputs fixed within each Stage-A comparison. Raw-output and candidate-prompt results are reported separately. Stage-B arms consume the same image, instruction, candidate inventory, checkpoint, decoding configuration, executor, and renderer. Gold references, patches, success flags, and repaired outputs are unavailable to both the binder and downstream editor. Exact canonical identifier equality determines binding success; executable success is measured after the fixed executor and renderer.

Table 10: Information conditions for the comparisons. Each comparison holds case, checkpoint, parser, decoding, scoring, and (where applicable) executor fixed; no arm receives gold bindings or a different executor.

Comparison	Treatment difference	Target claim
Stage-A	Raw output vs. selective binding	frozen binding correction
Candidate prompt	direct candidate choices vs. ViCoB gate	within-prompt correction
Stage-B	typed constraints plus capsule	downstream use
Attention	evidence keys plus signed bias	local margin effect

Table 11: Stage-B field-level prompt contract. "Inventory" denotes the same executable candidate list in all three arms.

Arm	Added directive	Added source-derived content
Raw	Directly request the executable patch from image, instruction, and inventory.	None beyond the shared inventory.
Typed	Raw directive plus operation type, ordered endpoint semantics where applicable, and required patch schema.	None beyond the shared inventory.
ViCoB	Exactly the Typed directive.	Frozen predicted canonical ID(s), primitive type, normalized text, bounding box, and relation tags.

External SVGEditBench conversion check. The external check uses 100 source-SVG-disjoint SVGEditBench V2 conversions and a frozen LLaVA-Med-v1.5-Mistral-7B checkpoint. The held-out source filenames are excluded from the adapter-development set before deterministic sampling. Because this is an EVG-compatible conversion, its target is exact canonical binding rather than SVGEditBench V2's original full-edit score. The Raw response and both adapter arms use the same candidate inventory; appearance descriptors and isolated raster boxes are computed only from `before.svg`. The visual-anchor arm adds a numbered-mark VLM call and repairs an invalid raw response only when a positive numbered mark is parsed. The final replay contains 100 held-out cases per arm, with 16/19/20 exact cases for Raw, appearance, and visual-anchor, respectively; paired transitions are 3/0 and 4/0 repairs/regressions. These cases are a source-disjoint external confirmation of the binding interface and are not pooled into the primary Stage-A matrix.

Stage-B field-level interface contract. The three arms differ only in the textual interface shown in Table 11; the checkpoint, visual input, candidate inventory, decoding, executor, and renderer remain fixed. In particular, the predicted capsule names one selected canonical reference (or ordered endpoint pair) and its source-derived fields. It is produced from that checkpoint's frozen Stage-A output and carries no gold reference or executable result.

Statistics and statistical unit. The primary Stage-A matrix contains 1,750 underlying cases, each evaluated under three frozen backbones, yielding 5,250 paired model–case observations. We report model–case rates for descriptive per-backbone and pooled summaries, but the statistical unit for Stage-A uncertainty is the underlying dataset–sample case: each bootstrap draw samples these case clusters with replacement and carries all available backbone observations for a selected case, so the 5,250 observations are never treated as independent replicates. The pooled cross-backbone point estimate averages the paired model–case observations, while its uncertainty remains case-clustered. All 10,000-draw percentile intervals for Stage-A rates, differences, retrieval, and pre-decision strata use this same case-clustered bootstrap; conditional retrieval uses the subset of case clusters represented by complete frozen exports. Stage-B comparisons are matched within checkpoint and case on the 960-case holdout, with paired intervals over cases. Wilson intervals in the rollout association figure use individual rollout as the unit, and the attention sign tests use the paired endpoint-correction case as the unit. McNemar tests are reported only for matched binary comparisons and use their declared

Table 12: **Interface invariants and schema adapters.** Invariants define the method contract; the right-hand columns list the SVG and DOM instantiations used in evaluation.

Invariant	Contract	SVG instantiation	DOM instantiation
Typed candidate projection	Emit only operation-compatible executable references.	Primitive type, endpoint order, and SVG operation rules.	Element role, DOM type, and action compatibility.
Visible evidence comparison	Score only source-visible candidate evidence.	ID, text, labels, geometry, and relation tags from source SVG.	Text, attributes, role, and tree-local metadata.
Selective intervention	Change Raw only inside the declared correction set.	Validity, type, identifier regime, and score-margin gate.	The corresponding DOM validity and compatibility gate.
Exact fallback	Outside the correction set, return canonicalized Raw exactly.	Raw SVG identifier or endpoint pair.	Raw DOM element reference.
Deterministic output	Resolve ties by the fixed stable rule.	Candidate order after canonical projection.	Candidate order after DOM projection.

matched unit; they complement, rather than replace, the case-clustered bootstrap for pooled Stage-A inference.

C Implementation details

C.1 Configuration-selection ledger

The reported Stage-A matrix is a replay of historical frozen model outputs under one historically frozen, hand-specified ordinal VICOB-ZERO configuration. The configuration is training-free: no weight is estimated by minimizing error on the primary reported labels. Coefficients encode the intended ordering of evidence sources, operation overrides encode executable semantics, and the same values are applied to all six suites and all three backbones. The frozen configuration record declares no split-specific or model-specific override. Table 13 records the freeze boundary. We do not describe this history as preregistration, and we do not claim that aggregate results were never inspected; the claim is the performance of this fixed cross-suite configuration. The selection boundary is therefore a single canonical recipe, not a per-suite optimum. We write its ordered components as $\theta_{\text{bind}} = (w_{\text{label}}, w_{\text{text}}, w_{\text{geom}}, w_{\text{vlm}}, w_{\text{type}}, w_{\text{raw}}; \tau_t, \tau_g, \tau_r) = (0.50, 1.00, 0.75, 1.00, 2.00, 0.15; 0.50, 0, 0.25)$. These are hand-specified ordinal priorities and gate constants, not parameters fitted or selected by minimizing error on a reported test set; the tuple is copied unchanged to every Stage-A row. Novel-SVG, DOM, candidate-prompt, external-SVG, and WebMMU results use separate schema/interface adaptations; risk-gate margins are held-out calibration operating points, not changes to this recipe. The 0.5x/1.5x weight replays and raw-prior sensitivity rows are stability diagnostics only.

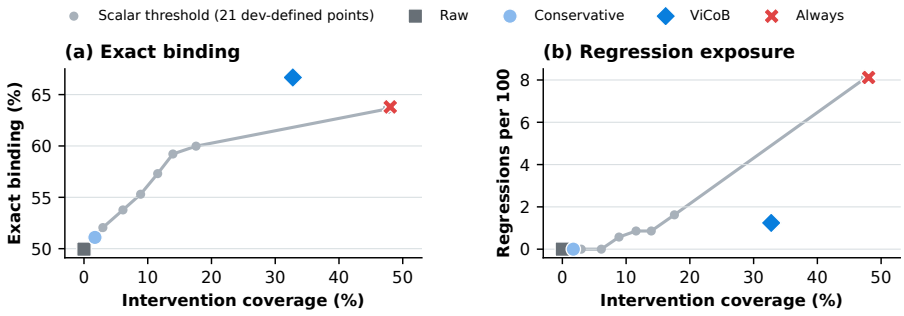
The candidate parser enumerates SVG identifiers and normalizes text, primitive types, bounding boxes, parent-child relations, and existing edges. Candidate scores use the fixed weights in Equation 4; a candidate must satisfy operation compatibility before it can win. The raw prior is 0.15, compared with 1.00 for explicit text and 2.00 for type compatibility. For multi-endpoint operations, joint decoding applies the 2.0 existing-edge and 3.0 self-loop penalties before the 1.5 pair-geometry bonus and deterministic tie break. The raw-preserving gate uses the fixed 0.5 valid-raw type threshold, 0.0 generic-ID margin, and 0.25 non-generic rerank margin. For candidate-prompt outputs, valid object selections are retained, while group and endpoint rules remain the same. The WebMMU component-root adapter is reported separately: it resolves the canonical component operation before the generic type aliases, uses the strongest local executable field rather than subtree-length-sensitive text aggregation, and treats zero-sized geometry as unavailable. Canonicalization is deterministic, so any frozen model output can be audited against the emitted reference without rerunning the VLM.

Table 13: Configuration-selection and freezing ledger for the reported Stage-A replay. "Labels used" refers to selection of the fixed value, not final offline scoring.

Quantity	Selection basis	Labels used	Freeze / reuse boundary
Text, label, type, raw weights	ordinal evidence design; type compatibility dominates direct text, raw is a tie-break	no fit	one value set for all suites/models
Geometry weight	normalized spatial-relation score on the same ordinal scale	no fit	one value; operation overrides below
Move / style overrides	executable semantics: displacement is not label matching; style has no unary spatial target	no	fixed by operation, not split/model
Endpoint penalties / bonus	reject self-loops and existing edges; prefer plausible ordered endpoints	no	fixed for every connection case
Selective gate	validity, type evidence, generic-ID regime, and positive margin only	no gold at inference	same gate across suites; candidate-prompt object IDs receive exact retention
Reported matrix	frozen raw outputs and cached preferences	gold only after emission	primary matrix: 1,750 cases / 5,250 observations; no new VLM inference or per-split override

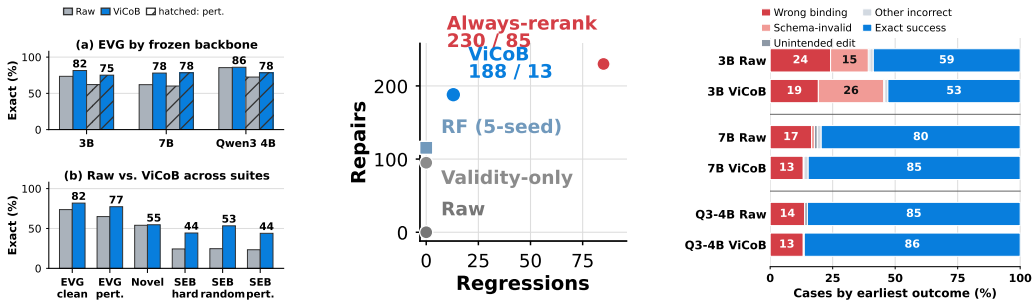
Table 14: Coarse frozen-logit replay surface (exact binding, %). Entries vary the Raw prior and columns jointly scale the symbolic block $(w_{\text{label}}, w_{\text{text}}, w_{\text{geom}})$ and endpoint geometry while holding the VLM preference fixed. All values exceed Raw (49.1%). This is a current-route stability replay, kept separate from the historical 64.4% primary matrix because valid-endpoint retention differs between the replay and historical route; it is not a search used to replace the declared recipe.

Raw prior	Symbolic 0.5×	Symbolic 1.0×	Symbolic 1.5×
0.00	63.2	64.0	**64.2**
0.15	62.5	63.3	63.6
0.30	61.4	62.2	62.5



Policy	Exact (%)	Coverage (%)	Repairs	Regressions
Raw	50.0	0.0	0	0
Conservative	51.1	1.7	12	0
Default ViCoB	66.7	32.8	188	13
Always-rerank	63.8	48.0	230	85

Figure 5: **Development-defined margin replay on fixed held-out evidence.** The top panels show exact binding and regressions per 100 observations versus intervention coverage for all 21 development-quantile thresholds; overlapping thresholds are retained in the accompanying CSV. The bottom table reports the four interpretable operating points on the same 1,047 observations. The conservative point is the development-selected q95 margin, not a test-selected optimum. It yields 12 repairs and zero regressions here. Across five deterministic split seeds, the conservative rule has mean coverage $3.48 \pm 3.70\%$, exact binding $51.72 \pm 3.01\%$, repairs 25.2 ± 24.2, and regressions 0.8 ± 1.3 (mean$\pm$SD). The scalar replay and default ViCoB are distinct policies.



(a) Stage-A binding **(b) Fixed retrieval** **(c) Downstream consumption**

Figure 6: Supplementary overview of primary binding, fixed-retrieval intervention, and downstream consumption. Panel (b) is complemented by the main fixed-retrieval comparison in Figure 2.

Table 14 gives a deliberately coarse rather than local perturbation region. The replay remains positive throughout this 3×3 surface; type weights $0.5\times$ and $1.5\times$ leave its 63.3% center unchanged, and four joint gate settings $(\tau_g, \tau_t) \in \{0.25, 0.50\} \times \{0.25, 0.75\}$ obtain 62.2%. Thus this diagnostic does not identify a narrow optimum. These post-freeze replays are stability checks only: the historical configuration remains the shared recipe for Table 1.

C.2 Fixed-retrieval overwrite-policy controls

Every overwrite policy receives the same candidate inventory, candidate scores, frozen VLM preference, and retrieved top alternative. Policies differ only in whether that alternative is authorized to replace the incumbent binding. The deterministic split assigns 598 `dataset::base_sample_id` case groups to calibration and 152 groups to test, giving 4,203 and 1,047 model–case observations. Margin, VLM-margin, and learned overwrite-gate thresholds maximize calibration repairs subject to zero observed calibration regressions; no test label is used for threshold selection. The test rows below are the complete ledger underlying Table 3; paired intervals resample case groups and intervention precision uses Wilson intervals.

Table 15: Complete case-disjoint selection-policy comparison with retrieval fixed. Accuracy gains are paired-bootstrap 95% intervals over Raw; precision is Wilson 95% in the machine-readable report.

Policy	Acc. (%)	Gain [95% CI]	Repairs	Regr.	Cov. (%)	Prec. (%)
Raw	50.0	+0.0 [0.0,0.0]	0	0	0.0	—
Always-rerank	63.8	+13.8 [7.2,20.4]	230	85	48.0	73.0
Validity-only	59.0	+9.1 [6.7,11.6]	95	0	19.0	100.0
Margin-only	59.6	+9.6 [7.2,12.4]	101	0	19.6	100.0
VLM-margin	59.0	+9.1 [6.7,11.6]	95	0	19.0	100.0
Learned logistic gate	59.9	+9.9 [6.7,13.7]	104	0	15.7	100.0
ViCoB reported gate	**66.7**	+16.7 [11.7,22.2]	**188**	13	32.8	93.5

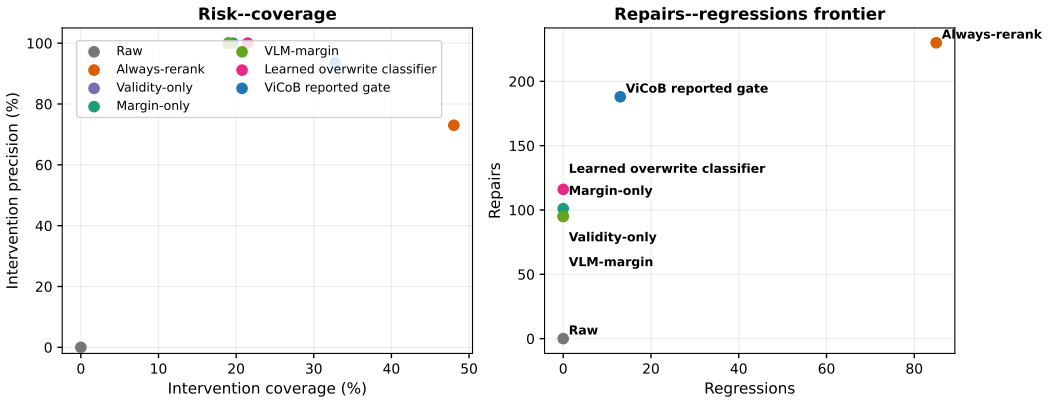


Figure 7: Full fixed-retrieval policy frontier, including the coverage–precision panel omitted from the main-paper crop.

The comparison isolates the replacement decision from retrieval: every non-Raw row receives the same frozen top alternative. ViCoB has the highest tested accuracy and a favorable high-coverage/low-regression position, while the zero-regression policies trade away recovery. The evidence supports an operating-point comparison, not a claim of Pareto dominance over all possible policies.

C.3 Target-recovery and strong-verifier controls

To test whether a stronger learned post-verifier can explain the Stage-A effect, we evaluate Qwen3-VL-8B-Instruct on every frozen Raw row. Let I_i be the rendered image, x_i the instruction, o_i the operation, b_i the frozen Raw binding, C_i the typed executable candidate inventory, and M_i the source-derived SVG metadata (symbolic elements, groups, edges, and ports). The verifier receives $(I_i, x_i, o_i, b_i, C_i, M_i)$ and returns exactly one object

$$d_i \in \{\texttt{retain}, \texttt{correct}\}, \qquad \tilde{b}_i \in \mathrm{Schema}(o_i). \tag{11}$$

Its final binding is b_i for `retain`, and $\tilde{b}_i$ only when every emitted identifier belongs to C_i and the task schema is valid; malformed or invalid outputs deterministically fall back to b_i. Gold targets, executor outcomes, and any result-derived feature are removed before candidate construction and generation. The evaluator records one call for each of the 5,250 paired model–case observations, exact binding only after generation, the paired Raw transition, generated-token count, and wall-clock inference time. The completed run has no invalid responses, averages 1,024 input and 21.6 output tokens, and costs 0.727 seconds per observation (3,815 seconds total). It reaches 58.9% exact binding (+9.8 from Raw; 1,162/649 repairs/regressions), compared with 64.4% and 902/100 for ViCoB on the same primary matrix. It is a stronger learned verifier with the complete executable input contract, while ViCoB retains its frozen candidate-preference score rather than adding another VLM call.

Table 16: Alternative executable-binding controls omitted from the compact main-paper table. All rows use the same 1,750-case primary matrix; the verifier adds one VLM call per observation.

Control	Cases	Raw	ViCoB/control	Δ
Text+Type rule	1,750	55.1	64.4	+9.3
Text+Type+Geom rule	1,750	54.7	64.4	+9.7
Strong verifier	1,750	49.1	58.9	+9.8
Candidate-only	1,750	70.5	72.8	+2.25
Raw+Candidates	1,750	65.0	67.4	+2.42

Table 17: Reported Stage-A configuration for the training-free VICOB reranker. These fixed ordinal weights and structural penalties are shared across every reported Stage-A suite and backbone.

Group	Fixed quantity	Value
Candidate score	exact label / text-symbolic / unary geometry	.50 / 1.00 / .75
Candidate score	VLM token preference / type compatibility / raw prior	1.00 / 2.00 / .15
Raw retention	minimum valid-raw type score	.50
Raw retention	generic-ID override / non-generic rerank margin	0.0 / 0.25
Endpoint pair	existing-edge / self-loop penalty	2.0 / 3.0
Endpoint pair	pair-geometry bonus	1.5
Tie break	lexicographically largest element-ID tuple	fixed

C.4 REPRODUCIBLE SCORING, GATE, AND METRIC COMPUTATION

All symbolic component scores are in $[0, 1]$. For a finite set of raw values z_i, the VLM preference score uses

$$\mathrm{mm}(z_i) = \begin{cases} (z_i - \min_j z_j)/(\max_j z_j - \min_j z_j), & \max_j z_j - \min_j z_j > 10^{-8}, \\ 0, & \text{otherwise}, \end{cases} \tag{12}$$

and symbolic scores are kept in their native $[0, 1]$ range. The lexical text score is the maximum, over query strings and candidate fields (element ID, linked object ID, and source-derived label), of exact match, substring match (0.85), token Jaccard overlap, and 0.6 times normalized edit similarity. The label score is a binary exact/substring compatibility indicator. Geometry scores normalize horizontal, vertical, or anchor-distance values to $[0, 1]$ only when the instruction contains the corresponding spatial cue; otherwise the geometry term is zero. Type scores are task- and role-specific indicators described by the candidate filter in Table 17.

For every candidate e, the reported training-free score is exactly

$$S(e) = 0.50 s_{\text{label}} + 1.00 s_{\text{text}} + 0.75 s_{\text{geom}} + 1.00 s_{\text{vlm}} + 2.00 s_{\text{type}} + 0.15 s_{\text{raw}}, \tag{13}$$

where $s_{\text{raw}} = \mathbb{1}[e = b_{\text{raw}}]$. The fixed operation overrides are $(w_{\text{label}}, w_{\text{text}}) = (0, 0)$ for `move` and $w_{\text{geom}} = 0$ for `set_style`; no other weight is changed. For endpoint pairs, the implemented score is

$$S_{\text{pair}}(i, j) = S_s(i) + S_t(j) + 1.5 g(i, j) - 3\mathbb{1}[i = j] - 2\mathbb{1}[(o_i, o_j) \in E], \tag{14}$$

where $g(i, j) = 0.75 p(i, j) + 0.25 \ell(i, j)$, p is min–max normalized inverse Euclidean distance, and $\ell(i, j) = \mathbb{1}[x_j > x_i]$. Type-incompatible candidates are removed before either argmax. Ties use the lexicographically largest element-ID tuple, which makes the replay deterministic.

For a valid single-target Raw candidate, let $\Delta = S(e_1) - S(b_{\text{raw}})$ and let t_{raw} be its type score. The fixed gate parameters are $\tau_g = 0$, $\tau_t = 0.5$, and $\tau_r = 0.25$. A valid raw object is retained unless it is a generic ID with $\Delta > \tau_g$, or it has $t_{\text{raw}} < \tau_t$ and $\Delta \geq \tau_r$ outside that generic-ID override. Invalid Raw objects are reranked to e_1; valid endpoint pairs are retained under the reported endpoint policy. Thus the gate is a deterministic function of inference-visible validity, type, identifier pattern, and score margin; no gold label is used to choose a row-level action.

Table 18: Complete frozen-observation risk–coverage ledger underlying Fig. 8. The margin is the selected-vs-runner-up candidate-score difference required before changing Raw. Reported ViCoB is the paper policy; margin settings are post-hoc cost-sensitive operating points, not selection runs.

Operating point	Corrections	Coverage (%)	Accuracy (%)	Δ	R/R
Reported ViCoB	1,814	34.6	**64.4**	+15.3	902/100
Margin ≥ 0	436	8.3	54.0	+4.9	257/0
Margin ≥ 0.001	364	6.9	53.8	+4.6	244/0
Margin ≥ 0.10	327	6.2	53.5	+4.4	232/0
Margin ≥ 0.25	292	5.6	53.1	+4.0	212/0
Margin ≥ 0.50	248	4.7	52.5	+3.4	176/0
Margin ≥ 0.75	74	1.4	50.5	+1.4	74/0
Margin ≥ 1.00	24	0.5	49.6	+0.5	24/0

Table 19: Rebind-confidence diagnostic by frozen backbone on the fixed-retrieval overwrite candidates. A beneficial overwrite is a repair rather than a regression or no-benefit replacement.

Backbone	N	Repairs	Regr.	AUROC	AUPRC
Qwen2.5-VL-3B	156	75	32	.519	.570
Qwen2.5-VL-7B	211	91	18	.816	.660
Qwen3-VL-4B	136	64	35	.587	.590

For N matched model–case observations with Raw correctness $r_i \in \{0,1\}$ and ViCoB correctness $v_i \in \{0,1\}$, all reported quantities are computed as

$$\begin{aligned} \mathrm{Acc}(a) &= N^{-1}\sum_{i=1}^{N} a_i, & \Delta &= \mathrm{Acc}(v) - \mathrm{Acc}(r), \\ R &= \sum_i \mathbb{1}[r_i = 0, v_i = 1], & G &= \sum_i \mathbb{1}[r_i = 1, v_i = 0], \\ \mathrm{Coverage} &= N^{-1}\sum_i \mathbb{1}[v_i \neq r_i]. \end{aligned} \tag{15}$$

The same definitions apply to exact executable-patch success in Stage-B. For pooled Stage-A estimates, bootstrap intervals resample dataset–sample case clusters and carry every available backbone observation within each sampled case; each of 10,000 draws recomputes the row-weighted rate or difference before taking the 2.5th and 97.5th percentiles. For a single-checkpoint Stage-B estimate, the cluster is the matched holdout case. McNemar's two-sided exact test uses only the discordant counts (R, G) at the declared matched unit and the binomial null with success probability $1/2$. For the five risk splits, the margin is selected on calibration cases by maximizing R_{cal} subject to $G_{\mathrm{cal}} = 0$; the selected margin is then frozen and evaluated once on the untouched test cases.

For `move`, label and text weights are zero; for `set_style`, unary geometry is zero. The reported replay evaluates the operation-aware candidate filter, all three fixed gate thresholds, deterministic endpoint decoding, and stable tie-break without label fitting, split-specific tuning, or additional VLM inference. The 0.5x and 1.5x ordinal-weight variants in the supplemental control are sensitivity replays, not alternative selected configurations; no result row changes the reported tuple.

This ledger supplies the exact counts used in the main risk–coverage analysis. It separates the high-coverage scientific result from conservative deployment choices: every positive-margin row eliminates observed raw-correct regressions on this matrix while retaining a positive aggregate gain. The operating points therefore expose an intervention-cost continuum without changing the reported policy.

For the attention suite, every attempted intervention, response, applied patch, reward component, and final selection is written to a trace. Completion verification requires all 50 paired endpoint-correction cases under all seven conditions (350 condition–case observations), and rejects truncated or legacy artifacts. The reported attention statistic is the defined target-token attention measure for the first divergent identifier decision. At each selected layer, decoder self-attention is averaged over heads

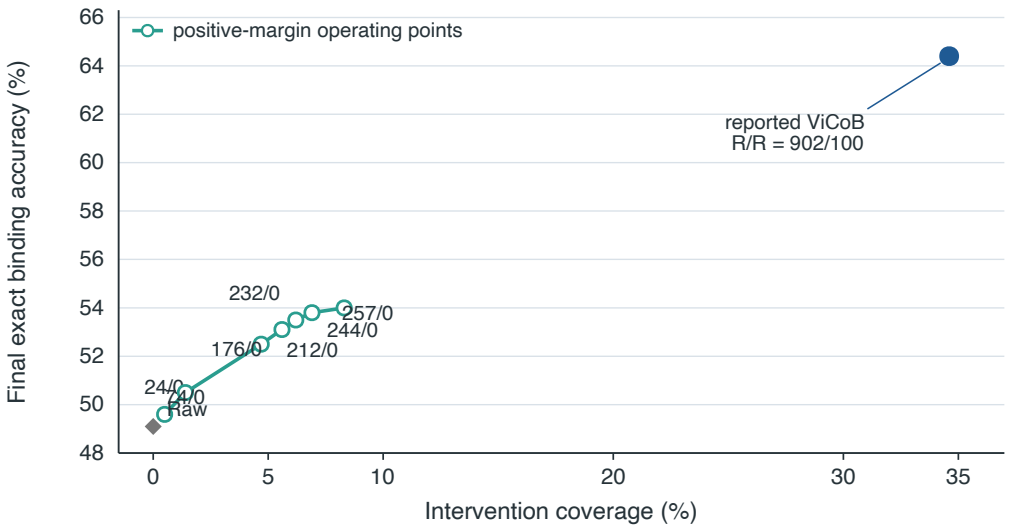


Figure 8: Frozen risk–coverage replay moved from the main paper. Coverage is the fraction of Raw bindings changed; labels denote repairs/regressions.

at the teacher-forced first-divergent identifier token. If K_{tv}, K_{ti}, K_{dv}, and K_{di} denote target-visual, target-SVG-ID, distractor-visual, and similar-wrong-ID prompt positions, respectively, the trace statistic is

$$A_{\mathrm{target}} = \sum_{k \in K_{\mathrm{tv}} \cup K_{\mathrm{ti}}} \bar{A}_k, \qquad A_{\mathrm{distractor}} = \sum_{k \in K_{\mathrm{dv}} \cup K_{\mathrm{di}}} \bar{A}_k, \qquad M_{\mathrm{attn}} = A_{\mathrm{target}} - A_{\mathrm{distractor}}, \tag{16}$$

where $\bar{A}_k$ is the head-mean attention mass at key position k. The intervention estimand remains the paired logit margin $M = \log p(t_{\mathrm{canonical}}) - \log p(t_{\mathrm{contrast}})$; attention mass is a guard statistic and is not substituted for the logit estimand.

Post-training implication. The post-training experiments use the same early-interface decomposition as a training diagnostic. Terminal executable failure alone can conflate several causes: the patch may be malformed, the binding may point to the wrong executable object, or the final patch may be executable but semantically wrong. We therefore record schema validity $v(y)$, exact executable binding $a(y)$, and exact patch equality $e(y)$ as separate variables. This supports the paper's motivation that binding is a useful intermediate interface variable for weak-model optimization. The matched executor-only and intermediate-credit 3B controls are interpreted exactly as reported in Table 36: both improve the development gate by the same amount, so the current evidence does not isolate the intermediate reward as the sole cause of final improvement.

Existing rollout traces. The complete rollout-variation ledger and adaptive-resampling counts are retained in Appendix F; the main text focuses on the directly measurable binding–execution association below.

Binding-conditioned trace audit. We additionally compare the exact logged binding indicator $a(y)$ with downstream outcomes in the same traces (Appendix G). Exact-binding rollouts are schema-valid at 100.0% for all four tracks, versus 69.5%, 74.0%, 78.7%, and 97.1% when binding is wrong; patch-exactness gaps are +88.5, +87.3, +69.4, and +48.8 points. This is a strong observational association with later executable progress and supports the layered binding–consumption relationship.

Post-training RL protocol. Each RL track fixes the model checkpoint, Raw anchor predictions, case selection, executor, renderer, and reward decoder before training. The 7B track uses 53 selected cases, three epochs, and eight rollouts per group; the selective-aware 3B and Qwen3 tracks use 207

and 80 selected cases, one epoch, and four rollouts per case. All tracks use raw-anchored advantages, skip groups that are already uniformly correct, and permit at most two adaptive resampling rounds at temperatures 1.1×1.25^r capped at 1.8 when a group has zero reward variance. The final 3B continuation adds a binding/schema intermediate reward with weight 0.35 to the executor reward while preserving the same raw anchor and decoder. A locked run is authorized only when the untouched 40-case development slice has a positive patch-exact delta and more recoveries than regressions. Locked evaluations use a disjoint 200-case slice and retain every paired regression. The Qwen3 training manifest additionally records six finite optimizer steps, 80 raw anchors, and a passed encoding/parameter-validity audit.

Post-training objective and fixed hyperparameters. For a sampled patch y on case x, let $u(y)$ be the unintended-edit rate, $v(y)$ indicate schema validity, $a(y)$ indicate exact executable binding, and $e(y)$ indicate exact patch equality. The terminal reward used by every track is

$$r(y) = -0.20 + 0.10v(y) + 0.35a(y) + 1.00e(y) - 0.30\min\{u(y), 1\}, \tag{17}$$

and the non-terminal binding/schema potential is

$$q(y) = 0.10v(y) + 0.35a(y) - 0.30\min\{u(y), 1\}. \tag{18}$$

With a deterministic pre-update Raw anchor (r_0, q_0), the raw-anchored advantage for rollout j is

$$A_j = \mathrm{clip}_{[-1.5,1.5]}\big(r(y_j) - r_0 + \lambda[q(y_j) - q_0]\big), \qquad \mathcal{L}_{\mathrm{RL}} = -K^{-1}\sum_{j=1}^{K} A_j \log \pi_\theta(y_j \mid x), \tag{19}$$

where $K = 4$, $\lambda = 0.35$ for intermediate-credit 3B and $\lambda = 0$ for the executor-only control. Uniformly zero-variance groups receive REINFORCE only when a nonzero A_j exists; uniformly exact groups are skipped, and uniformly non-exact groups use $0.25\,\mathcal{L}_{\mathrm{SFT}}$ as the explicitly logged fallback. Adaptive resampling is deterministic in the temperature schedule $T_r = \min(1.8, 1.1 \cdot 1.25^r)$ for $r \in \{0, 1, 2\}$ and is invoked only for a uniformly non-exact, zero-variance group. All reported 3B/Qwen3 runs use one epoch, AdamW with learning rate 5×10^{-6} and weight decay 0.01, $p = 0.98$, 160 generated tokens, gradient-norm clipping at 1.0, and seed 13; the 7B track differs only in its declared three epochs and eight rollouts per group. These values were frozen in the run manifests before locked evaluation and are reused across the matched executor-only control.

D TRANSFER AND REPRESENTATION BREAKDOWN

The DOM supplement uses the generated `MiniWoB-Binding` split at `data/dom_transfer/miniwob_binding.jsonl`. It contains 500 single-step cases across click-button, click-link, enter-text, select-option, click-checkbox, and click-radio tasks. Candidates are visible, in-viewport, non-disabled executable elements with stable `dom_id`, tag, role, text/aria/placeholder evidence, and bounding box. Raw VLM-DOM directly receives the screenshot, instruction, and candidate list; invalid IDs count as errors and no repair is applied. ViCoB-DOM uses the fixed DOM adaptation of text, type/schema, geometry, and selective retention, with no tuning on the test split. The DOM adaptation is also fixed before evaluation: for candidate e, $S_{\mathrm{DOM}}(e) = 1.00s_{\mathrm{text}} + 1.00s_{\mathrm{type}} + 0.35s_{\mathrm{geom}} + 1.00s_{\mathrm{raw}}$, with $\tau_{\mathrm{DOM}} = 0$ and minimum raw type score 0.5; the VLM term is zero. It retains a valid raw ID unless the type score is below 0.5 or the top candidate exceeds the raw score by a positive margin. Candidate ranking is by descending score, then original inventory order, then DOM ID, so candidate-order perturbations test the same deterministic policy rather than a tuned variant. Table 20 gives the per-task breakdown; the aggregate result in the main text is therefore not driven by a single task family.

The two raw VLMs (Qwen3-VL-4B and Qwen2.5-VL-3B) have 74.0% and 75.2% exact accuracy, respectively, with 100% candidate validity and zero wrong-type outputs. The surface-text control uses only `visible_text` and `placeholder`; it excludes ARIA/name/value, operation-type calibration, geometry, raw retention, and the selective gate. Its 0/83 `select_option` result is deterministic: those target `select` elements have empty visible text and placeholder, while the requested option is stored in `value`. In paired transitions, ViCoB-DOM retains 370/370 and 376/376 raw-VLM correct predictions for the two models, while correcting 126 and 120 raw-VLM errors. Candidate-order seeds 42, 43, and 44 give ViCoB-DOM accuracies 99.2/99.2/99.2%. The result therefore supports transfer of executable-reference calibration to DOM representations under this single-step template protocol.

Table 20: DOM transfer by task (exact DOM-reference accuracy, %).

Task	n	Raw VLM-DOM	Surface text	ViCoB-DOM
click_button	84	100.0	100.0	100.0
click_checkbox	83	25.9	14.5	**100.0**
click_link	84	100.0	100.0	100.0
click_radio	83	27.1	9.6	**95.2**
enter_text	83	94.0	89.2	**100.0**
select_option	83	100.0	0.0	100.0

Table 21: Exploratory WebMMU existing-ID component-root stress test ($N = 95$). All policies use identical Raw and candidate-retrieval artifacts. The conversion may include descendant code changes within an existing ID root; it is not pooled with the atomic DOM transfer result.

Policy	Exact (%)	Coverage (%)	Repairs	Regressions	Precision (%)
Raw	27.4	0.0	0	0	—
Always-rerank	32.6	69.5	18	13	27.3
Validity-only	27.4	0.0	0	0	—
ViCoB	33.7	61.1	15	9	25.9

D.1 WebMMU real-web code-editing boundary

We also ran a deliberately bounded real-web stress test on WebMMU Code Editing (Awal et al., 2025). The component-root adapter maps the changed DOM IDs to their nearest common pre-existing, visible, unique stable HTML ID and requires at least two visible ID candidates. This produces 95 component-root bindings across the four released languages. Descendant insertion or multi-property changes may occur inside a root, so this is a component-root binding evaluation rather than an atomic leaf-edit benchmark and is not pooled with the DOM portability result.

All policies consume the same Qwen2.5-VL-7B Raw binding and the same frozen DOM retrieval artifact. The adapter uses only generic component-operation dispatch, local component evidence aggregation, and rejection of zero-sized geometry placeholders; no WebMMU coefficient, threshold, learned retriever, or overwrite rule is tuned. Table 21 reports the complete transition accounting. Always-rerank repairs 18 roots but introduces 13 regressions; ViCoB repairs 15 while introducing only 9 regressions. Raw exact binding is 27.4%, compared with 32.6% for Always-rerank and 33.7% for ViCoB. Thus the corrected contract provides directional external replication of the repair–regression balance and a modest final exact-binding gain for ViCoB under this frozen protocol. We do not promote it to a full portability claim because the root mapping is automatic and the model-assisted review accepts only 8 high-confidence single-component cases; human double annotation of an atomic real-web set remains the required confirmation.

D.2 Learned overwrite-gate details

This experiment is the fixed-retrieval learned overwrite-gate counterpart to Table 3. Candidate retrieval is frozen exactly as in the main held-out overwrite-policy comparison: the candidate inventory, candidate scores, VLM preference, and top alternative are fixed. The learned model predicts only whether that frozen alternative should replace the incumbent. Thus these are learned authorization baselines, not learned rerankers. Features are restricted to inference-visible pre-decision variables: Raw validity, the frozen Raw score, generic-ID indicator, top-versus-Raw score margin, top-versus-runner-up margin, frozen VLM margin, candidate count, and Raw rank. Gold correctness is used only to form calibration labels $y = \mathbb{1}[b_{\mathrm{raw}}$ is wrong $\wedge\, e_1$ is exact$]$; no gold feature or test label is available at inference. Logistic, MLP, and random-forest gates are fit separately on five case-disjoint calibration seeds and thresholded subject to zero observed calibration regressions. Table 22 reports their discrimination and selected operating points.

The random forest provides the strongest discrimination among the learned overwrite models, with AUROC 0.887 and AUPRC 0.661, but its selected operating point reaches $60.97 \pm 0.53\%$ final exact binding. This distinction illustrates that ranking overwrite benefit across thresholds and selecting

Table 22: Five-seed learned overwrite-gate summary on the fixed-retrieval test split. AUROC/AUPRC measure evidence discrimination across thresholds; Exact/Cov./Rep./Reg. describe the selected operating point, with differences computed from unrounded values.

Gate	AUROC	AUPRC	ECE	Exact	Cov.	Repairs	Regr.
Logistic	.824±.000	.575±.000	.176±.000	59.89±.00	15.66±.00	104.0±0.0	0.0±0.0
MLP	.861±.011	.621±.020	.053±.007	58.40±1.93	11.10±3.28	88.4±20.2	0.0±0.0
Random forest	.887±.001	.661±.002	.164±.002	60.97±.53	15.45±.81	115.4±5.5	0.0±0.0

Table 23: Calibration-selected asymmetric overwrite utility on the untouched test split, $U_\lambda = R - \lambda G$.

Policy	$\lambda = 1$	$\lambda = 2$	$\lambda = 5$	$\lambda = 10$	$\lambda = 20$
Margin-only	145	122	103	101	101
Learned logistic	**191**	**177**	**127**	**122**	**104**
Always-rerank	145	60	−195	−620	−1,470
ViCoB reported gate	175	162	123	58	−72

a high-utility executable intervention point are different objectives. Under the tested data budget, feature contract, and calibration protocol, the reported ViCoB gate achieves a stronger final repair–regression operating point. The evidence supports this favorable ViCoB operating point under the tested protocol, rather than a general claim that deterministic rules dominate learned authorization.

The cost table makes the deployment choice explicit: λ is the relative cost assigned to a harmful overwrite, not a fitted natural constant. Figure 9 visualizes the same measured operating points. Some policies are preferable at particular costs; the evidence supports a repair–regression tradeoff rather than universal policy optimality.

E Learned-ranker control details

This is a candidate-ranking control, distinct from the overwrite gates in Appendix D.2. It uses the current full candidate-ranking export and the same candidate inventories as deterministic ViCoB. For each $(\texttt{sample_id}, \texttt{model}, \texttt{operation}, \texttt{role})$ group, candidate correctness supplies the supervised training label, while prediction uses only the five inference-visible numeric fields $S_{\text{text}}, S_{\text{type}}, S_{\text{geom}}, S_{\text{raw}}, S_{\text{label}}$. Task, model, candidate ID, and gold-derived input fields are excluded. A group-shuffled 80/20 split by sample ID prevents candidate or scene leakage; selection is the highest predicted probability within each held-out inventory. The nonlinear comparator is a class-balanced random forest with 300 trees and minimum leaf size five. A logistic comparator is included as a linear structured control.

At seed 17, deterministic ViCoB corrects 228 of 545 raw-wrong groups while the learned forest corrects fewer groups and disagrees with ViCoB on both recoveries and regressions. The five-seed

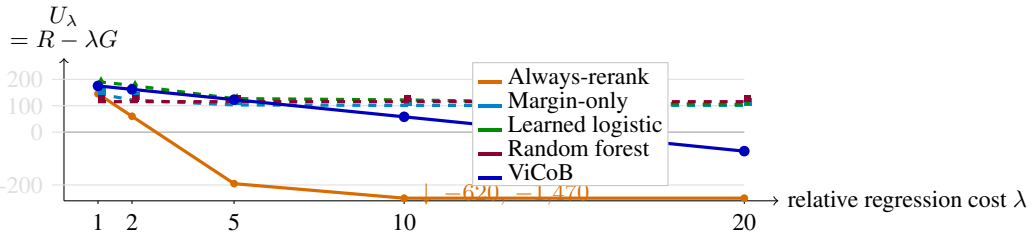


Figure 9: **Overwrite utility as regression cost varies.** $U_\lambda = R - \lambda G$ is evaluated on the fixed-retrieval test split. Margin-only and learned-logistic points use calibration-selected thresholds for each displayed λ; the random-forest line is its five-seed mean at the selected zero-calibration-regression operating point. Fixed-policy segments connect measured operating points only. The downward labels denote Always-rerank utilities outside the plotted range.

Table 24: Five-seed case-disjoint learned-ranker control. Values are held-out exact binding accuracy (%).

Method	seed 11	seed 17	seed 23	seed 31	seed 47	mean ± std
Raw anchor	52.6	54.2	50.2	50.8	50.2	51.6 ± 1.6
Deterministic ViCoB	67.3	71.8	68.6	68.9	67.2	68.8 ± 1.6
Learned random forest	65.7	67.5	65.0	65.2	64.0	65.5 ± 1.1

Table 25: Local neural-routing sensitivity diagnostic. Paired effects use the specified attention-bias operator on selected endpoint-correction decisions; this is an evaluation-only 50-case diagnostic.

Paired intervention	Exact decode	Δ logit margin [95% CI]	ΔM	Sign-test p
No bias → target evidence	74→76	+.220 [.145,.305]	+.0140	1.19e−7
Target evidence → ablate target	76→74	−.283 [−.395,−.180]	−.0172	1.57e−4
Visual evidence → SVG evidence	—	+.268 [.150,.390]	+.0023	2.98e−6
Distractor contrast → target	70→76	+.430 [.293,.580]	—	2.98e−8
Random evidence → target	74→76	−.083 [−.223,.055]	+.0139	.672

mean gap is 3.3 points in favor of ViCoB. This result favors the explicit ordinal design under the tested data budget: it transfers the same operation constraints without estimating parameters, while the supervised forest must learn them from the disjoint training groups. Table 24 reports the complete five-seed comparison.

F Expanded results

F.1 Evaluation-only local intervention diagnostic

The local attention suite is an appendix diagnostic on 50 selected endpoint decisions where frozen Raw is wrong and ViCoB is correct. Offline gold identifies the canonical target and contrast positions for measurement; normal ViCoB inference never uses these positions. With a fixed guided prefix, additive attention bias is applied only to finite entries,

$$B'_{\ell,h,q,k} = B_{\ell,h,q,k} + \alpha, \tag{20}$$

preserving causal and padding exclusions. The estimand is the fixed-prefix canonical-target–contrast logit margin. This diagnostic characterizes the specified operator on its selected decision set.

Trace-backed binding cases. Figure 10 illustrates the two gate outcomes used throughout the paper: correction of an invalid Raw endpoint and retention of a valid Raw reference. Aggregate claims use the paired Stage-A matrix.

Gate sensitivity. The reported gate supplies a construction-level fallback property. Under the frozen replay, macro accuracy is .442 for Raw, .555 when the gate always permits reranking, and .588 for the reported gate. These comparisons complement the pooled primary result in Table 1: the gate consumes inference-visible validity, type, identifier regime, and score margin.

Schema-adapted Novel-SVG replay. The internal 50-case Novel-SVG row in the main table is retained as the historical boundary check. To test whether its small gain reflected an interface mismatch, we replayed the fixed frozen model outputs on a separate 300-case Novel-SVG expansion (900 model–case observations) whose instructions include "draw ... toward ...", repeated visible labels, and explicit replacement names such as "`Stage::clean_b`" and "`svc-auth_shadow`". The adapter (i) parses the endpoint operation, (ii) enables a source-derived ordered-pair rerank when duplicate labels make raw retention ambiguous, and (iii) uses a unique longest executable-ID suffix when the replacement name disambiguates a duplicate object. It uses no gold reference, executor feedback, or additional VLM call. The same margin (0.1) is used for all three checkpoints. As shown in Table 26, the adapted interface improves every checkpoint and produces no Raw-correct → adapted-wrong transitions. This is a schema-adaptation analysis, reported separately from the primary Stage-A matrix, rather than a replacement of the historical main result.

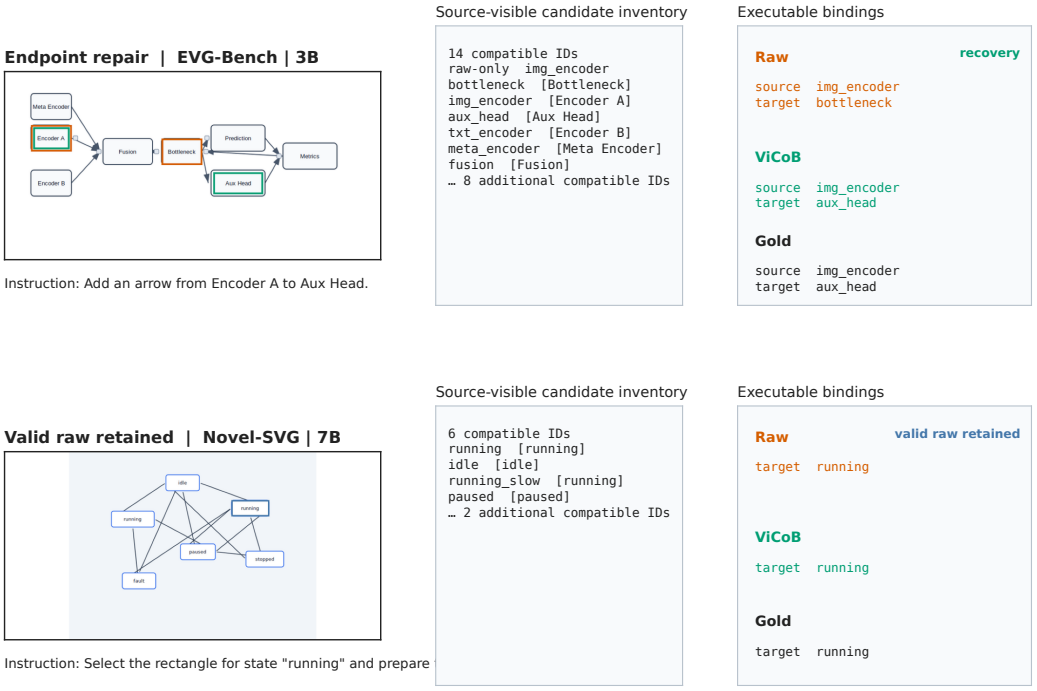


Figure 10: **Trace-backed Stage-A cases.** Each row shows the source SVG, source-visible candidates, and Raw, ViCoB, and evaluation-only gold bindings. Coral marks Raw, teal a correction, and blue retained Raw.

Table 26: Schema-adapted Novel-SVG replay on 300 held-out cases per checkpoint. Results are deterministic replays of fixed Raw outputs; R/R denotes recoveries/regressions.

Checkpoint	Cases	Raw	Adapted	Δ	R/R
Qwen2.5-VL-3B	300	56.7	**69.7**	+13.0	39/0
Qwen2.5-VL-7B	300	54.3	**65.3**	+11.0	33/0
Qwen3-VL-4B	300	62.7	**68.3**	+5.7	17/0
All checkpoints	900	57.9	**67.8**	+9.9	89/0

Table 27: Macro Stage-A gate sensitivity across the six benchmark strata. Macro accuracy is an unweighted suite average that reports the behavior of the retention policy.

Gate policy	Macro acc.	Δ vs. Raw	Interpretation
Raw output	44.2	—	no correction
No raw-retention gate	55.5	+11.4	rerank all compatible cases
Reported raw-preserving gate	**58.8**	+14.6	retain unless the correction set permits replacement

Table 28: Primary Stage-A transition accounting (1,750 underlying cases; 5,250 paired model–case observations). "Correct" means exact canonical binding.

Raw state	n	ViCoB correct	ViCoB wrong
Correct	2,578	2,478 (96.1%)	100 (3.9%)
Wrong	2,672	902 (33.8%)	1,770 (66.2%)

Table 29: Candidate-prompt Stage-A results (exact binding accuracy, %). Every column uses the same frozen direct output for the corresponding case and model.

Suite	Candidate-only	ViCoB	Raw+Candidates	ViCoB
EVG-Bench	75.8	**80.0**	74.2	**77.8**
EVG-Perturbed	72.9	**76.8**	70.9	**75.3**
Novel-SVG	57.3	57.3	60.7	60.7
SEB-hard	66.3	66.3	55.7	55.7
SEB-random	72.7	72.7	58.7	58.7
SEB-perturbed	66.3	66.3	55.4	55.4
All primary observations	70.5	**72.8**	65.0	**67.4**

Transition accounting. Table 28 gives the complete all-split transition accounting stored inside the primary Stage-A observations. It makes both recovery and harm visible: 902 raw-wrong bindings become correct, while 100 raw-correct bindings become wrong. Neither direction is filtered from the aggregate.

Candidate-prompt results. Table 29 reports the candidate-prompt controls omitted from the compact main-paper table. Gains appear on the EVG and EVG-Perturbed cases, while the other splits remain unchanged.

Stage-B error accounting. Table 31 separates unguarded frozen capsule transfer from the execution-validity fallback used in the main downstream table. The guard is applied after generation and before scoring: if the ViCoB patch is malformed or schema-invalid under the fixed executor, the Raw patch is returned. This uses no gold patch, exact-success label, render similarity, or unintended-edit label.

The frozen-capsule trace gives a more explicit view of where the downstream interface fails. Table 30 classifies each holdout row by its earliest observed failure layer. ViCoB reduces wrong-binding failures for all three checkpoints, while the unguarded 3B capsule shifts a substantial fraction of remaining failures into schema-invalid patches. This is the operational reason for reporting the execution-validity bridge separately: the bridge protects the downstream editor boundary rather than changing the Stage-A binding score.

Complete mechanism conditions. Table 33 gives the seven condition means from the verified 50-case intervention suite. Absolute margins are reported for reconstruction; the intervention estimands are the paired contrasts and intervals in Table 25. SVG-only and visual-only conditions are intentionally fixed-prefix-only, avoiding a second decoded-output confound.

Table 30: First observed Stage-B failure layer on the frozen 960-case holdout (percent). The 960 matched cases are classified after generation; "schema" includes parse or fixed-executor validity failure, and the unguarded ViCoB arm is shown to expose the interface boundary.

Model	Arm	Wrong binding	Schema	Unintended	Patch incorrect	Exact
Qwen2.5-VL-3B	Raw	24.2	15.1	0.6	1.4	58.8
	Frozen ViCoB	19.3	26.1	0.4	1.3	52.9
Qwen2.5-VL-7B	Raw	16.6	1.1	1.1	1.6	79.6
	Frozen ViCoB	13.2	0.1	0.2	1.7	84.8
Qwen3-VL-4B	Raw	13.8	0.1	1.0	0.0	85.1
	Frozen ViCoB	13.2	0.1	0.3	0.0	86.4

Table 31: Independent Stage-B guard metrics (%) on the matched 960-case holdout. Guarded ViCoB uses execution-validity fallback to Raw before scoring; executor score is the mean fixed-executor render score.

Checkpoint	Arm	Parse	Valid	Patch exact	Unintended edit	Executor score
Qwen2.5-VL-3B	Raw	99.1	84.9	**58.8**	**39.2**	**85.3**
	Typed	99.1	80.4	53.8	44.1	80.7
	Frozen ViCoB	96.3	73.9	52.9	42.7	77.5
	Guarded ViCoB	**99.5**	**89.0**	**64.9**	**33.5**	**88.9**
Qwen2.5-VL-7B	Raw	99.8	98.9	79.6	18.5	98.7
	Typed	99.8	99.0	79.5	18.5	98.8
	Frozen ViCoB	**100.0**	**99.9**	**84.8**	**14.1**	**99.8**
	Guarded ViCoB	**100.0**	**99.9**	**84.8**	**14.1**	**99.8**
Qwen3-VL-4B	Raw	**100.0**	99.9	85.1	14.8	99.8
	Typed	**100.0**	**100.0**	84.6	15.3	**99.9**
	Frozen ViCoB	**100.0**	99.9	86.4	13.9	99.8
	Guarded ViCoB	**100.0**	**100.0**	**86.5**	**13.8**	**99.9**

Table 32: Stage-B split results (exact executable patch success, %). "Guard" is execution-validity fallback to Raw; W/L are recoveries/regressions relative to the corresponding Raw anchor.

Model / variant	Split	Cases	Raw	Typed	ViCoB	W/L
Qwen2.5-VL-3B / Frozen	clean	180	69.4	62.8	58.3	19/39
	perturbed	780	56.3	51.7	51.7	92/128
Qwen2.5-VL-3B / Guard	clean	180	69.4	62.8	**73.3**	19/12
	perturbed	780	56.3	51.7	**62.9**	92/40
Qwen2.5-VL-7B / Frozen	clean	180	79.4	79.4	**87.8**	17/2
	perturbed	780	79.6	79.5	**84.1**	48/13
Qwen2.5-VL-7B / Guard	clean	180	79.4	79.4	**87.8**	17/2
	perturbed	780	79.6	79.5	**84.1**	48/13
Qwen3-VL-4B / Frozen	clean	180	86.7	86.7	**89.4**	7/2
	perturbed	780	84.7	84.1	85.6	17/10
Qwen3-VL-4B / Guard	clean	180	86.7	86.7	**89.4**	7/2
	perturbed	780	84.7	84.1	**85.8**	17/9

Table 33: Full 50-case mechanism suite. M is the fixed-prefix canonical-target–contrast logit margin and M_{attn} is target-minus-contrast attention mass. Dashes denote fixed-prefix-only conditions.

Condition	Decoded exact (%)	M	M_{attn}
Guided, no bias	74.0	2.3675	.00234
Target evidence (all)	76.0	2.5875	.01631
Target evidence ablated	74.0	2.3050	−.00088
Target SVG only	—	2.6100	.01093
Target visual only	—	2.3425	.00866
Raw-contrast evidence	70.0	2.1575	−.00070
Random evidence	74.0	2.6700	.00236

Table 34: Guarded post-training extension across the three model-specific RL tracks. Each block reports its declared baseline entry; Δ and R/R are relative to that baseline.

Model / Evaluation	Variant	Success	Δ vs. baseline	R/R
7B / Main ($n = 200$)	Raw anchor	.830	—	—
	SFT	.830	.000	—
	Guarded RL	.860	+.030	6/0
7B / Holdout ($n = 200$)	Raw anchor	.815	—	—
	Guarded RL	.835	+.020	4/0
3B / Locked ($n = 200$)	Raw anchor	.810	—	—
	Intermediate-credit RL ViCoB	.820	+.010	4/2
Qwen3 / Locked ($n = 200$)	Raw anchor	.815	—	—
	Risk-gated RL	.820	+.005	4/3

Table 35: Post-training estimand and slice ledger. Each result is interpreted only within its declared entry; development gates authorize evaluation and are not pooled with locked or composite results.

Track / checkpoint	Train	Dev	Locked	Composite	Reported estimand
Qwen2.5-VL-7B guarded RL	53	—	200	—	trained Raw vs. guarded RL, same locked checkpoint/slice
Qwen2.5-VL-3B intermediate-credit	207	40	200	960	locked: trained Raw vs. risk-gated ViCoB; composite: frozen Raw plus accepted adapter deployment
Qwen3-VL-4B risk-gated RL	80	40	200	—	trained Raw vs. risk-gated ViCoB, same locked checkpoint/slice
Qwen2.5-VL-3B executor-only	207	40	—	—	interface-matched development control; no locked claim

Table 36: Matched 3B post-training development gate. Both variants use the same initialization, selected examples, optimizer, sampling, 40-case protocol, and risk-gated evaluation interface; only the intermediate-reward weight differs. Raw anchors are generated within each trained checkpoint, so the entries test within-checkpoint improvement and their absolute levels are not a cross-checkpoint comparison.

Variant	Int. reward	Raw	ViCoB	Δ	R/R	Decision
Intermediate-credit RL	.35	.950	**.975**	+.025	1/0	authorize locked
Executor-only RL	0.00	.925	**.950**	+.025	1/0	matched gate

Table 37: Compact development-provenance chronology.

Component	Inspected data	Frozen before	Later use
Stage-A weights	ordinal design	primary replay	none
Operation rules	SVG semantics	primary replay	shared across suites
Gate thresholds	validity and margin	primary replay	risk frontier diagnostic
Novel-SVG adapter	held-out expansion	historical matrix	separate schema analysis
DOM adapter	DOM split	SVG matrix	single-step portability

Post-training provenance. The 7B track starts from the initial SFT adapter and trains on 53 selected EVG/EVG-Perturbed curriculum cases for three epochs; its completed trace contains 159 groups, 88 REINFORCE updates, and 71 all-correct skips. Its expanded sensitivity run evaluates the same checkpoint on 500 matched cases under the split manifest (Raw 83.0% versus ViCoB 85.0%, 19/9 recoveries/regressions) and is stored separately from the primary disjoint locked slice. The final selective-aware 3B continuation uses 207 training examples, 207 traced rollout groups, one epoch, raw-anchored advantages, intermediate reward weight 0.35, and two adaptive resampling rounds. The matched executor-only control uses the same 207 examples and traced groups, intermediate reward weight 0.00, 91 REINFORCE updates, and a verifier-passed training trace; its corrected risk-gated 40-case gate passes and the earlier un-gated evaluation is retained only as a debug artifact. The Qwen3 track uses 80 training examples, 80 traced rollout groups, one epoch, six finite optimizer steps, raw anchors for all 80 cases, and the same two-round adaptive resampling rule. Its expanded 600-case sensitivity run uses the same trained checkpoint and matched evaluator (300 cases per split; Raw 87.83% versus ViCoB 88.00%, 6/5 recoveries/regressions) and is stored separately from the primary disjoint locked slice. All tracks use four samples per case for the new 3B/Qwen3 runs, the fixed executor reward, and no gold patch in the policy prompt. The traces record every rollout, resampling decision, reward component, patch, and final update mode, including zero-variance groups and fallback-SFT decisions. These traces substantiate the post-training design's handling of binding/schema credit; they are not used as evidence for a measured GRPO/DPO sample-efficiency gain. The 3B and Qwen3 locked runs were authorized only after positive 40-case gates; machine-readable result paths are supplied with the repository artifacts.

Development provenance disclosure. The primary Stage-A matrix uses one frozen VICOB-ZERO recipe and replayed model outputs; no coefficient or rule was fitted on those observations. The 300-case Novel-SVG expansion is a later schema-adaptation analysis: its parser and source-visible endpoint/replacement rules test compatibility with that schema, and its results are reported separately rather than folded into the primary matrix. Other supplemental controls retain their stated manifests and are labelled as diagnostics or downstream validations. This chronology separates the invariant interface claim from adapter-specific engineering choices.

> **Development provenance.** The primary matrix (1,750 underlying cases; 5,250 paired model–case observations) uses one frozen recipe; the Novel-SVG and DOM changes are separate schema adapters; risk-gate margins are held-out operating points. This chronology distinguishes the invariant intervention contract from later schema-specific engineering.

Source-only replay note. A source-only attribution replay reconstructed 3,350 of the 5,250 paired observations; the remainder was unavailable because of a structural source-snapshot mismatch. The missingness is therefore non-random, and the partial replay is excluded from the main text and from any claim about visual contribution. It is retained only as a reproducibility note for locating the incomplete artifact.

G BINDING-CONDITIONED REWARD OBSERVABILITY

This post-hoc analysis uses the four frozen traces listed in `results/interface_trace_analysis_20260826/manifest.json`; it requires no model inference, training, or prediction rewrite. For each rollout y, we define $a(y) = \mathbf{1}[b_{\text{pred}} = b_{\text{ref}}]$

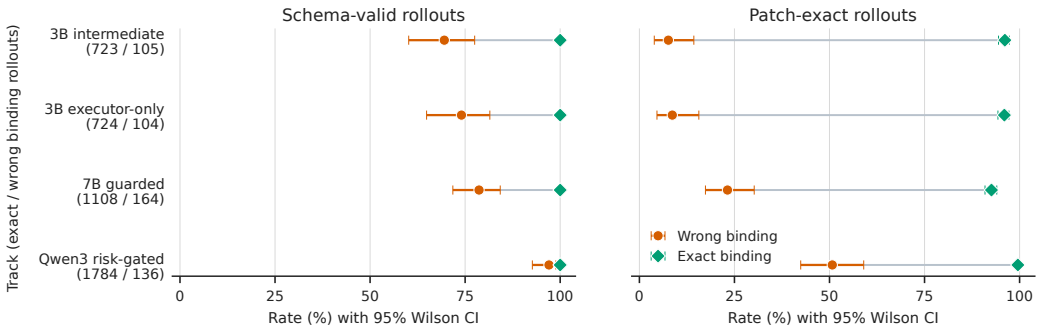


Figure 11: Binding-conditioned downstream association in stored rollout traces. This descriptive post-hoc association does not estimate a causal effect of binding quality on execution.

Table 38: Binding-conditioned downstream observability in stored rollout traces. Values are exact counts with percentages in parentheses; the group-level result is reported separately because mixed groups can expose reward variation for reasons unrelated to binding.

Track	$a = 1$	$v = 1 \mid a = 1$	$a = 0$	$v = 1 \mid a = 0$	Δv	Fisher p
3B intermediate	723	723 (100.0)	105	73 (69.5)	+30.5	1.86×10^{-31}
3B executor-only	724	724 (100.0)	104	77 (74.0)	+26.0	1.78×10^{-26}
7B guarded	1108	1108 (100.0)	164	129 (78.7)	+21.3	2.33×10^{-33}
Qwen3 risk-gated	1784	1784 (100.0)	136	132 (97.1)	+2.9	2.42×10^{-5}

for nonempty reference bindings, schema validity $v(y) = \mathbf{1}[\texttt{valid}]$, and exact patch success $e(y) = \mathbf{1}[\texttt{patch_exact}]$. For group g with terminal rewards R_g, $O_g = \mathbf{1}[|\{R_{gj}\}| > 1]$ is the binary reward-observability indicator and $B_g^{\text{any}} = \mathbf{1}[\exists j : a(y_{gj}) = 1]$ is the declared primary group-level binding statistic. Wilson intervals and two-sided Fisher exact tests are used for the rollout-level proportions; these are descriptive within-track associations and do not replace case-clustered inference for the primary Stage-A matrix.

Figure 11 visualizes the corresponding descriptive binding–execution association; Table 38 gives the underlying observability counts.

The corresponding patch-exactness rates for $a = 1$ versus $a = 0$ are 96.1%/7.6%, 96.0%/8.7%, 92.6%/23.2%, and 99.6%/50.7%, respectively. At the group level, $P(O = 1 \mid B^{\text{any}} = 1)$ versus $P(O = 1 \mid B^{\text{any}} = 0)$ is 31.1% versus 90.9% (3B intermediate), 34.3% versus 88.9% (3B executor-only), 52.4% versus 91.7% (7B), and 6.7% versus 13.3% (Qwen3). Thus the traces show a consistent binding–schema association but no uniform positive group-level reward-variation effect. This is observational evidence for a layered executable interface, not a causal estimate of ViCoB-induced reward variance, sample efficiency, convergence, or trajectory length. The exact computation and raw output are archived in `BINDING_CONDITIONED_REWARD_OBSERVABILITY.md`.

H Claim–Evidence Status

The primary frozen binding-recovery claim is directly demonstrated by the paired Stage-A matrix, transition accounting, risk–coverage replay, and matched verifier/ranker controls. The retrieval/authorization decomposition is directly isolated by the case-disjoint fixed-retrieval replay: all policies receive the same top alternative, while final accuracy changes with the overwrite rule. Portability is directly measured on the tested GLM, DOM, source-disjoint SVG, and separately reported Novel-SVG adapter. Stage-B directly measures a second binding-consumption interface under the declared execution-validity guard. Binding/schema variables as intermediate credit and local neural routing are supported interpretations of the post-training traces and selected fixed-prefix intervention, respectively; neither establishes sample-efficiency, convergence, or a population-wide

Table 39: Compact claim–evidence status and boundary.

Claim	Status	Existing evidence	Boundary
Retrieval / authorization	Direct	frozen-retrieval replay; same top alternative	tested operating points, not general decision theory
Overwrite authorization	Direct	188/13 vs. 230/85 on 1,047 held-out rows; valid-but-wrong challenge	one frozen evidence distribution
Frozen exact binding	Direct	1,750 cases / 5,250 paired model–case observations and transitions	SVG benchmark contract
Raw-preserving gate	Direct	exact fallback invariant and risk/coverage	behavioral, not correctness guarantee
Tested representation transfer	Direct	GLM, DOM, disjoint SVG, adapted Novel-SVG	adapters and exposed evidence are schema-specific
WebMMU real-web code editing	Directional	95 existing-ID component roots; ViCoB 33.7% vs. Raw 27.4%	8 model-adjudicated high-confidence cases; human atomic set still required
Downstream bridge	Direct	matched Stage-B and validity guard	patch consumption is a second bottleneck
Intermediate credit	Supported	decomposed rewards; binding-conditioned schema/patch association	no sample-efficiency claim
Local neural routing	Supported	selected 50-case intervention	local diagnostic only
Longer trajectories / lower cost	Hypothesis	no direct measurement	requires new multi-step traces

mechanism. Longer trajectories, reduced repeated actions, and lower rollout cost remain hypotheses requiring new multi-step traces and cost accounting.

Prioritized optional analyses. The lowest-cost extension is a binding-conditioned reward-observability report from stored rollout traces: compute the fraction of groups with nonzero informative reward variation, stratified by binding status, without changing training or evaluation. If existing multi-step logs contain action histories and valid case identifiers, a second analysis can report first-failure and repeated-action strata. Untouched-schema transfer and a stronger learned candidate-ranker control are lower-priority extensions; neither is needed to interpret the current frozen Stage-A result.

I Reproducibility artifacts

The repository provides machine-readable tables, paired identifiers, audits, interface contracts, traces, completion verifiers, and regeneration scripts. Reproduction requires the 1,750-case, three-backbone Stage-A matrix (5,250 paired model–case rows), the 960-case-per-checkpoint Stage-B holdout, the verified 50-case mechanism suite, and the declared RL traces and promotion manifests. The corresponding control tables and invariants are exported before the paper tables are regenerated.