BUDGETED ERROR CORRECTION IN LLM-BASED MULTI-AGENT SYSTEMS FROM OBSERVABLE INTERVENTION RESPONSES

Anonymous authors
Paper under double-blind review

ABSTRACT

Similar failures in LLM-based multi-agent systems can require different repairs, but locating a failure does not by itself determine which repair will help. Under a shared budget, selection must further weigh uncertain repair gains against costs across systems. We study whether already acquired intervention responses can guide this selection before repair success is known. To investigate this question, we introduce EvidenceGraph, which compares intervention responses and repair outcomes across failure mechanisms in systems matched on task and topology. We further characterize theoretically when these observations improve repair selection and bound the gap to full-information control, separating repair coverage and information limits from prediction and allocation errors. To turn observable intervention responses into repair decisions, we introduce Observable Action-Effect Prediction (OAE), which uses observed answers and evidence margins to predict sequence-specific replay agreement as a proxy for repair selection. Its greedy allocator prioritizes predicted gains over no intervention per unit cost under the shared budget. In independent evaluation, OAE improves task success by 8.18 percentage points over a prespecified learned comparator at equal realized terminal spend.

1 INTRODUCTION

Errors in LLM-based multi-agent systems can propagate between agents and compromise final answers (Xie et al., 2026). Repeated agreement may reflect inherited evidence or a faulty shared source, so similar failures can require different repairs. Figure 1 illustrates this distinction. Masking the analyst–synthesizer connection can interrupt peer propagation, while a shared-source error may persist until the source is replaced. The visible failure alone need not distinguish these explanations (Pearl, 2009). Repair therefore requires assessing how the system responds to a proposed intervention and whether that response can improve the final answer.

Existing failure analysis and debugging research provides ways to examine where errors arise and what interventions change. Failure taxonomies and step attribution characterize and locate failures (Cemri et al., 2025; Zhang et al., 2025c), while replay-based debugging and communication control examine intervention responses (Ma et al., 2026; Shah, 2026; Huang et al., 2026). Locating a failure, however, does not by itself rank candidate repairs by their benefit, and validating a repair for one system does not determine its priority relative to repairs for other systems. In addition, repairing multiple task instances requires accounting for total repair expenditure. Tasks differ in how much they benefit from extra computation, and assigning the same allowance to every task limits reallocation toward those that would benefit more (Damani et al., 2025). We therefore adopt a total repair budget shared across task instances to control aggregate expenditure while allowing allocations to vary with each instance's expected repair gain.

To address this allocation problem, we investigate whether already acquired intervention responses can inform estimates of repair gains before success outcomes are known and guide selection under a shared budget. Estimating these gains requires considering both the baseline and the candidate sequence: a system likely to fail may be difficult to repair, while a sequence likely to succeed may add little to an already strong baseline. Moreover, gains depend on operation order, and additional steps can disrupt correct answers. These differences motivate examining how each system responds

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.

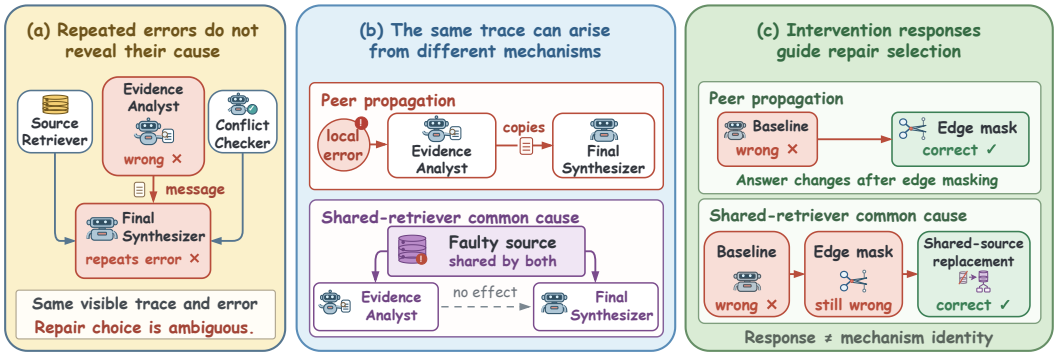

Figure 1: Similar failures can require different repairs: illustrative peer-propagation and shared-source cases. The lower sequence applies edge masking then shared-source replacement. Mechanism and correctness labels are reader-only; responses are not universal signatures.

to specific interventions. To characterize these responses, we compare answers and evidence margins before and after selected interventions. Tracking the margin alongside the answer captures shifts in the balance of support and opposition even when the final decision remains unchanged. These measurements inform sequence-specific predictions without being treated as unique signatures of the underlying failure mechanism.

To evaluate whether observable intervention responses can guide repair selection under a shared budget, we construct EvidenceGraph. The benchmark draws on Controlled EvidenceGraph, MATH-derived (Hendrycks et al., 2021b), and MMLU-derived (Hendrycks et al., 2021a) tasks, with MATH and MMLU adapted to assess evidence support for claims rather than original benchmark accuracy. For controlled repair comparisons, we construct six matched systems per task with different failure mechanisms but the same task, reference answer, wording, and topology. Within each system, ordered repair sequences replay separately from the same snapshot, so candidate repairs are compared from a common starting point. However, comparing repair outcomes does not by itself establish whether the available observations suffice to guide repair selection. To address this question, we characterize when observations improve repair selection and bound the gap to full-information selection under a shared budget, separating repair coverage and information limits from prediction and allocation errors.

To implement observation-based repair selection in this setting, we introduce OAE (Observable Action-Effect Prediction), which compares candidate sequences without requiring mechanism identification. OAE compares answers and evidence margins from baseline, edge-mask, and conditional source-swap responses to characterize how each system reacts to intervention. Based on these observations, it predicts whether each sequence's replayed answer matches its expected reducer answer and uses this agreement as a proxy for repair selection. Building on these predictions, OAE-GREEDY prioritizes repairs across systems by the ratio of predicted gain over no intervention to repair cost, then selects repairs within the shared budget.

We make three contributions.

- We construct a three-stratum matched EvidenceGraph benchmark that compares intervention responses across failure mechanisms in systems matched on task and topology. It evaluates whether these observable responses can guide repair selection under a shared budget.
- We characterize repair-response geometry through fixed, observable, and full-response values. We establish conditions for observations to improve selection and bound shared-budget regret, separating limits of repair coverage and available information from prediction and allocation errors.
- We introduce OAE and OAE-GREEDY for response-based repair selection. Sequence-specific predictors learn replay agreement from observable interventions, and the allocator uses predicted gains over no intervention per unit cost to select repairs under a shared budget.

Table 1: Coverage of reported protocols. ✓: explicit; △: narrower; –: not specified, not a capability exclusion. Inject: controlled faults; Matched: fixed task/surface/topology across mechanisms (△: paired or controlled comparisons); Repair: selectable repair bank (△: recovery/debugging/feedback); Ordered: alternative ordered repair sequences; Hidden: outcomes withheld until policy lock. Shared budget: across systems.

Benchmark	Inject	Matched	Repair	Ordered	Hidden	Shared budget
MAST (Cemri et al., 2025)	–	△	△	–	–	–
Who & When (Zhang et al., 2025c)	–	–	–	–	–	–
AgentProcessBench (Fan et al., 2026)	–	–	–	–	–	–
AgenTracer (Zhang et al., 2026)	✓	△	△	–	–	–
MAS-FIRE (Jia et al., 2026)	✓	△	△	–	–	–
From Spark to Fire (Xie et al., 2026)	✓	△	△	–	–	–
EvidenceGraph	✓	✓	✓	✓	✓	✓

2 Related Work and Benchmark Comparison

Failure studies locate component, agent, and process defects (Chowdhury & D'Souza, 2026; Cemri et al., 2025; Zhang et al., 2025c; Fan et al., 2026; Zhang et al., 2026); injection studies test propagation (Xie et al., 2026; Jia et al., 2026; Shen et al., 2025). Table 1 compares their coverage with EvidenceGraph.

Correlated errors (Kim et al., 2025), hallucination dynamics (Jamshidi et al., 2026), and harmful debate shifts (Wynn et al., 2025) make agreement difficult to interpret. Free-MAD discourages conformity (Cui et al., 2026); Hidden Anchors models persistent beliefs (Pokharel & Dantu, 2026). Routing prominence need not reflect functional importance (Ghosh et al., 2026). Transfer entropy measures directed dependence (Schreiber, 2000), which CASPIAN uses for cascade monitoring (Venkatesh et al., 2026).

Replay methods edit traces or plans (Ma et al., 2026; Bonagiri et al., 2026; Shah, 2026). MARS searches repairs on replayable StateMAS trajectories (Lu & Zhang, 2026), while SymTrace anchors downstream replay and SymFail supplies annotated failures (Luan et al., 2026). Zero-replay debugging predicts which trace events merit a limited replay budget (Kang et al., 2026); graph-constrained joint replay recovers minimal singleton/pair repair families within a declared graph domain (Li et al., 2026). EvidenceGraph instead studies which repair sequence to assign to each system from an acquired replay bank under a shared terminal budget. Other work optimizes clarification (Lin et al., 2026), graphs (Zhuge et al., 2024; Zhang et al., 2025b), pruning (Zhang et al., 2025a; Wang et al., 2025), and routing (Lu et al., 2026); CAGE-Select uses CAGE-Cal confidence from matched communication contrasts to select topology outputs per query (Huang et al., 2026).

Task-based learning and predict-then-optimize distinguish prediction error from decision loss (Donti et al., 2017; Elmachtoub & Grigas, 2022). Our allocator comparisons hold scores fixed to examine this distinction in budgeted repair.

3 EvidenceGraph and Repair Responses

3.1 Task and construction

EvidenceGraph evaluates a four-role system comprising a source retriever, an evidence analyst, a conflict checker, and a final synthesizer. Tasks are constructed from Controlled EvidenceGraph (synthetic rule-compositional tasks), MATH-derived solution graphs, and MMLU-derived evidence capsules (Hendrycks et al., 2021b;a), as shown in Figure 2.

For each base task, we construct six systems with different failure mechanisms: peer propagation, terminal corruption, source mismatch, shared-retriever common cause, independent failure, and mixed peer/shared failure. These systems form a matched block with the source, reference decision, semantic surface, and chain/star topology held fixed.

Each adapted task asks whether evidence supports a candidate claim, rather than requesting an original MATH solution or MMLU answer. During replay, the behavior model reads the query and indexed evidence records and labels their relations to the claim, without receiving repair-

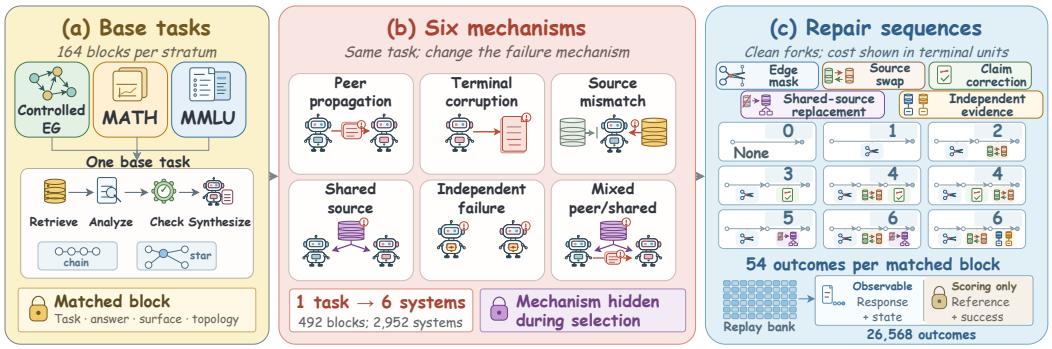

Figure 2: EvidenceGraph construction from three source strata. Matched blocks fix task, reference, surface, and topology across six mechanisms; each system runs nine clean-fork sequences. Cohort locks separate observable replay states from hidden scoring keys.

success labels. A fixed deterministic reducer aggregates these relations into the final decision $\hat{y}_i(h) \in \{\text{ACCEPT}, \text{REJECT}\}$.

Repairs modify the replay inputs: edge masking blocks a designated communication link, source swapping replaces a designated source input, and claim correction replaces an intermediate claim. Shared-source replacement changes the common upstream source, while independent evidence adds a separate evidence source. These five operations define nine ordered repair sequences $h \in \mathcal{H}$, including no intervention ($\emptyset$). Their assigned costs are 1, 1, 2, 4, and 4 units, respectively; $c(h)$ sums these costs, not tokens or runtime. All sequences replay from the same frozen snapshot, so one replay cannot modify another. Fixed-task success is $S_i(h) = \mathbf{1}\{\hat{y}_i(h) = y_i^\star\}$, where $y_i^\star$ is the hidden reference decision. The reducer and reference decision remain fixed across sequences, so a changed answer is not automatically a successful repair.

The 492 matched blocks are balanced across strata: 42 for development fitting, 18 for configuration selection, and 216 each for disjoint Primary and Fresh evaluation cohorts. Six systems per block and nine sequences per system yield 2,952 systems and 26,568 system–sequence evaluations. Blocks are the independent units for uncertainty assessment.

For repair selection, the controller observes O_i, comprising traces, topology, surface metadata, model-produced relations, and reducer summaries from acquired replays. Answers and margins are visible; correctness labels, planted mechanisms, and reference-answer scoring fields remain hidden. Scoring keys are released after predictions, selections, and the analysis plan are fixed. Primary supports selection-adjusted comparisons among fitted policies; Fresh independently evaluates the prespecified comparison. Neither supplies fitting labels or hyperparameter choices. The cohorts are never pooled.

The same model family constructed and separately reviewed all records; there was no human or independent-model review. Audits checked schema validity, balance, source overlap, near duplicates, and deterministic state generation. Across strata, the benchmark measures controlled repair responses; it does not estimate natural failure frequencies. The released numeric package supports offline policy evaluation on the acquired bank.

3.2 Repair information and achievable value

For mechanism m, the response signature $r_m = (p_m(h))_{h \in \mathcal{H}}$ collects expected terminal success under each sequence. At fixed costs, mechanisms with identical signatures are decision-equivalent, so predicting repair value need not classify mechanisms. This equivalence guarantees neither signature recovery from observations nor mechanism identification.

Successful repairs need not be selectable from observations. For the same per-system feasible set $\mathcal{H}_c = \{h : c(h) \le c\}$, define best-fixed value (one sequence for all systems), observable-adaptive Bayes value (selection using O), and full-response coverage (selection knowing all outcomes):

$$F_c = \max_{h \in \mathcal{H}_c} \mathbb{E}S(h), \quad A_c = \mathbb{E} \max_{h \in \mathcal{H}_c} \mathbb{E}[S(h) \mid O], \quad C_c = \mathbb{E} \max_{h \in \mathcal{H}_c} S(h). \tag{1}$$

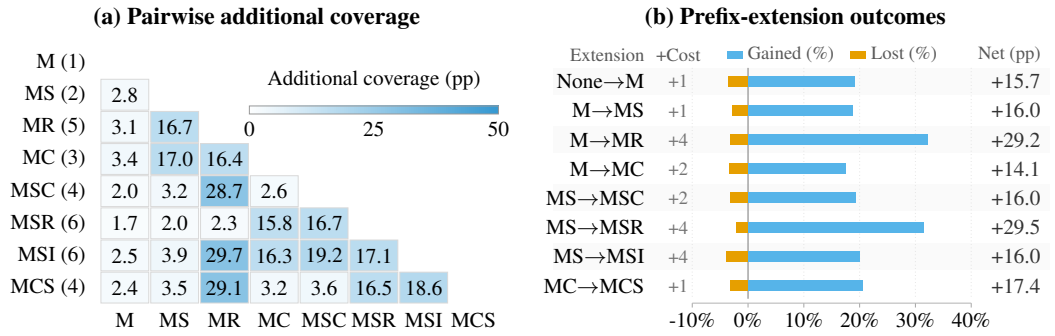

Figure 3: Complementary coverage and bidirectional outcome changes on Fresh ($N = 1{,}296$; 216 matched blocks). (a) Coverage beyond the better sequence: $100\min(n_{01}, n_{10})/N$ (pp). (b) Success gained (right) and lost (left), divided by N; net changes in pp. Parentheses: terminal costs; +Cost: added costs. M: edge mask; S: source swap; R: shared-source replacement; C: claim correction; I: independent evidence. This retrospective coverage is not deployable policy value or an equal-spend controller comparison; clean-fork losses are not causal harm estimates.

Theorem 1 (Repair-value hierarchy)**.** *$F_c \le A_c \le C_c$, with $C_c - F_c = (A_c - F_c) + (C_c - A_c)$. The two terms represent observable targeting opportunity and remaining hidden response information, respectively. Equality $F_c = A_c$ holds exactly when a best-fixed sequence is conditionally optimal almost surely; $A_c = C_c$ holds exactly when observable information supports a response-maximizing selection almost surely.*

Observations help only when they change optimal repairs. These per-system values do not give a shared-budget controller's achievable value.

In Figure 3, n_{ab} counts systems with outcomes $(S(h), S(k)) = (a, b)$. These retrospective clean-fork outcomes do not estimate A_c or identify causal harm.

For shared-budget selection, let observations and outcomes share a joint law and define $p_i(h) = \mathbb{E}[S_i(h) \mid \mathbf{O}]$, where $\mathbf{O} = (O_1, \ldots, O_N)$. Let $\mathcal{F}_B(\mathcal{H})$ contain one-sequence-per-system assignments satisfying $\sum_i c(h_i) \le B$ and the per-system cap. An observable policy $\pi(\mathbf{O}) \in \mathcal{F}_B(\mathcal{H})$ has value $V_B^O(\pi) = N^{-1}\sum_i p_i(h_i^\pi)$, with optimum $V_B^O(\mathcal{H}) = \max_\pi V_B^O(\pi)$. A full-information oracle selects knowing all candidate outcomes but is evaluated conditional on the same $\mathbf{O}$:

$$V_B^{\mathrm{FI}}(\mathcal{H}) = \mathbb{E}\left[\max_{\mathbf{h}\in\mathcal{F}_B(\mathcal{H})} N^{-1}\sum_i S_i(h_i) \,\middle|\, \mathbf{O}\right].$$

For $\mathcal{H}^+ \supseteq \mathcal{H}$ under the same constraints, define $\kappa_{\mathcal{H}} = V_B^{\mathrm{FI}}(\mathcal{H}^+) - V_B^{\mathrm{FI}}(\mathcal{H})$ and $\Delta_O = V_B^{\mathrm{FI}}(\mathcal{H}) - V_B^O(\mathcal{H})$. These nonnegative gaps separate repair coverage from information limits: expanding the feasible set or revealing outcomes before selection cannot reduce optimal expected value.

Prediction and allocation introduce two further errors. Define $\epsilon = \max_{i,h} |\widehat{p}_i(h) - p_i(h)|$ and let δ_{alloc} be the gap of a feasible allocator $\widehat{\pi}$ from the optimal mean predicted value.

Theorem 2 (Observable budgeted-control regret)**.** *Conditional on $\mathbf{O}$, almost surely,*

$$V_B^{\mathrm{FI}}(\mathcal{H}^+) - V_B^O(\widehat{\pi}) \le \kappa_{\mathcal{H}} + \Delta_O + 2\epsilon + \delta_{\mathrm{alloc}}. \tag{2}$$

Optimizing predicted value reduces allocation error but leaves prediction error ϵ unchanged, so task success need not improve. We therefore evaluate predictions against task success and evaluate allocators with scores fixed (Section 5.5). The bound gives no greedy optimality or universal approximation guarantee.

4 Observable Action-Effect Prediction

Figure 4 summarizes OAE: observed answers and evidence margins form 11 features, standardized to predict each sequence's replay agreement. OAE-Greedy retains one candidate per system by gain over no intervention per unit cost, then ranks candidates to allocate the shared budget.

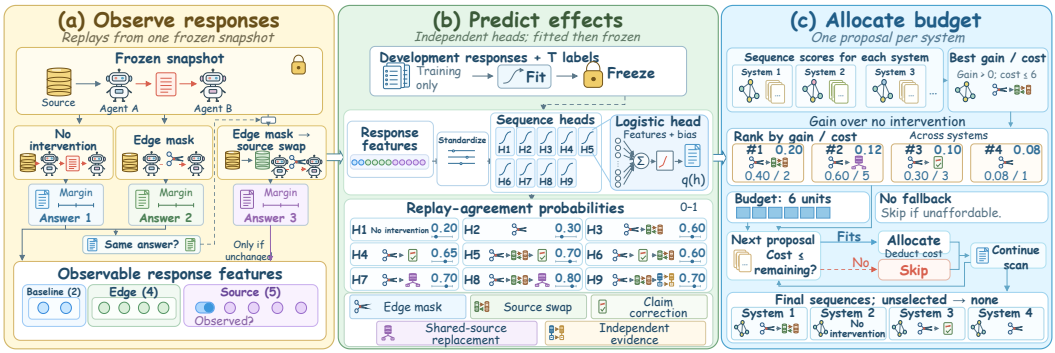

Figure 4: **OAE workflow.** (a) Clean-fork probes; source swap follows an unchanged edge-mask answer. (b) Nine logistic heads predict replay agreement. (c) One candidate per system is ranked by gain per cost; unaffordable candidates are skipped. Scores are selection surrogates, and observations are already acquired.

4.1 Observable intervention responses

Evidence may change without changing the answer, so we observe both the reducer's answer and its evidence margin: weighted support minus opposition after filtering. Each replay in Figure 4(a) starts from the same snapshot and cannot alter another. We compare the no-intervention and edge-mask answers, observing the edge-mask–source-swap result only when they agree. Figure 1 illustrates a peer failure and a shared-source failure that call for different repairs; their hidden mechanisms are not predictor inputs.

The vector x_i summarizes these observations O_i in 11 features. The baseline supplies its signed and absolute margins. The sign indicates whether support or opposition dominates, while the absolute value measures the size of the imbalance. Edge masking adds four features: both margins, the signed change from baseline, and whether the answer changed. Source swapping adds the same four measurements relative to edge masking, plus an indicator of whether the response was observed. If unobserved, its four response entries are zeroed before scaling. The observation indicator distinguishes these missing responses from observed zero margins, where weighted support and opposition are equal. The predictor receives these features without access to answer correctness.

4.2 Sequence-specific action-effect prediction

Because repair sequences can produce different outcomes on the same system, we score each sequence separately. The predictor in Figure 4(b) standardizes all 11 features, including binary indicators, with development-fit means and population standard deviations; zero-variance scales are set to one. OAE fits nine independent logistic heads, one for each $h \in \mathcal{H}$, including no intervention ($\emptyset$). All receive the standardized vector z_i, but each learns its own coefficients from its sequence's labels:

$$q_i(h) = \sigma\big(b_h + \boldsymbol{\beta}_h^\top z_i\big)\,, \qquad \widehat{p}_i(h) = \mathrm{clip}_{[0.01,0.99]}\, q_i(h). \tag{3}$$

Each sequence's expected answer is the reducer output for its reference evidence relations. OAE predicts agreement with this answer: $T_i(h) = 1$ for a match and $T_i(h) = 0$ otherwise. Task success $S_i(h)$ instead compares the actual answer with the fixed task answer $y_i^\star$. Because these references can differ, $\widehat{p}_i(h)$ is not an established calibrated task-success probability. Several sequences can match their respective expected answers on the same system, giving positive labels to multiple heads for that training example. The nine heads therefore answer separate binary questions; scores need not sum to one or increase with sequence length. They guide repair selection, which is evaluated by task success.

We estimate the scaler and fit heads with regularized binary cross-entropy on development-fit blocks, leaving intercepts unpenalized. Regularization minimizes stratum-macro Brier score against replay-agreement labels on disjoint selection blocks; ties favor lower macro log loss, lower calibration error, then stronger regularization. Heldout predictions use the fixed model without refitting, recalibration, or threshold selection.

Table 2: Fresh terminal success (%) by mechanism and sequence (retrospective). M: edge mask; S: source swap; C: claim correction; R: shared-source replacement; I: independent evidence.

Mechanism	None	M	MS	MR	MC	MSC	MSR	MSI	MCS
Independent failure	6.5	4.2	2.8	4.2	8.3	4.2	6.9	**97.2**	4.2
Mixed peer/shared	6.0	8.3	11.6	94.0	9.7	17.1	**97.2**	14.4	10.6
Peer propagation	3.7	94.4	**97.7**	95.8	93.5	97.2	**97.7**	94.0	96.8
Shared-retriever cause	7.9	5.1	10.6	94.0	6.5	11.1	**95.4**	10.6	13.9
Source mismatch	4.2	6.9	94.0	5.6	1.9	**95.4**	94.9	**95.4**	**95.4**
Terminal corruption	6.5	9.7	7.9	10.2	93.5	95.8	9.3	8.8	**97.2**

4.3 Shared-budget action selection

A high sequence score alone does not justify spending: the same system may already score highly without intervention, and repair costs differ. The allocation objective in Figure 4(c) is the largest total predicted score under shared budget B, with one repair sequence per system and a per-system cost cap of six:

$$\max_{h_i \in \mathcal{H}_6} \sum_i \widehat{p}_i(h_i) \quad \text{subject to} \quad \sum_i c(h_i) \le B, \qquad \mathcal{H}_6 = \{h \in \mathcal{H} : c(h) \le 6\}. \tag{4}$$

OAE-Greedy uses predicted gain per cost in Equation 5 to prioritize spending: subtracting the no-intervention score measures the additional predicted benefit, and dividing by cost accounts for the budget it consumes:

$$\widehat{g}_i(h) = \widehat{p}_i(h) - \widehat{p}_i(\emptyset), \qquad \widehat{\rho}_i(h) = \widehat{g}_i(h)/c(h), \quad c(h) > 0. \tag{5}$$

OAE-Greedy retains each system's highest-ratio candidate after excluding over-cap costs and non-positive gains. Ties favor larger gain, lower cost, then lexical sequence order. Cross-system ranking uses the same criteria, with system identifiers before sequences. One full scan accepts affordable candidates and deducts their cost, skipping others without substituting a second-ranked sequence. Every unselected system executes no intervention. Evaluation is offline on an already acquired shared replay bank; $B = 2{,}592$ covers terminal repairs, excluding replay-bank acquisition.

5 Experimental Evaluation

5.1 Evaluation protocol

Three disjoint cohorts serve fitting and tuning (Development), policy selection (Primary), and independent evaluation (Fresh). Using DeepSeek V4 Flash replays, we evaluate repair policies under a prespecified 2,592-unit terminal budget. PARE is the prespecified equal-spend comparator; it uses additional intervention responses to fit residual corrections to OAE scores, with the allocator unchanged.

Intervals use 50,000 stratum-balanced matched-block bootstrap draws. Each draw keeps all six systems and their sequence outcomes together within a task block. Primary's ten-policy family and Fresh's four retrospective equal-spend contrasts receive separate max-T adjustments. Ablations, budget sweeps, and mechanism slices are also retrospective.

Retrospective LLM selector adaptations (Table 3) estimate task success from evidence text using the same sequences and allocator. Their unequal-spend results lie outside the confirmatory family.

5.2 Benchmark action-response landscape

Without budget constraints, at least one sequence succeeds for 1,293/1,296 Fresh systems (99.77%), versus 867 (66.90%) for the best single sequence (Table 2; retrospective). Realizing this coverage therefore requires selecting different sequences for different systems.

5.3 OAE performance and repair targeting

On Fresh, OAE-Greedy achieved 42.44% success at a spend of 2,592 units, ahead of five equal-spend baselines (Table 3). Primary supported OAE over nine alternatives after selection adjustment (Table 4).

Table 3: Fresh success (%; 1,296 systems, 432 per stratum) under a shared 2,592-unit cap. Recovery uses 1,221 failures. Strata slice the global allocation, not original MATH/MMLU accuracy.

	Success (%)					
Policy	Overall	Controlled EG	MATH-derived	MMLU-derived	Intervention rate (%)	Error recovery (%)
No intervention	5.79	3.94	12.96	0.46	0.00	0.00
Cost-matched random	26.70	27.31	29.40	23.38	61.96	24.16
Best fixed	21.45	17.13	30.32	16.90	100.00	20.31
Budgeted verify-all	28.78	26.16	36.57	23.61	50.00	25.55
Baseline-risk priority	28.01	27.08	32.18	24.77	50.00	25.55
Feature-union model	23.53	22.45	30.79	17.36	63.81	21.21
PARE-GREEDY	34.26	30.56	39.58	32.64	87.35	32.51
Retrospective LLM selectors (adaptations)						
Mapper	21.99	21.30	25.93	18.75	43.83	18.92
Self-Refine-style selector	18.21	6.25	42.59	5.79	25.23	14.33
Reflection-memory selector	17.75	5.32	40.97	6.94	23.61	13.84
OAE-GREEDY	**42.44**	**39.58**	**42.82**	**44.91**	61.96	**40.62**

Table 4: Primary success (%); population and cap as in Table 3. Recovery uses 1,234 failures. Spend: none 0, best fixed 1,296, others 2,592.

	Success (%)					
Policy	Overall	Controlled EG	MATH-derived	MMLU-derived	Intervention rate (%)	Error recovery (%)
No intervention	4.78	1.39	12.96	0.00	0.00	0.00
Cost-matched random	28.32	27.08	34.03	23.84	87.58	26.82
Best fixed	21.45	17.59	30.09	16.67	100.00	20.34
Budgeted verify-all	29.94	28.01	35.65	26.16	50.00	27.63
Baseline-risk priority	27.93	26.16	32.18	25.46	50.00	25.36
Feature-union model	23.84	23.61	28.94	18.98	63.43	22.37
PARE-GREEDY	34.26	30.79	39.12	32.87	87.58	33.31
OAE-GREEDY	**43.06**	**39.81**	**40.97**	**48.38**	60.88	**41.57**

The prespecified contrast yielded an 8.18-point gain (95% matched-block interval: $[6.02, 10.34]$), with all three preregistered strata favoring OAE. Simultaneous intervals also favored OAE across four retrospective equal-spend contrasts. Within-block reassignment preserves repair counts and total cost but changes which system receives each sequence; the result supports repair matching ($p = 0.000020$).

5.4 OPERATING REGIMES AND REPAIR BEHAVIOR

In the retrospective budget sweep (Figure 5a), OAE trailed PARE at 324 and 1,296 units but led at 2,268 and 2,592; Primary showed the same reversal. Panel (b) shows that higher yield need not imply higher success: yield depends on selection and is not cost-normalized.

Figure 6 shows low success on independent and mixed peer/shared failures. OAE trailed budgeted verify-all on source mismatch and cost-matched random on mixed failures; these global-allocation slices cannot isolate causal effects or distinguish system-selection from repair-choice errors.

5.5 COMPONENT AND ALLOCATOR ABLATIONS

Across six retrospective ablations, OAE-Greedy led on Primary and Fresh; all Fresh simultaneous intervals excluded zero. Removing source responses reduced Fresh success by 17.36 points and sharing coefficients by 12.35 points. Ranking predicted scores per cost reduced Fresh success by 20.99 points at a spend of only 1,296 units, so this is not an equal-spend comparison.

Prediction quality. Fresh OAE achieved system-macro AUROC 0.533 and top-3 coverage 85.49% over eight nonempty sequences; task-success expected calibration error (ECE) across all nine sequences was 0.428. Top-3 coverage records whether at least one of the three highest-ranked sequences succeeds; budgeted allocation must still choose and pay for a single repair.

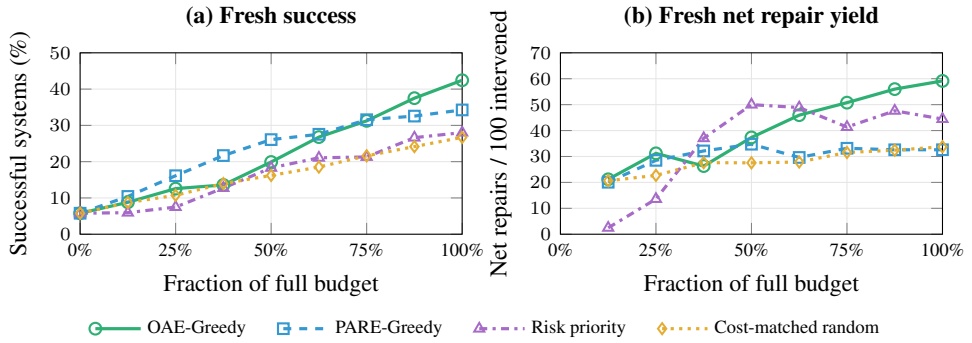

Figure 5: Fresh budget sensitivity (retrospective; replay acquired). The axis shows cap/2,592, not realized spend. (a) Success over 1,296 systems. (b) $100(\text{fixed} - \text{harmed})/\text{intervened}$: baseline failures recovered minus successes lost per 100 interventions; undefined at zero budget.

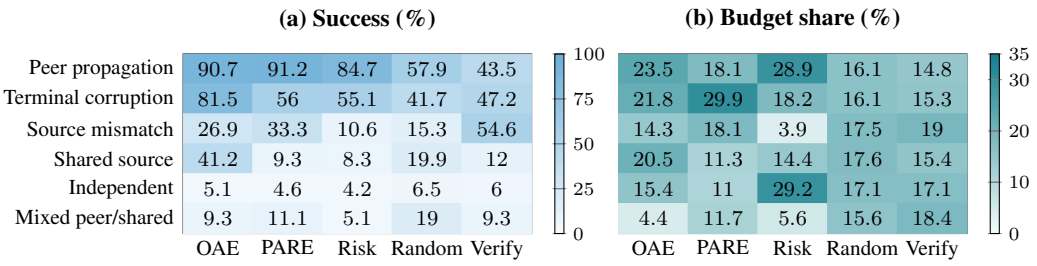

Figure 6: Fresh retrospective allocation slices. (a) Success for 216 systems per mechanism (blue: 0–100%). (b) Share of global 2,592-unit spend (teal: 0–35%); each column sums to 100%. Panel scales differ; these slices do not identify causal effects.

Table 5: Frozen-score allocators at equal 2,592-unit spend. Success (%) by stratum (432 systems each) and overall (1,296); bold marks maxima. Descriptive slices, no within-stratum reallocation; not original MATH/MMLU test accuracy. LocalSearch allows at most 128 exchanges.

	Primary				Fresh			
Allocator	Controlled	MATH	MMLU	Overall	Controlled	MATH	MMLU	Overall
OAE-LocalSearch	34.26	37.27	37.27	36.27	34.03	39.81	35.42	36.42
OAE-Lagrangian	35.65	34.26	36.11	35.34	34.72	35.65	31.48	33.95
OAE-Exact	34.26	36.57	35.42	35.42	34.26	35.65	31.71	33.87
OAE-Greedy	**39.81**	**40.97**	**48.38**	**43.06**	**39.58**	**42.82**	**44.91**	**42.44**

Allocation quality. With scores and constraints frozen, we test whether better optimization of predicted gains improves realized repairs. Exact predicted-gain allocation lost 8.56 task-success points to Greedy on Fresh at equal spend (Table 5). LocalSearch refines Greedy through feasible replacements; Lagrangian balances predicted gains against costs using a common penalty across systems. Both also trailed in descriptive comparisons by stratum. For each allocator, the stratum results report subsets of its global decisions under the shared budget.

6 Conclusion

We study observation-based repair selection under a shared budget. EvidenceGraph supports controlled repair comparisons, while our theory separates repair coverage and information limits from prediction and allocation errors. In this setting, OAE predicts replay agreement from intervention responses to prioritize repairs by predicted gain per cost. The scope is offline control on an already acquired shared replay bank.

AI USE STATEMENT

Generative AI assisted language polishing, drafting of manuscript sections, generation of synthetic datasets for EvidenceGraph, and development of proofs for mathematical claims. Dataset assistance included generating synthetic task and evidence records from MATH and MMLU source material. The benchmark's constructor and separate reviewer used the same model family; Section 3 describes those reviews and automated data checks. The authors take responsibility for the final manuscript, mathematical arguments, and data, including AI-assisted content.

REPRODUCIBILITY STATEMENT

Our code and data are available at https://anonymous.4open.science/r/EvidenceGraph-9EC4/. The anonymous OAE implementation and the numeric dataset provided in the repository support development-only fitting, frozen prediction, budget allocation, and offline evaluation on the released replay bank. The repository provides dependencies, runnable tests, reproduction commands, and expected results. Sections 4 and 5 describe the method and evaluation protocol; the supplementary document provides construction details, model specifications, and theoretical proofs.

REFERENCES

Akash Bonagiri, Devang Borkar, Gerard Janno Anderias, Setareh Rafatirad, and Houman Homayoun. CausalFlow: Causal attribution and counterfactual repair for LLM agent failures. *arXiv preprint arXiv:2605.25338*, 2026. URL https://arxiv.org/abs/2605.25338.

Mert Cemri, Melissa Z. Pan, Shuyi Yang, Lakshya A. Agrawal, Bhavya Chopra, Rishabh Tiwari, Kurt Keutzer, Aditya Parameswaran, Dan Klein, Kannan Ramchandran, Matei A. Zaharia, Joseph E. Gonzalez, and Ion Stoica. Why do multi-agent LLM systems fail? In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-4082. URL https://papers.nips.cc/paper_files/paper/2025/hash/b1041e52d3be19f0a9bc491657488e4a-Abstract-Datasets_and_Benchmarks_Track.html. Datasets and Benchmarks Track.

Sayak Chowdhury and Meenakshi D'Souza. SETA: Statistical fault attribution for compound AI systems. In *Proceedings of the IEEE/ACM 5th International Conference on AI Engineering – Software Engineering for AI*, pp. 54–61, 2026. doi: 10.1145/3793653.3793764. URL https://doi.org/10.1145/3793653.3793764.

Yu Cui, Hang Fu, Haibin Zhang, Licheng Wang, and Cong Zuo. Free-MAD: Consensus-free multi-agent debate. In *Findings of the Association for Computational Linguistics: ACL*, pp. 31977–31997, 2026. doi: 10.18653/v1/2026.findings-acl.1600. URL https://aclanthology.org/2026.findings-acl.1600/.

Mehul Damani, Idan Shenfeld, Andi Peng, Andreea Bobu, and Jacob Andreas. Learning how hard to think: Input-adaptive allocation of LM computation. In *International Conference on Learning Representations*, pp. 102783–102802, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/ff414825df833edb8b1839e3d5d495e9-Paper-Conference.pdf.

Priya L. Donti, Brandon Amos, and J. Zico Kolter. Task-based end-to-end model learning in stochastic optimization. In *Advances in Neural Information Processing Systems*, volume 30, 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/hash/3fc2c60b5782f641f76bcefc39fb2392-Abstract.html.

Adam N. Elmachtoub and Paul Grigas. Smart "predict, then optimize". *Management Science*, 68(1): 9–26, 2022. doi: 10.1287/mnsc.2020.3922. URL https://pubsonline.informs.org/doi/10.1287/mnsc.2020.3922.

Shengda Fan, Xuyan Ye, Yupeng Huo, Zhi-Yuan Chen, Yiju Guo, Shenzhi Yang, Wenkai Yang, Shuqi Ye, Jingwen Chen, Haotian Chen, Xin Cong, and Yankai Lin. AgentProcessBench: Diagnosing step-level process quality in tool-using agents. In *Proceedings of the 32nd ACM*

SIGKDD Conference on Knowledge Discovery and Data Mining V.2, pp. 8823–8834, 2026. doi: 10.1145/3770855.3817494. URL https://doi.org/10.1145/3770855.3817494.

Sudipto Ghosh, Sujoy Nath, Sunny Manchanda, and Tanmoy Chakraborty. Disentangling intrinsic importance from emergent structure in multi-expert orchestration. *Transactions on Machine Learning Research*, 2026. URL https://openreview.net/forum?id=4W7sgat04A.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *International Conference on Learning Representations*, 2021a. URL https://openreview.net/forum?id=d7KBjmI3GmQ.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the MATH dataset. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1, 2021b. URL https://datasets-benchmarks-proceedings.neurips.cc/paper_files/paper/2021/hash/be83ab3ecd0db773eb2dc1b0a17836a1-Abstract-round2.html.

Jiatan Huang, Mingchen Li, Ziming Li, Sunjae Kwon, Hong Yu, and Chuxu Zhang. Counterfactual graph for multi-agent LLM calibration. *arXiv preprint arXiv:2605.30653*, 2026. URL https://arxiv.org/abs/2605.30653.

Saeid Jamshidi, Arghavan Moradi Dakhel, Kawser Wazed Nafi, and Foutse Khomh. Hallucination cascade: Analyzing error propagation in multi-agent LLM systems. *arXiv preprint arXiv:2606.07937*, 2026. URL https://arxiv.org/abs/2606.07937.

Jin Jia, Zhiling Deng, Zhuangbin Chen, Yingqi Wang, and Zibin Zheng. MAS-FIRE: Fault injection and reliability evaluation for LLM-based multi-agent systems. *arXiv preprint arXiv:2602.19843*, 2026. URL https://arxiv.org/abs/2602.19843.

Dong Ho Kang, Hyeonjeong Cha, and Daein Weon. Knowledge-based zero-replay debugging of multi-agent LLM traces. *arXiv preprint arXiv:2606.14805*, 2026. URL https://arxiv.org/abs/2606.14805.

Elliot Myunghoon Kim, Avi Garg, Kenny Peng, and Nikhil Garg. Correlated errors in large language models. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 30038–30066, 2025. URL https://proceedings.mlr.press/v267/kim25e.html.

Bingjie Li, Yumeng Song, Zhongming Yao, and Tianyi Li. Localizing emergent failures in agentic AI: Recovering minimal repair families via counterfactual replay. *arXiv preprint arXiv:2608.29228*, 2026. URL https://arxiv.org/abs/2608.29228.

Bohan Lin, Kuo Yang, Zelin Tan, Yingchuan Lai, Chen Zhang, Guibin Zhang, Xinlei Yu, Miao Yu, Xu Wang, Yudong Zhang, and Yang Wang. AgentAsk: Multi-agent systems need to ask. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 28055–28077, 2026. doi: 10.18653/v1/2026.acl-long.1294. URL https://aclanthology.org/2026.acl-long.1294/.

Hanxiao Lu and Tianyi Zhang. Autonomous repair for multi-agent systems via Monte-Carlo tree search. *arXiv preprint arXiv:2607.29055*, 2026. URL https://arxiv.org/abs/2607.29055.

Yuxing Lu, Yucheng Hu, Xukai Zhao, and Jiuxin Cao. DyTopo: Dynamic topology routing for multi-agent reasoning via semantic matching. *arXiv preprint arXiv:2602.06039*, 2026. URL https://arxiv.org/abs/2602.06039.

Zhongwen Luan, Xiaoyu Zhang, Ming Hu, Yue Yang, Jiongchi Yu, and Xiaohong Chen. Repair or resample? Rethinking failure debugging in LLM multi-agent systems. *arXiv preprint arXiv:2608.25920*, 2026. URL https://arxiv.org/abs/2608.25920.

Ming Ma, Jue Zhang, Fangkai Yang, Yu Kang, Qingwei Lin, Saravan Rajmohan, and Dongmei Zhang. DoVer: Intervention-driven auto debugging for LLM multi-agent systems. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=mrEK16Jy6h.

Judea Pearl. *Causality: Models, Reasoning, and Inference*. Cambridge University Press, second edition, 2009. ISBN 978-0-521-89560-6. URL https://assets.cambridge.org/97805218/95606/frontmatter/9780521895606_frontmatter.pdf.

Apurba Pokharel and Ram Dantu. Hidden anchors in multi-agent LLM deliberation. *arXiv preprint arXiv:2606.19494*, 2026. URL https://arxiv.org/abs/2606.19494.

Thomas Schreiber. Measuring information transfer. *Physical Review Letters*, 85(2):461–464, 2000. doi: 10.1103/physrevlett.85.461. URL https://journals.aps.org/prl/abstract/10.1103/PhysRevLett.85.461.

Jaineet Shah. Causal agent replay: Counterfactual attribution for LLM-agent failures. *arXiv preprint arXiv:2606.08275*, 2026. URL https://arxiv.org/abs/2606.08275.

Xu Shen, Yixin Liu, Yiwei Dai, Yili Wang, Rui Miao, Yue Tan, Shirui Pan, and Xin Wang. Understanding the information propagation effects of communication topologies in LLM-based multi-agent systems. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 12347–12361, 2025. doi: 10.18653/v1/2025.emnlp-main.623. URL https://aclanthology.org/2025.emnlp-main.623/.

Kavana Venkatesh, Jafar Isbarov, Saad Amin, Murat Kantarcioglu, and Jiaming Cui. CASPIAN: Online detection and attribution of cascade attacks in LLM multi-agent systems via cross-channel causal monitoring. *arXiv preprint arXiv:2605.19240*, 2026. URL https://arxiv.org/abs/2605.19240.

Zhexuan Wang, Yutong Wang, Xuebo Liu, Liang Ding, Miao Zhang, Jie Liu, and Min Zhang. AgentDropout: Dynamic agent elimination for token-efficient and high-performance LLM-based multi-agent collaboration. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 24013–24035, 2025. doi: 10.18653/v1/2025.acl-long.1170. URL https://aclanthology.org/2025.acl-long.1170/.

Andrea Wynn, Harsh Satija, and Gillian Hadfield. Talk isn't always cheap: Understanding failure modes in multi-agent debate. *arXiv preprint arXiv:2509.05396*, 2025. URL https://arxiv.org/abs/2509.05396.

Yizhe Xie, Congcong Zhu, Xinyue Zhang, Tianqing Zhu, Dayong Ye, Minfeng Qi, Huajie Chen, and Wanlei Zhou. From spark to fire: Modeling and mitigating error cascades in LLM-based multi-agent collaboration. *arXiv preprint arXiv:2603.04474*, 2026. URL https://arxiv.org/abs/2603.04474.

Guibin Zhang, Yanwei Yue, Zhixun Li, Sukwon Yun, Guancheng Wan, Kun Wang, Dawei Cheng, Jeffrey Xu Yu, and Tianlong Chen. Cut the crap: An economical communication pipeline for LLM-based multi-agent systems. In *International Conference on Learning Representations*, 2025a. URL https://openreview.net/forum?id=LkzuPorQ5L.

Guibin Zhang, Yanwei Yue, Xiangguo Sun, Guancheng Wan, Miao Yu, Junfeng Fang, Kun Wang, Tianlong Chen, and Dawei Cheng. G-Designer: Architecting multi-agent communication topologies via graph neural networks. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 76678–76692, 2025b. URL https://proceedings.mlr.press/v267/zhang25cu.html.

Guibin Zhang, Junhao Wang, Junjie Chen, Wangchunshu Zhou, Kun Wang, and Shuicheng Yan. AgenTracer: Who is inducing failure in the LLM agentic systems? In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=l05DseqvuD.

Shaokun Zhang, Ming Yin, Jieyu Zhang, Jiale Liu, Zhiguang Han, Jingyang Zhang, Beibin Li, Chi Wang, Huazheng Wang, Yiran Chen, and Qingyun Wu. Which agent causes task failures and when? On automated failure attribution of LLM multi-agent systems. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 76583–76599, 2025c. URL https://proceedings.mlr.press/v267/zhang25cq.html.

Mingchen Zhuge, Wenyi Wang, Louis Kirsch, Francesco Faccio, Dmitrii Khizbullin, and Jürgen Schmidhuber. GPTSwarm: Language agents as optimizable graphs. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 62743–62767, 2024. URL https://proceedings.mlr.press/v235/zhuge24a.html.