FlowRSI: Benchmarking Recursive Self-Improvement in Flowchart Parsing

Anonymous authors
Paper under double-blind review

ABSTRACT

Improving after feedback is not the same as learning how to improve: an agent may fix serialization without recovering more visual structure, overwrite a useful strategy, or fail to transfer it to new appearances. We introduce FLOWRSI, an offline benchmark of frozen-weight, agent-directed strategy revision in flowchart parsing. Agents inspect labeled examples, receive feedback from two successive mock exams, and commit prompt, memory, or skill states; we independently replay all four states on the same hidden 1,348-image test. Exact graph supervision, strict and content-aware scores, statewise replays, artifact interventions, and clean, corrupted, and unseen-writer strata distinguish feedback attribution, revision stability, and visual transfer. Across nine open systems, final strict relation-set F1 improves for four and declines for five. Qwen3.5-27B gains 55.47 points, yet its content F1 gains only 6.88 while valid output rises from 6.08% to 77.30%: much of the measured gain is contract repair. Qwen3.5-9B instead loses 25.78 points in its last revision, and removing its final artifacts restores 27.48 points in a matched replay. Even the improved 27B state scores 65.87 on held-out scan corruption but only 26.06 on unseen handwriting. The central limitation is thus not merely whether agents can change themselves, but whether they can diagnose the right failure, retain prior competence, and transfer a correction. FLOWRSI turns these mechanisms into observable outcomes and suggests that future multimodal self-improvement should couple error attribution with revision safeguards and shift-aware validation.

1 INTRODUCTION

An agent that scores higher after receiving feedback has not necessarily learned to improve itself. The system may have repaired its output format, memorized visible answers, consumed more inference, or selected a favorable intermediate state. Endpoint evaluation collapses these mechanisms into one number. It therefore cannot establish whether the agent found informative evidence, diagnosed its failures, formed a reusable correction, persisted that correction, and applied it to unseen inputs.

This distinction matters for multimodal agents. Visual errors arise from several coupled capabilities: reading local content, tracing spatial relations, choosing tools, following a structured-output contract, and controlling a long interaction. Intrinsic self-correction without external feedback can fail or even degrade reasoning accuracy (Huang et al., 2024). With gold feedback available, the separate question is whether an agent converts it into a persistent change that helps on unseen inputs. Conversely, machine-readable validity can improve while the recovered visual structure remains unchanged. Evaluating multimodal improvement requires both a statewise hidden test and a trace that connects visible feedback to persistent behavior.

Existing recursive self-improvement research makes iterative change executable through code, prompts, memories, skills, synthesized data, or agent harnesses (Zelikman et al., 2024; Robeyns et al., 2025; Zhang et al., 2026; Lee et al., 2026; Wu et al., 2026; Meng et al., 2026). These evaluations primarily target coding, terminal use, or data research. One-shot multimodal benchmarks instead evaluate a fixed solver. Neither setting isolates whether a multimodal agent can convert exact visual feedback into a bounded strategy revision whose effect survives on a common hidden test.

FLOWRSI operationalizes this question as the trajectory in Figure 1. Model weights, the host, and the task remain fixed. The evaluated agent chooses which labeled cases to inspect, which self-tests to

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

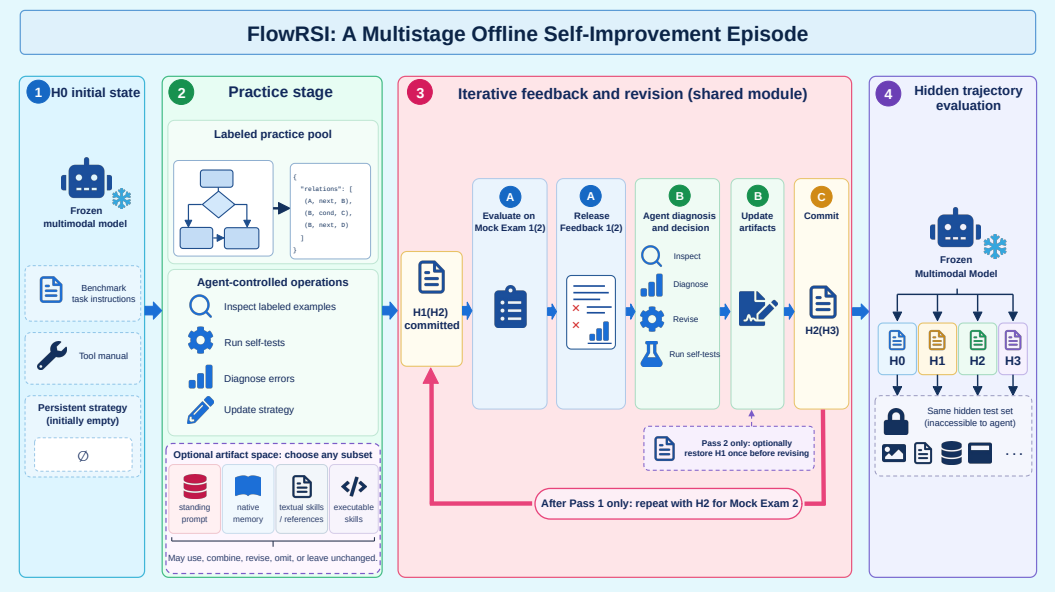


Figure 1: The FLOWRSI improvement episode. A frozen multimodal model moves from H_0 to three committed persistent strategies. The mock-exam, feedback, and revision module runs twice, first producing H_2 and then H_3. The second revision may restore H_1 once before editing. The forward edge into hidden evaluation follows the second pass; the host independently replays H_0–H_3 on the same test set without returning test feedback to the agent.

run, and which optional persistent artifacts to update. It commits three strategy states across practice and two mock-feedback revisions. The first mock asks whether an initial diagnosis becomes a useful revision; the second tests whether the agent can diagnose the revised strategy itself, preserve its gains, or use a one-time rollback. A hidden evaluator then replays H_0, H_1, H_2, H_3 independently on the same test images. The final state is mandatory, so a harmful last revision cannot be replaced post hoc by the best checkpoint.

Flowchart relation extraction supplies the supervision needed to inspect this process. The task requires node-text recognition, directed-edge tracing through branches and crossings, relation typing, and complete structured serialization. Program-derived graphs expose missing, spurious, reversed, mislabeled, and containment relations without an LLM judge. Multiple renderers, scan corruption, and writer-disjoint hand drawings test whether a learned strategy transfers beyond the examples that produced it.

Our contributions are:

- We formulate multimodal recursive self-improvement as an auditable, frozen-weight episode: agents control evidence acquisition and persistent revisions, while two mock exams and same-test replay expose the consequences of every commitment.

- We build a 5,484-image, exactly scored flowchart benchmark that separates structural parsing from output validity and familiar visual conditions from held-out corruption and human handwriting.

- We identify three distinct bottlenecks across nine systems—feedback interpretation, revision stability, and transfer—using coupled outcome measures, paired repair/break analysis, action traces, and controlled artifact-removal replays. These distinctions point to more targeted improvement objectives than endpoint score alone.

2 Related Work

2.1 Recursive Self-Improvement

Parametric self-improvement changes the model or a learned component through self-training, synthetic-data generation, fine-tuning, reinforcement learning, or optimizer and training-loop search. RSIBench-Data evaluates an agent that improves a student through autonomous data research under a fixed post-training stack (Meng et al., 2026); MetaRSI organizes improvement operators across model, data, and harness levels (Tan et al., 2026). These systems can change task competence directly through weight updates, but their outcomes combine the quality of the agent's research decisions with the efficacy of the selected training procedure.

Non-parametric self-improvement instead modifies the computation surrounding frozen weights. STOP revises a code-generation scaffold (Zelikman et al., 2024); self-improving coding agents, the Darwin Gödel Machine, RHI, and ModularRSI extend the editable object to monolithic or modular agent implementations (Robeyns et al., 2025; Zhang et al., 2026; Lee et al., 2026; Wu et al., 2026). Reflection and experience systems retain verbal feedback or reusable procedures across attempts (Shinn et al., 2023; Zhao et al., 2023), while Voyager and SkillLearnBench study executable or iteratively generated skills (Wang et al., 2023; Zhong et al., 2026). SelfMem and PAST-Bench focus on persistent memory and later-episode reuse (Yang et al., 2026; Xue et al., 2026). FlowRSI belongs to this branch: the model weights remain frozen, while the agent selects evidence and versions a bounded persistent inference strategy. Its central measurement is not whether an artifact was written, but whether the submitted artifact state changes unseen visual predictions when independently replayed.

2.2 Multimodal Structure Understanding

Multimodal agent benchmarks often combine visual grounding with long-horizon environment execution. OSWorld, for example, measures whether an agent can interpret interfaces and complete computer tasks (Xie et al., 2024). Such environments are realistic, but state changes and execution variance make it difficult to replay several learned strategies on exactly the same visual inputs. VideoHarness-RSI studies executable context construction around a frozen video-language model (Xu & Chen, 2026); FlowRSI shares the frozen-backbone perspective but evaluates how the model itself acquires evidence and persists a visual parsing strategy.

Diagram and flowchart understanding offer deterministic structural targets. FlowGen generates source graphs through multiple rendering backends and controls topology and difficulty (Shi et al., 2026). FC-B contributes human-drawn diagrams whose writer variation is unavailable from programmatic renderers (Bresler et al., 2016). Prior work uses these data to assess one-shot recognition or parsing. FlowRSI uses their exact supervision and complementary appearance shifts to study a different capability: whether feedback from visible examples is transformed into an effective persistent strategy across successive hidden replays.

3 The FlowRSI Benchmark

3.1 Task Definition

Let M denote a fixed multimodal model and A the fixed host runtime. At stage t, the evaluated system is

$$\mathcal{S}_t = (M, A, H_t), \qquad H_t = (P_t, R_t, T_t, E_t), \tag{1}$$

where H_t is the committed persistent strategy. Its optional components are a standing prompt P_t, native memory R_t, textual skills or references T_t, and executable skills E_t. Any component may be created, revised, retained, or omitted. Writing an artifact, loading it, and invoking it are distinct recorded events.

For one flowchart image, the solver must return exactly one JSON array whose elements are triples $[\texttt{source}, \texttt{relation}, \texttt{target}]$. The array represents an unordered relation set; duplicate triples are collapsed. Relations include directed transitions and direct `partOf` membership when present in the canonical graph. A correct prediction must recover node text, trace the relevant directed edges,

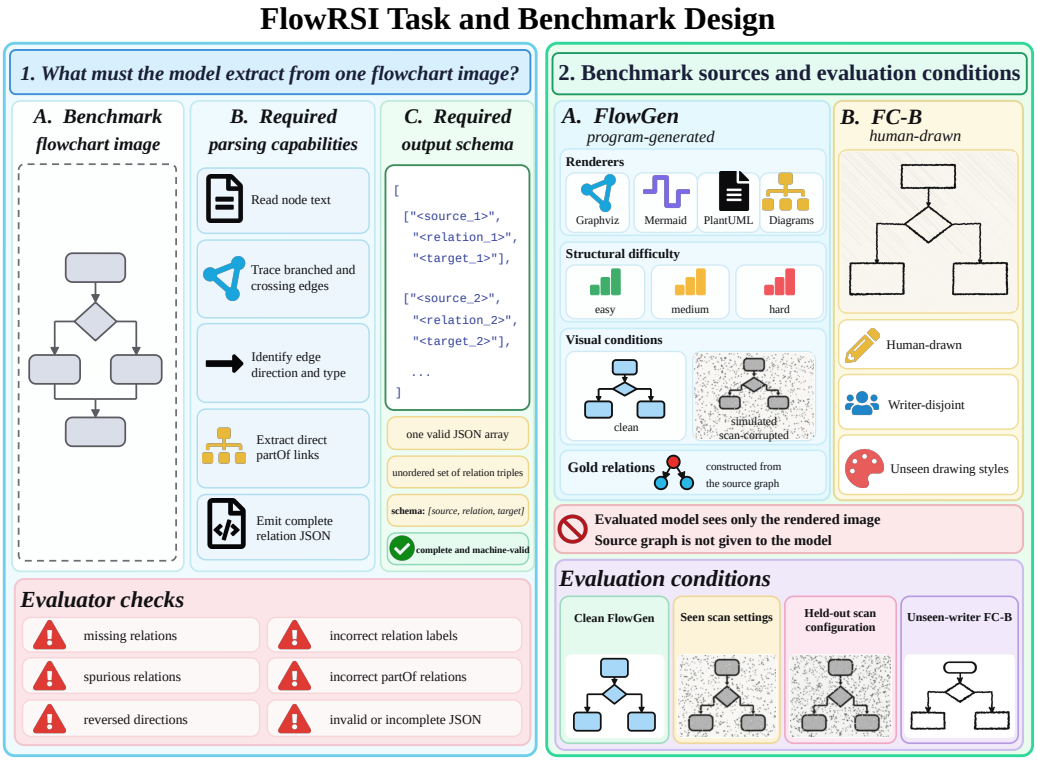


Figure 2: Task and benchmark design. The solver maps one flowchart image to its complete unordered gold relation set. FlowGen supplies controlled renderer, topology, and scan variation; FC-B supplies writer-disjoint human drawings. The test separates clean FlowGen, seen scan corruption, held-out scan corruption, and unseen-writer FC-B.

resolve relation type and direction, and serialize the complete set. Isolated-node recovery is outside the current contract.

The agent observes stage-appropriate evidence F_t, executes an action trace τ_t, and proposes the next strategy. The host never exposes hidden-test images, predictions, labels, or scores. We call this setting *bounded non-parametric self-improvement*: the evaluated model autonomously changes its reusable inference strategy, but does not modify weights or the improvement algorithm itself.

3.2 Dataset Design

Figure 2 summarizes the two data sources. FlowGen forms 12 renderer–difficulty cells from Graphviz, Mermaid, PlantUML, and Diagrams crossed with easy, medium, and hard structural bands. Its source programs permit deterministic reconstruction of relation labels. We audit source/render correspondence, edge endpoints, relation vocabulary, duplicate text, image identity, and cross-split structural collisions; inconsistent hierarchy labels are resolved as direct `partOf` membership from the canonical programs. Here `partOf` is a relation type, not an output format.

FC-B (Bresler et al., 2016) adds human-drawn flowcharts under writer-disjoint splits. Practice uses the original training group; its validation group is divided into two writer-disjoint mock groups; the original test writers remain hidden until final evaluation. The final test therefore measures transfer from synthetic renderers and earlier handwriting to previously unseen writers. FlowGen test images add two corruption axes: a corruption family represented during the visible episode and a held-out corruption configuration that is not available in practice or either mock.

Table 1 reports the frozen v11/r3 composition. The practice pool is available from episode start. Mock 1 is inaccessible until H_1 is committed, and Mock 2 is inaccessible until H_2 is committed.

Table 1: Frozen v11/r3 benchmark composition and information release. Counts are images. The test total comprises 576 clean FlowGen, 288 seen-scan FlowGen, 288 held-out-scan FlowGen, and 196 unseen-writer FC-B images.

Split	FlowGen	FC-B	Total	Agent visibility
Practice	2700	280	2980	episode start
Mock 1	480	84	564	after committing H_1
Mock 2	480	112	592	after committing H_2
Test	1152	196	1348	never
Total	4812	672	5484	–

Test data remain inaccessible throughout the episode. Mocks are feedback stages, not researcher-controlled validation checkpoints: every submitted state is retained, and the final report does not replace H_3 with the best intermediate state.

3.3 Improvement Episode

The episode produces the state sequence

$$H_0 \xrightarrow{\text{practice}} H_1 \xrightarrow{\text{Mock 1 feedback}} H_2 \xrightarrow{\text{Mock 2 feedback}} H_3. \tag{2}$$

The initial state contains no learned persistent artifact. During practice, the agent can inspect labels, request predictions from its current solver, analyze failures, update any subset of persistent artifacts, and submit H_1. Mock 1 then evaluates H_1 on a previously unavailable set. Only after the complete exam does the host release aggregate metrics and queryable per-example evidence, including predictions, gold relations, literal differences, and images. The agent performs its own diagnosis and commits H_2. Mock 2 repeats this process. During the final revision, the agent may continue from H_2 or restore H_1 once before producing H_3.

Each learning or revision stage permits 80 controller decisions. The agent sees both the total and remaining budget, with the final ten calls identified as a submission reserve. Across the episode it may request 600 practice predictions. Prompt, memory, loaded references, and executable-skill descriptions share a 4,096-token persistent-state budget. The largest final loaded state in this suite uses 2,688 tokens, so this cap does not truncate any reported final strategy. If the model exhausts its disclosed decision budget without committing, the host freezes the last legal state and records forced submission.

The agent may combine action classes rather than select one fixed learning mode (Table 7 in the appendix). The standing prompt is injected into each solver conversation; memory uses the runtime's native persistence semantics; textual skills are reusable procedures or references; executable skills are scripts with declared entry points. Revision rollback reconstructs memory and the loaded-artifact cache from the restored state before further edits.

The operating manual specifies the task, files, tools, action parameters, budgets, feedback schedule, artifact semantics, submission rule, and failure behavior. It provides no demonstration trajectory. The host negotiates advertised tools from each runtime's actual schema while retaining family-native image handling and tool serialization. Thus all systems receive the same scientific action space without being prompted to call tools that their runtime does not expose.

4 Evaluation Methodology

4.1 Metrics

The primary metric is macro-averaged *strict JSON relation-set F1*. The entire response must be exactly one schema-valid JSON array, after which triples are compared as unordered sets. This metric jointly tests recovered structure and compliance with the machine-consumable output contract. We also report exact-set match, precision, recall, and strict validity.

Content F1 is diagnostic. A deterministic parser may recover one unique complete array surrounded by a Markdown fence or short prose, but it never repairs endpoints, edge direction, relation labels,

truncation, or multiple candidate arrays. The format gap at state H_t is

$$G_{\text{fmt}}(H_t) = F_1^{\text{content}}(H_t) - F_1^{\text{strict}}(H_t). \tag{3}$$

A falling gap with unchanged Content F1 indicates output-contract repair. Rising Content F1 with stable validity provides stronger evidence that the recovered visual structure improved.

For the hidden trajectory, we report

$$\Delta_{\text{final}} = S_{\text{test}}(H_3) - S_{\text{test}}(H_0), \qquad L_{\text{bf}} = \max_{t \in \{0,1,2,3\}} S_{\text{test}}(H_t) - S_{\text{test}}(H_3), \tag{4}$$

along with the four-state trajectory AUC. Paired item bootstrap intervals quantify state differences for a fixed agent trajectory. We retain negative and forced trajectories and never select the best intermediate state as the endpoint.

The analysis further tracks repair and destruction on paired hidden items, mock-to-test transfer, artifact dependency, and clean, seen-scan, held-out-scan, and unseen-writer scores. The full observable-to-capability mapping is in Table 6 of the appendix.

4.2 Evaluated Systems

The formal suite covers Qwen3.5 at 9B, 27B, 122B-A10B, and 397B-A17B scales; Qwen3.8-27B; InternVL3.5-38B; GLM-4.6V variants; and Gemma-4-31B. This range includes small dense models, mid-scale open VLMs, and large mixture-of-experts systems. Model-size comparisons are descriptive because architecture, multimodal pretraining, active parameters, and tool tuning vary across families.

Hermes provides the common controller loop, typed action validation, workspace isolation, checkpointing, and trace schema for *all* evaluated systems, including Qwen; Qwen-Agent is not a separate evaluation path here. Each model retains its supported chat template, visual preprocessing, stop behavior, and tool serialization. Every system receives the same task, stage schedule, action semantics, data, output contract, and scorer. This is a bounded, non-parametric operationalization of recursive self-improvement: the agent chooses evidence and revises the strategy consumed by its future solver, but neither model weights nor the revision algorithm itself change.

Before formal scoring, each configuration must pass qualification on held-out real images, every advertised action class, persistent-state loading, executable-skill invocation when supported, rollback cache refresh, strict-output parsing, per-item failure isolation, and atomic submission. A malformed or repeated tool call with a healthy service is model behavior. A model-side failure for one sample produces an invalid prediction for that item without terminating the remaining evaluation. Per-model serving and decoding configurations are retained with the executable evaluation records.

5 Results

5.1 Main Results

Table 2 and Figure 3 report the strict hidden-test trajectory. Each state score is averaged over three evaluations on the same 1,348-image test set. Initial task competence and ability to improve it are strikingly different. Gemma-4-31B has the strongest H_0 (53.05) but ends slightly lower (52.47); Qwen3.5-27B starts at 3.36 and reaches 58.83. Four of nine systems improve at H_3. Five systems completed three autonomous strategy commitments; InternVL and Gemma reached an early final state after two, and the two GLM systems after one. Their scored endpoint states remain in the table, but those rows do not demonstrate two completed autonomous feedback revisions. The paired-item intervals exclude zero for Qwen3.5-27B, Qwen3.5-122B, and Qwen3.5-397B, conditional on the observed agent trajectory; they do not quantify between-episode variability.

The statewise view changes the ranking. The 9B system partially recovers between H_1 and H_2 but loses 25.78 points at the last revision; its 28.30-point best-to-final loss would disappear in a best-checkpoint report. Qwen3.5-122B improves during practice, loses 7.89 points after Mock 1, and recovers 7.44 points after Mock 2. Qwen3.5-397B undergoes the same improve–regress–recover pattern at smaller magnitude. Consequently, improvement is neither monotonic nor explained by model scale. Gemma's high initial score but early finalization further separates base task competence from the ability to complete the revision protocol.

Table 2: Hidden-test trajectories on the same 1,348 images, with each state score averaged over three evaluations. All values are percentage points of strict JSON relation-set F1. $\Delta = H_3 - H_0$; the conditional 95% interval is a 1,000-resample paired item bootstrap of this episode's predictions, not an interval across agent episodes. AUC is trapezoidal area over three unit-spaced transitions, divided by three; best-to-final loss is $\max_t H_t - H_3$. InternVL and Gemma ended early after two autonomous commits, and both GLM systems after one.

System	H_0	H_1	H_2	H_3	Δ [95% CI]	AUC	Loss
Qwen3.5-9B	30.88	25.52	28.36	2.58	-28.30 [$-30.00, -26.57$]	23.54	28.30
Qwen3.5-27B	3.36	31.81	29.80	58.83	$+55.47$ [53.41,57.69]	30.90	0.00
Qwen3.5-122B-A10B	8.14	25.14	17.24	24.68	$+16.54$ [14.63,18.59]	19.60	0.46
Qwen3.5-397B-A17B	19.83	20.51	18.90	21.89	$+2.05$ [0.09,4.17]	20.09	0.00
Qwen3.8-27B	5.28	0.00	0.00	0.00	-5.28 [$-6.49, -4.20$]	0.88	5.28
InternVL3.5-38B	39.92	39.04	41.93	40.84	$+0.92$ [$-0.30, 2.20$]	40.45	1.09
GLM-4.6V-Flash	0.12	0.13	0.18	0.10	-0.02 [$-0.20, 0.16$]	0.14	0.08
GLM-4.6V	0.05	0.00	0.07	0.04	-0.01 [$-0.16, 0.13$]	0.04	0.03
Gemma-4-31B	53.05	52.65	52.85	52.47	-0.58 [$-1.20, 0.02$]	52.75	0.58

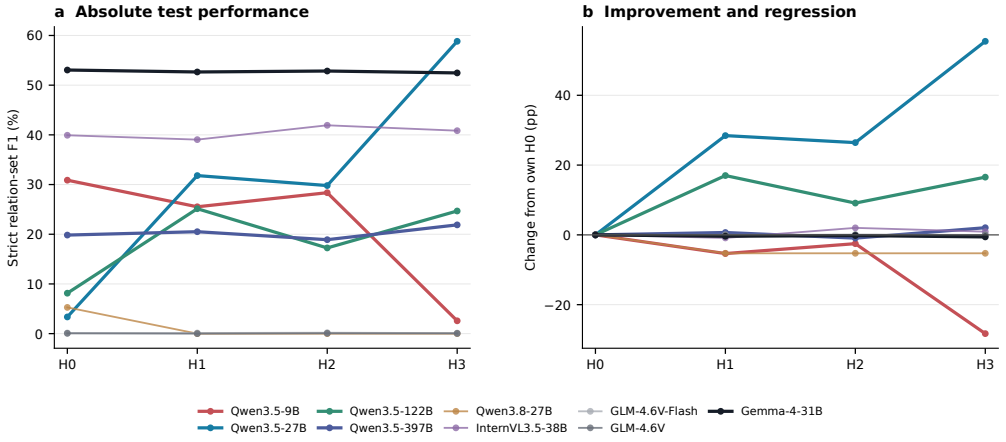


Figure 3: Four-state learning trajectories. Panel (a) compares absolute strict relation-set F1. Panel (b) centers every system at its own H_0, separating base task competence from subsequent improvement and regression.

Strict F1 alone cannot determine whether a gain comes from better graph recovery or improved serialization. Table 3 therefore contrasts Content F1, strict validity, and the format gap at H_0 and H_3. The joint pattern supports three distinct conclusions: content improvement with stable validity, contract repair with stable content, or a mixed change affecting both. We do not attribute all strict-F1 movement to visual reasoning.

The contrast between strict and content metrics changes the interpretation of the trajectories. Qwen3.5-27B gains 55.47 strict points, but its Content F1 gains only 6.88; most of the strict gain accompanies the shift from 6.08% to 77.30% valid responses. Qwen3.5-122B improves both content (+13.79) and validity (+32.27). Qwen3.5-397B gains 5.70 content points while validity falls 3.56 points. GLM-4.6V's near-zero strict score alongside 42.41 Content F1 illustrates why the diagnostic parser is needed even when strict F1 remains primary.

5.2 Feedback Interpretation

Appendix Figure 5 and Table 10 connect what agents inspected to what they committed. The API records overview requests, self-tests, status queries, submissions, and rollback; native edits are preserved in snapshots. Crucially, receiving gold answers is only evidence availability. The agent must infer whether the failure concerns the image, relation semantics, or the output contract, then encode the right correction in a loaded artifact.

Table 3: Content and output validity at the initial and final states. The format gap is Content F1 minus strict F1.

	Content F1		Valid (%)		Gap	
System	H_0	H_3	H_0	H_3	H_0	H_3
Qwen3.5-9B	33.79	22.06	68.18	5.49	2.91	19.48
Qwen3.5-27B	61.95	68.83	6.08	77.30	58.59	10.00
Qwen3.5-122B	50.14	63.93	10.09	42.36	42.00	39.25
Qwen3.5-397B	56.85	62.55	35.09	31.53	37.02	40.66
Qwen3.8-27B	14.58	2.26	11.35	0.00	9.30	2.26
InternVL3.5-38B	41.74	42.35	91.25	91.69	1.82	1.51
GLM-4.6V-Flash	23.37	24.34	0.59	0.45	23.25	24.24
GLM-4.6V	42.40	42.41	0.07	0.07	42.34	42.37
Gemma-4-31B	53.09	52.59	96.22	94.66	0.04	0.12

Table 4: Mock-to-test transfer for both revisions. Mock post-scores are computed host-side after each new state is committed and are hidden from the agent.

System	Revision	Mock before	Mock after	Δ Mock	Δ Test
Qwen3.5-9B	$H_1 \to H_2$	24.52	26.95	$+2.43$	$+2.84$
	$H_2 \to H_3$	27.11	2.55	-24.57	-25.78
Qwen3.5-27B	$H_1 \to H_2$	31.48	31.46	-0.01	-2.01
	$H_2 \to H_3$	27.26	54.17	$+26.91$	$+29.03$
Qwen3.5-122B	$H_1 \to H_2$	25.41	19.07	-6.33	-7.89
	$H_2 \to H_3$	17.03	26.14	$+9.11$	$+7.44$
InternVL3.5-38B	$H_1 \to H_2$	38.46	41.02	$+2.56$	$+2.89$
	$H_2 \to H_3$	39.86	39.43	-0.43	-1.09

The trace gives a concrete attribution contrast. Qwen3.5-27B starts with 61.95 Content F1 but only 6.08% strictly valid responses. Its prompt revisions increasingly insist that the first output character be `[`, the last be `]`, and no Markdown or prose surround the array; final validity reaches 77.30%, while Content F1 rises by 6.88 points. This is a mixed gain dominated by contract repair, not 55.47 points of new diagram understanding. Qwen3.8-27B makes the opposite diagnosis: its committed prompt specifies a JSON *object* with a `relations` key, contradicting the benchmark's required top-level array, and carries that instruction through H_3. Its final strict validity and F1 are both zero. This observed schema mismatch explains the strict-score collapse more directly than an assertion that the model lost all visual perception; its final Content F1 remains 2.26. GLM-4.6V likewise retains 42.40–42.41 Content F1 while strict validity stays below 0.1%, illustrating a persistent attribution/contract bottleneck. Future improvement agents need to separate parseable-content feedback from contract feedback, and verify that a proposed rule is compatible with the actual evaluator before promoting it to persistent strategy.

5.3 Revision Stability

The first mock tests whether the agent can revise its initial strategy; the second asks a harder question: after a revision, can it detect and repair damage in the revised strategy? For revision j, Table 4 compares the host-side change on the completed mock with that of the same state transition on the hidden test. These four systems have completed two-stage host-side mock replays; they are not a random sample of the nine systems. Post-revision mock replay is analysis-only and is never returned to the agent. The comparison measures observed direction agreement, not a causal effect of feedback: there is no no-feedback or shuffled-feedback controller. Figure 4 separates beneficial repair from revision damage and reports the same transition across visual strata.

All eight measured mock revisions have the same change direction on mock and hidden test, *including harmful revisions*. The 9B final prompt expands to 2,688 loaded tokens and contains a long catalog of text, relation, and format rules; yet after this commitment Mock 2 drops 24.57 points and the hidden test drops 25.78. Its final validity falls from 68.18% at H_0 to 5.49%; a matched no-artifact replay recovers 27.48 points (Table 5). The evidence identifies a destructive persistent state, although it does not isolate which sentence caused the damage. Qwen3.5-122B instead loses 6.33 mock and

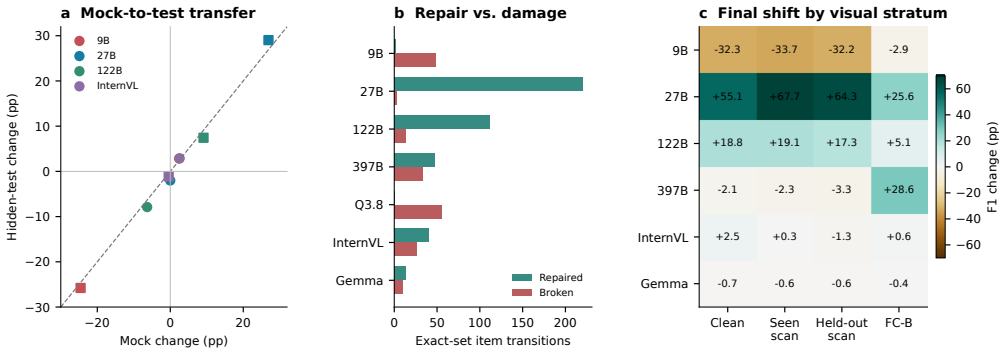


Figure 4: Feedback transfer and revision damage. (a) Eight revision-level mock/test change pairs from four systems; the diagonal is equal change. (b) $H_0 \rightarrow H_3$ exact-set wrong-to-right and right-to-wrong counts on paired items. (c) Strict-F1 change in clean, seen-scan, held-out-scan, and unseen-writer subsets.

7.89 test points after the first feedback, uses the allowed rollback in its second revision, then gains 9.11 mock and 7.44 test points. That recovery shows the value of preserving an earlier state, but its H_3 still trails its own H_1 slightly. At the item level, 27B repairs 220 initially wrong exact sets and breaks 3, whereas 9B repairs 2 and breaks 48. Thus even when mock feedback tracks hidden-test direction, it does not guarantee the agent will act on that signal safely. A revision mechanism should test proposed artifact changes against a retained canary set, compare repairs with regressions, and support state restoration before final commitment.

5.4 Strategy Transfer and Artifact Dependence

Transfer has two meanings here: carrying a feedback-derived rule into the solver and applying it outside the domain that supplied the evidence. Within FlowGen, the seen-to-held-out scan comparison isolates a visual corruption shift. FC-B instead measures a broader cross-source, unseen-writer shift: drawing style, graph structure, text, and annotation conventions can all differ, so its gap cannot be attributed solely to appearance. At H_3, 27B scores 61.09 on clean, 69.55 on seen-scan, and 65.87 on held-out-scan FlowGen, but only 26.06 on unseen-writer FC-B; Gemma scores 61.68 versus 5.47 on clean versus FC-B. The larger cross-source gap is a harder transfer stress test, not a controlled estimate of handwriting's isolated effect. The 397B state is different: FC-B rises from 5.91 to 34.49 while all three FlowGen strata fall modestly. Its aggregate gain therefore masks cross-source specialization rather than uniformly improved graph parsing.

Artifact-dependency replay tests whether a learned artifact actually helps the final solver. We chose 9B, 27B, and 122B as diagnostic cases with different artifact bundles and trajectory shapes, not as a population-wide estimate. On the same hidden images, removing all artifacts raises 9B by 27.48 points but lowers 27B by 56.43; removing the 122B textual skill and script together raises its score by 9.68 (Appendix Table 5). These conditional, non-additive interventions expose both beneficial dependencies and harmful interactions. Authored artifacts should therefore be tested for marginal utility and retained competence, not counted as successful learning merely because they exist.

6 Conclusion

FlowRSI treats multimodal self-improvement as accountable decisions, not an endpoint gain. Across nine frozen-weight agents, the same feedback opportunity produces contract repair, persistent schema error, regression recovery, destructive revision, and cross-source specialization. Two mock exams expose whether an agent can form a correction and then revise that correction; same-test replay, content/validity diagnostics, paired repair and damage, and artifact interventions distinguish their consequences. Future agents should attribute errors to the right layer, check revisions against retained competence before commitment, and retain only strategies that survive domain shift. The benchmark measures these capabilities while keeping initial task accuracy separate from improvement behavior.

AI USE STATEMENT

Generative AI tools assisted with evaluation-code implementation, data processing, and language polishing. The authors conceived the research questions, designed the benchmark, analyzed the results, wrote the manuscript, and are responsible for its scientific claims and final content.

REFERENCES

Martin Bresler, Daniel Prusa, and Vaclav Hlavac. Online recognition of sketched arrow-connected diagrams. *International Journal on Document Analysis and Recognition*, 19(3):253–267, 2016. doi: 10.1007/s10032-016-0269-z.

Jie Huang, Xinyun Chen, Swaroop Mishra, Huaixiu Steven Zheng, Adams Wei Yu, Xinying Song, and Denny Zhou. Large language models cannot self-correct reasoning yet. In *International Conference on Learning Representations*, 2024.

Hyunin Lee, Jinglue Xu, Jeffrey Seely, Donghyun Lee, Matei Zaharia, and Yujin Tang. Recursive harness self-improvement. *arXiv preprint arXiv:2607.15524*, 2026.

Fanqing Meng, Lingxiao Du, Qiguang Chen, Ziqi Zhao, Haocheng Lu, Mengkang Hu, and Michael Qizhe Shieh. RSIBench-Data: Benchmarking data-centric research for recursive self-improvement. *arXiv preprint arXiv:2607.25886*, 2026.

Maxime Robeyns, Martin Szummer, and Laurence Aitchison. A self-improving coding agent. *arXiv preprint arXiv:2504.15228*, 2025.

Kaiwen Shi, Sichen Liu, Ziyue Lin, Hangrui Guo, and Gong Cheng. FlowGen: Synthesizing diverse flowcharts to enhance and benchmark MLLM reasoning. In *International Conference on Learning Representations*, 2026.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. In *Advances in Neural Information Processing Systems*, 2023.

Zihan Tan, Leixin Sun, Zitong Shi, Yitao Liu, Jiajun Wu, Nathaniel Brooks, Jiaru Qian, Xiaoran Shang, Suyuan Huang, Yi Ding, et al. MetaRSI / RSI2: A meta-recursive self-improving system for recursive self-improving systems themselves. *arXiv preprint arXiv:2609.06396*, 2026. URL https://arxiv.org/abs/2609.06396.

Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Voyager: An open-ended embodied agent with large language models. *arXiv preprint arXiv:2305.16291*, 2023.

Siwei Wu, Jincheng Ren, Yizhi Li, Haau-Sing Li, Chengran Yang, Yuxuan Zhang, Weicheng Gu, Jian Yang, Riza Batista-Navarro, Chuanyi Zhang, Xianglong Liu, Ming Zhou, Bryan Dai, and Chenghua Lin. ModularRSI: Modular and generalizable recursive harness self-improvement. *arXiv preprint arXiv:2609.14857*, 2026.

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.

Guoyang Xu and Hao Chen. VideoHarness-RSI: Recursive harness self-improvement for long-video understanding with frozen vision-language models. *arXiv preprint arXiv:2608.24302*, 2026. URL https://arxiv.org/abs/2608.24302.

Shuhan Xue, Zixin Ding, Yichen Shen, Yinjie Wang, Zhenfei Yin, Yingcheng Wu, Yuxin Chen, Mengdi Wang, and Ling Yang. PAST-Bench: Benchmarking the foundations of recursive self-improvement in personal agents. *arXiv preprint arXiv:2608.04003*, 2026.

Table 5: Artifact-dependency replay on the 1,348-example hidden test. Strict relation-set F1 is shown as a percentage; Δ is the percentage-point change from the same model's full final state. Each reduced state removes exactly the named component, except *no skill bundle*, which jointly removes the textual skill and its executable script. Effects are conditional on retained components, not additive attributions.

Model	Retained-state intervention	Strict F1 (%)	Δ (pp)
Qwen3.5-9B	Full state	2.47	–
	No standing prompt	14.65	+12.18
	No native memory	6.27	+3.81
	No textual skill	4.81	+2.34
	No artifacts	29.95	+27.48
Qwen3.5-27B	Full state	60.13	–
	No standing prompt	4.55	−55.58
	No textual skill	49.04	−11.09
	No artifacts	3.71	−56.43
Qwen3.5-122B	Full state	24.81	–
	No standing prompt	2.74	−22.07
	No textual skill	32.21	+7.40
	No executable script	27.57	+2.76
	No skill bundle	34.49	+9.68
	No artifacts	8.17	−16.64

Shu Yang, Junchao Wu, Derek F. Wong, and Di Wang. SelfMem: Self-optimizing memory for AI agents. *arXiv preprint arXiv:2607.03726*, 2026.

Eric Zelikman, Eliana Lorch, Lester Mackey, and Adam Tauman Kalai. Self-taught optimizer (STOP): Recursively self-improving code generation. In *Conference on Language Modeling*, 2024.

Jenny Zhang, Shengran Hu, Cong Lu, Robert Lange, and Jeff Clune. Darwin Gödel machine: Open-ended evolution of self-improving agents. In *International Conference on Learning Representations*, 2026.

Andrew Zhao, Daniel Huang, Quentin Xu, Matthieu Lin, Yong-Jin Liu, and Gao Huang. ExpeL: LLM agents are experiential learners. *arXiv preprint arXiv:2308.10144*, 2023. URL `https://arxiv.org/abs/2308.10144`.

Shanshan Zhong, Yi Lu, Jingjie Ning, Yibing Wan, Lihan Feng, Yuyi Ao, Leonardo F. R. Ribeiro, Markus Dreyer, Sean Ammirati, and Chenyan Xiong. SkillLearnBench: Benchmarking continual learning methods for agent skill generation on real-world tasks. *arXiv preprint arXiv:2604.20087*, 2026. URL `https://arxiv.org/abs/2604.20087`.

Interpretation. The 27B standing prompt carries most of its measured gain: removing it costs 55.58 points, while removing its textual skill costs 11.09 points conditional on the rest of the final state. The 122B standing prompt is also helpful (22.07 points lost when removed), but removing its textual skill and executable script together improves F1 by 9.68 points. Each delta uses that intervention suite's own full-state replay as its baseline, so it is neither an additive share of the episode-level gain nor an estimate over repeated autonomous episodes. The smaller 122B component differences are diagnostic rather than a stable population effect.

A Reproducibility Statement

The release includes frozen data manifests, label reconstruction, the agent manual and action schemas, full trajectories, committed artifact snapshots, per-state predictions, evaluator outputs, and runtime receipts. Hidden-test information is isolated from the agent; configurations and audit rules are detailed below.

Table 6: Diagnostic dimensions and observables. These analyses localize different links in the improvement process; none replaces strict relation-set F1 as the primary outcome.

Dimension	Observable	Scientific question	Localized capability
Budget use	actions, inspected examples, practice predictions, artifact tokens	does the agent acquire useful evidence before resources expire?	acquisition and efficiency
Artifact lifecycle	authored, loaded, invoked, and replay-ablated artifacts	which persistent representation changes inference?	persistence and harness use
Feedback transfer	mock change versus same-revision hidden-test change	does visible feedback induce reusable behavior?	diagnosis and induction
Revision damage	wrong-to-right and right-to-wrong items or relations	does a repair preserve previously correct capability?	negative transfer
Visual transfer	clean, seen scan, held-out scan, unseen-writer changes	does the strategy generalize beyond visible appearance?	multimodal execution
Governance	submission timing, forced freeze, rollback, best-to-final loss	can the agent stop, revise, or restore effectively?	control policy

B Evaluation Contract and Statistical Details

Primary score. For image i with gold set G_i and strict prediction P_i, we calculate $\mathrm{F1}_i = 2|P_i \cap G_i|/(|P_i| + |G_i|)$, with the standard exact-empty-set convention. Invalid JSON, an invalid schema, or an unrecoverable model-side sample failure has $\mathrm{F1}_i = 0$. The benchmark score is the arithmetic mean across images rather than micro-averaging all triples. The same strict parser and relation normalizer apply to every model and state. Exact-set accuracy requires $P_i = G_i$ and a valid response. Content F1 applies only a bounded extraction of one complete array from otherwise non-conforming output; it does not repair the predicted relations.

Statewise comparison. Every hidden state uses the same frozen test ID list and gold relations; its reported score averages three evaluations. We pair records by ID. The reported $H_3 - H_0$ interval resamples the 1,348 paired item differences 1,000 times with replacement, using seed 2027, then takes the 2.5 and 97.5 percentiles. Trajectory AUC is $[S(H_0)/2 + S(H_1) + S(H_2) + S(H_3)/2]/3$. Best-to-final loss is $\max_t S(H_t) - S(H_3)$. Exact-set repair counts items incorrect at H_0 and correct at H_3; destruction is the reverse. Their difference is not the macro-F1 change, because partially correct relation sets also contribute to F1.

Information boundary. Practice labels are available from the start. Each mock set and its gold relations remain inaccessible until its preceding state has been committed and the entire mock completed. The hidden test is inaccessible during the episode. Host-side mock replays and artifact interventions occur after state submission and do not enter the agent's feedback stream. This preserves a single learning trajectory while allowing controlled analysis of its submitted states.

C Data Construction and Provenance

FlowGen contributes 4,812 images across the four renderers Graphviz, Mermaid, PlantUML, and Diagrams, and its easy, medium, and hard structural bands. Practice contains 2,700 FlowGen images, the mocks 480 each, and the hidden test 1,152. Of the latter, 576 are clean, 288 use scan-corruption families represented in the visible episode, and 288 use a held-out scan configuration. FC-B contributes 672 human-drawn diagrams: 280 practice, 84 Mock 1, 112 Mock 2, and 196 hidden test. The FC-B groups are writer-disjoint across stages. The test therefore contains 576 clean, 288 seen-scan, 288 held-out-scan, and 196 unseen-writer examples.

The graph-to-label transformation retains visible node strings, directed source and target endpoints, edge labels, and direct node-to-group `partOf` membership. An unlabeled edge uses `connectedTo`. Blank routing junctions are collapsed rather than treated as semantic nodes. Triple duplicates are removed. The data build checks graph/source/render identity, image hashes, split membership, relation vocabulary, endpoint existence, and cross-split structure identifiers before freezing the manifests. Inconsistent hierarchy labels are reconstructed as direct membership from canonical source programs before scoring any model; this is not model-dependent adjudication.

D FROZEN AGENT INTERFACE

Table 7: Agent action surface. The named classes are a semantic grouping of typed benchmark calls and native workspace/file operations; exact typed controller calls are listed below.

Class	Agent-controlled choice	Host guarantee	Recorded evidence
Orient	inspect dataset cards, budgets, stage state, and summaries	complete counts, paths, and remaining budgets	calls and viewed state cards
Query	filter IDs by source, renderer, difficulty, corruption, writer, or score	stable IDs and file export	predicates and selected IDs
Inspect	open images, labels, predictions, and released feedback	native image transport and file manifests	viewed items and order
Practice	run current solver on a selected batch	charge completed predictions against 600-item allowance	batch composition, outputs, errors
Persist	revise prompt, memory, textual reference, or executable skill	validate paths, token limits, and loader visibility	artifact diff, state ID, invocation
Govern	submit, continue from H_2, or restore H_1 once	atomic commits, cache refresh, immutable lineage	submissions and rollback

The benchmark's typed controller API has five calls: `get_overview()` returns the current phase, data paths, score coverage, and remaining budgets; `practice(sample_ids | ids_file)` starts an asynchronous self-test on selected training IDs; `practice_status(job_id)` waits for that job and returns paths to its predictions and summary; `submit_stage(wait_for_practice=true)` atomically freezes the current strategy; and `rollback()` restores the H_1 artifacts once, only in the second revision. The agent uses its native file and terminal interface to inspect images and indices, filter examples, and edit prompt, memory, or skills in the workspace. Thus a model may query thousands of examples through files without an arbitrary 20- or 50-row tool limit.

The fixed allowance is 80 controller decisions per learning/revision stage, 600 practice predictions over the episode, and 4,096 loaded persistent-artifact tokens. The prompt, native memory, textual skill descriptions, and executable-skill descriptions share the latter allowance. Artifact files also have a 262,144-byte cap, and native terminal operations have a 60-second timeout. The operating manual exposes budgets, valid parameters, stage transitions, and submission semantics before the agent acts. The final ten controller decisions are identified as a submission reserve; this is not an extra budget. A practice request can provide distinct IDs directly or a workspace-local JSON array file. The agent may inspect the catalog and index files before selecting IDs.

Artifacts are content-addressed at each submission. Prompt injection, native memory reload, textual skill discovery, and executable-script availability are reconstructed from the committed snapshot. Removing a component in Table 5 changes only that submitted component and then reruns inference on the same hidden ID set. During rollback, both persisted files and the live loader cache are restored before further changes. The episode can therefore combine artifacts, leave any unchanged, or submit none; it is not a three-way choice among prompt, memory, and skill.

E Additional Quantitative Results

Table 8: Strict relation-set F1 by final-test visual stratum. Each cell gives $H_0 \rightarrow H_3$ in percent; stratum sizes are clean 576, seen scan 288, held-out scan 288, FC-B 196. The difficulty bands are upstream structural labels, not a separately calibrated human difficulty scale.

System	Clean	Seen scan	Held-out scan	Unseen-writer FC-B
Qwen3.5-9B	35.27→2.97	37.52→3.78	34.47→2.31	2.94→0.05
Qwen3.5-27B	6.03→61.09	1.84→69.55	1.53→65.87	0.42→26.06
Qwen3.5-122B	11.81→30.61	8.28→27.33	6.05→23.34	0.19→5.32
Qwen3.5-397B	24.38→22.24	23.31→21.03	16.73→13.45	5.91→34.49
Qwen3.8-27B	4.99→0.00	6.82→0.00	5.79→0.00	3.10→0.00
InternVL3.5-38B	46.91→49.37	46.05→46.35	45.08→43.74	2.78→3.40
GLM-4.6V-Flash	0.21→0.22	0.00→0.06	0.00→0.00	0.22→0.00
GLM-4.6V	0.13→0.00	0.00→0.21	0.00→0.00	0.00→0.00
Gemma-4-31B	62.33→61.68	60.69→60.06	58.97→58.42	5.85→5.47

The FC-B gap is not a minor degradation in many systems: both InternVL and Gemma retain roughly 40–60 F1 on FlowGen while scoring below 6 on unseen writers. Qwen3.5-27B transfers more successfully to FC-B than those systems but still has a 35-point gap between clean and human-drawn diagrams at H_3. The 397B final state reverses the usual direction, trading a small loss in each synthetic condition for a large FC-B gain. This is why an aggregate test score alone is insufficient to characterize the learned strategy.

Table 9: Paired $H_0 \rightarrow H_3$ exact-set transitions on 1,348 test images. Repaired means wrong-to-right; broken means right-to-wrong.

System	Repaired	Broken
Qwen3.5-9B	2	48
Qwen3.5-27B	220	3
Qwen3.5-122B	111	13
Qwen3.5-397B	47	33
Qwen3.8-27B	0	55
InternVL3.5-38B	40	26
GLM-4.6V-Flash	0	0
GLM-4.6V	0	0
Gemma-4-31B	14	10

F Artifact and Action Trace Interpretation

The Qwen3.5-9B final snapshot contains a standing prompt, native memory, and a textual flowchart-parsing skill (2,688 loaded tokens). The 27B final state contains a standing prompt and a textual edge-extraction skill (1,579 tokens). The 122B final state contains a standing prompt, a textual flowchart skill, and a script in the skill directory (1,793 tokens). All final loaded states are below the 4,096-token cap. These file-level facts are distinct from whether an individual test solver invoked a native tool. Table 5 is the direct dependency test: a component that is present but ineffective or harmful cannot be credited as a successful learned skill simply because it was authored.

For the benchmark-controller API, 9B records seven overview calls, 11 practice requests, ten practice-status calls, and three submissions; 27B records 6, 21, 19, and 3 respectively; 122B records 11, 17, 14, and 3 plus one rollback; 397B records 10, 14, 16, and 3. These counts exclude native file reads and edits, which are separately present in the full tool transcript. Two agents attempted one 2,980-ID practice request and received a budget rejection while cumulative practice allowance remained. Several early-final events occurred with controller calls still available. These are control decisions rather than proof that a larger fixed allowance would have helped.

Table 10: Trace-grounded cases. Controller-call counts exclude native file reads and edits. Artifacts are loaded final-state snapshots; metric changes compare H_0 and H_3 unless a specific revision is indicated. The table separates an observed action or committed rule from its measured consequence, without assigning causality to one prompt sentence.

System	Observed trajectory or committed state	Measured consequence	Diagnostic reading
Qwen3.5-27B	21 practice requests, 19 status checks; final prompt repeatedly demands a bare array beginning with `[` and ending with `]`	Validity 6.08%→77.30%; Content F1 +6.88; strict F1 +55.47	Predominantly contract repair, with some content gain
Qwen3.8-27B	Final prompt requires a top-level `{"relations":...}` object despite the array-only contract; no new practice request after the initial stage	Final strict validity and F1 0; final Content F1 2.26	An incompatible rule persisted across both revisions
Qwen3.5-9B	Final state loads 2,688 tokens across prompt, memory, and text skill; last revision expands prescriptive parsing rules	Mock 2 -24.57 and test -25.78; no-artifact replay +27.48 versus full state	Destructive revision, not a shortage of written rules
Qwen3.5-122B	Uses the one-time rollback in revision 2; final state combines prompt, textual skill, and executable script	$H_1 \rightarrow H_2$ test -7.89, $H_2 \rightarrow H_3$ +7.44; removing skill bundle +9.68	State restoration helps, yet artifact components conflict

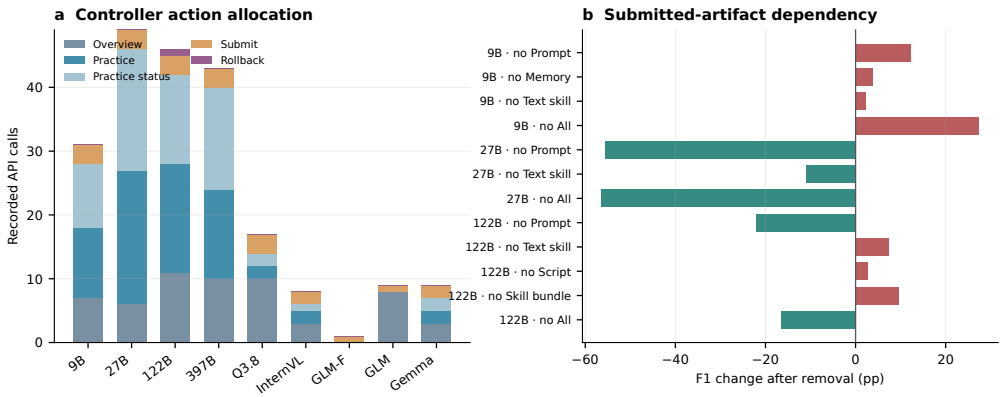


Figure 5: Observed learning behavior and artifact effects. (a) Counts of benchmark-API overview, practice, practice-status, submission, and rollback calls; native file operations are excluded. (b) Strict-F1 change when one submitted artifact or a valid dependency bundle is removed from the same final state. Positive removal effects identify harmful artifacts.

G Runtime Integrity and Scope

All systems use the same frozen splits, action meanings, state-transition rules, strict scorer, hidden test IDs, and no-test-feedback boundary. Model adapters retain the checkpoint's supported visual preprocessing and tool serialization. Most inference services use a 65,536-token serving context; InternVL3.5-38B uses 32,768. Scores characterize each complete open model and compatible runtime under the common scientific protocol. The four hidden states are loaded separately from immutable strategy snapshots, with no learning state carried from one test evaluation to the next.

A verified sample-specific model failure contributes an invalid prediction for that item and does not terminate the rest of the test. Transport and service faults are distinguished from model decisions

and recovered from immutable states or predeclared, non-overlapping test partitions. Final metrics require exactly one scored record for every expected test ID, matching evaluation contracts, and an unchanged scorer. The 122B no-script, no-skill-bundle, and no-artifact interventions use a versioned per-item handling rule for a model-generated image crop rejected by the processor; the affected item is scored invalid, and the original full-state result is retained. This rule is applied without inspecting whether it raises or lowers the reported comparison.

Scope. FLOWRSI evaluates bounded, non-parametric strategy improvement in one visual-structure task family. It does not measure weight-level recursive self-improvement, autonomous redesign of the improvement algorithm, or general-agent competence across unrelated tasks. Item-bootstrap intervals do not replace multi-seed controller replication. The results motivate broader task families and repeated episodes, but the evidence here is the observed flowchart episode and its exact hidden replays.