EVALOPT: EVOLVING EVALUATORS FOR OPEN-ENDED TASKS WITH HUMAN FEEDBACK

Anonymous authors
Paper under double-blind review

ABSTRACT

Evaluating open-ended outputs requires task-dependent judgments across multiple aspects of quality that must be balanced in context. Existing proxy metrics and model-based evaluators do not consistently align with human preferences. Humans can often tell which of two outputs is better, but turning that judgment into an effective evaluation procedure is harder. Evaluator self-evolution offers a way to learn such procedures from human feedback, but the direction of improvement is often ambiguous. The same evaluation error can suggest several plausible corrections, while a useful direction may require multiple revisions before producing a measurable gain. An unsuccessful early attempt therefore provides limited evidence about whether a direction is ineffective or simply underdeveloped. In the default model-driven evolution setup we tested, unsuccessful early candidates were often followed by shifts to alternative directions. We present EvalOpt, a framework for self-evolving evaluators across open-ended tasks with fixed base-model weights. EvalOpt combines disagreement-based preference mining with depth-first evaluator evolution, explicitly maintaining an improvement direction across revisions until diagnostics support moving on or a rejection or resource limit is reached. Unsuccessful candidates guide further implementation without replacing the best evaluator retained on validation data. Across 19 domains, EvalOpt reaches 69.6% held-out decisive-pair accuracy, exceeding the initial Seed evaluator and GEPA, the strongest baseline, by 8.0 and 5.5 percentage points under matched feedback and search budgets. On four analysis domains, it exceeds fixed-patience continuation by 3.9 points, while preference mining adds 1.8 points over random sampling in existing-label replay at the same feedback budget. Independent blinded judgments favor EvalOpt-guided over Seed-guided outputs at tie-aware preference rates of 60.8% for critique-guided refinement and 59.1% for reward-guided training.

1 INTRODUCTION

Open-ended evaluation requires case-specific judgments across quality dimensions (Ye et al., 2024; Kim et al., 2024b). A summary may require completeness or brevity; a slide for live discussion may prioritize clarity over detail. Metrics and model-based evaluators do not consistently match human preferences (Fabbri et al., 2021; Xu et al., 2023; Zheng et al., 2023; Hu et al., 2024), yet humans often cannot express their judgments as an executable evaluation procedure (Shankar et al., 2024b).

Existing approaches use human feedback to train evaluator models (Xu et al., 2023), learn or refine evaluation criteria (Liu et al., 2024; Lv et al., 2026), and select evaluation functions (Shankar et al., 2024b). Prompt and skill optimizers offer a complementary mechanism for revising instructions from execution feedback (Agrawal et al., 2026; Yang et al., 2026a). EvalOpt uses preferences to revise an explicit evaluation procedure—its prompts, tool use, component connections, and scoring rules—without updating the base model. This raises two challenges: obtaining useful human preferences and turning them into improvements to the procedure.

Under a limited feedback budget, comparison selection affects learning efficiency (Muldrew et al., 2024; Oh et al., 2025). Hand-designed comparisons reflect anticipated failures, while broad collections can contain easy or redundant cases. Disagreement-based preference mining ranks comparisons by conflicting signals within the same quality dimension and retains a random anchor 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.

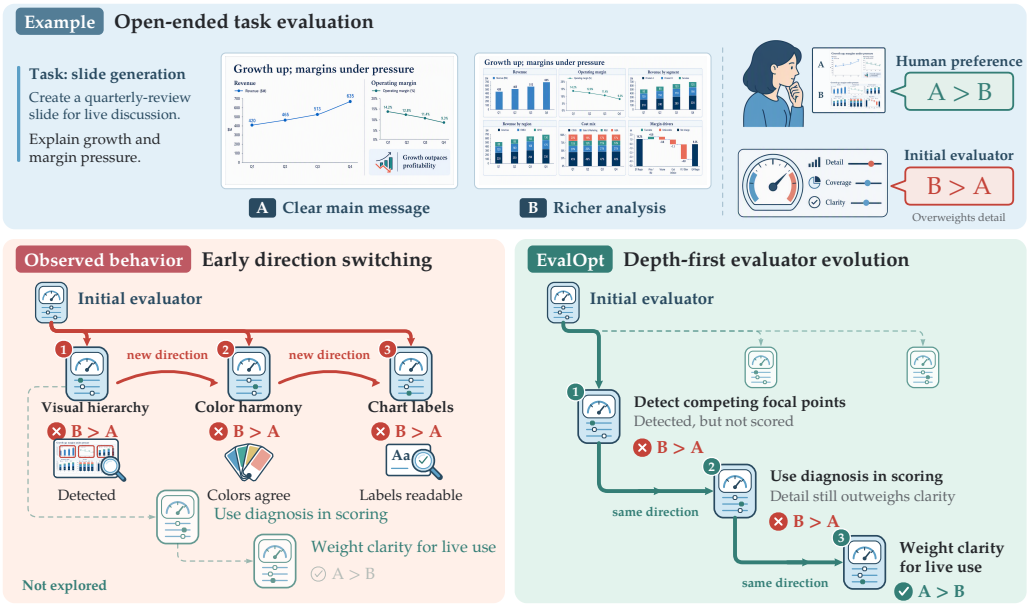

Figure 1: **An unsuccessful revision need not end an improvement direction.** In this slide-evaluation example, the human prefers clarity for live discussion, whereas the initial evaluator favors detail. In the lower-left panel, search switches directions while the disagreement persists. EvalOpt instead uses execution diagnostics to guide the next revision within the same direction.

expose errors shared by those signals. Selected feedback comes from compatible existing labels or new human annotation; disagreement determines what to query, not the correct preference.

Pairwise preferences do not specify the fine-grained criteria behind a choice (Ye et al., 2024; Lv et al., 2026). A useful correction may require several revisions: a proposed tool call may not execute, or its evidence may not affect scoring. An unsuccessful candidate therefore need not invalidate the correction. In the default evolution setup we tested, the model proposed revisions from previous candidates and outcomes, but often switched directions after unsuccessful early candidates.

Depth-first evaluator evolution maintains an improvement direction—a target failure and proposed correction—across revisions (Figure 1, EvalOpt panel; Figure 2, Evaluator evolution panel). Diagnostics guide repair or reconsideration even after candidate rollback. Continuation is separate from candidate acceptance and global retention at validation (Val) checkpoints. Directions end on diagnostic-based decisions, rejection limits, or resource limits.

We compare EvalOpt with Seed, learned signal fusion, SkillOpt, GEPA, and Meta-Harness across 19 equally weighted domains spanning four task families (Table 1). Held-out decisive-pair accuracy measures whether independently scored outputs match human preferences under common feedback and search budgets. Webpage generation, image captioning, summarization, and research-idea generation support search controls, sampling comparisons, and independently judged downstream applications.

EvalOpt reaches 69.6% accuracy across 19 domains, exceeding Seed by 8.0 percentage points and the strongest baseline, GEPA, by 5.5 points. On the four analysis domains, it reaches 72.1%, versus 68.2% for fixed patience and 67.4% for history-based search (Table A8). EvalOpt and fixed patience average 3.8 and 3.6 revisions per direction; similar mean depths are consistent with a benefit from diagnostic-guided continuation but do not establish its cause.

On the same four domains, mined feedback raises EvalOpt accuracy from 70.3% to 72.1% at 100 revealed input groups, with total model-and-tool costs of 106 versus 102 units for random sampling (Table A10). Blinded judgments favor EvalOpt-guided over Seed-guided outputs at tie-aware rates of 60.8% for critique-guided refinement and 59.1% for reward-guided training (Figure 4b). These downstream comparisons assess output quality independently of the optimized evaluator.

Contributions. We contribute preference-guided revision of executable evaluators with fixed base-model weights, combining disagreement-based mining with depth-first evolution that retains diagnostics after rejection. A 19-domain evaluation and four-domain controls assess agreement, search, feedback efficiency, and downstream utility.

2 Related work

Learning evaluators from human feedback. Human feedback supports both training evaluator models and constructing explicit evaluation rules. Improving these judgments matters beyond assessment: when evaluators guide generator optimization, reward hacking can exploit their errors rather than improve human-perceived quality (Wang et al., 2026). Learned reward models and trained judges adapt model weights to produce scores or critiques aligned with human judgments (Xu et al., 2023; Kim et al., 2024a; Ke et al., 2024). Other approaches use feedback to generate, select, or refine explicit evaluation criteria, including preference-grounded query-specific rubrics (Shankar et al., 2024a; Liu et al., 2024; Shankar et al., 2024b; Lv et al., 2026). Agent-as-a-Judge and WebGrader extend evaluation through tool-based inspection and evolving verifier skills, respectively (Zhuge et al., 2025; Chen et al., 2026a). EvalOpt builds on this move toward agentic evaluators: with the base model fixed, it uses human feedback to revise both how evidence is gathered and how that evidence determines judgments.

Program optimization and agent self-evolution. Automated optimization treats prompts, modules, or agent code as editable state, allowing adaptation without changing the underlying model weights. DSPy and TextGrad optimize components of predefined programs (Khattab et al., 2024; Yuksekgonul et al., 2025), while GEPA and SkillOpt use execution feedback and model-generated reflection to revise textual state (Agrawal et al., 2026; Yang et al., 2026a). Agent-level search expands the editable structure and preserves alternatives through archives of designs or programs in ADAS and DGM, distinct harnesses in HarnessBank, and a persistent revision tree in SkillHEX (Hu et al., 2025; Zhang et al., 2026; Luo et al., 2026; Feng et al., 2026). SkillLens distinguishes extracting a skill from successfully using it, motivating attention to executed behavior rather than proposed changes alone (Huang et al., 2026). Within this line of work, EvalOpt focuses on whether to continue an improvement direction after a candidate fails to improve: it maintains the direction across revisions and decides its continuation separately from candidate acceptance.

Active preference acquisition for evaluation. Active preference learning selects which comparisons receive feedback, with disagreement sampling rooted in query-by-committee (Seung et al., 1992; Muldrew et al., 2024; Oh et al., 2025). Benchmark^2 addresses a related selection problem for measurement, constructing test sets for ranking consistency, discriminability, and capability alignment (Qian et al., 2026). EvalOpt instead selects feedback to improve an evaluator, reducing the priority of repetitive comparisons on which existing signals agree without treating agreement as correctness. It can mine existing labels or request new annotations; its endpoint is evaluator improvement under matched label-access or annotation budgets, respectively, rather than disagreement in the selected set.

3 EvalOpt

A failed revision may reflect an incompletely implemented correction rather than evidence against the correction itself. EvalOpt separates direction continuation from evaluator retention.

3.1 Problem setup and overview

Starting from a seed evaluator E_0, EvalOpt revises an agentic evaluator E_θ using human feedback, with the base model M fixed. Its editable configuration θ specifies prompts, tool use, component connections, and scoring rules. For a task x, it evaluates each output y separately, without seeing the competing output or human preference, and records intermediate observations alongside its score. An LLM-based *evolution agent* diagnoses execution records and proposes revisions; each candidate is a complete revised evaluator. Appendix B.1 specifies the shared implementation settings.

Within a domain D sharing an evaluation standard, the objective is agreement with human preferences on unseen inputs,

$$\max_{\theta} J_D(\theta), \qquad J_D(\theta) = \Pr_{(x,y^+,y^-)\sim\mathcal{P}_D,\,\xi}\big[\bar{s}_\theta(x,y^+) > \bar{s}_\theta(x,y^-)\big]. \tag{1}$$

Here y^+ is the human-preferred output and $\mathcal{P}_D$ is the domain distribution. The deployment score $\bar{s}_\theta$ aggregates repeated executions by their median under a protocol frozen before search, with execution randomness ξ (Appendix A.3). Original-input groups are split into Train, Val, and Test before mining. Only Train supplies labels, cases, and traces for revision. Val selects the retained evaluator, and Test assesses it after search is frozen. Both held-out cohorts remain fixed (Appendix A.3).

Disagreement-based preference mining selects which comparisons receive human feedback; depth-first evaluator evolution uses that feedback and accumulated execution evidence to decide what to test next. Within one direction, revisions are executed and diagnosed, then confirmed, staged, or rolled back. Candidate decisions and direction continuation are separate: an unsuccessful revision can still guide the next attempt at the same correction.

3.2 Disagreement-based preference mining

Mining prioritizes disagreements among metrics and judges, while retaining a random anchor to check for errors shared by the available signals. The Train pool contains pairs $r = (x, y_a, y_b)$ from existing datasets or outputs generated for Train inputs. Each signal j yields $u_j(r) \in \{-1, 0, +1, \perp\}$: preference for y_b, a tie, preference for y_a, or unavailable evidence. Score orientation, tie margins, and mappings to evaluation dimensions are fixed before human preferences are revealed. To avoid counting related signals as independent votes, signals sharing a model or underlying metric form a source group. Within each quality dimension q, a group contributes at most one vote; internal opposition yields abstention. We rank pairs by the number of dimensions with opposing source-group preferences,

$$P(r) = \sum_{q} I_{\text{within},q}(r). \tag{2}$$

The indicator equals one when distinct source groups favor opposite outputs on the same dimension q. Ties and missing signals do not count as opposition. Preferences on different dimensions, or on overall quality versus a dimension, are not compared for conflict.

Original-input hashes select the random anchor before ranking, keeping outputs from the same input together. We fill the remaining feedback budget in decreasing order of $P(r)$, with pair identifiers breaking ties. Selected preferences come from compatible existing labels or blinded annotation. Reuse requires the same task and output pair and matching evaluation criteria and annotation protocol, or an explicitly audited harmonization. Both routes preserve label provenance and missing feedback fields; unselected labels remain hidden. Appendix A.2 details signal configuration, source grouping, and anchor selection. Mining is assessed by downstream evaluator improvement, not disagreement yield.

3.3 Depth-first evaluator evolution

Depth-first evolution develops one correction across attempts, using diagnostics to assess whether an unsuccessful revision reflects incomplete implementation or evidence against the correction. The candidate decision is *confirm*, *stage*, or *roll back*; the direction decision is *continue* or *end*. Diagnostics persist across both decisions, including candidate rejection.

Direction and evaluator states. A direction d pairs a target failure pattern with a proposed correction. Revisions that develop or test that correction remain within the direction; adopting a different correction begins a new one. Its history H retains observed outcomes, the evolution agent's interpretations, and unresolved questions across revisions.

Evaluator states separate provisional exploration, within-direction progress, and global retention. The *working evaluator* W determines what to try next and may contain provisional changes. The *reference evaluator* R is the within-direction comparison and rollback point, updated by locally confirmed Train progress. The *best evaluator* B is the Val-selected evaluator retained across directions and returned after search. Each new direction restarts from the globally retained evaluator, setting

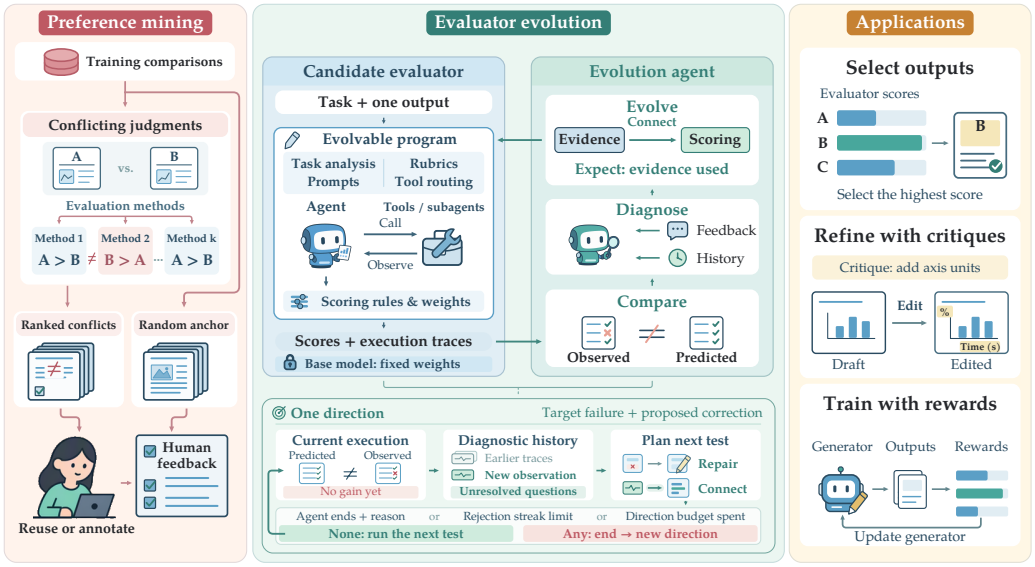

Figure 2: **EvalOpt workflow.** **Preference mining** combines comparisons ranked by within-dimension disagreement with a random anchor for human feedback. The **evolution agent** revises the evaluation program with fixed base-model weights, separating candidate decisions from direction continuation. Execution diagnostics persist after rejection and guide the next edit or check within the same correction. A direction ends on the agent's recorded decision or when rejection or resource limits are reached. Val selects the retained **evolved evaluator** for output selection, critique-guided refinement, and reward-guided training.

$R \leftarrow B$ and $W \leftarrow B$. Changes from one direction carry into the next only through Val retention. A controller enforces the state updates; the evolution agent interprets diagnostics and chooses revisions.

Revising and diagnosing. The evolution agent revises W within a controller-bounded modification scope to produce candidate C, using prior attempts to identify what remains unresolved. The agent identifies the failures to address and the behavior intended to change or remain unchanged. Declared contrast cases identify judgments whose ordering accuracy should not decline. Declared predictions about intermediate outputs are recorded before execution and evaluated as mechanism checks against the resulting traces.

Diagnosis links the intended correction to its execution, the use of resulting evidence in scoring, and the observed ordering changes. An unrealized change motivates an implementation repair; realized but unused evidence motivates a revision to how scoring consumes it. When the intended behavior is realized, case-level gains and regressions inform whether the correction needs refinement or reconsideration. Observed records remain separate from the evolution agent's interpretation and survive rejection.

Measuring candidate progress. Parent and candidate are compared on the same cases under a fixed scoring protocol and matched execution conditions. The local comparison set combines a fixed Train panel $\mathcal{D}_d$ with the revision's target and contrast cases and additional Train cases for broader coverage. Local gain compares C with its parent on this set, whereas accumulated gain compares C with R on $\mathcal{D}_d$. For each case, accuracy is the fraction of repeated executions that rank the human-preferred output strictly higher, and gain averages the differences in these accuracies over the comparison cases. A gain or loss is supported when its magnitude exceeds the empirical repeat-variability buffer; the buffer is a decision heuristic, not a significance test. An otherwise eligible candidate with positive but uncertain gain over R may receive one prespecified fresh measurement on the same cases with the same repeat count to determine its candidate decision. Appendix B specifies the estimator, buffer, fresh-measurement eligibility, and execution-integrity rules.

Candidate and direction decisions. These measurements and declared checks determine the candidate-level decision. With complete measurements, rejection checks take precedence over confirmation and staging.

Decision	Evidence or condition	State update
Confirm	Supported gain over R, nonnegative local gain, and declared checks satisfied	$R \leftarrow C, W \leftarrow C$
Stage	Execution-valid, unconfirmed candidate passing rejection checks within provisional limits	$W \leftarrow C$; R unchanged
Roll back	Rejection condition or provisional limit reached	$W \leftarrow R$

For confirmation, each declared contrast must avoid a decrease in mean accuracy, and every declared mechanism check must pass. A contrast decline blocks confirmation but does not alone trigger rollback. Staging requires execution validity, no rejection condition, and remaining allowance under the shared loss and step limits. Zero or uncertain gain alone does not trigger rejection. An edit that leaves the scoring computation unchanged is rejected unless predeclared as scaffolding and validated by a separate execution preflight. A scaffolding step declares its downstream use and success condition; once connected to scoring, staging requires passed intermediate mechanism checks. Successive staged candidates form a provisional chain. The chain shares a fixed exploration window and loss limit relative to R, neither reset by staging another candidate. For edits that change scoring, a supported loss relative to the parent or R, or a failed declared mechanism check, triggers rollback while retaining diagnostics.

The evolution agent records a continue-or-end decision and its reason based on accumulated diagnostics. This is an agent judgment, not a deterministic test that the correction is ineffective. While the direction remains active, the next revision or diagnostic addresses the same correction. A direction ends on the recorded ending decision, the consecutive-rejection limit, or resource exhaustion. Rollback alone neither ends the direction nor erases H; reaching a limit does not establish that the correction is ineffective.

Global retention. Val/Test evaluate the deployment protocol, which aggregates each output's repeated scores before comparing outputs. Train's repeated ordering frequencies therefore guide search rather than directly estimating the deployment objective J_D. At prespecified checkpoints, an execution-valid W is compared with B on the complete frozen Val cohort. Train confirmation governs local development; Val governs which evaluator to retain across directions. A staged W is therefore eligible, but replaces B only under the prespecified Val improvement criterion. Promotion changes only B, not R, W, or the current direction decision. Val cases and traces remain sealed; the evolution agent receives only the promotion indicator and retained best score. Withholding per-case diagnostics follows the restricted-feedback motivation of the Ladder (Blum & Hardt, 2015). This limits feedback but does not eliminate adaptive-selection risk. After the final checkpoint, search returns B for one locked Test assessment. Algorithm 1 gives the full loop.

4 EXPERIMENTS

We evaluate whether EvalOpt improves evaluator–human agreement under matched budgets (C1), whether diagnostic continuation and disagreement-based mining contribute to that improvement, and whether evolved evaluators improve critique-guided refinement (C2) and reward-guided training (C3).

Tasks and assessment. C1 covers 19 equally weighted domains in four families, with a separate evaluator per domain (Figure 3a). SV1 uses our human-annotated slides; Appendix A specifies the other label sources. Original-input groups are split before Train-only mining; Val and Test remain fixed. The primary endpoint is decisive-pair accuracy: each output is scored independently, and a predicted tie counts as an error (Appendix A.3). Val/Test scores use the median of three executions per output, distinct from independent search runs.

Comparisons and resources. We compare EvalOpt with Seed, learned signal fusion, SkillOpt (Yang et al., 2026a), GEPA (Agrawal et al., 2026), and Meta-Harness (Lee et al., 2026). All op-

timizers start from the same domain-specific Seed evaluator, informed by published rubrics and protocols (Table A1).

We preserve each optimizer's native search policy while matching human-feedback and search budgets. Search costs include diagnostics, rejected candidates, and Val checkpoints; end-to-end costs additionally include mining and amortized controller tuning where applicable. GEPA can edit complete evaluation programs, while text-only EvalOpt matches SkillOpt's editable space. GPT-5.6 Sol executes Seed and evolved evaluators and powers all optimizers. We run five independent searches per adaptive condition. C1 accuracy for every method, including Seed and learned signal fusion, averages five scoring blocks on the same Test pairs. Mining combines rubric judgments from GPT-5.5, GPT-5.6 Sol, GPT-6 Astra, and Claude Sonnet 4.6 with applicable dimensional checks. Appendix A.3 details resources and assessment.

4.1 Evaluator quality under matched budgets

The Seed comparison measures improvement over the starting evaluator; learned signal fusion tests whether reweighting fixed signals suffices without revising the procedure. Optimizer comparisons test search effectiveness with the same feedback and budget (Table 1). All methods score the same outputs independently, without access to human labels or competing outputs.

Table 1: **C1: final held-out accuracy across domains.** Decisive-pair accuracy (%, ↑), averaged equally over domains within each family and over all 19 domains for Overall. SV: structured visuals; IG: images and 3D; TG: text generation; AR: automated research. Bold marks the highest value in each column.

Evaluator	SV (5)	IG (6)	TG (5)	AR (3)	Overall
Seed evaluator	61.7	58.6	67.6	57.0	61.5
Learned signal fusion	63.7	60.2	69.0	57.5	63.0
SkillOpt	64.8	60.7	70.3	58.3	63.9
GEPA	64.7	61.6	69.4	59.3	64.1
Meta-Harness	64.2	60.9	69.9	57.8	63.7
EvalOpt	**70.6**	**67.2**	**74.6**	**64.2**	**69.6**

Figure 3c checks whether aggregate differences extend across domains or concentrate in a few tasks. Appendix C reports per-domain means and denominators, Overall SDs, and method differences.

4.2 Direction continuation and diagnostics

We isolate diagnostic continuation from retaining history or making more revisions. Search and mining analyses use webpage generation (SV3), image captioning (IG1), summarization (TG1), and research-idea generation (AR1), prespecified for diversity and measurement feasibility and weighted equally.

History-based search retains diagnostics but may adopt a different correction after each revision. *Fixed-patience continuation* develops each direction for a Train-tuned number of attempts, subject to common rejection, integrity, and resource limits. EvalOpt instead conditions continuation on accumulated diagnostics for the current correction. All three policies share confirmation, staging, and Val-retention rules. New directions share initialization from the globally retained evaluator, resetting the panel and scope.

Component ablations test EvalOpt's internal mechanisms (Figure 4a). *No retained history* removes prior-attempt diagnostics from the evolution agent, while *reduced feedback* withholds case-level traces but preserves aggregate outcomes and controller-required checks. *No staging* requires confirmation before advancing the working evaluator. *Immediate branching* switches direction after the first execution-valid nonconfirming candidate. *Fixed scope* disables scope adaptation, and *text-only edits* restrict revisions to SkillOpt's editable fields (Appendix C.2).

Controlled cases test repair of unrealized tool calls, integration of unused evidence, and termination of harmful corrections. Independent reviews examine these responses in natural trajectories, including abandoned directions.

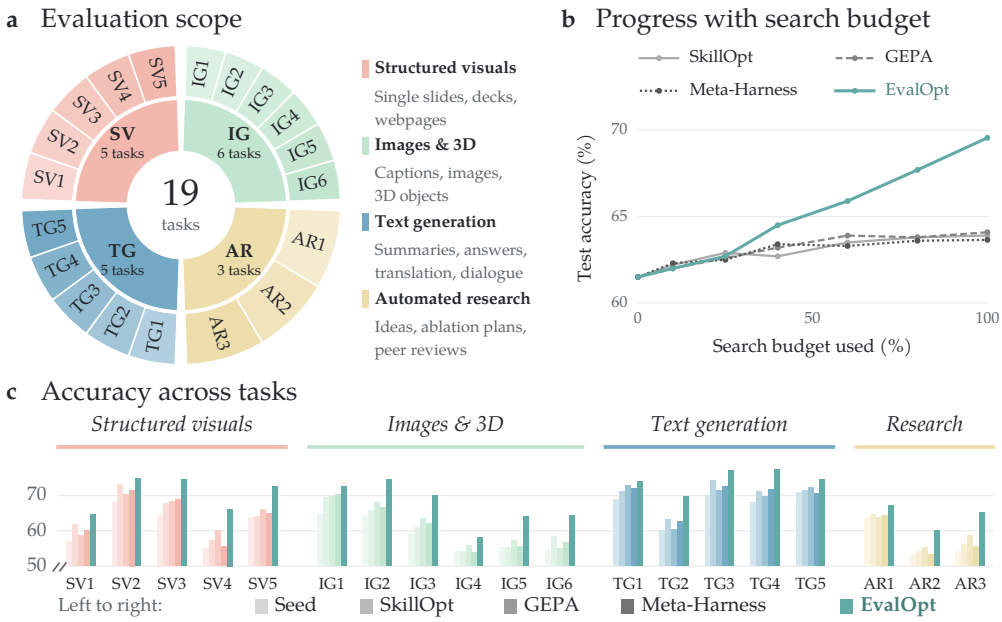

Figure 3: **C1 task scope and evaluator accuracy.** (a) Four task families in equal quadrants, with individual domains in the outer ring. (b) Test accuracy of Val-retained checkpoints versus consumed search budget, evaluated only after search is frozen and averaged equally over 19 domains; endpoints match Table 1. (c) Per-domain Test accuracy for Seed and four optimizers on a common 50–80% scale: EvalOpt in lake green, baselines in family shades.

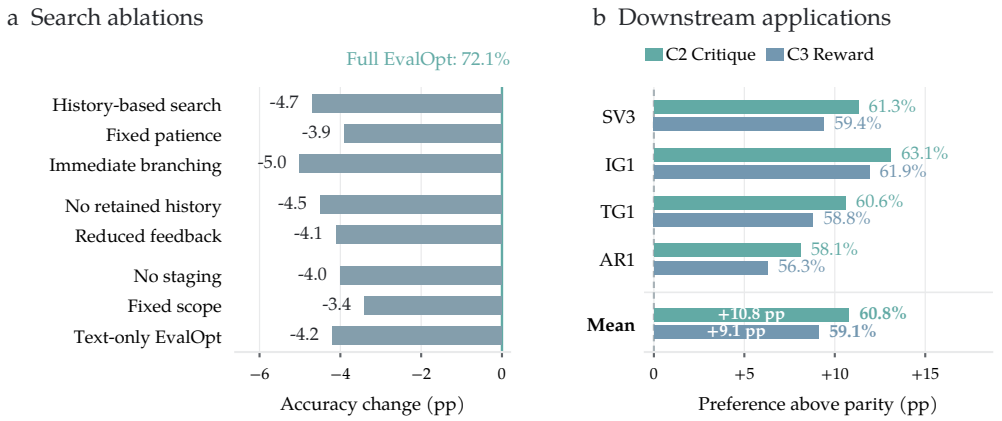

Figure 4: **Search ablations and downstream utility.** (a) Terminal Test accuracy changes relative to full EvalOpt at matched search and feedback budgets. (b) Tie-aware human preference for EvalOpt-guided versus Seed-guided outputs, using critique for C2 refinement and reward for C3 training. Bar lengths give percentage points above 50% parity; endpoint labels give preference rates. Both panels use SV3, IG1, TG1, and AR1, weighted equally in aggregate.

4.3 Feedback efficiency

We assess mining by Test accuracy at matched feedback budgets. Random sampling, uncertainty sampling, cross-model disagreement, and our mining share the pool and feedback format, restarting evolution at each budget (Figure A1a). The anchor ablation compares full mining with spending

all labels on ranked disagreements (Figure A1b). A 2×2 comparison crosses random/mined feedback with fixed-patience/EvalOpt search to describe their separate effects and interaction, without a significance claim (Table A10).

The SV1 collection analysis compares random and mined question selection at the same annotation budget, requesting only sampled dimensions and overall preference. It reports decisive preferences, ties, and unresolved judgments for the same data used in C1 (Appendix C.4).

4.4 Independent downstream utility

C2 and C3 use the same four domains. C2 fixes the GPT-5.6 Sol generator; C3 updates Qwen3.5-9B with a frozen GPT-5.6 Sol reward evaluator.

For each fresh output y_0, C2 compares no revision with three resource-matched revision arms: no-critic (Madaan et al., 2023), Seed-guided, and EvalOpt-guided. The no-critic arm tests revision without evaluator feedback. Blinded judgments compare outputs and assess regressions from y_0.

C3 tests whether offline agreement gains translate into better policies under reward optimization. It compares Base with policies trained using frozen Seed or EvalOpt rewards under the same within-domain recipe (Hu et al., 2022; Shao et al., 2024).

Blinded, rubric-guided human preference is the primary downstream endpoint, complemented by reference-free browser tests, image–caption alignment (VQAScore), and source–summary consistency (AlignScore; Table A14). Figure 4b reports domain results and equal-domain means; Appendix D details configurations, controls, and regressions.

5 Discussion

Retaining evidence beyond a candidate. EvalOpt separates the lifetime of an implementation from that of the evidence it produces. Rollback discards provisional changes but preserves diagnostics. Depth-first evolution uses this evidence to decide whether further work is warranted and what to test next, rather than prescribing more attempts.

Scope and limitations. Diagnostic continuation relies on records that help assess why a revision failed. Aggregate outcomes alone may not justify further development; reduced feedback tests the additional value of case-level traces. The evolution agent's interpretations can be wrong, motivating independent trajectory review. We do not evaluate cross-domain transfer of evolved evaluators. Higher evaluator–human agreement need not improve downstream outputs, especially when reward optimization amplifies residual evaluator errors (Wang et al., 2026). C2 and C3 therefore assess outputs through independent blinded judgments, not the evolved evaluator.

6 Conclusion

EvalOpt learns evaluation procedures from human preferences through disagreement-based preference mining and depth-first evaluator evolution. It separates direction continuation from Train confirmation and global Val retention, allowing rejected candidates to inform further revisions without carrying their changes forward. Held-out agreement, controlled search analyses, feedback efficiency, and independently judged downstream outputs test whether this approach uses limited human feedback effectively.

AI USE STATEMENT

Generative AI assisted manuscript drafting and revision, submission-format checking, literature organization, and the preparation of editable LaTeX conceptual diagrams and plotting code.

REPRODUCIBILITY STATEMENT

Sections 3.1–4 define semantic mappings, feedback, the evaluation program, evolution, and controlled evaluation. Appendix A specifies data sources, annotation, access boundaries, and measurement. Appendix B describes baseline fidelity, resource accounting, and the reference implementation. Appendices C–D provide domain-level reporting, ablations, trajectory analysis, and downstream protocols. An executable implementation and experiment manifests are included in this manuscript bundle.

REFERENCES

Lakshya A Agrawal, Shangyin Tan, Dilara Soylu, Noah Ziems, Rishi Khare, Krista Opsahl-Ong, Arnav Singhvi, Herumb Shandilya, Michael J Ryan, Meng Jiang, Christopher Potts, Koushik Sen, Alex Dimakis, Ion Stoica, Dan Klein, Matei Zaharia, and Omar Khattab. GEPA: Reflective Prompt Evolution Can Outperform Reinforcement Learning. In *International Conference on Learning Representations*, volume 2026, pp. 8479–8565, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/0e9e708b6f48e14fd0ac29e167413f76-Abstract-Conference.html.

Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. SPICE: Semantic Propositional Image Caption Evaluation. In *Computer Vision – ECCV 2016*, pp. 382–398. Springer International Publishing, 2016. doi: 10.1007/978-3-319-46454-1_24. URL https://doi.org/10.1007/978-3-319-46454-1_24.

A. S. Bandeira, K. Scheinberg, and L. N. Vicente. Convergence of trust-region methods based on probabilistic models. *SIAM Journal on Optimization*, 24(3):1238–1264, 2014. doi: 10.1137/130915984. URL https://doi.org/10.1137/130915984.

Jose Blanchet, Coralia Cartis, Matt Menickelly, and Katya Scheinberg. Convergence rate analysis of a stochastic trust-region method via supermartingales. *INFORMS Journal on Optimization*, 1(2): 92–119, 2019. doi: 10.1287/ijoo.2019.0016. URL https://doi.org/10.1287/ijoo.2019.0016.

Avrim Blum and Moritz Hardt. The ladder: A reliable leaderboard for machine learning competitions. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 1006–1014. PMLR, 2015. URL https://proceedings.mlr.press/v37/blum15.html.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging Properties in Self-Supervised Vision Transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9650–9660, 2021.

Boshui Chen, Huiping Liu, and Shaolei Zhang. WebGrader: Training LLMs for web development with self-evolving programmatic grader. *arXiv preprint arXiv:2608.06474*, 2026a. URL https://arxiv.org/abs/2608.06474.

Ruobing Chen, Matt Menickelly, and Katya Scheinberg. Stochastic optimization using a trust-region method and random models. *Mathematical Programming*, 169(2):447–487, 2018. doi: 10.1007/s10107-017-1141-8. URL https://doi.org/10.1007/s10107-017-1141-8.

Xin-Sheng Chen, Jiayu Zhu, Pei lin Li, Hanzheng Wang, Shuojin Yang, and Meng-Hao Guo. PresentBench: A Fine-Grained Rubric-Based Benchmark for Slide Generation. *arXiv preprint arXiv:2603.07244*, 2026b. URL https://arxiv.org/abs/2603.07244.

Zhiyu Chen, Keyu Zhao, Jigao Fu, Dong Liang, Yanbiao Wu, Jiaoyang Li, Haidong Xue, Xinhua Zeng, Yuanyi Zhen, Fengli Xu, and Yong Li. Ideation Arena: Evaluating LLM Generated Research Ideas with Battle-style Human Expert Assessment. arXiv preprint, 2026c. URL https://arxiv.org/abs/2608.29696.

Kanzhi Cheng, Wenpo Song, Jiaxin Fan, Zheng Ma, Qiushi Sun, Fangzhi Xu, Chenyang Yan, Nuo Chen, Jianbing Zhang, and Jiajun Chen. CapArena: Benchmarking and Analyzing Detailed Image Captioning in the LLM Era. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 14077–14094. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.findings-acl.724. URL https://doi.org/10.18653/v1/2025.findings-acl.724.

Jaemin Cho, Yushi Hu, Roopal Garg, Peter Anderson, Ranjay Krishna, Jason Baldridge, Mohit Bansal, Jordi Pont-Tuset, and Su Wang. Davidsonian Scene Graph: Improving Reliability in Fine-grained Evaluation for Text-to-Image Generation. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=ITq4ZRUT4a.

Andrew R. Conn, Nicholas I. M. Gould, and Philippe L. Toint. *Trust Region Methods*. Society for Industrial and Applied Mathematics, 2000. ISBN 9780898719857. doi: 10.1137/1.9780898719857. URL https://doi.org/10.1137/1.9780898719857.

Dylan Ebert. 3D Arena: An Open Platform for Generative 3D Evaluation. arXiv preprint, 2025. URL https://arxiv.org/abs/2506.18787.

Alexander R. Fabbri, Wojciech Kryściński, Bryan McCann, Caiming Xiong, Richard Socher, and Dragomir Radev. SummEval: Re-evaluating Summarization Evaluation. *Transactions of the Association for Computational Linguistics*, 9:391–409, 2021. doi: 10.1162/tacl_a_00373.

Yuchen Fan, Chen Ling, Xin Zhong, Shuo Zhang, Heng Zhou, Yuchen Zhang, Mingyu Liang, Chengxing Xie, Ermo Hua, Zhizhou He, Cheng Huang, Gang Chen, Ning Ding, and Bowen Zhou. LFQA-E: Carefully Benchmarking Long-form QA Evaluation. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=bJYm4v0Spr.

Yuru Feng, Yaoqi Chen, Beidi Zhao, Qianxi Zhang, Xinjiang Wang, Jianan Lu, Zhirui Wang, Shusen Xu, Zengzhong Li, and Qi Chen. SkillHEX: Improving agent skills via hypothesis-driven autonomous exploration and exploitation. *arXiv preprint arXiv:2608.05628*, 2026. URL https://arxiv.org/abs/2608.05628.

Markus Freitag, George Foster, David Grangier, Viresh Ratnakar, Qijun Tan, and Wolfgang Macherey. Experts, Errors, and Context: A Large-Scale Study of Human Evaluation for Machine Translation. *Transactions of the Association for Computational Linguistics*, 9:1460–1474, 2021. doi: 10.1162/tacl_a_00437. URL https://doi.org/10.1162/tacl_a_00437.

Tianyu Gao, Howard Yen, Jiatong Yu, and Danqi Chen. Enabling Large Language Models to Generate Text with Citations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 6465–6488. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.emnlp-main.398. URL https://doi.org/10.18653/v1/2023.emnlp-main.398.

Dhruba Ghosh, Hannaneh Hajishirzi, and Ludwig Schmidt. GenEval: An object-focused framework for evaluating text-to-image alignment. In *Advances in Neural Information Processing Systems*, volume 36, pp. 52132–52152, 2023. doi: 10.52202/075280-2270.

Yuze He, Yushi Bai, Matthieu Lin, Wang Zhao, Yubin Hu, Jenny Sheng, Ran Yi, Juanzi Li, and Yong-Jin Liu. T^3Bench: Benchmarking Current Progress in Text-to-3D Generation. arXiv preprint, 2023. URL https://arxiv.org/abs/2310.02977.

Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. CLIPScore: A Reference-free Evaluation Metric for Image Captioning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 7514–7528. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.emnlp-main.595. URL https://aclanthology.org/2021.emnlp-main.595/.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Shengran Hu, Cong Lu, and Jeff Clune. Automated Design of Agentic Systems. In *International Conference on Learning Representations*, volume 2025, pp. 21344–21377, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/36b7acf6f6010652b3f2a433774a66fe-Abstract-Conference.html.

Xinyu Hu, Mingqi Gao, Sen Hu, Yang Zhang, Yicheng Chen, Teng Xu, and Xiaojun Wan. Are LLM-based evaluators confusing NLG quality criteria? In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 9530–9570. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.acl-long.516. URL https://aclanthology.org/2024.acl-long.516/.

Yushi Hu, Benlin Liu, Jungo Kasai, Yizhong Wang, Mari Ostendorf, Ranjay Krishna, and Noah A. Smith. TIFA: Accurate and Interpretable Text-to-Image Faithfulness Evaluation with Question Answering. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 20349–20360. IEEE, 2023. doi: 10.1109/iccv51070.2023.01866. URL https://doi.org/10.1109/iccv51070.2023.01866.

Zisu Huang, Jingwen Xu, Yifan Yang, Ziyang Gong, Qihao Yang, Muzhao Tian, Xiaohua Wang, Changze Lv, Xuemei Gao, Qi Dai, Bei Liu, Kai Qiu, Xue Yang, Dongdong Chen, Xiaoqing Zheng, and Chong Luo. From Raw Experience to Skill Consumption: A Systematic Study of Model-Generated Agent Skills. *arXiv preprint arXiv:2605.23899*, 2026. URL https://arxiv.org/abs/2605.23899.

Dongfu Jiang, Max Ku, Tianle Li, Yuansheng Ni, Shizhuo Sun, Rongqi Fan, and Wenhu Chen. GenAI Arena: An Open Evaluation Platform for Generative Models. In *Advances in Neural Information Processing Systems 37*, pp. 79889–79908. Neural Information Processing Systems Foundation, Inc. (NeurIPS), 2024. doi: 10.52202/079017-2538. URL https://doi.org/10.52202/079017-2538.

Pei Ke, Bosi Wen, Andrew Feng, Xiao Liu, Xuanyu Lei, Jiale Cheng, Shengyuan Wang, Aohan Zeng, Yuxiao Dong, Hongning Wang, Jie Tang, and Minlie Huang. CritiqueLLM: Towards an informative critique generation model for evaluation of large language model generation. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 13034–13054. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.acl-long.704. URL https://aclanthology.org/2024.acl-long.704/.

Omar Khattab, Arnav Singhvi, Paridhi Maheshwari, Zhiyuan Zhang, Keshav Santhanam, Sri Vardhamanan, Saiful Haq, Ashutosh Sharma, Thomas T. Joshi, Hanna Moazam, Heather Miller, Matei Zaharia, and Christopher Potts. DSPy: Compiling Declarative Language Model Calls into Self-Improving Pipelines. In *International Conference on Learning Representations*, volume 2024, pp. 54928–54958, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/f1cf02ce09757f57c3b93c0db83181e0-Paper-Conference.pdf.

Seungone Kim, Jamin Shin, Yejin Cho, Joel Jang, Shayne Longpre, Hwaran Lee, Sangdoo Yun, Seongjin Shin, Sungdong Kim, James Thorne, and Minjoon Seo. Prometheus: Inducing Fine-grained Evaluation Capability in Language Models. In *International Conference on Learning Representations*, volume 2024, pp. 29927–29962, 2024a. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/803485352e61e3ebf41221e4776c9fd4-Paper-Conference.pdf.

Tae Soo Kim, Yoonjoo Lee, Jamin Shin, Young-Ho Kim, and Juho Kim. EvalLM: Interactive evaluation of large language model prompts on user-defined criteria. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*, pp. 1–21. ACM, 2024b. doi: 10.1145/3613904.3642216. URL https://doi.org/10.1145/3613904.3642216.

Yuval Kirstain, Adam Polyak, Uriel Singer, Shahbuland Matiana, Joe Penna, and Omer Levy. Pick-a-Pic: An Open Dataset of User Preferences for Text-to-Image Generation. In *Advances in Neural Information Processing Systems*, volume 36, pp. 36652–36663, 2023. doi: 10.52202/075280-1594.

Philippe Laban, Tobias Schnabel, Paul N. Bennett, and Marti A. Hearst. SummaC: Re-Visiting NLI-based Models for Inconsistency Detection in Summarization. *Transactions of the Association for Computational Linguistics*, 10:163–177, 2022. doi: 10.1162/tacl_a_00453. URL https://aclanthology.org/2022.tacl-1.10/.

Yoonho Lee, Roshen Nair, Qizheng Zhang, Kangwook Lee, Omar Khattab, and Chelsea Finn. Meta-Harness: End-to-end optimization of model harnesses. *arXiv preprint arXiv:2603.28052*, 2026. URL https://arxiv.org/abs/2603.28052.

Chin-Yew Lin. ROUGE: A Package for Automatic Evaluation of Summaries. In *Text Summarization Branches Out*, pp. 74–81. Association for Computational Linguistics, 2004. URL https://aclanthology.org/W04-1013/.

Zhiqiu Lin, Deepak Pathak, Baiqi Li, Jiayao Li, Xide Xia, Graham Neubig, Pengchuan Zhang, and Deva Ramanan. Evaluating text-to-visual generation with image-to-text generation. In *Computer Vision – ECCV 2024*, pp. 366–384. Springer Nature Switzerland, 2024. doi: 10.1007/978-3-031-72673-6_20. URL https://doi.org/10.1007/978-3-031-72673-6_20.

Yuxuan Liu, Tianchi Yang, Shaohan Huang, Zihan Zhang, Haizhen Huang, Furu Wei, Weiwei Deng, Feng Sun, and Qi Zhang. Calibrating LLM-based evaluator. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pp. 2638–2656. ELRA and ICCL, 2024. doi: 10.63317/3posqwchnb6g. URL https://aclanthology.org/2024.lrec-main.237/.

Renze Lou, Hanzi Xu, Sijia Wang, Jiangshu Du, Ryo Kamoi, Xiaoxin Lu, Jian Xie, Yuxuan Sun, Yusen Zhang, Jihyun Janice Ahn, Hongchao Fang, Zhuoyang Zou, Wenchao Ma, Xi Li, Kai Zhang, Congying Xia, Lifu Huang, and Wenpeng Yin. AAAR-1.0: Assessing AI's Potential to Assist Research. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 40361–40383. PMLR, 2025. URL https://proceedings.mlr.press/v267/lou25c.html.

Zimu Lu, Yunqiao Yang, Houxing Ren, Haotian Hou, Han Xiao, Ke Wang, Weikang Shi, Aojun Zhou, Mingjie Zhan, and Hongsheng Li. WebGen-Bench: Evaluating LLMs on Generating Interactive and Functional Websites from Scratch. In *Advances in Neural Information Processing Systems 38*, pp. 80192–80236. Neural Information Processing Systems Foundation, Inc. (NeurIPS), 2025. doi: 10.52202/085713-2419. URL https://doi.org/10.52202/085713-2419.

Xiaotian Luo, Dizhan Xue, Fengxingyu Wang, Chuanrui Hu, and Yafeng Deng. HarnessBank: Semantic gene-bank search with gated verification for agent-harness self-evolution. *arXiv preprint arXiv:2607.13683*, 2026. URL https://arxiv.org/abs/2607.13683v2.

Changze Lv, Jie Zhou, Wentao Zhao, Jingwen Xu, Shihan Dou, Zisu Huang, Muzhao Tian, Xiaohua Wang, Zhengkang Guo, Yang Liu, Pluto Zhou, Tao Gui, Le Tian, Xiao Zhou, Xiaoqing Zheng, Xuanjing Huang, and Jie Zhou. Learning Query-Specific Rubrics from Human Preferences for DeepResearch Report Generation. *arXiv preprint arXiv:2602.03619*, 2026. URL https://arxiv.org/abs/2602.03619.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. Self-Refine: Iterative Refinement with Self-Feedback. In *Advances in Neural Information Processing Systems*, volume 36, pp. 46534–46594, 2023. doi: 10.52202/075280-2019.

William Muldrew, Peter Hayes, Mingtian Zhang, and David Barber. Active preference learning for large language models. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 36577–36590. PMLR, 2024. URL https://proceedings.mlr.press/v235/muldrew24a.html.

Minhyeon Oh, Seungjoon Lee, and Jungseul Ok. Comparison-based active preference learning for multi-dimensional personalization. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 33145–33166. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.acl-long.1590. URL https://aclanthology.org/2025.acl-long.1590/.

Yasumasa Onoe, Sunayana Rane, Zachary Berger, Yonatan Bitton, Jaemin Cho, Roopal Garg, Alexander Ku, Zarana Parekh, Jordi Pont-Tuset, Garrett Tanzer, Su Wang, and Jason Baldridge. DOCCI: Descriptions of connected and contrasting images. In *Computer Vision – ECCV 2024*, pp. 291–309. Springer Nature Switzerland, 2025. doi: 10.1007/978-3-031-73027-6_17. URL https://doi.org/10.1007/978-3-031-73027-6_17.

Maja Popović. chrF: character n-gram F-score for automatic MT evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pp. 392–395. Association for Computational Linguistics, 2015. doi: 10.18653/v1/W15-3049. URL https://aclanthology.org/W15-3049/.

Matt Post. A Call for Clarity in Reporting BLEU Scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pp. 186–191. Association for Computational Linguistics, 2018. doi: 10.18653/v1/W18-6319. URL https://aclanthology.org/W18-6319/.

Qi Qian, Chengsong Huang, Jingwen Xu, Changze Lv, Muling Wu, Wenhao Liu, Xiaohua Wang, Zhenghua Wang, Zisu Huang, Muzhao Tian, Jianhan Xu, Kun Hu, He-Da Wang, Yao Hu, Xuanjing Huang, and Xiaoqing Zheng. Benchmark^2: Systematic Evaluation of LLM Benchmarks. *arXiv preprint arXiv:2601.03986*, 2026. URL https://arxiv.org/abs/2601.03986.

Yusu Qian, Jiasen Lu, Tsu-Jui Fu, Xinze Wang, Chen Chen, Yinfei Yang, Wenze Hu, and Zhe Gan. GIE-Bench: Towards Grounded Evaluation for Text-Guided Image Editing. arXiv preprint, 2025. URL https://arxiv.org/abs/2505.11493.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning Transferable Visual Models From Natural Language Supervision. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 8748–8763. PMLR, 2021. URL https://proceedings.mlr.press/v139/radford21a.html.

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. COMET: A Neural Framework for MT Evaluation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 2685–2702. Association for Computational Linguistics, 2020. doi: 10.18653/v1/2020.emnlp-main.213. URL https://aclanthology.org/2020.emnlp-main.213/.

Finn Schmidt, Jan Philip Wahle, Terry Ruas, and Bela Gipp. Who Watches the Watchmen? Humans Disagree With Translation Metrics on Unseen Domains. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 22822–22841. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.findings-acl.1145. URL https://doi.org/10.18653/v1/2026.findings-acl.1145.

H. S. Seung, M. Opper, and H. Sompolinsky. Query by committee. In *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, pp. 287–294. ACM, 1992. doi: 10.1145/130385.130417. URL https://doi.org/10.1145/130385.130417.

Shreya Shankar, Haotian Li, Parth Asawa, Madelon Hulsebos, Yiming Lin, J. D. Zamfirescu-Pereira, Harrison Chase, Will Fu-Hinthorn, Aditya G. Parameswaran, and Eugene Wu. spade: Synthesizing data quality assertions for large language model pipelines. *Proceedings of the VLDB Endowment*, 17(12):4173–4186, 2024a. doi: 10.14778/3685800.3685835. URL https://www.vldb.org/pvldb/vol17/p4173-shankar.pdf.

Shreya Shankar, J.D. Zamfirescu-Pereira, Bjoern Hartmann, Aditya Parameswaran, and Ian Arawjo. Who Validates the Validators? Aligning LLM-Assisted Evaluation of LLM Outputs with Human Preferences. In *Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology*, pp. 1–14, 2024b. doi: 10.1145/3654777.3676450.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024. URL https://arxiv.org/abs/2402.03300.

Chenglei Si, Diyi Yang, and Tatsunori Hashimoto. Can LLMs generate novel research ideas? a large-scale human study with 100+ NLP researchers. In *International Conference on Learning Representations*, volume 2025, pp. 94003–94092, 2025a. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/ea94957d81b1c1caf87ef5319fa6b467-Abstract-Conference.html.

Chenglei Si, Yanzhe Zhang, Ryan Li, Zhengyuan Yang, Ruibo Liu, and Diyi Yang. Design2Code: Benchmarking Multimodal Code Generation for Automated Front-End Engineering. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 3956–3974, 2025b. doi: 10.18653/v1/2025.naacl-long.199.

Yushuai Song, Weize Quan, Weining Wang, Jiahui Sun, Jing Liu, Pengbin Yu, Zhentao Chen, Meng Li, Wei Shen, Lunxi Yuan, and Dong-Ming Yan. RefReward-SR: LR-Conditioned Reward Modeling for Preference-Aligned Super-Resolution. In *Computer Vision – ECCV 2026*, pp. 94–112. Springer Nature Switzerland, 2026. doi: 10.1007/978-3-032-37324-3_6. URL https://doi.org/10.1007/978-3-032-37324-3_6.

Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. Learning to summarize with human feedback. In *Advances in Neural Information Processing Systems*, volume 33, pp. 3008–3021. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/1f89885d556929e98d3ef9b86448f951-Paper.pdf.

Songjun Tu, Yiwen Ma, Jiahao Lin, Qichao Zhang, Xiangyuan Lan, Junfeng Li, Nan Xu, Linjing Li, and Dongbin Zhao. PaperAudit-Bench: Benchmarking Error Detection in Research Papers for Critical Automated Peer Review. arXiv preprint, 2026. URL https://arxiv.org/abs/2601.19916.

Ramakrishna Vedantam, C. Lawrence Zitnick, and Devi Parikh. CIDEr: Consensus-based image description evaluation. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4566–4575. IEEE, 2015. doi: 10.1109/cvpr.2015.7299087. URL https://doi.org/10.1109/cvpr.2015.7299087.

Xiaohua Wang, Muzhao Tian, Yuqi Zeng, Zisu Huang, Jiakang Yuan, Bowen Chen, Jingwen Xu, Mingbo Zhou, Wenhao Liu, Muling Wu, Zhengkang Guo, Qi Qian, Yifei Wang, Feiran Zhang, Ruicheng Yin, Shihan Dou, Changze Lv, Tao Chen, Kaitao Song, Xu Tan, Tao Gui, Xiaoqing Zheng, and Xuanjing Huang. Reward Hacking in the Era of Large Models: Mechanisms, Emergent Misalignment, Challenges. *arXiv preprint arXiv:2604.13602*, 2026. URL https://arxiv.org/abs/2604.13602.

Zhilin Wang, Yi Dong, Olivier Delalleau, Jiaqi Zeng, Gerald Shen, Daniel Egert, Jimmy Zhang, Makesh Sreedhar, and Oleksii Kuchaiev. HelpSteer 2: Open-source dataset for training top-performing reward models. In *Advances in Neural Information Processing Systems 37*, pp. 1474–1501. Neural Information Processing Systems Foundation, Inc. (NeurIPS), 2024. doi: 10.52202/079017-0047. URL https://doi.org/10.52202/079017-0047.

Zhilin Wang, Alexander Bukharin, Olivier Delalleau, Daniel Egert, Gerald Shen, Jiaqi Zeng, Oleksii Kuchaiev, and Yi Dong. HelpSteer2-Preference: Complementing Ratings with Preferences. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=MnfHxPP5gs.

Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004. doi: 10.1109/tip.2003.819861. URL https://doi.org/10.1109/tip.2003.819861.

Bosi Wen, Yilin Niu, Cunxiang Wang, Xiaoying Ling, Ying Zhang, Pei Ke, Hongning Wang, and Minlie Huang. IF-RewardBench: Benchmarking Judge Models for Instruction-Following Evaluation. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 23816–23843. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.acl-long.1092. URL https://doi.org/10.18653/v1/2026.acl-long.1092.

Keming Wu, Sicong Jiang, Max Ku, Ping Nie, Minghao Liu, and Wenhu Chen. EditReward: A Human-Aligned Reward Model for Instruction-Guided Image Editing. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=eZu358JOOR.

Xiaoshi Wu, Yiming Hao, Keqiang Sun, Yixiong Chen, Feng Zhu, Rui Zhao, and Hongsheng Li. Human Preference Score v2: A Solid Benchmark for Evaluating Human Preferences of Text-to-Image Synthesis. arXiv preprint, 2023. URL https://arxiv.org/abs/2306.09341.

Jiazheng Xu, Xiao Liu, Yuchen Wu, Yuxuan Tong, Qinkai Li, Ming Ding, Jie Tang, and Yuxiao Dong. ImageReward: Learning and Evaluating Human Preferences for Text-to-Image Generation. In *Advances in Neural Information Processing Systems*, volume 36, pp. 15903–15935, 2023. doi: 10.52202/075280-0700.

Yifan Yang, Ziyang Gong, Weiquan Huang, Qihao Yang, Ziwei Zhou, Zisu Huang, Yan Li, Xuemei Gao, Qi Dai, Bei Liu, Kai Qiu, Yuqing Yang, Dongdong Chen, Xue Yang, and Chong Luo. SkillOpt: Executive strategy for self-evolving agent skills. *arXiv preprint arXiv:2605.23904*, 2026a. URL https://arxiv.org/abs/2605.23904v2.

Yunqiao Yang, Wenbo Li, Houxing Ren, Zimu Lu, Ke Wang, Zhiyuan Huang, Zhuofan Zong, Mingjie Zhan, and Hongsheng Li. SlidesGen-Bench: Evaluating Slides Generation via Computational and Quantitative Metrics. arXiv preprint, 2026b. URL https://arxiv.org/abs/2601.09487.

Seonghyeon Ye, Doyoung Kim, Sungdong Kim, Hyeonbin Hwang, Seungone Kim, Yongrae Jo, James Thorne, Juho Kim, and Minjoon Seo. FLASK: Fine-grained language model evaluation based on alignment skill sets. In *International Conference on Learning Representations*, volume 2024, pp. 55361–55414, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/f41b4a6b202adcd8e150a9d4f124d8f6-Abstract-Conference.html.

Mert Yuksekgonul, Federico Bianchi, Joseph Boen, Sheng Liu, Pan Lu, Zhi Huang, Carlos Guestrin, and James Zou. Optimizing generative AI by backpropagating language model feedback. *Nature*, 639(8055):609–616, 2025. doi: 10.1038/s41586-025-08661-4.

Yuheng Zha, Yichi Yang, Ruichen Li, and Zhiting Hu. AlignScore: Evaluating factual consistency with a unified alignment function. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 11328–11348. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.acl-long.634. URL https://aclanthology.org/2023.acl-long.634/.

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*, volume 2026, pp. 104223–104294, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/aa5f5e6eb6f613ec412f1d948dfa21a5-Abstract-Conference.html.

Kai Zhang, Lingbo Mo, Wenhu Chen, Huan Sun, and Yu Su. MagicBrush: A Manually Annotated Dataset for Instruction-Guided Image Editing. In *Advances in Neural Information Processing Systems 36*, pp. 31428–31449. Neural Information Processing Systems Foundation, Inc. (NeurIPS), 2023. doi: 10.52202/075280-1365. URL https://doi.org/10.52202/075280-1365.

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The Unreasonable Effectiveness of Deep Features as a Perceptual Metric. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 586–595. IEEE, 2018. doi: 10.1109/cvpr.2018.00068. URL https://doi.org/10.1109/cvpr.2018.00068.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. BERTScore: Evaluating Text Generation with BERT. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=SkeHuCVFDr.

Yuhan Zhang, Mengchen Zhang, Tong Wu, Tengfei Wang, Gordon Wetzstein, Dahua Lin, and Ziwei Liu. 3DGen-Bench: Comprehensive Benchmark Suite for 3D Generative Models. arXiv preprint, 2025a. URL https://arxiv.org/abs/2503.21745.

Yujie Zhang, Bingyang Cui, Qi Yang, Zhu Li, and Yiling Xu. Benchmarking and Learning Multi-Dimensional Quality Evaluator for Text-To-3D Generation. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 18563–18574. IEEE, 2025b. doi: 10.1109/iccv51701.2025.01725. URL https://doi.org/10.1109/iccv51701.2025.01725.

Zhuohao (Jerry) Zhang, Ruiqi Chen, Mingyuan Zhong, and Jacob O. Wobbrock. SlideAudit: A Dataset and Taxonomy for Automated Evaluation of Presentation Slides. In *Proceedings of the 38th Annual ACM Symposium on User Interface Software and Technology*, pp. 1–23. ACM, 2025c. doi: 10.1145/3746059.3747736. URL https://doi.org/10.1145/3746059.3747736.

Yilun Zhao, Weiyuan Chen, Zhijian Xu, Manasi Patwardhan, Chengye Wang, Yixin Liu, Lovekesh Vig, and Arman Cohan. AbGen: Evaluating Large Language Models in Ablation Study Design and Evaluation for Scientific Research. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 12479–12491. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.acl-long.611. URL https://doi.org/10.18653/v1/2025.acl-long.611.

Hao Zheng, Xinyan Guan, Hao Kong, Wenkai Zhang, Jia Zheng, Weixiang Zhou, Hongyu Lin, Yaojie Lu, Xianpei Han, and Le Sun. PPTAgent: Generating and Evaluating Presentations Beyond Text-to-Slides. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 14402–14418, 2025. doi: 10.18653/v1/2025.emnlp-main.728.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, Hao Zhang, Joseph Gonzalez, and Ion Stoica. Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena. In *Advances in Neural Information Processing Systems*, volume 36, pp. 46595–46623, 2023. doi: 10.52202/075280-2020.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. Instruction-Following Evaluation for Large Language Models. arXiv preprint, 2023. URL https://arxiv.org/abs/2311.07911.

Mingchen Zhuge, Changsheng Zhao, Dylan R. Ashley, Wenyi Wang, Dmitrii Khizbullin, Yunyang Xiong, Zechun Liu, Ernie Chang, Raghuraman Krishnamoorthi, Yuandong Tian, Yangyang Shi, Vikas Chandra, and Jürgen Schmidhuber. Agent-as-a-judge: Evaluate agents with agents. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 80569–80611. PMLR, 2025. URL https://proceedings.mlr.press/v267/zhuge25a.html.

A Data and evaluation settings

A.1 Task scope, resources, and evaluation dimensions

A domain has a fixed quality standard and a separately adapted evaluator. Table A1 lists the 19 equally weighted C1 domains, their resources, Seed criterion sources, and candidate signals. Rubric judges can supply dimensional and overall judgments. The executed signal subset is frozen before mining; reference-based metrics require permitted references, and reserved assessment signals remain outside search.

Seed construction and attribution. Seed is our domain-specific implementation, not a released evaluator copied from a benchmark. We adapt published rubric criteria where available and otherwise write criteria from the task requirements and cited evaluation protocols. The table identifies these sources alongside our additions; borrowing a criterion does not imply reusing its original judge prompt, model, or score scale. We assemble the prompts, tool routing, and score aggregation, freezing one initial checkpoint per domain before comparative search, without Val/Test preference feedback. All optimizers receive that checkpoint, subject to their declared editable spaces (Table A5).

Table A1: Task scope, data resources, Seed criterion sources, and candidate signals across the 19 domains. Numbers link methods to dimensions within each row. Sources inform criteria or checks; listed signals are not all required Seed components.

Domain, task and data sources	Seed criteria and sources	Corresponding methods and signals
Structured visual artifacts		
TD-SV1 Brief-conditioned single-slide generation Our human-annotated slide comparisons	(1) Brief adherence; (2) content correctness; (3) readability; (4) visual hierarchy; (5) layout consistency *Our brief/content rubric; visual checks informed by SlideAudit (Zhang et al., 2025c).*	(1–2) Brief-to-text coverage and factuality checks; (3) font-size and contrast diagnostics; (4–5) PPTX/render geometry, overlap and alignment checks, supplemented by visual rubric judgments.
TD-SV2 Screenshot-conditioned webpage reconstruction Design2Code; human-pair release	(1) Text fidelity; (2) position and structure; (3) color and style; (4) visual detail *Adapted from Design2Code's reconstruction metrics; visual detail added.*	(1–3) Design2Code text, position, color and block-match scores (Si et al., 2025b); (4) screenshot SSIM (Wang et al., 2004) and CLIP/DINO similarity (Radford et al., 2021; Caron et al., 2021) as visual proxies.
TD-SV3 Specification-conditioned webpage generation WebDev Arena; WebGen-Bench (Lu et al., 2025)	(1) Requirement fulfillment; (2) functionality; (3) usability; (4) visual quality *Adapted from WebGen-Bench's functionality and visual evaluation; usability added.*	(1–2) Browser interaction and requirement checklists; (3) axe-core accessibility checks; (4) WebGen-Bench visual evaluation.
TD-SV4 Source-grounded slide-deck generation PresentBench; PPTAgent (Zheng et al., 2025)	(1) Source fidelity and coverage; (2) organization; (3) visual consistency and readability; (4) editability *PresentBench content checklists and PPTAgent/SlidesGen-Bench quality criteria.*	(1) PresentBench checklists (Chen et al., 2026b); (2) deck-level organization and repetition checks; (3–4) SlidesGen-Bench (Yang et al., 2026b) aesthetics/editability protocols and PPTX diagnostics.

Table A1 continued

Domain, task and data sources	Seed criteria and sources	Corresponding methods and signals
TD-SV5 Topic-conditioned slide-deck generation Slides-Align (Yang et al., 2026b); SlidesGen-Bench	(1) Topic relevance and correctness; (2) narrative organization; (3) aesthetics and readability; (4) cross-page consistency *SlidesGen-Bench visual criteria; topic correctness and organization added.*	(1–2) Deck-level rubric judgments; (3) SlidesGen-Bench aesthetics protocol; (4) cross-page typography, palette and layout checks.
Image generation and understanding		
TD-IG1 Detailed image-caption generation CapArena (Cheng et al., 2025); DOCCI (Onoe et al., 2025)	(1) Factuality and relations; (2) content coverage; (3) detail; (4) conciseness and organization *Adapted from CapArena's precision, informativeness, hallucination, and conciseness guidance.*	(1,3) Atomic visual-claim checks; (2) reference-based CIDEr/SPICE (Vedantam et al., 2015; Anderson et al., 2016); (4) rubric judgments. CLIPScore (Hessel et al., 2021) supplies a complementary image–text alignment signal.
TD-IG2 Text-to-image generation HPDv2 (Wu et al., 2023); ImageRewardDB	(1) Prompt alignment; (2) compositional relations; (3) visual quality; (4) overall preference *Our rubric combines ImageReward/HPDv2 preference criteria with compositional checks.*	(1) CLIP similarity; (1–2) TIFA (Hu et al., 2023)/DSG (Cho et al., 2024) visual questions, GenEval checks (Ghosh et al., 2023); (3) visual rubric judgments; (4) PickScore (Kirstain et al., 2023), ImageReward (Xu et al., 2023), HPSv2.
TD-IG3 Single-source instruction-based image editing GenAI Arena (Jiang et al., 2024); MagicBrush (Zhang et al., 2023)	(1) Instruction fulfillment; (2) content preservation; (3) edit quality and naturalness *GIE-Bench instruction/preservation checks; our visual-quality rubric.*	(1–2) GIE-Bench (Qian et al., 2025) visual questions and preservation checks; (2) masked DINO/LPIPS; (3) visual rubric judgments. EditReward (Wu et al., 2026) supplies a separate preference signal.
TD-IG4 LR-conditioned image super-resolution RefSR-18K (Song et al., 2026); RefReward-SR	(1) Fidelity; (2) detail and texture; (3) artifacts and naturalness *Our fidelity/quality rubric, informed by RefReward-SR's evaluation setting.*	(1) Reference-based PSNR/SSIM; (2–3) LPIPS (Zhang et al., 2018) and visual rubric judgments. RefReward-SR supplies a separate preference signal.
TD-IG5 Text-to-3D object generation MATE-3D (Zhang et al., 2025b); 3DGen-Bench (Zhang et al., 2025a)	(1) Text alignment; (2) geometry and completeness; (3) texture; (4) multiview consistency *T3Bench alignment and multiview quality protocol; our geometry/texture checks.*	(1) T3Bench (He et al., 2023) text–3D alignment protocol; (2) mesh diagnostics; (2–4) T3Bench multiview quality evaluation and fixed-view visual rubric judgments.
TD-IG6 Image-to-3D object generation 3D Arena (Ebert, 2025); 3DGen-Bench	(1) Source-image fidelity; (2) geometry; (3) texture; (4) multiview consistency *Our source-conditioned adaptation of 3D quality checks; 3D Arena supplies comparisons.*	(1) Camera-aligned source-view CLIP/DINO similarity; (2) mesh diagnostics; (3–4) visual rubric judgments across standardized renders.

Table A1 continued

Domain, task and data sources	Seed criteria and sources	Corresponding methods and signals
Text generation		
TD-TG1 Source-grounded document summarization summarize-from-feedback (Stiennon et al., 2020)	(1) Source faithfulness; (2) coverage; (3) relevance; (4) coherence and conciseness *Adapted from summarization feedback and SummEval criteria (Fabbri et al., 2021).*	(1) SummaCConv (Laban et al., 2022); (2–3) reference-based ROUGE (Lin, 2004)/BERTScore (Zhang et al., 2020) similarity and source-coverage checks; (4) rubric judgments.
TD-TG2 RAG-supported long-form answer generation LFQA-E (Fan et al., 2026); ALCE (Gao et al., 2023)	(1) Correctness and coverage; (2) evidence support; (3) citation precision and completeness; (4) clarity *ALCE evidence/citation checks; our answer-quality rubric informed by LFQA-E.*	(1,4) Question-conditioned rubric judgments; (2) claim–evidence entailment; (3) ALCE citation precision/recall.
TD-TG3 Machine translation WMT MQM (Freitag et al., 2021); CD-ESA (Schmidt et al., 2026)	(1) Accuracy; (2) fluency; (3) terminology; (4) omissions and additions; (5) style *Adapted from MQM's error taxonomy; omissions/additions made explicit.*	(1–5) MQM-style error judgments; COMET (Rei et al., 2020) supplies aggregate translation quality, with SacreBLEU/chrF (Post, 2018; Popović, 2015) as reference-overlap signals.
TD-TG4 Instruction-constraint-following response generation IF-RewardBench (Wen et al., 2026)	(1) Explicit constraint satisfaction; (2) correctness; (3) usefulness and relevance; (4) expression *IFEval/IF-RewardBench constraint checks; our general response-quality criteria.*	(1) IFEval (Zhou et al., 2023) strict/loose checks and task-specific constraint checkers; (2–4) rubric judgments, informed by IF-RewardBench checklists.
TD-TG5 Open-ended dialogue-response generation HelpSteer2 (Wang et al., 2024; 2025)	(1) Helpfulness; (2) correctness; (3) contextual consistency; (4) relevance and expression; (5) safety *HelpSteer2 helpfulness, correctness, and coherence; relevance/expression adapted, safety added.*	(1–5) Dimensional dialogue rubric judgments; (3) history–response contradiction checks. A compatible public reward model supplies a separate overall signal.
Automated research		
TD-AR1 Research-idea generation Research Ideation Arena (Chen et al., 2026c)	(1) Novelty; (2) feasibility; (3) significance; (4) specificity *Adapted from Ideation Arena's four expert-review criteria.*	(1–4) Ideation Arena human-review rubric; (1,3) comparison with the supplied literature and claim–source support checks.

Table A1 continued

Domain, task and data sources	Seed criteria and sources	Corresponding methods and signals
TD-AR2 ML/NLP ablation-plan generation AbGen (Zhao et al., 2025)	(1) Importance; (2) faithfulness; (3) soundness, including controls and executability *Adapted from AbGen-Eval's three criteria; controls/executability operationalize soundness.*	(1–3) AbGen-Eval rubric judgments; (2) proposal–paper alignment; (3) intervention/control/outcome checks and confound diagnostics.
TD-AR3 Peer-review critique generation AAAR (Lou et al., 2025); PaperAudit (Tu et al., 2026)	(1) Correctness and evidence; (2) significance and coverage; (3) actionability; (4) fairness and clarity *Our critique-quality rubric; AAAR/PaperAudit inform evidence and error checks.*	(1) Paper-evidence localization and PaperAudit error checks on annotated subsets; (2–4) critique-quality rubric judgments.

SV1 uses our own human-annotated slide comparisons; SlideAudit informs visual criteria rather than supplying these preference labels. For public resources, human dimensional labels can support criterion-specific evaluation; they are not relabeled as overall preferences. Cohort construction must match each judgment to its artifacts, criterion, and permitted evidence. SV5 uses aesthetic ranks, IG5 overall MOS, TG3 segment scores, and AR2 a three-dimension composite. These score-derived orderings differ from direct choices but contribute to the reported aggregate; separate sensitivity results for these label types are not reported here. Admission requires a frozen source version, permission for the intended processing, executable inputs, and an independent test cohort.

References, dimensional scores, and metric-training resources do not automatically supply independent overall preferences. DOCCI supplies caption references (Onoe et al., 2025) and SummEval supplies CNN/DailyMail dimensional ratings, not TL;DR pair labels (Fabbri et al., 2021). Resources used to train a metric require an overlap audit before independent evaluation.

A.2 Feedback and preference mining

Human feedback has four levels: H_0 gives overall preferences; H_1 gives dimensional preferences; H_2 identifies error type, severity, and location; H_3 optionally supplies a free-form rationale. Missing feedback remains absent. Existing-label replay counts revealed source groups and pairs; new annotation additionally records requested ratings, person-time, abstentions, and adjudication. Freeze rubrics, permitted evidence, tie rules, rater qualifications, compensation, and review requirements. Development uses two independent judgments with adjudication, final assessment uses three, and both hide method identities, earlier winners, and model scores while randomizing presentation.

Mining uses deduplicated Train groups under the signal normalization and voting rules in Section 3.2. Signal definitions, source groups, dimension mappings, tie margins, and repeats are frozen before label access. The hash-selected random anchor is independent of scores and counts toward the total label budget. The four judges in Table A4 form four source groups; repeats, prompt variants, and the Seed's use of the same model do not add groups or establish independent errors. Within-dimension opposition determines rank, with pair IDs breaking ties.

A model signal uses the median of same-repeat score differences; this differs from comparing each artifact's median final score in Equation 3. The native runner uses three repeats and a two-point tie margin for model scores; metric bundles specify their own margins. A missing required repeat makes a signal unavailable, not an agreement. Metrics reserved for final assessment are excluded from mining and evaluator tools. The signal roster and anchor fraction are specified in Table A4.

A.3 Data allocation and evaluation protocol

Splits and allocations. Before mining, split original-input groups and declared dependence groups into Train, Val, and Test without using labels or mining scores. Keep document derivatives, website templates, related images, and research topics together where applicable. All methods share the frozen source pool, eligibility rules, and held-out pairs, with no later Val/Test additions or replacements. Train supplies feedback and traces for revision; Val permits only checkpoint retention, releasing the promotion indicator and retained score to the optimizer. Test is assessed offline after search and selection rules are frozen. The deployed scorer sees one output and permitted evidence, not opponents, human labels, generator identity, or split metadata.

Table A2 counts deduplicated original-input groups, each contributing one C1 pair selected by fixed hash before label inspection. Tune is included in Train but excluded from preference mining; Feedback counts revealed groups. Val/Test allocations precede annotation, with decisive pairs, ties, abstentions, and missing assets reported separately. Source versions, hashes, grouping keys, filters, and pair IDs are recorded in the release manifest. Insufficient eligible groups require collection or a protocol revision, not extra dependent pairs counted as new inputs. For the four analysis domains, the public files contain 6,041 single-turn WebDev requests, 3,400 CapArena images with strict preferences, 20,986 TL;DR posts with comparisons, and 1,782 ideation contexts with strict overall preferences. These are source capacities before grouping and eligibility checks, not final cohorts. Appendix D specifies separate C2/C3 inputs and cross-source exclusions.

Table A2: **C1 data allocation by domain.** Train includes Tune; Feedback is drawn from the remaining Train pool. Every count is in original-input groups. Resource roles and evaluation dimensions are in Table A1.

ID	Cohort source	Train	Tune	Feedback	Val	Test
SV1	Our human-annotated slide comparisons	200	20	100	40	120
SV2	Design2Code human-pair release	240	32	100	80	160
SV3	WebDev Arena	256	32	100	64	200
SV4	PresentBench; new deck pairs	120	16	80	40	78
SV5	Slides-Align	90	10	50	30	67
IG1	CapArena human comparisons	256	32	100	64	200
IG2	HPDv2	256	32	100	64	200
IG3	MagicBrush sources; new edited pairs	128	16	100	32	64
IG4	RefSR-18K	256	32	100	64	200
IG5	MATE-3D	80	10	50	30	50
IG6	3D Arena; new overall judgments	128	16	100	32	64
TG1	Summarize-from-feedback, TL;DR only	256	32	100	64	200
TG2	LFQA-E with fixed evidence bundles	256	32	100	64	200
TG3	WMT MQM; document-level grouping	256	32	100	64	200
TG4	IF-RewardBench	256	32	100	64	200
TG5	HelpSteer2 preference release	256	32	100	64	200
AR1	Research Ideation Arena	256	32	100	64	200
AR2	AbGen-Eval	50	10	40	20	30
AR3	AAAR; new overall judgments	128	16	100	32	64

Experimental allocation and replication. Table A3 specifies data, replication, and allowances. Four-domain analyses reuse C1 Val/Test groups but run fresh searches for changed feedback or policies; the mining interaction reuses matching 100-group runs without counting them as new replications. Evaluator search, refinement, and training have separate budgets.

The five optimizer runs use search seeds 11/22/33/44/55, distinct from the shared initial Seed evaluator. Each run starts anew from that evaluator with the same data split, revealed C1 feedback, and search cap; seeds control data ordering and search randomization, not model weights. API sampling is logged separately and need not be deterministic under a seed. Each run retains its own Val-selected checkpoint, and all runs enter assessment without choosing the best run by Test accuracy. Mining comparisons additionally vary acquisition across the five blocks, sharing the pool, split, and budgets rather than the selected feedback.

Table A3: **Allocation across experiments.** The four-domain subset is SV3, IG1, TG1, and AR1. A search cap is 100 units as defined in Table A4.

Experiment	Data per domain and condition	Replication and allowance
C1 optimizer comparison	Table A2; identical revealed feedback for every optimizer	5 seeds; 100 units/search
Search ablations	Four domains; 100 revealed groups; common 64 Val and 200 Test groups	Full plus 8 controls; 5 seeds; 100 units/search
Controller tuning	32 Tune groups per analysis domain, disjoint from preference mining	$k \in \{2, 4, 6\}$; 2 seeds; 10 units per setting/seed
Mining replay	Four domains; 20/50/100/200 revealed groups from the 224-group sampling pool	Random, uncertainty, cross-model, full; 5 seeds and 100 search units at each budget
Anchor and interaction	Four domains; 100 revealed groups; 20% versus no anchor; random/mined × fixed-patience/EvalOpt	5 seeds; common search cap; reuse matching cells
Controlled diagnostics	Four domains; 20 Train-derived instances per failure type; 3 types	5 seeds; at most 6 attempts and 10 units per instance
Natural trajectories	30 directions/domain from the full-method searches, stratified by confirmation and ending outcome	2 independent reviewers; retain failed and exhausted directions
SV1 collection	Same C1 pool and splits; 100 acquired pairs/arm	900 individual ratings/arm; shared C1 assessment
C2 refinement	Separate 64 development and 200 final-assessment inputs/domain	3 paired seeds; 2 refinement rounds
C3 reward training	Separate 512 training, 64 development, and 200 final-assessment inputs/domain	3 paired seeds; 128 updates/reward arm

Fixed patience is tuned on two complementary held-out halves of Tune, never C1 Val/Test. Its six short runs cost 60 additional units per analysis domain, amortized across five searches as 12 units per search and included in end-to-end accounting (Table A10). Other arms use common defaults without outcome-guided tuning.

Accuracy and missing outcomes. Train uses mean correct-ordering frequencies across repeated executions for each case, as in Equation 5. Val/Test instead uses a frozen scoring protocol with $K \geq 3$ executions per artifact and median aggregation. For strict human pairs P_g within source group g,

$$\begin{aligned} \bar{s}_\theta(x, y) &= \mathrm{median}_{r=1,\ldots,K}\, s_\theta^{(r)}(x, y), \\ v_{gj} &= \mathbb{I}[\bar{s}_\theta(x_g, y_{gj}^+) > \bar{s}_\theta(x_g, y_{gj}^-)], \\ \widehat{J}(\theta) &= \frac{1}{|\mathcal{G}|} \sum_{g \in \mathcal{G}} \frac{1}{|P_g|} \sum_{j \in P_g} v_{gj}. \end{aligned} \tag{3}$$

Predicted ties count as errors; human ties and groups without strict preferences are counted separately. Score-tie rates use the same strict pairs and weights as accuracy. With fixed weights $w_{gj} = 1/(|\mathcal{G}||P_g|)$ and completion indicators c_{gj}, define $M_{\mathrm{cov}} = \sum w_{gj} c_{gj}$ and $K_{\mathrm{corr}} = \sum w_{gj} c_{gj} v_{gj}$. Report conditional accuracy $K_{\mathrm{corr}}/M_{\mathrm{cov}}$, completion counts, and the full-cohort missing-outcome bounds

$$[K_{\mathrm{corr}},\ K_{\mathrm{corr}} + 1 - M_{\mathrm{cov}}]. \tag{4}$$

These are bounds, not confidence intervals. Comparisons use a common complete cohort, with requested-cohort bounds alongside them; missing outcomes are never replaced by zeros or selectively favorable retries. Best-of-N and regret require a separate fully ranked cohort. Native correlations, score-derived preferences, and direct human choices retain their distinct definitions.

Aggregation and descriptive statistics. First average within-domain accuracy over five scoring blocks, then average domains equally. For the optimizers, each block assesses one independent search's Val-retained evaluator; Seed and learned signal fusion are also scored in five blocks, without treating those blocks as five adaptive searches. Overall therefore weights the four family summaries by $5/19$, $6/19$, $5/19$, and $3/19$. All methods share the endpoint's domain roster, and in-

complete runs follow the prespecified failure policy rather than being dropped. Search replication follows Table A3; scoring repeats do not increase that count. With one decisive pair per group and the same N_d complete pairs in each of the five scoring blocks, domain accuracy in percent is $100 \sum_{s=1}^{5} c_{ds}/(5N_d)$, where c_{ds} is the integer number of correctly ordered pairs in block s. The three scoring executions yield one median score per output, not three correctness observations. Compute block averages, domain averages, and between-method differences at full precision and round only the final displayed value, using nearest-tenth rounding with ties to even for percentages and percentage-point differences. An auditable result export must identify the common pair cohort and give the requested, decisive, and complete counts and integer correct counts for each method and search; allocation counts alone do not specify the effective denominator. For descriptive Overall SD, first average the 19 domain accuracies equally within each block, then compute the mean and sample SD of the five block averages (variance denominator 4). Within-domain SD likewise uses the five block accuracies, not individual scoring executions. Seed and fusion repeat assessment of their respective fixed evaluators, not five optimization searches. Matching block identifiers alone does not establish paired searches across methods. These SDs describe the available block arrays, not input-sampling uncertainty or demonstrated search stability. Table A7 reports Overall summaries and method differences.

Checkpoint assessment. Test is opened after search and all selection rules are frozen. Figure 3b uses checkpoints at 0, 10, 25, 40, 60, 80, and 100% of the common cost cap. The four-domain mechanism protocol additionally permits 25/50/75/100% checkpoint summaries. Each boundary uses the last retained B, carried forward after an early stop, never a Test-selected evaluator. Additional checkpoint assessment costs are separate from search.

B Method and implementation details

B.1 Shared configuration and baselines

Shared execution settings. Table A4 fixes models and search defaults; downstream policy settings are separate. The reference implementation combines atomic judgments and four criterion ratings by a weighted mean; a separate holistic diagnostic score does not enter that mean. Table A1 specifies domain quality dimensions, rather than a universal number of aggregation fields. Prompts, tools, component connections, and scoring rules remain editable, with evaluator weights fixed. Train-derived examples cannot contain case-ID lookups or whole-case score tables.

The requested API identifiers are `gpt-5.6-sol`, `gpt-5.5-2026-04-23`, `gpt-6-astra`, and `claude-sonnet-4-6`. Record requested and returned identifiers, provider, date, reasoning settings, and complete call configuration. Model routing is fixed across arms; reasoning-enabled calls omit temperature and top-p overrides. Mining judges share task evidence and dimensional rubrics, with at most 8,192 generated tokens per artifact and repeat. Their scores select feedback, not human labels or independent assessment endpoints. All required evidence and outputs must fit without truncation, checked before splitting with each model's tokenizer and image accounting. Optimizers share input caps while retaining native history policies.

Baseline adaptation. Table A5 specifies native optimizer semantics and editable spaces. Adapters pin source commits, model routing, and Train/Val access before comparison. Learned fusion fits a regularized pairwise logistic model to frozen signal-feature differences on Train, then scores outputs independently. Feature normalization, missingness handling, and regularization are fitted or selected on Train. Text-only EvalOpt uses SkillOpt's editable fields. Adapters change transport only as needed for the matched model; unmatched model-access configurations are excluded from the controlled comparison.

Resource matching. Within each domain, all methods use the same Tune-calibrated Seed pair cost and frozen model/tool price schedule to convert measured use into search units. One-pass pair cost covers both artifacts and their evaluator calls and tools, but no optimizer call. The shared cap charges proposals, reflections, diagnostics, failed and rejected attempts, repeats, retries, and Val selection; mining and terminal Test assessment are itemized separately, and end-to-end comparisons include mining. Annotation effort is reported separately from model/tool cost. Deployment shares

Table A4: **Shared models and evaluator-search settings.** Evaluator weights remain fixed. Downstream policy updates are specified separately in Table A13.

Setting	Configuration
Evaluator execution	GPT-5.6 Sol; shared by Seed and all candidate and retained evaluators; low reasoning effort
Evolution and optimization	GPT-5.6 Sol for EvalOpt, SkillOpt, GEPA, and Meta-Harness; medium reasoning effort
Mining judges	GPT-5.5, GPT-5.6 Sol, GPT-6 Astra, and Claude Sonnet 4.6; fixed dimensional rubrics; 3 repeats/model; medium effort, with adaptive thinking for Claude
Other mining signals	Applicable dimensional checks from Table A1, excluding reserved assessment measures; 20% random anchor
Input and generation caps	65,536 input tokens per call; at most 8,192 generated tokens per optimizer call or across one artifact evaluation, including reasoning tokens
Tools	At most 8 external tool calls per artifact evaluation; 30 seconds per call; fixed rendering, tool versions, and image preprocessing across arms
Local measurement	28 cases, up to 4 targets and 8 sampled contrasts; fixed direction panel of at most 8 cases; missing panel cases added; 2 repeats initially and for a permitted fresh confirmation
Search budget	100 units = cost of 10,000 one-pass Seed pair evaluations, calibrated on Tune; reserve 20 units for Val; at most 60 charged candidate attempts
Direction limits	At most 18 charged steps; 2 provisional steps; 3 consecutive charged rejections; scope starts at level 2
Retention and assessment	Val every 3 charged steps, at direction exit, and finally; 3 repeats/artifact on Val/Test
Randomization	Split seed 42; search seeds 11/22/33/44/55 for data ordering and search randomization; API sampling is recorded separately

Table A5: Main baseline contract. All adaptive rows use the same permitted human feedback. Native search semantics are preserved; differing editable spaces are explicit.

Method	Optimized object	Required fidelity and interpretation
Seed evaluator	None; initial full evaluation program	Same seed, tools, routes, and score interface used to initialize program evolution
Learned signal fusion	Weights over fixed signal features	Fit on Train preferences; hold features, model, and observation path fixed
SkillOpt (Yang et al., 2026a)	Skill text	Preserve bounded edits, validation gate, rejected-edit buffer, and specified meta updates; pin paper-style variant
GEPA (Agrawal et al., 2026)	Full program through a declared adapter	Use official engine and program-capable interface; preserve reflection, population selection, and configured merge policy
Meta-Harness (Lee et al., 2026)	Harness code	Preserve filesystem access to prior code, scores, and traces; do not restrict parents to the current best
EvalOpt	Full evaluation program	Active direction, Train acceptance, periodic Val selection, and stopping rules

input, context, tool, time, and token caps. Record actual attempts, input/output/reasoning tokens, calls, API charges including cache and service tier, local accelerator costs, and wall time. Attempt and resource ceilings do not guarantee completed runs.

B.2 CONTROLLER AND MEASUREMENT

The controller separates execution validity, Train control, and Val retention. Algorithm 1 gives the loop; numerical settings below specify the reference implementation rather than tuned optima.

Algorithm 1 Direction development with reference updates and bounded exploration

Require: Seed E_0, Train, sealed Val, frozen measurement and control settings
Measure E_0 on Val; set $B \leftarrow E_0$; initialize history
while search resources remain **do**
  Propose direction d; initialize $R = W = B$, scope, and direction limits
  Freeze direction panel $\mathcal{D}_d$ from the selected Train cases
  while d is active and its resources remain **do**
    Save parent $P \leftarrow W$; submit candidate C and pre-execution predictions
    Validate the edit and measure P, C, R under matched Train conditions
    if required executions are incomplete **then**
      Record the interruption; **return** the last valid B
    end if
    Record effective changes, declared-check outcomes, gains, and declared contrast-case losses
    if C qualifies for the single fresh confirmation measurement **then**
      Remeasure all compared evaluators on the same Train cases under its frozen scope
      Use the fresh measurement, including check outcomes, for the decision
    end if
    Apply the rules below to confirm $R = W = C$, stage $W = C$, or roll back $W = R$
    Update pending age, rejection count, and scope
    Retain observations and interpretations separately, including unsuccessful attempts
    if a checkpoint is due and W is execution-valid **then**
      Compare W, B on full Val; apply Equation 7
      Release only promotion and retained score; leave Train states unchanged
    end if
    End d on the evolution agent's recorded decision or a direction limit
  end while
  Perform the scheduled exit checkpoint; archive the direction before reset
end while
Complete any outstanding final checkpoint; freeze and return B

State and declarations. Algorithm 1 separates Train updates to R/W from Val updates to B. A new direction initializes $R = W = B$, so its reference need not be Train-confirmed; all inherited changes must have passed Val retention. The agent records the target failure and correction, and a change of correction opens a new direction. Each edit identifies changed components and intended effects. Optional numerical predictions specify metric, reference, cohort, and units; predictions and checks cannot be relabeled after measurement. Cross-component edits are measured as bundles unless separate interventions are valid. A scaffold also declares its downstream use and intermediate success condition, without requiring connection on the next revision.

An intermediate check specifies a case, structured field, and predicate. Eligible predictions must hold on every repeat, with a predicted field changing or node being created on every repeat. Configuration, calls, and artifacts distinguish unrealized changes from produced-but-unused evidence, but fulfilled checks alone do not establish preference gain. Reason-text-only, final-score-only, purely semantic, and all-preservation checks remain unresolved. For scoring-active edits, failed checks reject the candidate and unresolved checks block confirmation. Absent checks do not veto measured gain and are not recorded as passed. Scoring-active scaffolds need passed intermediate checks to stage; disconnected scaffolds instead need the execution preflight below.

Measurement identity and panel. The immediate parent $P = W$ and candidate C use matched cases, scoring, resources, and unaffected inputs. Cache identity covers the evaluator, dependencies, inputs, model/tools, scoring configuration, and replicate; replay is not a fresh repeat. Edits

invalidate dependent artifacts while permitting reuse of compatible intermediate outputs. Task-only preprocessing cannot inspect the evaluated output. Train repeats rerun output-dependent judgments conditional on compatible fixed preprocessing; frozen Val/Test runs remain end-to-end. Required incomplete seed or candidate measurements interrupt search and retain the last valid B, without automatic evidence recovery. The fixed direction panel $\mathcal{D}_d$ contains the first at most eight selected Train IDs. The default local roster has 28 cases, including up to four targets and eight sampled contrasts, with additional Train cases filling the remainder. Missing panel cases are added, so the roster can exceed 28. Compare C with P on the full roster and with R on the panel.

Gain, contrast-case losses, and uncertainty. For each case i and evaluator E, let $o_{i,r}(E)$ be one if repeat r ranks the human-preferred output strictly higher, and zero for a wrong ordering or tie. Unavailable executions have no correctness value. With m complete repeats, $\widehat{p}_i(E)$ is their mean. For evaluators E, F on a prespecified cohort $\mathcal{C}$, the paired mean gain is

$$\widehat{g}_{\mathcal{C}}(E, F) = \frac{1}{|\mathcal{C}|} \sum_{i \in \mathcal{C}} [\widehat{p}_i(E) - \widehat{p}_i(F)]. \tag{5}$$

The implementation uses the sum $G_{\mathcal{C}} = |\mathcal{C}|\widehat{g}_{\mathcal{C}}$, equivalent to Equation 5 after rescaling. Its empirical uncertainty buffer for comparing E and F is

$$\mathrm{SE}_{\mathcal{C}}(E, F) = \left[\frac{1}{m} \sum_{i \in \mathcal{C}} \big(s_i^2(E) + s_i^2(F) \big) \right]^{1/2}, \tag{6}$$

where s_i^2 is the within-case sample variance across repeats. The gain is paired by case, not by arbitrarily aligned stochastic repeat slots. Independent permutations of repeat order therefore leave every comparison unchanged.

The buffer is a search heuristic, not a confidence interval, significance test, or estimate of generalization. Zero empirical variance does not establish noiseless execution, and shared evidence or correlated calls limit the independent-arm approximation. Only explicitly declared contrasts are guarded; randomly sampled contrasts are not. A decrease in any guarded case's mean correctness blocks confirmation, but does not alone require rollback or prohibit staging within provisional limits. Margins, traces, ranking flips, score variation, cost, and failures remain diagnostics rather than alternative confirmation criteria.

Fresh confirmation. With two initial repeats per evaluator and case, positive panel gain against R within the uncertainty buffer permits one fresh measurement if execution is effective and complete, local and declared contrast gains are nonnegative, and checks pass or are absent. Remeasure P, C, R on the same cases with the same repeat count and conditional execution protocol. The initial batch determines eligibility; only the fresh outcomes and checks determine the candidate decision, even when less favorable. Both batches remain in the diagnostic history, and equivalent execution identities cannot request another confirmation draw. This conditional remeasurement is a search heuristic, not a calibrated test or a guarantee against selection bias.

Train decision rules. After completeness checks, an effective-change check compares deterministic identities of the parent and candidate scoring computations. The identity includes scoring inputs, active prompts, routes, tool dependencies, and aggregation, excluding unused configuration; it tests computation changes, not accuracy gains. An unchanged identity triggers rejection unless the edit is a predeclared scaffold with a passed execution preflight. The preflight independently executes the disconnected capability, allowing bounded staging but not confirmation until the scoring computation changes. For a changed scoring computation, a failed declared mechanism check rejects the candidate without staging it, followed by the rules below.

First, supported harm relative to either P or R triggers rollback to R and narrows the scope by one level. Harm means at least two repeats with $G < -\mathrm{SE}$, or the stop-loss condition $G + \mathrm{SE} \leq -2$ correct pairs. Second, confirmation requires at least two repeats, panel gain $G_{\mathcal{D}_d}(C, R) > \mathrm{SE}_{\mathcal{D}_d}(C, R)$, no declared contrast-case regression in either comparison, nonnegative local mean gain, and no failed or unresolved declared check. Confirmation sets $R = W = C$.

Otherwise, a valid unresolved ordinary edit can remain provisional, including one with zero or uncertain gain or a declared contrast decline, provided neither comparison triggers the aggregate

harm rule. A scaffold that changes the scoring computation additionally needs passed intermediate checks. Staging sets $W = C$ without moving R. The chain permits two charged provisional steps; a third unresolved step restores $W = R$ and records expiration. Confirmation or rollback clears pending age; charged rejection restores $W = R$ and clears the provisional window. Uncharged generation retries retain W and pending age, subject to the separate failure limit. Comparison against R keeps accumulated loss visible as the parent changes.

Three consecutive charged rejections end the direction; confirmation or provisional continuation resets that rejection count. A direction also ends at 18 charged steps or on the agent's recorded diagnostic judgment. These limits bound exploration, not the correctness of that judgment; a single rejection or Val non-promotion does not itself end the direction.

Scope adaptation. Permitted edits have three levels: one component, multiple components within one functional group, and coordinated changes across groups. Scope constrains each edit from W, not cumulative distance from R. The initial level is two; two consecutive supported reference updates expand it by one, while supported harm contracts it by one, within levels one to three. The 24,000-character guard and call limits bound resources, not semantic distance. The categorical policy draws on trust-region adaptation (Conn et al., 2000), without establishing the assumptions needed for stochastic convergence guarantees (Bandeira et al., 2014; Chen et al., 2018; Blanchet et al., 2019).

Val retention. The reference cadence is every three charged steps, plus direction-exit and final checkpoints. The cadence is frozen before search; the number of checkpoints depends on realized steps and direction endings within the reserved Val budget. Compare eligible W and B on complete frozen Val under identical scoring and coverage requirements:

$$B \leftarrow \begin{cases} W, & \widehat{J}_{\mathrm{Val}}(W) - \widehat{J}_{\mathrm{Val}}(B) \geq \tau_{\mathrm{Val}}, \\ B, & \text{otherwise}, \end{cases} \tag{7}$$

where $\tau_{\mathrm{Val}} > 0$ is prespecified. The agent receives only promotion and retained score, never an unpromoted candidate's score; Val does not alter Train states or direction control. This restriction follows the Ladder's motivation (Blum & Hardt, 2015), without its theoretical bound or elimination of adaptive-selection risk. Reserve the last eligible W for the scheduled final comparison before reset; an incomplete comparison retains the last valid B.

C Complete results and ablations

C.1 Per-domain evaluator quality

Table A6 preserves all six means for every C1 domain and adds the stated per-block denominator. Differences use unrounded means: $69.55765 - 61.51504 = 8.04261$ pp, displayed as 8.0, rather than subtracting 69.6 and 61.5. The authors report requested = complete = decisive and zero missing outcomes within the eligible C1 count cohort. Under that condition, Equation 4 reduces to the accuracy point value. This does not establish the absence of human ties, exclusions, or missing annotations in the source data.

C.2 Search ablations and trajectory diagnostics

Search and mining diagnostics use SV3, IG1, TG1, and AR1, without changing their C1 weights. All search arms share selected feedback, seed, model, scoring, Val access, and cost except for the declared intervention. Table A8 supplements Figure 4a with absolute accuracy and revisions per direction.

Policy controls. History-based search can replace the correction after each revision while retaining diagnostics. A switch initializes $R = W = B$, a new panel, and initial scope; continuation keeps the common controller. Fixed patience uses $k \in \{2, 4, 6\}$ attempts selected under a shared Train-only tuning allowance. The agent cannot voluntarily end early, but shared rejection, integrity,

Table A6: **C1 accuracy in every domain.** Values average five scoring blocks per method. N is the stated decisive-pair count per block (11/22/33/44/55). All domains have equal weight in Overall. Fusion: learned signal fusion; Meta: Meta-Harness.

Domain	N	Seed	Fusion	SkillOpt	GEPA	Meta	EvalOpt
SV1	120	57.0	58.2	61.7	58.8	60.2	64.7
SV2	160	68.0	73.0	73.1	70.4	71.6	74.8
SV3	200	64.5	65.0	67.8	68.4	69.0	74.7
SV4	78	54.9	57.4	57.2	60.0	55.6	66.2
SV5	67	63.9	64.8	64.2	66.0	64.8	72.5
IG1	200	65.0	64.5	69.4	69.9	70.3	72.5
IG2	200	64.0	66.8	65.8	68.1	66.6	74.5
IG3	64	59.1	61.6	60.9	63.4	62.2	70.0
IG4	200	54.0	56.3	54.2	55.8	53.9	58.0
IG5	50	55.2	56.8	55.2	57.2	55.6	64.0
IG6	64	54.4	55.3	58.4	55.0	56.6	64.4
TG1	200	69.0	68.5	71.1	72.9	71.9	74.0
TG2	200	60.0	61.9	63.3	60.4	62.7	69.8
TG3	200	70.0	72.6	74.4	71.5	72.7	77.2
TG4	200	68.0	70.1	71.2	69.7	71.7	77.3
TG5	200	71.0	72.0	71.4	72.3	70.7	74.6
AR1	200	63.5	64.5	64.6	63.9	64.4	67.2
AR2	30	53.3	52.7	54.0	55.3	53.3	60.0
AR3	64	54.1	55.3	56.2	58.8	55.6	65.3

Table A7: **C1 Overall descriptive summaries.** Accuracy is in percent, differences in percentage points (pp). Author-supplied SDs use five equal-domain block averages, not 19 domain means.

Method	Overall mean $\pm$ SD	EvalOpt minus method (pp)
Seed	61.5 ± 1.8	8.0
Learned signal fusion	63.0 ± 1.7	6.5
SkillOpt	63.9 ± 1.1	5.7
GEPA	64.1 ± 1.5	5.5
Meta-Harness	63.7 ± 2.0	5.9
EvalOpt	69.6 ± 0.9	—

and resource limits remain active. Immediate branching ends after the first execution-valid nonconfirming candidate and applies the same new-direction initialization. These contrasts include the initialization consequences of switching, not just a change of direction identifier.

Information and state controls. Reduced feedback retains aggregate gain, loss, uncertainty, checks, transitions, and cost while withholding case-level traces. No retained history exposes only the current diagnostic record, removing earlier records and summaries from every proposer context. Both retain all controller checks and charge actual context/proposal costs. No staging requires reference confirmation before advancing W; fixed scope removes adaptive scope changes; text-only EvalOpt matches SkillOpt's edit fields. Shared repair allowances and Val cadence remain unchanged. Fresh online searches, not replay alone, test these interventions.

Table A8: **Matched-cost search comparison.** Pair accuracy is in percent; revisions per direction measure search depth. Every condition uses the same search cap of 100 units.

Condition	Pair acc.	Revisions/direction
History-based search	67.4	1.6
Fixed patience	68.2	3.6
No retained history	67.6	3.1
EvalOpt	72.1	3.8
Reduced feedback	68.0	3.5
No staging	68.1	3.4
Fixed scope	68.7	3.8
Immediate branching	67.1	1.0
Text-only EvalOpt	67.9	3.0

Controlled diagnostics. Table A9 separates useful evidence awaiting integration, an unrealized tool call, and a realized harmful correction. Each policy uses the same seed, available diagnostics, and prespecified allowance. Integration requires actual execution and downstream use; harmful-correction termination requires realized execution and independently specified loss. The three percentages have different denominators and are not additive.

Table A9: **Controlled diagnostic decisions.** Entries are percentages under each scenario's prespecified allowance. Integrate: complete a useful downstream evidence connection; repair: realize the intended tool call; end: stop a realized harmful correction.

Policy	Integrate ↑	Repair ↑	End ↑
History-based search	36	47	66
Fixed patience	63	59	43
EvalOpt	71	74	76

Trajectory assessment. Sample 30 directions per domain by stable IDs, ten each with confirmed progress, nonconfirmed agent ending, and forced ending, pooling searches without replacement and reporting shortages. Two independent reviewers assess semantic continuity and link failure, realized or unused evidence, subsequent edits, $R/W/B$ transitions, and cost. Retain failed, exhausted, and censored trajectories, including attempts without follow-up, and allow multiple or unknown causes. These descriptions distinguish observations from agent interpretations and do not establish causality from later success.

C.3 Preference-mining comparisons

At each label budget in Table A3, restart evolution from the same Seed with fixed Test cohorts, a common deduplicated pool, and identical feedback detail. Only selected labels are revealed under the group-level access protocol in Appendix A.2. Figure A1 compares label budgets and reallocates the anchor's 20% budget to ranked pairs in the no-anchor condition. Table A10 gives the secondary random/mined $\times$ fixed-patience/EvalOpt comparison, with mining, search, and amortized Tune costs itemized separately.

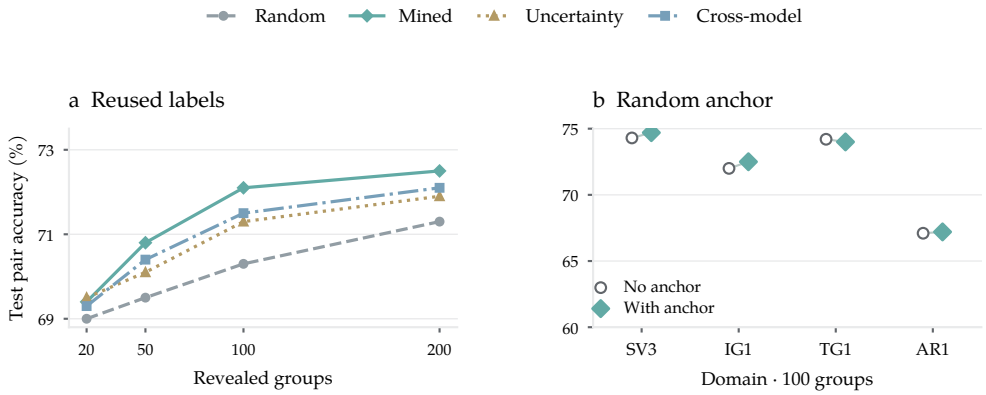

Figure A1: **Preference-mining efficiency.** (a) Held-out pair accuracy versus revealed groups in existing-label replay. (b) Mined feedback with and without the random anchor at the same 100-group budget, by domain.

Table A10: **Mining and end-to-end comparison.** The two-by-two design uses 100 revealed groups in every cell. Decisive-pair accuracy is in percent. Total model-and-tool cost per search includes search, mining, and Tune amortization in relative units, not human labor. Fixed-patience tuning costs 60 units per domain, or $60/5 = 12$ units per search; EvalOpt has no such tuning charge.

Feedback	Policy	Pair acc.	Search	Mining	Tune	Total
Random	Fixed patience	67.0	100	2	12	114
Mined	Fixed patience	68.2	100	6	12	118
Random	EvalOpt	70.3	100	2	0	102
Mined	EvalOpt	72.1	100	6	0	106

Mining and continuation effects. From Table A10, mined minus random feedback is 1.2 pp under fixed patience and 1.8 pp under EvalOpt. EvalOpt minus fixed patience is 3.3 pp with random feedback and 3.9 pp with mined feedback. The descriptive difference-in-differences is $(1.8-1.2) = (3.9-3.3) = 0.6$ pp. These contrasts are calculated from the reported cell accuracies, without adding runs or changing costs. These are descriptive contrasts, not evidence of a significant interaction.

What disagreement mining selects. We examine SV3, IG1, TG1, and AR1 at the same 100-group feedback budget, combining 80 ranked pairs and 20 random-anchor pairs into the selected feedback set. Each domain has 224 eligible Train groups with decisive overall preference labels, one pair per group, and five selection seeds. Existing labels retain their source semantics; source ties and missing labels are outside this diagnostic cohort. This composition diagnostic uses a separate two-rubric-source configuration, not the four-judge experimental roster. Both sources cover four dimensions and overall quality, supplemented by browser requirement checks (SV3), CIDEr (IG1), or SummaC (TG1); AR1 has no additional checker. For each dimension, comparison coverage is the fraction of pairs with at least two available source-group votes, including tied votes. Across the four dimensions, its ranges for Pool / Selected are SV3 95.1–100.0% / 95.6–100.0%; IG1 94.6–100.0% / 95.2–100.0%; TG1 94.6–99.6% / 95.6–99.0%; AR1 93.8–98.7% / 95.2–98.6%. Missing votes remain unavailable rather than counting as agreement.

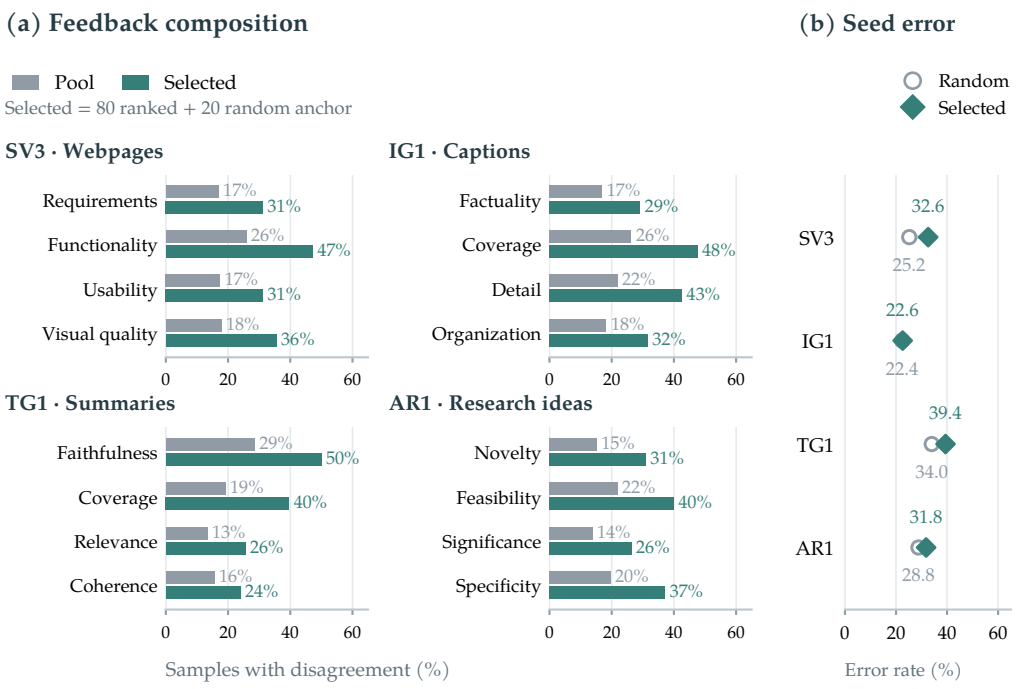

Figure A2: **Feedback composition and Seed errors.** (a) Pairs with observed evaluator disagreement on each dimension, as a percentage of all pairs in the Pool (gray) or Selected feedback (teal). Selected combines 80 ranked and 20 random-anchor pairs. A pair can contribute to several dimensions, so percentages need not sum to 100%. (b) Seed error for Selected versus 100 uniformly sampled pairs, using decisive overall labels and counting evaluator score ties as errors. Pool percentages use 224 pairs; Selected and Random percentages pool 500 appearances across five selections, not independent samples.

Figure A2a describes selection's effect on observed disagreements, which depend on the evaluators and their coverage rather than intrinsic dimension difficulty. Panel b audits Seed errors using overall preference labels revealed only after selection, without dimensional human labels. Learning benefit is assessed separately in Figure A1.

For the anchor audit, we deduplicate pairs across selection seeds and retain those with decisive human preferences and matching non-tied overall votes from both rubric judges. The counts opposing

this shared model preference are SV3 4/55, IG1 5/59, TG1 5/51, AR1 4/47. These conditional counts describe shared errors accessible to random sampling; the fixed-budget anchor comparison tests their contribution to learning.

C.4 SV1 DATA COLLECTION AND ANNOTATION YIELD

This section documents the collection of SV1's C1 feedback and compares its question-selection policy with random sampling under the same annotation budget. Both arms use the same source pool, initial Seed evaluator, and frozen Val/Test cohorts as C1, not an additional slide dataset. For collection, we use briefs from PPTBench-Generation, with two single-slide outputs per brief generated by GPT-5.6 Sol under a shared prompt and budget with different generation seeds. The public resource supplies inputs, not the pairwise human labels for this study. The C1 allocation is 200 Train groups, including 20 Tune groups excluded from acquisition, with 40 Val and 120 Test groups (Table A2). Each acquisition arm selects 100 pairs from the remaining 180 Train groups; this count includes pairs whose questions receive ties or remain unresolved. Source-document and near-duplicate groups must remain within one split, and the source manifest, generation settings, renderer, and output hashes must be frozen before mining. Eligibility requires a self-contained brief, accessible supporting material, and two inspectable renders; exclusions precede both sampling arms.

Question selection and blinding. The mined arm selects 20 random-anchor pairs before ranking and 80 pairs by the within-dimension disagreement count in Equation 2. For each ranked pair, it samples one or two conflicting dimensions and one or two additional non-conflicting dimensions, uniformly without replacement within each set. The requested counts are chosen by independent input hashes before annotation and capped by the available dimensions in each set. Anchor pairs receive three uniformly sampled dimensions. Every selected pair also receives an overall question. The random arm samples 100 pairs uniformly from the same Train pool and matches the mined arm's per-pair question-count distribution, but samples dimensional questions uniformly. The criteria in Table A1 form a question bank, not a mandatory checklist: the interface displays only the sampled questions, with no fields for unqueried dimensions. Each arm requests 400 questions (20 pairs with 3, 60 with 4, and 20 with 5 questions), including 100 overall and 300 dimensional questions, for 800 initial individual ratings. The comparison tests the combined pair-and-dimension sampling policy.

Raters see the brief, supporting material, and equally sized slide renders in randomized A/B positions, with model identities, signal votes, and sampling status hidden. Two independent raters answer overall preference first, followed by the sampled dimensional questions in randomized order. Each question offers A, B, tie, or unable to judge; tie means comparable quality, whereas inability indicates insufficient evidence. Raters must be fluent in the brief's language and familiar with presentation review, with rubric calibration on separate practice items before annotation. An identical non-missing response from both raters is accepted, including a tie. Otherwise, a third blinded rater answers that question, and two matching non-missing responses determine its label; without that agreement it remains unresolved. Each arm reserves 100 ratings for adjudication after the 800 initial ratings; if demand exceeds this reserve, a prespecified hash order determines which questions are adjudicated, leaving the remainder unresolved. Unused reserve is not spent on selecting additional pairs. Unasked dimensions remain absent rather than being filled by overall preference or model votes.

Annotation yield. A decisive label is a resolved A or B preference; a tie is resolved feedback but supplies no strict ordering for decisive-pair accuracy. An unresolved question lacks sufficient rater agreement and is distinct from a tie. Table A11 separates these outcomes for overall and sampled dimensional questions, without treating a dimensional preference as an overall label. The primary yield is decisive labels per 100 allocated individual ratings, using the same 900-rating budget for both arms; actual rating costs are also reported. This comparison tests whether disagreement-based selection spends fewer questions on equal-quality comparisons, rather than assuming that model disagreement guarantees a decisive human preference. Shared pair–question labels are reused across arms, while each arm is charged the ratings needed to obtain its own feedback set. Val and Test annotation costs are common to both arms and reported separately from Train acquisition.

Connection to C1. The mined feedback is the SV1 feedback shared by the C1 optimizers; the random arm is an acquisition control, not a second test cohort. Evaluator accuracy is reported in the

Table A11: **SV1 annotation yield at matched budgets.** Both arms use the C1 Train pool and 900 individual ratings of allocated budget, including adjudication. Decisive preferences, ties, and unresolved outcomes partition the requested questions; dimensional labels retain their criterion identities.

Measure	Random	Mined
Allocated individual ratings	900	900
Pairs / requested questions	100 / 400	100 / 400
Initial + adjudication ratings	800 + 84 = 884	800 + 98 = 898
Overall: decisive / tie / unresolved	54 / 40 / 6	78 / 14 / 8
Dimensions: decisive / tie / unresolved	182 / 104 / 14	246 / 38 / 16
All decisive labels / questions	236 (59.0%)	324 (81.0%)
All ties / questions	144 (36.0%)	52 (13.0%)
All unresolved / questions	20 (5.0%)	24 (6.0%)
Decisive labels per 100 allocated ratings	26.2	36.0

C1 results using the common Seed, search budgets, and scoring protocol, without a separate accuracy estimate here. Only acquired Train labels enter revision, preserving dimension identities, ties, and missing fields. Three blinded raters judge overall preference for every Val and Test pair, requiring two matching non-missing responses under the same rule as Train. These held-out annotations require 120 Val and 360 Test ratings shared by both arms. The acquisition comparison concerns one feedback collection per arm, not five independent annotation studies.

D Downstream applications

D.1 Shared setup and assessment

C2 tests critique-guided refinement and C3 tests reward-guided training, using the shared assessment in Table A12 for Figure 4b.

Table A12: **Downstream assessment by domain.** Human preference is the primary endpoint in C2/C3. Computable measures assess specific attributes, not a shared cross-domain quality score.

ID / inputs	Human rubric source	Human assessment	Computable complement
SV3 / WebDev	Adapted WebGen-Bench appearance criteria and operation–expected-result checks (Lu et al., 2025)	Requirement satisfaction, usability and appearance; pairwise overall choice	Render success; reserved requirement-test pass rate
IG1 / DOCCI	CapArena annotation guidelines, Appendix H (Cheng et al., 2025)	Precision, salient information, hallucination and conciseness; pairwise choice	VQAScore image–caption alignment (Lin et al., 2024)
TG1 / TL;DR	TL;DR comparison and rating instructions, Appendix C.5 (Stiennon et al., 2020)	Overall choice; 1–7 coherence, accuracy, coverage and overall ratings	AlignScore source–summary consistency (Zha et al., 2023)
AR1 / Ideation Arena	Human-review criteria, Appendix A.2 (Chen et al., 2026c)	Overall choice; novelty, feasibility, significance and specificity preferences	None

Data allocation. Each domain uses 512 training, 64 development, and 200 final-assessment inputs, disjoint from evaluator Train/Val/Test. Freeze all roles before mining, without later additions. C2/C3 share final-assessment inputs but obtain separate outputs and new judgments, requiring 1,296 eligible groups per domain: 520 for C1 and 776 downstream, counted once across arms and seeds. Development inputs support generator checks and reward calibration, not evaluator evolution. Evaluator-search seeds 11/22/33 are paired with generator or training seeds in prespecified blocks, not selected by evaluator score.

Source overlap and eligibility. WebDev groups use normalized user specifications, with repeated requests and near-duplicate templates kept together; restrict the cohort to single-turn tasks with executable artifacts. CapArena's human comparisons reuse DOCCI images, so image IDs and near-duplicate scenes are grouped jointly across both releases. Use CapArena for C1, DOCCI Train for downstream training/development, and DOCCI Test images outside CapArena for downstream assessment; reference descriptions are withheld from generators. TL;DR groups use post IDs and normalized post text; downstream inputs come from the released source-and-reference corpus after removing all comparison-cohort posts. For ideation, group identical and near-duplicate literature contexts, using the same seed-paper group wherever its identity is available. Shared background citations alone do not merge otherwise distinct contexts; quantify this overlap and record unavailable seed identities in the manifest. The complete supplied literature context must fit both the generator and reward evaluator; exclude overlength inputs before allocation rather than truncating evidence underlying an existing preference label. Public-source counts precede these eligibility checks, and API processing also requires permission under each source's terms.

Table A13: **Domain-specific downstream inputs and generation limits.** C2 uses GPT-5.6 Sol; C3 uses Qwen3.5-9B. L is the output token cap. Shared allocations and study-specific settings are given in the text.

Domain	Input source	Input token cap	L
SV3	WebDev Arena	16,384	4,096
IG1	DOCCI	8,192	512
TG1	TL;DR	8,192	256
AR1	Ideation Arena	32,768	1,536

Models and inputs. Table A13 specifies domain-specific limits. GPT-5.6 Sol generates C2 initial outputs and revisions, with reasoning effort `none` and no temperature or top-p override. C3 uses the public post-trained Qwen3.5-9B checkpoint, revision `c20223623576`, in all four domains, with thinking disabled throughout training and assessment. The Base, Seed-reward, and EvalOpt-reward arms share this checkpoint and tokenizer; only the two reward arms receive policy updates. Seed and EvalOpt critiques and rewards use the GPT-5.6 Sol evaluator settings in Table A4. The captioner receives the image, but not the DOCCI reference. The webpage model generates a complete HTML/CSS/JavaScript artifact; browser execution supplies evaluator evidence rather than an extra trainable policy. Summarization uses the source post, and ideation uses a closed literature context with no online retrieval.

Human assessment. Three blinded raters assess each output pair under fixed, shared instructions after calibration on disjoint examples. They see the task, permitted evidence, and anonymized outputs, without critiques, rewards, automatic scores, or reference captions/summaries. Choices are A, B, tie, or unable to judge; AR1 also records Both Bad, counted as a tie and reported separately. AR1 raters must know the topic and supplied literature, deferring unsupported judgments. Tie-aware preference is $(W + T/2)/(W + L + T)$, with abstentions and missing outputs reported separately. Record rater backgrounds, calibration, agreement, and person-time.

Table A12 gives domain criteria and rubric sources. SV3 requires inspecting the running page, including interactions, with task-specific operation–expected-result checks. Its 1–5 appearance anchors adapt WebGen-Bench's model-grading prompt rather than an existing human pairwise instrument. IG1 follows CapArena's human guidelines, including not rewarding length alone; TG1 uses TL;DR's comparison instructions and separate 1–7 rating anchors. AR1 asks which proposal merits investment under the listed criteria, assessing a proposal rather than a completed finding (Si et al., 2025a).

Regression judgments. Each regression flag identifies a newly introduced material problem relative to the reference output: broken required interaction or unusable layout (SV3), a salient hallucination or lost essential description (IG1), a central factual distortion or omission (TG1), or an unsupported central premise or infeasible experimental dependency (AR1). Raters identify the affected requirement, image region, source passage, or proposal statement. Majority vote determines

the flag; minor stylistic changes do not qualify. Regression is measured against y_0 in C2 and Base in C3, not necessarily the pairwise comparison arm.

Reference-free automatic measures. The following specifies the assessment protocol for the results in Table A14. These measures use the task specification, image, or source document rather than a human-written reference output. SV3 runs prespecified assertions in isolated Chromium at 1280×800 with a 30-second load limit. Render success requires nonblank task content; requirement pass rate averages the fraction of passed assertions equally over tasks. Artifact-caused failures fail dependent assertions, while browser-service outages remain missing. IG1 uses VQAScore (Lin et al., 2024) with frozen `clip-flant5-xxl` in `t2v-metrics` 3.0, scoring the probability of "Yes" to the fixed image–caption compatibility question using the complete generated caption and its source image. No reference caption or generated replacement caption is supplied. TG1 uses AlignScore-large (Zha et al., 2023), with the released RoBERTa-large checkpoint and `nli_sp` mode, treating the original post as context and the generated summary as the claim. The released chunk–sentence aggregation handles the full source rather than comparing the summary with a human TL;DR. Both scores are reported on a 0–100 scale, not as percentages of correct outputs. VQAScore measures compatibility rather than exhaustive visual coverage, and AlignScore measures source support rather than coverage or writing quality. Their calibration on long DOCCI captions and Reddit summaries remains uncertain, so neither substitutes for human preference. AR1 retains expert preference and regression judgments without an automatic proxy for scientific novelty.

Assessment boundary and reporting. Reserve VQAScore, AlignScore, and final browser assertions from mining, evaluator tools, critique, reward, and checkpoint selection, while allowing general browser execution. SummaC remains an optimization-exposed mining diagnostic, not an independent downstream endpoint. Table A1 surveys alternatives, not the executed tool roster. Assess the same final outputs from every arm, recording versions, preprocessing, manifests, output lengths, completion counts, and missing outcomes. Empty or invalid outputs receive zero automatic score; infrastructure failures and inputs exceeding a metric's context limit are reported as missing, never silently truncated. Report missingness and output length alongside the scores to detect selective exclusion or gains from shorter descriptions. Average over inputs within each paired block, then equally over the three blocks, without averaging incompatible metric scales. For descriptive paired effects, subtract Seed from EvalOpt within each block and compute the mean and sample SD of the three differences (variance denominator 2). Per-arm SDs cannot be added or pooled under an independence assumption to recover this paired SD. For C2 requirement pass, the three block differences are 7.1, 3.7, and 5.4 percentage points, giving a mean of 5.4 and a sample SD of 1.7 percentage points. Table A15 reports the paired effects. Table A16 summarizes preference and per-arm outcomes.

Table A14: **Reference-free downstream checks.** Values are mean ± sample SD across three paired blocks of 200 inputs. C2 compares Seed- and EvalOpt-guided revisions; C3 compares their reward-trained policies. Browser outcomes are percentages; VQAScore and AlignScore are scaled scores ($\times 100$). All measures are higher-is-better.

		C2		C3	
Domain	Measure	Seed	EvalOpt	Seed	EvalOpt
SV3	Render success	98.2 ± 0.3	98.2 ± 0.3	96.5 ± 0.5	97.3 ± 0.6
SV3	Requirement pass	79.1 ± 1.0	84.5 ± 0.8	68.4 ± 1.2	74.3 ± 1.3
IG1	VQAScore	76.4 ± 0.7	79.5 ± 0.5	70.5 ± 0.7	75.5 ± 0.8
TG1	AlignScore	82.5 ± 1.0	85.1 ± 0.7	75.5 ± 0.9	79.1 ± 0.9

D.2 Critique-guided refinement

C2 branches each fresh initial output y_0 into no-critic, Seed-critic, and EvalOpt-critic arms, each with two successive revisions. Arms share the generator, settings, task evidence, and output cap L; critic arms additionally receive a frozen evaluator's critique, capped at 512 tokens per round under a common format. The current output and task are available in every round, including the same captioning image. Each branch generates at most $3L$ tokens including y_0, with critique and tool

Table A15: **Automatic-metric paired effects.** Values are mean ± sample SD of EvalOpt-minus-Seed differences across three paired blocks. Browser effects are in percentage points (pp); VQAScore and AlignScore effects are in score points on the 0–100 scale.

Measure	Unit	C2	C3
Render success	pp	0.0 ± 0.5	0.8 ± 0.8
Requirement pass	pp	5.4 ± 1.7	5.9 ± 2.5
VQAScore	points	3.1 ± 0.3	5.0 ± 1.1
AlignScore	points	2.6 ± 1.6	3.6 ± 1.7

costs charged separately. No intermediate human judgments or best-of-N selection are available. Seeds identify paired data/output blocks rather than deterministic API samples.

The primary comparison is EvalOpt versus Seed critique; secondary comparisons are EvalOpt versus no critic and each revised arm versus y_0. Five comparisons over 200 inputs, three seeds, and three raters allocate 9,000 pair judgments per domain, excluding adjudication and separate attribute ratings.

Table A16: **Downstream controls and per-arm outcomes.** Preference, major regression, and invalidity are percentages. Merged risk cells describe the first arm relative to y_0 (C2) or Base (C3), independently of the comparator. Values equally weight the four domains.

Contrast	Pref.	Major regression	Invalid
C2: Seed / y_0	57.5	10.0	1.0
C2: EvalOpt / y_0	63.6		
C2: EvalOpt / Seed	60.8	7.5	0.8
C2: EvalOpt / no critic	58.9		
C3: Seed / Base	57.5	12.5	1.3
C3: EvalOpt / Base	62.1	10.0	0.9
C3: EvalOpt / Seed	59.1		

D.3 Reward-guided training

C3 compares policies trained with frozen Seed or EvalOpt rewards against an untrained Base. Reward arms share the Base checkpoint, LoRA/GRPO settings, inputs, tools, and update budget; only the frozen evaluator differs. Both rewards score a common starting-policy pool for variance, ties, validity, and sensitivity to length or unsupported detail, under development-frozen preflight thresholds and stopping rules.

Only the preflight pool is shared: once policies diverge, each samples its own online trajectories.

Training recipe. Each reward arm starts from the same policy checkpoint with LoRA rank 16, scale 32, and dropout 0.05 on the language model's attention, gated-delta, and MLP projection layers. Embeddings, the vision encoder, and the vision–language merger remain frozen; exact adapter target names are fixed in the run manifest. GRPO uses eight distinct prompts per update and four on-policy completions per prompt, giving 32 sequences per optimizer update. Two seeded shuffled passes over the 512 training inputs give 128 updates and 4,096 completions per arm and seed. Generation uses temperature 0.8 and top-p 0.95, with the task-specific caps in Table A13. Use AdamW with learning rate 10^{-5}, $(\beta_1, \beta_2) = (0.9, 0.95)$, weight decay 0.01, and gradient-norm clipping at 1.0. The first 13 updates warm up linearly, followed by cosine decay to 10^{-6}. The clipped policy objective uses threshold 0.2, a fixed KL coefficient 0.02 against Base, and one optimization pass over each rollout batch. Advantages are standardized within each four-completion group with denominator floor 10^{-6}; zero-variance groups are retained in accounting but contribute zero policy-gradient advantage.

Reward and final assessment. Each frozen evaluator scores the same 256 Base completions (four per development input) to set its reward mean and standard deviation. Standardize scores with these

fixed calibration moments and clip to $[-3, 3]$, assigning invalid artifacts -3; use no additional correctness or length reward. One evaluator execution supplies each training reward, with the same decoding and tool limits in both arms. Required scorer failures interrupt the run rather than becoming fabricated rewards. Score ties, zero-variance groups, invalid outputs, and truncation rates are retained as diagnostics. Assess the terminal step-128 checkpoint; earlier training checkpoints are diagnostic only and are not selected by Test performance. Final generation uses temperature 0.7 and top-p 0.9 with one output per input and seed. Independent assessment compares EvalOpt-reward with Seed-reward and both with Base, allocating 5,400 pair judgments per domain across 200 inputs, three seeds, and three raters. Reward increases alone do not establish quality gains. The full training allocation is 24 runs (four domains, two reward arms, three seeds), with a 48-hour wall-clock ceiling per run; interruptions remain visible and are not silently replaced by successful seeds.