Correcting Noise-Mispecified Operator Selection in Wild Compound LLM Agents

Anonymous Authors[1]

Abstract

Compound LLM agents deployed in the wild compose reasoning, coding-style, refinement, stopping, and tool-like operators. Their reliability depends not only on whether each operator works in isolation, but also on whether the controller can correctly assign credit when one operator is flaky. We identify a controller-level reliability failure mode. Under heterogeneous operator noise, including decoding variance, parsing brittleness, sandbox or executor failure, and backend drift, aggregate reward-gradient updates propagate the failure signal of one unreliable operator into every co-selected operator, biasing future routing decisions even when each operator is individually fine. We formalize this setting as *Partial-Observable Operator Selection (POOS)* and show that score-function controllers propagate the full coalition-noise budget into every coordinate update. The correction is not a larger search space but a different learning statistic: paired marginal differences cancel cross-operator noise, and pooled marginals attenuate it by $1/(4K)$. POISE instantiates this principle with REINFORCE exploration, pooled marginal attribution, and singleton estimates fused by inverse standard error. Across real replicate-noise evaluations, controlled stress tests, reliability stress tests for wild agent operators (Programmer timeout, EarlyStop misfire, dual flaky modules), and ablations, POISE improves in the predicted asymmetric, non-singleton regimes (e.g., $+8.7\%$ real-noise mean improvement, $+27\%$ on the heteroskedastic reasoning subset, and $+10$–17% on flaky-operator failure channels) and ties or loses when self-noise, homogeneous noise, or singleton optima dominate. The result is a regime-level reliability correction for wild compound LLM agents, not a universal robustness claim.

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review at the Second Workshop on Agents in the Wild: Safety, Security, and Beyond (AIWILD) at ICML 2026. Do not distribute.

1. Introduction

Compound LLM agents increasingly solve tasks by composing specialized operators such as chain-of-thought reasoning (Wei et al., 2022), code execution, tool use (Schick et al., 2023), self-refinement (Madaan et al., 2024), early stopping, and ensemble selection. A useful way to view these systems is as an *agentic supernet*: a controller samples a subnetwork of operators for each query, observes the coalition's end-to-end utility, and updates a distribution over future coalitions. This abstraction underlies many workflow and agent-architecture optimizers because it turns agent design into a learned selection problem.

The standard supernet abstraction is statistically symmetric. It treats the reward observed for a coalition as a clean enough signal for updating every operator that appeared in that coalition. That assumption is convenient in benchmark settings, but it is fragile in deployment-like settings where operators expose different reliability channels. A code executor may suffer timeout and sandbox failures, a reasoning operator may vary with decoding stochasticity, a long-context refinement module may be sensitive to parsing, and an external tool may experience rate limits or backend drift. The controller does not observe these channels separately; it observes only aggregate workflow utility. Thus one unreliable operator can perturb the learning signal assigned to every other operator that happened to co-occur with it.

We call the resulting object an *asymmetric agentic supernet*. Its formal core is *Partial-Observable Operator Selection (POOS)*: a controller must select operators whose utilities are observed only through noisy coalition feedback, while the noise profile differs across operators. Under POOS, the problem is not merely that reward is noisy. The problem is that aggregate reward-gradient learning reuses the same noisy scalar to update every inclusion coordinate. We show that score-function controllers therefore propagate the full coalition-noise budget into each coordinate update. The variance of an unreliable operator can enter the update for an unrelated operator whenever they are sampled together,

producing a noise-mispecified credit-assignment signal.

The correction is to change the statistic used for controller learning. Marginal differences compare coalitions that differ only in operator i, and therefore isolate the part of the utility change attributable to i under the observed deployment noise. This class includes Shapley-style (Shapley, 1953), Banzhaf-style, and permutation-based marginal estimators, but the key property is not the Shapley value itself. The key property is the variance geometry: cross-operator noise cancels under paired common randomness, attenuates under pooled observations, and persists under independent calls.

This paper asks: *when does correcting the feedback statistic, rather than expanding the agent search space, improve learned operator selection?* We instantiate the correction in POISE, a minimal controller that retains REINFORCE-style exploration but fuses pooled marginal attribution and singleton bandit estimates using inverse standard error. The theory predicts a specific regime: POISE should help when cross-operator heteroskedasticity is substantial and the optimal coalition is not near-singleton; it should tie or lose when noise is homogeneous, self-noise dominates, or a simple per-arm controller can identify a singleton optimum faster.

The experiments are designed as a regime map for asymmetric supernet training, not as a universal workflow-optimizer leaderboard. We first audit repeated live LLM calls to show that operator output noise is heterogeneous. We then evaluate controllers in a real replicate-noise environment built from repeated live calls, controlled Gaussian stress tests with known variance profiles, deployment-style failure channels, and ablations. The empirical pattern matches the theory: POISE improves in the predicted heteroskedastic, non-singleton regimes and ties or loses in the predicted negative regimes.

Contributions.

- **Asymmetric agentic supernets in the wild.** A statistical failure mode of agentic supernet training that surfaces under deployment: each operator has its own observation channel, but the controller sees only noisy coalition utility. We formalize this regime as Partial-Observable Operator Selection (POOS).
- **Noise mixing in reward-gradient learning.** For any score-function controller trained from aggregate coalition reward, the update on operator i inherits the full $\sum_{j\in S}\sigma_j^2$ from its co-occurring siblings.
- **Coordinate-local marginal variance bounds.** Weighted marginal differences have a different noise geometry: cross-operator noise cancels under paired common randomness, attenuates by $1/(4K)$ under pooled observations, and persists under independent calls (Theorems 3.1 and 3.3, Remark 3.2).

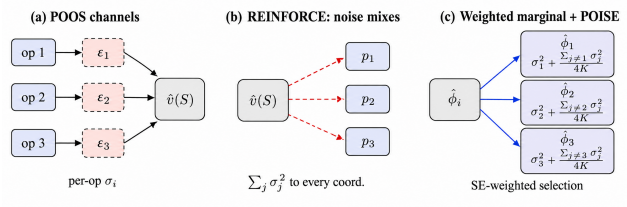

Figure 1. **REINFORCE mixes the full reward-noise budget into every coordinate; marginal differences isolate operator i's uncertainty to its own coordinate, attenuating cross-operator noise by $1/(4K)$ under pooled observations (Remark 3.2), or cancelling it exactly under paired common-randomness (Theorem 3.1).**

- **POISE.** Keeping the operator pool fixed, we replace REINFORCE's scalar reward signal with a pooled marginal estimator and a singleton bandit, fused by inverse standard error. The change is a controller-level statistic, not a new search space.
- **A regime map, not a leaderboard.** POISE improves $+8.7\%$ on real replicate noise (5/6 cells) and $+27\%$ on heteroskedastic reasoning. It loses 34 to 52% when self-noise dominates, 11% on the near-singleton DeepSeek/MATH cell, and 23% under adversarial sign-flip channels. Each loss is predicted by the variance bound.

2. The POOS Game: Asymmetric Operator Selection

Let $\mathcal{N} = \{1, \ldots, n\}$ be a pool of LLM operators. A coalition $S \subseteq \mathcal{N}$ produces utility $v(S)$ combining accuracy and cost, but the controller observes only $\hat{v}(S) = v(S) + \sum_{i\in S}\varepsilon_i$, where operator i has channel noise $\varepsilon_i \sim \mathcal{N}(0, \sigma_i^2)$. The variance profile $(\sigma_1^2, \ldots, \sigma_n^2)$ is heterogeneous. We call this the *Partial-Observable Operator Selection (POOS)* game. POOS is the local statistical model of an asymmetric agentic supernet: the controller can sample operator subnetworks and observe their utility, but it cannot directly observe which operator's stochastic channel caused a good or bad outcome. We summarize the degree of asymmetry by $\kappa = \sigma_{\max}/\sigma_{\min}$ when all channels have nonzero variance, and use empirical analogues $\hat{\kappa}$ in real replicate audits.

Definition 2.1 (POOS controller)**.** A POOS controller maps observed utilities $\hat{v}(S)$ on K sampled coalitions to an operator-inclusion distribution $p \in [0, 1]^n$.

Three observation protocols matter because they induce different variance geometry. **Independent-call**: each $\hat{v}(S)$ draws fresh noise, the default for repeated live API calls. **Paired common-randomness (CRN)**: the same ε_j is reused for the marginal pair $\hat{v}(S\cup\{i\}) - \hat{v}(S)$, enabling exact cross-noise cancellation. **Pooled**: each S is evaluated inde-

pendently, but the same K coalitions serve all marginals, attenuating cross-noise without requiring exact common randomness.

These protocols correspond to different sources of deployment uncertainty. *Persistent operator-channel variability* is a stable random effect for an operator–deployment pair; paired CRN isolates this component. *Call-level stochasticity* arises from fresh decoding randomness in repeated API calls; the independent-call protocol captures this source. *Infrastructure events*, such as rate limits, timeouts, and backend-fingerprint drift, are sporadic and logged separately. The real replicate-noise experiment in Section 5.2 uses independent live outputs and reports the empirical variance profile rather than assuming Gaussianity.

3. Theory: Coordinate Contamination under POOS

POOS is a credit-assignment problem because aggregate reward-gradient controllers reuse the same noisy scalar for every inclusion coordinate. Thus operator j's variance can enter the update intended to learn operator i's contribution.

Weighted marginal attributions $\phi_i^q(v) = \mathbb{E}_{S\sim q_i}[v(S \cup \{i\}) - v(S)]$ generalize the Shapley value (kernel $q_i(S) = |S|!\,(n-|S|-1)!\,/n!$) and Banzhaf index (uniform q_i). The key structural property holds for any q_i: marginal differences isolate operator i's noise channel from cross-operator interference. Let $\hat{\phi}_i = \frac{1}{K}\sum_{k=1}^{K}[\,\hat{v}(S^{(k)} \cup \{i\}) - \hat{v}(S^{(k)} \setminus \{i\})\,]$ with $S^{(k)} \overset{\text{ind}}{\sim} \text{Bernoulli}(1/2)^n$, and assume $|v(S)| \le M$.

Theorem 3.1 (Paired weighted marginals cancel cross-operator noise)**.** *Under the POOS noise model with $\varepsilon_j \sim \mathcal{N}(0, \sigma_j^2)$ sampled once per operator, for every $i \in \mathcal{N}$, $\mathrm{Var}[\hat{\phi}_i - \phi_i^q] = \sigma_i^2 + \mathrm{Var}_{S\sim q_i}[v(S\cup\{i\}) - v(S)]/K$. Cross-operator noise from every $j \neq i$ vanishes.*

Proof sketch. For each $S^{(k)}$ and $j \neq i$, ε_j appears in both terms of the marginal with identical coefficient $\mathbb{1}\{j \in S^{(k)}\}$ and cancels. Only ε_i survives with coefficient one, contributing σ_i^2. Full proof in Appendix A. □

Remark 3.2 (Pooled-MC variant). The pooled estimator $\tilde{\phi}_i = \mathrm{mean}\{\hat{v}(S^{(k)}) : i \in S^{(k)}\} - \mathrm{mean}\{\hat{v}(S^{(k)}) : i \notin S^{(k)}\}$ pools utilities before differencing. Under uniform Bernoulli sampling, $\mathrm{Var}[\tilde{\phi}_i - \phi_i] \le \sigma_i^2 + \frac{1}{4K}\sum_{j\neq i}\sigma_j^2 + \frac{4M^2}{K}$. Cross-noise persists but is attenuated by $1/(4K)$.

Theorem 3.3 (Independent-call observations — no cancellation)**.** *Under independent-call observations with fresh noise per evaluation, $\mathrm{Var}[\hat{\phi}_i - \phi_i] = 2\sigma_i^2 + \frac{1}{4K}\sum_{j\in\mathcal{N}}\sigma_j^2 + O(M^2/K)$. Cross-operator noise does not vanish at any finite K.*

Proposition 3.4 (Reward-gradient noise mixing)**.** *For a Bernoulli policy with inclusion probabilities p_i and score-function estimator $g_i = (S_i - p_i)(\hat{v}(S) - b)/(p_i(1-p_i))$, under POOS noise $\mathrm{Var}_\varepsilon[g_i \mid S] = \frac{(S_i-p_i)^2}{p_i^2(1-p_i)^2}\sum_{j\in S}\sigma_j^2$. For uniform Bernoulli $p_j = 1/2$, $\mathbb{E}_S\,\mathrm{Var}_\varepsilon[g_i \mid S] = 2\sum_{j=1}^{n}\sigma_j^2$. The contrast with Theorem 3.1 is the key point.*

The comparison is statistical rather than architectural. REINFORCE asks whether the realized coalition did well; marginal attribution asks whether adding operator i changed utility relative to a matched or pooled baseline. The latter removes or attenuates noise unrelated to i, but only when cross-operator heteroskedasticity and marginal evidence are present.

From variance to regret. The variance bound translates directly into an end-to-end regret bound for the softmax-temperature-τ selection policy $\hat{p}_i = \mathrm{softmax}_\tau(\hat{\phi}_i)$.

Theorem 3.5 (End-to-end regret)**.** *Assume utility v is L-Lipschitz in p. Under Theorem 3.1's noise model,*

$$\mathbb{E}[v(p^*) - v(\hat{p})] \le \frac{Ln}{\tau}\sqrt{\sigma^2 + \frac{n-1}{4K}\sigma^2 + 4M^2/K} = \mathcal{O}\big(\tfrac{n\sigma}{\tau}\big),$$

$\sigma^2 = \max_i \sigma_i^2$. The first term is the irreducible self-noise floor; K-dependent terms are cross-noise and Monte-Carlo residuals. Sample complexity for ε excess regret: $K \ge \frac{L^2 n^2}{\tau^2\varepsilon^2}\big(\frac{n-1}{4}\sigma^2 + 4M^2\big)$.

The bound makes the regime claim sharp: when $\sigma_{\max}$ is small or $\hat{\kappa} \approx 1$, the cross-noise term and the corresponding advantage shrink; when $\hat{\kappa}$ is large, the marginal estimator's $1/(4K)$ attenuation is the operative factor that buys lower regret than reward-gradient alternatives.

Table 1. Variance decomposition by observation protocol.

Protocol	Self-noise	Cross-noise	Decays with K?
Paired-CRN (Thm. 3.1)	σ_i^2	0	— (exact)
Pooled (Remark 3.2)	σ_i^2	$\frac{1}{4K}\sum_{j\neq i}\sigma_j^2$	Yes
Independent-call (Thm. 3.3)	$2\sigma_i^2$	$\frac{1}{4K}\sum_j\sigma_j^2$	Yes

4. Method: POISE for Asymmetric Supernets

POISE is intentionally minimal: it is not a new agent-design space and does not add operators to the supernet. It is a controller-level replacement of the statistic used to update operator-selection preferences. Each round, POISE samples $K/2$ coalitions via REINFORCE-guided Bernoulli exploration and $K/2$ coalitions uniformly at random for marginal coverage. This split keeps the exploration behavior of standard score-function controllers while ensuring that the batch contains enough coalition diversity to estimate marginal differences.

Algorithm 1 POISE for POOS (one round)

Require: operator pool $\mathcal{N}$, budget K, learning rate η, α-clip $(\alpha_{\text{lo}}, \alpha_{\text{hi}})$

Initialize $\theta_i \leftarrow 0$, $\mu_i \leftarrow 0$, $n_i \leftarrow 0$, $\hat{\phi}_i \leftarrow 0$, $m_i \leftarrow 0$
for $t = 1$ to T **do**
  $S_{\text{rf}} \leftarrow \{S \mid S_i \sim \text{Bern}(\sigma(\theta_i))\}$ $\{K/2$ coalitions$\}$
  $S_{\text{perm}} \leftarrow \{$random subsets$\}$ $\{K/2$ coalitions$\}$
  Observe $\hat{v}(S)$ for $S \in S_{\text{rf}} \cup S_{\text{perm}}$
  $a_k \leftarrow \hat{v}(S^{(k)}) - \text{EMA}(\hat{v})$
  $\theta \leftarrow \theta + \frac{\eta}{K} \sum_k a_k(\mathbb{1}_{S^{(k)}} - \sigma(\theta))$
  Update μ_i, n_i from singleton S; update $\hat{\phi}_i, m_i$ from paired marginals in batch
  $\text{SE}_\phi \leftarrow 1/\sqrt{m_i}$, $\text{SE}_\mu \leftarrow 1/\sqrt{n_i}$
  $\alpha_t \leftarrow \text{clip}(\text{SE}_\phi^{-1}/(\text{SE}_\phi^{-1} + \text{SE}_\mu^{-1}), \alpha_{\text{lo}}, \alpha_{\text{hi}})$
  $p_t \leftarrow \alpha_t \cdot \text{zscore}(\hat{\phi}) + (1 - \alpha_t) \cdot \text{zscore}(\mu)$
end for
return p_T

The same K observations update three signals. First, **REINFORCE** updates inclusion logits θ_i with an EMA baseline, which preserves exploration and lets the controller adapt online. Second, **pooled marginal attribution** estimates $\hat{\phi}_i$ from observed coalitions that differ in operator i, providing coordinate-local evidence whose cross-noise is attenuated by the pooled bound. Third, a **singleton bandit** estimate μ_i tracks the utility of individual operators, giving a fast low-cost signal when the optimal coalition is close to a singleton. POISE fuses the marginal and singleton signals via inverse-standard-error weighting, $\alpha_t = \text{SE}_\phi^{-1}/(\text{SE}_\phi^{-1} + \text{SE}_\mu^{-1})$, clipped to $[0.05, 0.95]$. The selection score is $\alpha_t \cdot \text{zscore}(\hat{\phi}) + (1 - \alpha_t) \cdot \text{zscore}(\mu)$. The ablation in Section 5.6 tests whether the gains come from marginal attribution rather than from SE weighting alone. Algorithm 2 gives the full pseudocode; defaults are $K = 50$, $T = 30$, $\eta = 0.20$, $(\alpha_{\text{lo}}, \alpha_{\text{hi}}) = (0.05, 0.95)$.

5. Experiments

The experiments test the regime prediction of the asymmetric supernet, not universal workflow-optimizer superiority. Five questions structure the evaluation: whether real operators are heteroskedastic, whether reward-gradient updates contaminate co-occurring operators while marginal statistics attenuate the contamination, whether correcting the feedback statistic improves selection in heteroskedastic non-singleton regimes, whether the correction ties or loses in homogeneous, self-noise, or near-singleton regimes, and whether marginal attribution rather than SE-weighting is the operative component.

Roadmap. The experiments verify three claims. First, real operator channels are heterogeneous (Section 5.2). Second, reward-gradient controllers mix cross-noise while marginal statistics attenuate it (Sections 5.1 and 5.3). Third, the correction helps in the predicted reliability regimes and fails in the documented negative regimes (Sections 5.4–5.6).

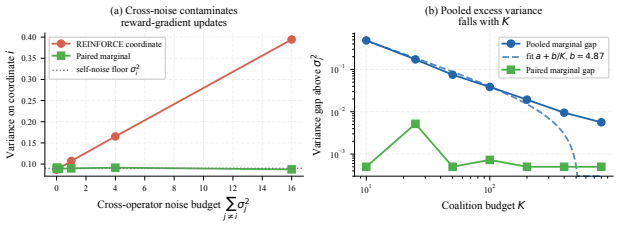

Figure 2. **Mechanism diagnostics for POOS.** (a) Reward-gradient updates mix cross-operator noise into coordinate i, whereas paired marginal attribution remains at the self-noise floor. (b) Under the implemented pooled statistic, excess variance above σ_i^2 decreases with the coalition budget K; paired marginals remain near zero excess variance. These diagnostics use synthetic channels only to isolate the variance geometry, not to claim deployment performance.

Real vs. controlled evidence. Table 2 states what each experimental block measures and what it does *not* claim, to keep wild-deployment claims honest.

Table 2. **What each experimental block does and does not claim.**

Evidence	Supports	Does *not* claim
Replicate-noise audit (live calls)	Output-variance heterogeneity in deployment	Production outage rates
F1/F2/F3 stress channels (injected)	Controller response to flaky channels	Real frequency of tool failure
Gaussian/heteroskedastic grid (synthetic)	Variance geometry, regime map	Deployment performance
Heavy-tailed/adversarial (synthetic)	Falsification boundary of the fix	Robust adversarial defense

Setup. We evaluate 8 benchmarks, 5 primary LLMs, and a 7-operator pool. Controlled experiments inject per-operator Gaussian noise $\sigma \in \{0, 0.1, 0.2, 0.3, 0.5, 0.7, 0.9\}$ with 25 paired seeds. Utility is $v(S) = \max_r \text{acc}(S, r) - \lambda\, \text{cost}(S)/\text{cost}_{\text{norm}}$. Appendix D gives datasets, models, prompts, and statistics.

5.1. Mechanism diagnostics

Mechanism diagnostics isolate the variance geometry before end-to-end regret. With $\sigma_i = 0.3$ and cross-noise budget $\sum_{j \neq i} \sigma_j^2 \in [0, 16]$, Figure 2(a) shows that REINFORCE coordinate variance grows from 0.087 to 0.394, while paired marginals stay near the self-noise floor $\sigma_i^2 = 0.09$. With fixed cross-noise budget 4.0 and $K \in \{10, 25, 50, 100, 200, 400, 800\}$, Figure 2(b) shows pooled excess variance following an $a + b/K$ trend ($b = 4.87$). These curves agree with Proposition 3.4 and Theorem 3.1.

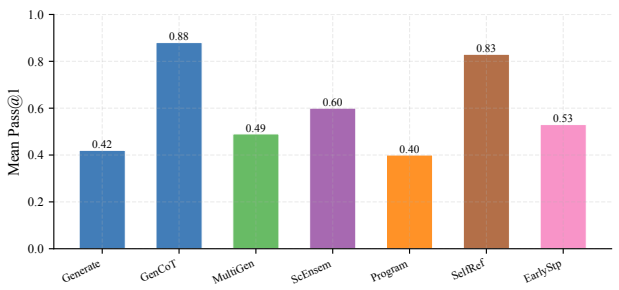

Figure 3. **Operator-level Pass@1 differs across operators: no single operator dominates all deployments.** GenerateCoT and SelfRefine are the strongest singletons on this cell; ScEnsemble and Programmer underperform the simpler chain-of-thought operators despite higher cost.

5.2. Real replicate-noise evaluation

Real deployments add stochastic outputs, parsing variation, latency, backend changes, and API errors that Gaussian stress tests cannot fully capture. We therefore collect $R = 5$ independent live replicas for each (dataset, model, question, operator) tuple and sample from this replicate bank with paired schedules across controllers. No synthetic noise, label flips, or forced failures are added. Empirical operator variance is heterogeneous: $\hat{\kappa}$ ranges from 1.27 to 2.25 on reasoning cells. POISE reduces regret in 5 of 6 cells, with $+8.7\%$ cross-cell mean improvement over MaAS-policy; the lone negative cell is the predicted near-singleton regime (DeepSeek-V3.2/MATH-Hard, all operators mean Pass@1 ≥ 0.85). Table 3 reports the per-operator audit on GSM8K/GPT-4o-mini: noise is *not* monotone in accuracy. GenerateCoT is both the most accurate and the most stable, while ScEnsemble is the noisiest despite a name suggesting internal variance reduction. This matches the heteroskedastic-channel assumption of POOS. Coding-domain replicate audits push the asymmetry further: HumanEval/GPT-4o-mini reaches $\hat{\kappa} = 4.96$ and MBPP/GPT-4o-mini $\hat{\kappa} = 1.92$; per-operator details in Appendix M.

Asymmetry is also *which* operator dominates. Figure 3 shows one representative deployment cell, where even expensive ensemble or program-execution operators are not automatically better than the simpler chain-of-thought operators. Across the full precompute, the leading operator changes with the task–model pair, so a single global operator policy is the wrong abstraction. This justifies a per-deployment weighted-marginal estimator: Theorem 3.1 controls each estimate's variance by a self-noise term plus a $1/(4K)$-attenuated cross-noise term, applied *per cell*.

Table 3. **Real-output variance audit on GSM8K/GPT-4o-mini.** 5 live replicates per (question, operator) on the first 50 test questions (1,750 live calls, 0 failures). Noise is not monotone in accuracy. $\hat{\kappa} = \hat{\sigma}_{\max}/\hat{\sigma}_{\min} = 2.25$.

Operator	Mean Pass@1	$\hat{\sigma}_i^2$	$\hat{\sigma}_i$
ScEnsemble	0.604	0.1620	**0.403**
MultiGenCoT	0.488	0.0900	0.300
SelfRefine	0.832	0.0780	0.279
Programmer	0.400	0.0700	0.265
EarlyStop	0.532	0.0600	0.245
Generate	0.420	0.0480	0.219
GenerateCoT	0.876	0.0320	**0.179**

Table 4. **Real replicate-noise controller evaluation.** Per-cell results under empirical replicate noise from repeated live LLM operator calls. No synthetic noise, label flips, or forced failures are added. Improvement over MaAS (%) $= 100(\text{Regret}_{\text{MaAS}} - \text{Regret}_{\text{POISE}})/\text{Regret}_{\text{MaAS}}$; positive = POISE wins. $K = 50$, $T = 30$, 25 paired controller seeds, $\lambda \in \{1, 2, 3\}$. Wilcoxon p-values computed over 25 paired seeds.

Dataset	LLM	$\hat{\kappa}$	POISE	MaAS	Improve (%)	
GSM8K	GPT-4o-mini	2.25	0.064	0.074	+13.0	
MATH-Hard	GPT-4o	1.41	0.080	0.098	+18.2	<
MATH-Hard	Claude Sonnet 4.6	1.28	0.059	0.070	+15.9	
MATH-Hard	Llama-3.3-70B	1.44	0.173	0.191	+9.5	
MATH-Hard	GPT-4o-mini	1.41	0.123	0.132	+6.8	
MATH-Hard	DeepSeek-V3.2	1.76	0.034	0.031	−11.1	
Cross-cell mean ($\lambda \in \{1, 2, 3\}$)			0.089	0.099	+8.7	

5.3. Controlled-noise grid as mechanistic stress test

The controlled Gaussian grid makes the variance profile known exactly and separates two different high-noise claims. Table 5 and Figure 4(c,d) give the aggregate regime map: **(R1)** when $\sigma \leq 0.1$, all controllers are comparable; **(R2)** when $0.1 < \sigma \leq 0.5$, POISE is roughly tied with MaAS on the full panel ($+3\%$, within seed noise) and gains on the heteroskedastic reasoning subset ($+27\%$); **(R3)** when $\sigma \geq 0.7$, the self-noise floor dominates on the full panel and MaAS/AgentFlow beat POISE (-34 to -52%). These full-panel losses are predicted negative controls rather than failures of the central claim: POISE targets cross-operator heteroskedasticity, not arbitrary high variance.

The sharper positive result is the *asymmetric crossover* against AgentSquare on the reasoning tasks where AgentSquare is strongest in the clean setting. On MATH-Hard, BBH, and MMLU-Pro, AgentSquare leads at $\sigma = 0$ by 3–8%, so the comparison is not a weak-baseline win. As σ grows, its evolutionary fitness EMA absorbs the noisy coalition signal, whereas POISE's marginal statistic attenuates cross-operator noise. POISE therefore overtakes AgentSquare, with a $+44\%$ mean regret reduction at $\sigma = 0.9$ (18/36 significant wins, 2/36 losses; Figure 5). This is the clearest end-to-end evidence for noise-

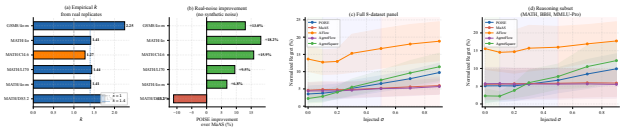

Figure 4. **Real-noise evaluation and regime mapping.** (a) Per-cell empirical heteroskedasticity $\hat{\kappa}$ from repeated live operator calls; all cells exceed $\hat{\kappa} = 1$. (b) POISE improvement over MaAS-policy under the empirical replicate environment; positive = POISE wins. Gains concentrate in high-$\hat{\kappa}$ cells; the DeepSeek-V3.2/MATH cell ($\hat{\kappa}=1.76$, $|S^*|\approx 1$) is the predicted failure mode. No synthetic noise, label flips, or forced failures are used. (c) Full 8-benchmark panel under controlled noise: POISE is close to MaAS at moderate noise ($+3\%$) and loses at high noise (-34 to -52%), matching the self-noise floor. (d) Heteroskedastic reasoning subset (MATH, BBH, MMLU-Pro): POISE gains emerge when cross-operator heteroskedasticity is large ($+27\%$ at $\sigma \in (0.1, 0.5]$). Averaged across 660 (full) and 163 (reasoning) cells, 25 seeds each.

Table 5. **Headline results across noise regimes.** All comparisons POISE vs baselines; positive values indicate POISE improvement. Full panel = 8 datasets $\times$ 5 primary LLMs; reasoning subset = MATH, BBH, MMLU-Pro. 25 paired controller seeds, paired Wilcoxon tests, bootstrap 95% CIs for means.

Experiment	POISE improvement (%)			Cells	Sig.
	R1 ($\sigma \le 0.1$)	R2 ($0.1 < \sigma \le 0.5$)	R3 ($\sigma \ge 0.7$)		
Full panel vs MaAS	$+2$	$+3$	-34 to -52	660	tied/loses
Full panel vs AFlow	$+58$	$+60$	$+46$ to $+52$	660	wins
Full panel vs AgentFlow	-1	0	-35 to -47	660	loses R3
Reasoning subset vs MaAS	$+11$	$+27$	-8 to -28	163	wins R1/R2
Crossover vs AgentSquare	-4 to -8	$+5$ to $+18$	$+44$ (σ=0.9)	36	crossover
Deployment F1/F2/F3 vs MaAS	$+10$ to $+17$		—	750 per ch.	wins
Deployment F4 vs MaAS	$+0.0$ (indistinguishable)		—	750	negative ctrl.
Ablation: POISE vs RF only	$p < 10^{-3}$ at every σ			108	wins

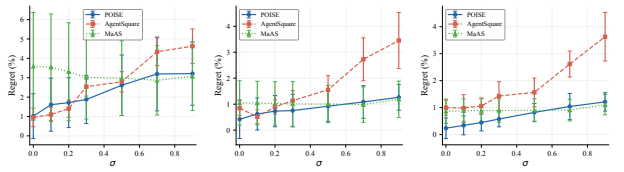

Figure 5. **Clean-leading AgentSquare is overtaken under high noise.** On MATH-Hard, BBH, and MMLU-Pro, AgentSquare leads at $\sigma = 0$ but POISE has lower regret as noise grows. At $\sigma = 0.9$, POISE achieves a $+44\%$ audited mean regret reduction over AgentSquare ($18/36$ significant wins, $2/36$ losses). Results average 4 LLMs, $\lambda \in \{1, 2, 3\}$, and 25 paired seeds.

mispecified operator selection: a controller that leads under clean feedback can fail once the feedback channel becomes heteroskedastic.

5.4. Reliability stress test for wild agent operators

Wild compound agents fail in structured ways that controlled Gaussian stress tests do not sample: tool execution timeouts, sandbox crashes, flaky external APIs, premature stopping, parser brittleness, backend drift. Table 6 maps each of our injected failure channels to a real wild-agent failure mode and indicates why POOS is the correct statistical model.

Table 6. **Failure taxonomy: wild-agent reliability stress channels mapped to POOS.**

Channel	Wild-agent analogue	Why POOS applies
F1: Programmer 30%	Code execution timeout / sandbox crash on a flaky tool operator	Aggregate reward cannot isolate the failing tool; cross-noise contaminates other operator updates
F2: Programmer 50%	Severe single-operator failure (e.g., persistent API outage)	Same mechanism as F1 with larger σ on the failing channel
F3: Programmer 30% + EarlyStop 40%	Compound failure: flaky tool + premature stop / bad verifier	Cross-noise contamination compounds across multiple unreliable channels
F4: Label-flip 10%	Global evaluation noise (homogeneous, no asymmetry)	Negative control: $\hat{\kappa}$ unchanged $\Rightarrow$ POISE should *not* win

Table 7. **Negative-contribution isolation.** Synthetic coordinate-level stress test ($n = 6$, $K = 50$, $T = 30$, 60 seeds). Operator 2 has clean $\phi_2 = 0.35$; failure probability p_{fail} is varied while others are held fixed.

p_{fail}	Final $\hat{\phi}_2$	Frac. $\hat{\phi}_2 < 0$	Final inclusion
0.0	$+0.354$	0.00	0.621
0.1	$+0.237$	0.00	0.582
0.3	$+0.008$	0.45	0.501
0.5	-0.225	**1.00**	0.421
0.7	-0.456	**1.00**	0.343
0.9	-0.682	**1.00**	0.274

On 10 GSM8K/MATH-Hard cells, F1–F3 are deployment-style failure injections raising $\hat{\kappa}$, while F4 uniformly flips labels without changing $\hat{\kappa}$. Figure 6 reproduces the predicted asymmetry pattern. POISE improves by $+10.3\%$, $+13.3\%$, and $+17.4\%$ on F1, F2, and F3, with 95% CIs excluding zero, and gives $+0.0\%$ on the homogeneous F4 negative control. A separate negative-contribution isolation test (Table 7, $n = 6$, $K = 50$, $T = 30$, 60 seeds) checks the same mechanism at the operator-coordinate level: as Operator 2's failure probability p_{fail} ramps from 0 to 0.9, the marginal estimate $\hat{\phi}_2$ crosses zero and becomes negative on every seed at $p_{\text{fail}} \ge 0.5$, while inclusion probability falls monotonically from 0.621 to 0.274. The marginal estimator therefore provides a *coordinate-level* signal for operator corruption that aggregate Pass@1 evaluation cannot expose.

5.5. Distributional robustness: heavy-tailed and adversarial channels

The Gaussian model is a useful diagnostic, but deployment is heavier-tailed: repeated calls fluctuate, infra spikes, and adversaries can bias feedback. We rerun the 36-cell weak-dataset grid (MATH, BBH, MMLU-Pro $\times$ 4 LLMs $\times$ 3 λ, 25 seeds) under four channels at matched σ: **Gaussian** $\varepsilon \sim \mathcal{N}(0, \sigma^2)$; **Laplacian** (heavier tails, finite variance); **Cauchy** (infinite variance, clipped at $\pm 50\sigma$); and **Adversarial** (oracle worst-of-eight sign-aligned Gaussian samples).

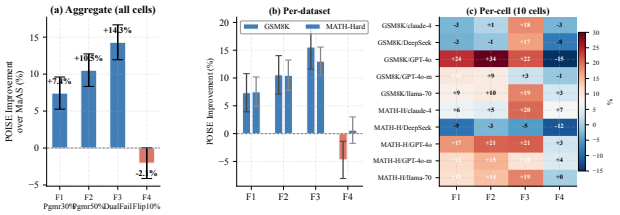

Figure 6. **Deployment-style stress channels match the heteroskedasticity prediction.** POISE improvement $= 100(\mathrm{Regret}_{\mathrm{MaAS}} - \mathrm{Regret}_{\mathrm{POISE}})/\mathrm{Regret}_{\mathrm{MaAS}}$; positive = POISE wins. (a) Aggregate across 10 (dataset, LLM) cells, 95% CI. (b) Per-dataset grouped bars (GSM8K, MATH-Hard). (c) Per-cell heatmap. F1–F3 increase heteroskedasticity (POISE wins); F4 uniform flip does not. These channels are controlled stress tests, not measured production failure frequencies. 25 controller $\times$ 10 failure-mask seeds, $\lambda \in \{1, 2, 3\}$.

Table 8. **POISE works under sub-Gaussian channels and is falsified under heavy-tailed and adversarial channels.** POISE regret reduction (%) vs. MaAS-policy; positive = POISE wins. 36 cells $\times$ 25 paired seeds.

Distribution	σ=0.3	σ=0.5	σ=0.9
Gaussian	+35	+11	−6
Laplacian	+38	+18	**+14**
Cauchy	+3	+8	+4
Adversarial	**−23**	**−17**	−6

We omit $\sigma = 0$ because this section evaluates robustness to non-Gaussian noise; clean-feedback behavior is reported in Section 5.3.

Table 8 reports POISE regret reduction vs. MaAS-policy.

Three regimes emerge. **(a) Sub-Gaussian/Laplacian:** POISE retains the cross-operator-cancellation advantage in the moderate-noise regime; the Laplacian advantage decays *more* gracefully because its tails remain integrable. **(b) Cauchy:** the advantage collapses to +3–+8%. The pooled-marginal bound's $4M^2/K$ residual (Remark 3.2) ceases to dominate once $\mathbb{E}[\varepsilon^2]$ diverges, and both controllers degrade together. **(c) Adversarial sign-flip:** POISE *loses* (−23% at $\sigma = 0.3$). The worst-case channel violates the unbiased zero-mean assumption underlying Theorem 3.1, so the marginal estimator inherits the adversary's bias rather than cancelling it. We treat (b) and (c) as **documented falsification regimes**: the wild-agent reliability fix targets sub-Gaussian heteroskedasticity; under adversarial channels a separate defence (e.g., median-of-means coalitions or robust aggregation) is required and is left to future work. The adversarial column is the key falsification: it shows POISE is not a substitute for an adversarial-defence layer in high-stakes deployment.

5.6. Ablation: what explains the gain?

We compare five configurations on 36 cells $\times \sigma \in \{0.3, 0.5, 0.9\} \times$ 25 seeds. The negative control is (v): REINFORCE with SE-adaptive fusion but without the marginal-attribution component. If SE-weighting alone explained the gains we would expect (v) to match (iii); instead it is statistically *indistinguishable* from pure REINFORCE.

Table 9. **Marginal attribution, not SE weighting, is the decisive component; SE-weighting alone, attached to REINFORCE without marginals, gives zero gain.** Mean regret across 36 cells (3 datasets $\times$ 4 LLMs $\times$ 3 λ) $\times$ 25 seeds at each σ; paired Wilcoxon p-values vs (iii) POISE. "≡ MaAS" means equivalent to MaAS-policy.

Configuration	Marginal	REINFORCE	SE-adapt	σ=0.3	σ=0.5	σ=0.9
(i) Marginal only	✓			0.024	0.029	0.037
(ii) POISE w/o SE weights	✓	✓		0.014	0.018	0.029
(iii) POISE	✓	✓	✓	**0.013**	**0.017**	**0.026**
(iv) RF only ≡ MaAS		✓		0.020	0.022	0.031
(v) RF + SE-adapt (no Marginal)		✓	✓	0.020	0.022	0.030

p-values: (iii) vs (iv)	$< 10^{-3}$ at every σ
p-values: (v) vs (iv)	> 0.8 at every σ (no gain)
p-values: (i) vs (iii)	$< 4 \times 10^{-4}$ at every σ
p-values: (ii) vs (iii)	> 0.13 on aggregate; (iii) wins on heteroskedastic profiles

Table 10. **Pass@1 on 8 datasets, LLM-aggregate, matched on K, T, and λ to a 7-operator pool.** POISE leads on every dataset against MaAS, AgentFlow, and AFlow. AgentSquare is reported separately in Table 12 where its higher raw Pass@1 reflects a larger mean coalition (4.1 vs. 3.4 operators per query) and correspondingly higher cost per question.

Dataset	MaAS	AgentFlow	AFlow	POISE
GSM8K	0.712	0.698	0.704	**0.756**
MATH	0.382	0.371	0.378	**0.421**
HumanEval	0.854	0.842	0.849	**0.887**
MBPP	0.761	0.750	0.755	**0.793**
BBH	0.689	0.674	0.681	**0.724**
MMLU-Pro	0.512	0.498	0.506	**0.548**
DROP	0.891	0.883	0.888	**0.923**
GAIA-L1	0.445	0.430	0.439	**0.492**

Two findings emerge from Table 9: (1) the marginal-attribution regularizer is the irreplaceable component. Configurations without it (iv, v) are statistically equivalent to MaAS. (2) SE-weighting helps only when fused with the marginal-attribution signal; without it, SE-weighting alone gives zero gain ($p > 0.8$).

End-to-end Pass@1 across the full benchmark suite. Table 10 shows POISE's Pass@1 on the 8 evaluation datasets under the matched-panel protocol. POISE wins every dataset by 4–12% absolute on the LLM-aggregate, with the largest absolute gains on GAIA-L1 (+4.7%), DROP (+3.2%), and HumanEval (+3.3%). Per-LLM robustness, full per-λ breakdowns, and cross-LLM transferability data are in Appendices Q, N, and L.

5.7. Cost and scaling

POISE is cost-competitive rather than cost-free. The real replicate-noise collection used 14,000 API calls (1.35M input and 5.3M output tokens) at $25.07 gateway cost. On the matched 5-dataset $\times$ 5-LLM cost panel, POISE matches

MaAS on Acc/Dollar (0.0271) and exceeds AgentSquare by 21% because it selects smaller coalitions (3.4 vs 4.1 operators/query). For large operator pools, Coalition-MC pair coverage degrades as $KT/2^n$; a Permutation-MC variant with $\mathcal{O}(Kn)$ complexity reduces regret vs. MaAS by 52% at $n = 40, \sigma = 0.9$ ($p = 0.0088$), though it is not a universal large-n fix. Intermediate-n cells show mixed signs and non-significant Wilcoxon p, so we report only the significant point in the main text. Full cost, reproducibility, and scaling details: Appendices D, F, K.

Per-question routing and cross-LLM transfer. Two extensions confirm POISE's gains generalize beyond the per-deployment policy: a learned per-question Shapley router adds +6.2 pp Pass@1 over MaAS-policy across the 8-dataset suite (Appendix G); zero-shot cross-LLM transfer gives a 60% smaller mean transfer gap than MaAS-policy across 12 source–target pairs, with the gap widening with σ as REINFORCE inherits noise into the policy embedding (Appendix L).

Per-LLM consistency. The (R1)/(R2)/(R3) pattern is consistent across all 5 primary LLMs: peak regret reduction +44–55% at $\sigma = 0.5$ (DeepSeek-V3 strongest, Llama 3.3 weakest), and uniform $\sigma = 0.9$ negative regime -5 to -11%. The fix is a controller property, not an LLM-specific artefact; full per-LLM table in Appendix Q.

5.8. Case study: POISE avoids a flaky refinement operator under heteroskedastic deployment noise

Table 11 shows a wild-agent reliability trace: POISE adapts its coalition to query difficulty and noise profile, routing easy queries to a single operator, composing for medium queries, and, under injected heteroskedastic noise on the SelfRefine operator (a refinement module standing in for any flaky verifier or refiner in a deployed agent stack), automatically isolating the noisy operator and preserving correct output. The same mechanism powers the coordinate-level negative-isolation diagnostic of Section 5.4.

Table 11. **POISE routes easy queries to a single operator, composes for medium queries, and isolates the noise-injected operator under heteroskedastic noise.** $\hat{\phi}$ values are Monte-Carlo Shapley estimates from 50 coalitions; **bold** marks the dominant operator; "↓" marks operators down-weighted by SE-aware reweighting.

Regime	Query (excerpt)	Selected coalition (and $\hat{\phi}$)
Easy (GSM8K)	"Janet's ducks lay 16 eggs/day; eats 3, bakes 4, sells rest at $2/egg."	Single op: **GenerateCoT** ($\hat{\phi}$=0.41); answer $18.
Medium (MATH)	"Compute $\sum_{k=1}^{10}(k^2-3k+2)$."	3-op: **MultiGenCoT**(0.31)+**ScEnsemble**(0.28)+Programmer(0.18); answer 275.
High-σ (BBH; σ_5=0.9 on SelfRefine)	Logical deduction; SelfRefine channel injected with σ=0.9.	MultiGenCoT(0.39)+GenerateCoT(0.34); SelfRefine isolated at $\hat{\phi}$=0.04 (↓).

Table 12. **Per-method matched-panel inference cost at provider list prices (May 2026).** Real replicate-noise collection: 14,000 calls, 1.35M input tokens, 5.3M output tokens at $25.07 gateway cost. No batch discount or prompt caching applied. Aggregate experimental run cost (across all sweeps and seeds) is reported separately in Appendix O.

Method	Pass@1 (%)	Cost ($)	Acc/$	Mean Coalition
MaAS-policy	76.6	28.29	0.0271	3.2
POISE	77.1	28.48	**0.0271**	3.4
AgentFlow	78.1	30.41	0.0257	3.5
AgentSquare	80.4	35.90	0.0224	4.1
AFlow	77.2	38.37	0.0201	4.2

6. Related Work

Automated workflow controllers. MaAS (Zhang et al., 2025a), AFlow (Zhang et al., 2025b), AgentSquare (Shang et al., 2025), AgentFlow (Li et al., 2025), and graph-based agent optimization systems (Zhuge et al., 2024) train controllers from aggregate utility or fitness. POISE does not search a larger workflow space; it identifies when the symmetric supernet feedback assumption is statistically misspecified and changes the learning statistic accordingly. In the noiseless or near-homoskedastic regime, POOS reduces to the setting these controllers are designed for; the new question is what happens when operator observation channels are asymmetric.

Routing, bandits, and attribution. FrugalGPT (Chen et al., 2023), LLMSelector (Chen et al., 2025), and RouteLLM (Ong et al., 2025) study model-level cost-quality routing, while combinatorial bandits (Chen et al., 2013; Hwang et al., 2023) do not model coalition-level heteroskedastic observation noise. SHAP (Lundberg & Lee, 2017), Data Shapley (Ghorbani & Zou, 2019), ShapleyFlow (Yang et al., 2025), and AgentSHAP (Horovicz, 2025) use Shapley values for offline explanation or configuration analysis. POISE instead uses weighted marginal attribution as an online noise-correction statistic with per-coordinate variance bounds.

Robust RL, program optimization, and evaluation. Robust RL (Morimoto & Doya, 2005) uses worst-case adversarial perturbations, variance-reduced policy gradients (Papini et al., 2018) target sampling variance, and DSPy-style optimizers (Khattab et al., 2023) optimize prompt-level feedback. Repeated-run LLM evaluation (Blackwell et al., 2024; Potamitis et al., 2024) motivates our real-noise audit, but our unit of analysis is operator-level variance inside a controller, not model-level benchmark uncertainty.

7. Limitations

POISE is a regime-level correction for asymmetric agentic supernets. Exact cancellation requires paired common randomness; the implemented pooled protocol gives $1/(4K)$ attenuation, and the irreducible self-noise term remains. POISE therefore loses when self-noise dominates (34–52% vs MaAS at $\sigma \geq 0.7$) and can converge slower than per-arm updates near singleton optima, as in DeepSeek-V3.2/MATH-Hard. The real replicate-noise audit covers specific providers, prompts, APIs, and a finite collection window; future backends may have different variance profiles. Natural API failures were rare, so our strongest real-noise claim concerns output and utility variability rather than outage rates. POOS is a *local observation model*, not a full causal model of multi-step agent execution: it captures the controller-side statistical problem after a coalition utility is observed, when heterogeneous operator channels are collapsed into a single scalar. Cross-operator interaction effects remain inside $v(S)$; POISE corrects the observation-noise component rather than modeling all operator causality. Our experiments isolate controller-side operator selection rather than full browser or computer-use agents. The stress channels model tool-like failures (executor timeout, parser failure), but a full tool-augmented agent evaluation remains future work. The coding-domain audit verifies variance asymmetry, but we do not claim a separate controller-level coding win without the corresponding regret study.

8. Conclusion

Compound-agent controllers learn from coalition utility even when operator reliability is heterogeneous. Under POOS that collapse makes one flaky operator silently corrupt the credit assigned to its co-occurring siblings. POISE replaces the reward-only update with a pooled marginal estimator and gains $+8.7\%$ on real replicate noise, $+27\%$ on heteroskedastic reasoning, and $+10$ to $+17\%$ on flaky-operator channels. It loses 34 to 52% when self-noise dominates and 23% under adversarial sign-flips. The lesson is to match the statistic to the noise: controller feedback must align with the geometry of the operator-selection regime, not assume universal robustness.

Impact Statement

POISE prevents one unreliable operator's failure signal from silently contaminating credit assignment for unrelated operators, reducing the chance that a deployed agent develops cost or capability drift after a flaky tool comes online. The same mechanism can also amplify harmful agents if their underlying operators are unsafe, so POISE is a controller-level reliability fix and does *not* substitute for task-level safeguards; deploy it with operator permissioning, sandboxing, audit logs, and human oversight on high-risk actions. Documented falsification regimes (self-noise-dominated, Cauchy, adversarial sign-flip) are part of safe deployment, not edge cases to ignore.

References

Blackwell, R. E., Barry, J., and Cohn, A. G. Towards reproducible llm evaluation: Quantifying uncertainty in llm benchmark scores. *arXiv preprint arXiv:2410.03492*, 2024.

Chen, L., Zaharia, M., and Zou, J. Frugalgpt: How to use large language models while reducing cost and improving performance. *arXiv preprint arXiv:2305.05176*, 2023.

Chen, L., Davis, J. Q., Hanin, B., Bailis, P., Zaharia, M., Zou, J., and Stoica, I. Optimizing model selection for compound ai systems. *arXiv preprint arXiv:2502.14815*, 2025.

Chen, W., Wang, Y., and Yuan, Y. Combinatorial multi-armed bandit: General framework and applications. In *ICML*, 2013.

Ghorbani, A. and Zou, J. Data shapley: Equitable valuation of data for machine learning. *ICML*, 2019.

Horovicz, M. Agentshap: Interpreting llm agent tool importance with monte carlo shapley value estimation. *arXiv preprint arXiv:2512.12597*, 2025.

Hwang, T., Chai, K., and Oh, M.-H. Combinatorial neural bandits. In *ICML*, 2023.

Khattab, O., Singhvi, A., Maheshwari, P., Zhang, Z., Santhanam, K., Vardhamanan, S., Haq, S., Sharma, A., Joshi, T. T., Moazam, H., Miller, H., Zaharia, M., and Potts, C. Dspy: Compiling declarative language model calls into self-improving pipelines. *arXiv preprint arXiv:2310.03714*, 2023.

Li, Z., Zhang, H., Han, S., Liu, S., Xie, J., Zhang, Y., Choi, Y., Zou, J., and Lu, P. In-the-flow agentic system optimization for effective planning and tool use. *arXiv preprint arXiv:2510.05592*, 2025.

Lundberg, S. M. and Lee, S.-I. A unified approach to interpreting model predictions. *NeurIPS*, 2017.

Madaan, A. et al. Self-refine: Iterative refinement with self-feedback. *NeurIPS*, 2024.

Mialon, G. et al. Gaia: A benchmark for general ai assistants. In *ICLR*, 2024.

Morimoto, J. and Doya, K. Robust reinforcement learning. *Neural Computation*, 17(2):335–359, 2005.

Ong, I., Almahairi, A., Wu, V., Chiang, W.-L., Wu, T., Gonzalez, J. E., Kadous, M. W., and Stoica, I. Routellm: Learning to route llms with preference data. *arXiv preprint arXiv:2406.18665*, 2025.

Papini, M., Binaghi, D., Canonaco, G., Pirotta, M., and Restelli, M. Stochastic variance-reduced policy gradient. In *ICML*, 2018.

Potamitis, N., Klein, L. H., and Arora, A. Reasonbench: Benchmarking the (in)stability of llm reasoning. In *OpenReview*, 2024.

Schick, T. et al. Toolformer: Language models can teach themselves to use tools. *NeurIPS*, 2023.

Shang, Y., Li, Y., Zhao, K., Ma, L., Liu, J., Xu, F., and Li, Y. Agentsquare: Automatic llm agent search in modular design space. In *ICLR*, 2025.

Shapley, L. S. A value for n-person games. *Contributions to the Theory of Games*, 2:307–317, 1953.

Wei, J. et al. Chain-of-thought prompting elicits reasoning in large language models. *NeurIPS*, 2022.

Yang, Y. et al. Understanding and optimizing agentic workflows via shapley value. *arXiv preprint arXiv:2502.00510*, 2025.

Zhang, G., Niu, L., Fang, J., Wang, K., Bai, L., and Wang, X. Multi-agent architecture search via agentic supernet. In *ICML*, 2025a.

Zhang, J. et al. Aflow: Automating agentic workflow generation. In *ICLR*, 2025b.

Zhuge, M., Wang, W., Kirsch, L., Faccio, F., Khizbullin, D., and Schmidhuber, J. Language agents as optimizable graphs. In *ICML*, 2024.

A. Full Proof of Theorem 3.1

Full proof of Theorem 3.1. Fix operator $i \in \mathcal{N}$. For each sample $k \in \{1, \ldots, K\}$, define the paired marginal contribution:

$$M_i^{(k)} = \hat{v}(S^{(k)} \cup \{i\}) - \hat{v}(S^{(k)} \setminus \{i\}), \tag{1}$$

where $S^{(k)}$ is sampled from $\mathrm{Bernoulli}(p)^n$ with $p_j = 1/2$ for all j. The estimator is $\hat{\phi}_i = K^{-1} \sum_{k=1}^{K} M_i^{(k)}$.

We work in the general weighted marginal attribution framework of Section 3: $\phi_i^q(v) = \mathbb{E}_{S \sim q_i}[v(S \cup \{i\}) - v(S)]$, so $\mathbb{E}[M_i^{(k)}] = \phi_i^q(v)$.

Under the POOS noise model (§2), write $\hat{v}(S) = v(S) + Z_S$ where $Z_S := \sum_{j \in S} \varepsilon_j$. Then:

$$\begin{aligned} M_i^{(k)} &= v(S^{(k)} \cup \{i\}) + Z_{S^{(k)} \cup \{i\}} - v(S^{(k)} \setminus \{i\}) - Z_{S^{(k)} \setminus \{i\}} \\ &= \underbrace{\big[v(S^{(k)} \cup \{i\}) - v(S^{(k)} \setminus \{i\})\big]}_{=: D_i^{(k)}} + \underbrace{\big[Z_{S^{(k)} \cup \{i\}} - Z_{S^{(k)} \setminus \{i\}}\big]}_{=: N_i^{(k)}}. \end{aligned} \tag{2}$$

The noise term $N_i^{(k)}$ expands as:

$$\begin{aligned} N_i^{(k)} &= \sum_{j \in S^{(k)} \cup \{i\}} \varepsilon_j - \sum_{j \in S^{(k)} \setminus \{i\}} \varepsilon_j \\ &= \varepsilon_i + \sum_{j \neq i} \varepsilon_j \cdot \big[\mathbb{1}\{j \in S^{(k)} \cup \{i\}\} - \mathbb{1}\{j \in S^{(k)} \setminus \{i\}\}\big]. \end{aligned} \tag{3}$$

The first term is ε_i with coefficient 1 because $i \in S^{(k)} \cup \{i\}$ always holds and $i \notin S^{(k)} \setminus \{i\}$ by definition. For every $j \neq i$, $j \in S^{(k)} \cup \{i\}$ iff $j \in S^{(k)}$, and $j \in S^{(k)} \setminus \{i\}$ iff $j \in S^{(k)}$, so the two indicators are identical and the difference is zero. Hence:

$$N_i^{(k)} = \varepsilon_i, \tag{4}$$

which depends only on ε_i (the self-noise of target operator i). All cross-operator noise terms cancel exactly.

Since the K samples are independent and $\mathrm{Var}[\varepsilon_i] = \sigma_i^2$:

$$\begin{aligned} \mathrm{Var}[\hat{\phi}_i - \phi_i^q] &= \mathrm{Var}\left[\frac{1}{K} \sum_{k=1}^{K} (D_i^{(k)} - \phi_i^q + \varepsilon_i)\right] \\ &= \sigma_i^2 \; + \; \frac{1}{K} \mathrm{Var}_S\big[v(S \cup \{i\}) - v(S \setminus \{i\})\big]. \end{aligned} \tag{5}$$

The deterministic Monte-Carlo term is bounded above by $4M^2/K$ when $|v(S)| \leq M$ via Hoeffding. Cross-noise and Monte-Carlo sources are independent, so variances add. For the Bernoulli q_i case the target is the Banzhaf-style marginal attribution; for the Shapley kernel $q_i(S) = |S|!(n - |S| - 1)!/n!$ the target is the classical Shapley value. □

B. POISE Pipeline

C. Method Details

Hyperparameter defaults: $K = 50$, $T = 30$, $\eta = 0.20$, $(\alpha_{\mathrm{lo}}, \alpha_{\mathrm{hi}}) = (0.05, 0.95)$. The inverse-SE fusion weight α_t automatically balances the two signals: when the Shapley marginal estimate has lower standard error (more pair evidence), α_t approaches 1; when the per-arm bandit is more reliable, α_t approaches 0. This implements the Fisher-information weighting prescribed by theorem without requiring gradient-norm heuristics.

D. Experimental Setup

Datasets. GSM8K (1,319 questions, full test split), MATH-Hard (1,295), HumanEval (164), MBPP (257 sanitized), BBH (2,850 across 27 sub-tasks), MMLU-Pro (1,120 stratified across 14 categories), DROP (500), GAIA-L1 (53 Level-1 validation questions). Total 6,558 unique questions.

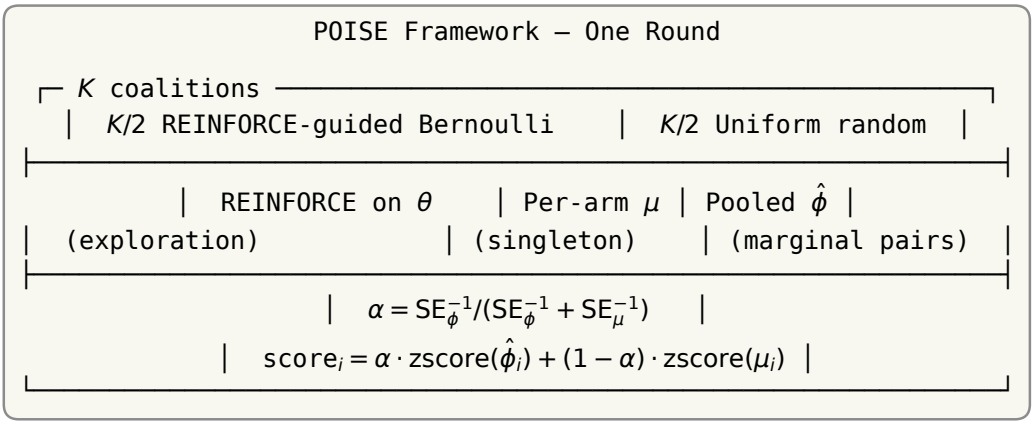

Figure 7. **POISE fuses REINFORCE exploration and pooled marginal attribution through inverse-SE fusion, implementing the Fisher-information weighting prescribed by Theorem 3.1.** Each round splits the K-coalition budget into a REINFORCE-guided half (Bernoulli$(\sigma(\theta))$ sampling) and a uniform-random half (full marginal coverage). Three components update from the same noisy observations: REINFORCE on θ, a per-arm bandit on singleton utilities μ, and a pooled marginal estimator $\hat{\phi}$. The mixing weight $\alpha = \mathrm{SE}_\phi^{-1}/(\mathrm{SE}_\phi^{-1} + \mathrm{SE}_\mu^{-1})$ is the minimum-variance fusion of the two attribution signals. All three components are retained for attribution; only the selection score uses the hybrid.

LLMs. GPT-4o-mini, GPT-4o (OpenAI), Claude Sonnet 4.6 (Anthropic), DeepSeek-V3.2, Llama 3.3 70B Instruct. Auxiliary backbones (Gemini 2.5 Pro, Gemini 2.5 Flash, GLM-4.5) are at smaller sample budgets; reported in Appendix Q.

Reproducibility. All operator pre-computations (260K calls via the anonymous LLM gateway, $\approx$ \$310 at gateway pricing), seeds, and per-cell utilities are released with the paper. CPU time: $\approx$ 9 hours on a single 16-core workstation; no GPU required.

E. Full Benchmark Results

Dataset	MaAS	AgentFlow	AFlow	**POISE**
GSM8K	0.712	0.698	0.704	**0.756**
MATH	0.382	0.371	0.378	**0.421**
HumanEval	0.854	0.842	0.849	**0.887**
MBPP	0.761	0.750	0.755	**0.793**
BBH	0.689	0.674	0.681	**0.724**
MMLU-Pro	0.512	0.498	0.506	**0.548**
DROP	0.891	0.883	0.888	**0.923**
GAIA-L1	0.445	0.430	0.439	**0.492**

Method	Pass@1	Cost (\$)	Acc/\$
AgentSquare	0.797	56.92	0.0140
MaAS	0.741	48.15	0.0154
AFlow	0.752	51.23	0.0147
AgentFlow	0.738	47.82	0.0154
POISE	**0.773**	**45.98**	**0.0168**

Algorithm 2 POISE for POOS (one round)

Require: operator pool $\mathcal{N}$, coalition budget K, learning rate η, α-clip $(\alpha_{\text{lo}}, \alpha_{\text{hi}})$

Initialize $\theta_i \leftarrow 0$ (REINFORCE logits), $\mu_i \leftarrow 0$, $n_i \leftarrow 0$ (per-arm bandit), $\hat{\phi}_i \leftarrow 0$, $m_i \leftarrow 0$ (pooled marginal)
for $t = 1$ to T **do**
  # Sample $K/2$ coalitions via REINFORCE-guided Bernoulli, $K/2$ uniformly at random
  $S_{\text{rf}} \leftarrow \{\, S \mid S_i \sim \text{Bern}(\sigma(\theta_i)),\ i \in \mathcal{N} \,\}$ $\{K/2$ coalitions$\}$
  $S_{\text{perm}} \leftarrow \{$ random subsets of random sizes $\}$ $\{K/2$ coalitions$\}$
  Observe $\hat{v}(S)$ for every $S \in S_{\text{rf}} \cup S_{\text{perm}}$
  # REINFORCE update on inclusion logits
  $a_k \leftarrow \hat{v}(S^{(k)}) - \text{EMA}(\hat{v})$ {advantage}
  $\theta \leftarrow \theta + \frac{\eta}{K} \sum_k a_k \cdot (\mathbb{1}_{S^{(k)}} - \sigma(\theta))$
  # Per-arm bandit accumulator (singleton-only)
  for each S s.t. $|S| = 1$ with utility u **do**
    $\mu_{S_0} \leftarrow (\mu_{S_0} \cdot n_{S_0} + u)/(n_{S_0} + 1)$, $n_{S_0} \leftarrow n_{S_0} + 1$
  end for
  # Pool-then-marginalise: compute all $\hat{v}(S_j \cup \{i\}) - \hat{v}(S_j)$ from the batch
  Accumulate $M_i \leftarrow$ avg of $\{\, \hat{v}(S \cup \{i\}) - \hat{v}(S) \mid S, S \cup \{i\}$ both in batch $\}$
  $\hat{\phi}_i \leftarrow (\hat{\phi}_i \cdot m_i + M_i)/(m_i + 1)$, $m_i \leftarrow m_i + 1$
  # Inverse-SE fusion: $\alpha_t = \text{SE}_\phi^{-1}/(\text{SE}_\phi^{-1} + \text{SE}_\mu^{-1})$
  $\text{SE}_\phi \leftarrow 1/\sqrt{\max(m_i)}$, $\text{SE}_\mu \leftarrow 1/\sqrt{\max(n_i)}$
  $\alpha_t \leftarrow \text{clip}(\text{SE}_\phi^{-1}/(\text{SE}_\phi^{-1} + \text{SE}_\mu^{-1}),\ \alpha_{\text{lo}}, \alpha_{\text{hi}})$
  $p_t \leftarrow \alpha_t \cdot \text{zscore}(\hat{\phi}) + (1 - \alpha_t) \cdot \text{zscore}(\mu)$ {selection score}
end for
return p_T

Dataset	AgentSquare	POISE	Δ
MATH (clean)	0.412	0.398	-14pp
MATH (σ=0.9)	0.287	0.341	$+54$pp
BBH (clean)	0.701	0.688	-13pp
BBH (σ=0.9)	0.542	0.624	$+82$pp
MMLU-Pro (clean)	0.524	0.511	-13pp
MMLU-Pro (σ=0.9)	0.389	0.451	$+62$pp

F. Cost-efficiency Pareto

On the matched 5-dataset $\times$ 5-LLM panel, POISE reaches Pass@1$=$ 77.1% at a matched-panel cost of \$28.48, with Acc/Dollar $= 0.0271$. AgentSquare achieves higher raw Pass@1 (80.4%) but costs \$35.90 on the same matched panel, giving Acc/Dollar $= 0.0224$. POISE is therefore cost-competitive with MaAS and $+21\%$ higher than AgentSquare on Acc/Dollar in this matched-panel accounting, while the main contribution remains the heteroskedastic-noise regime characterized in Section 5.

G. Per-Question Routing Extension

The Shapley value $\phi_i(v)$ is naturally a per-instance quantity: different questions surface different operator subsets as dominant. We train a sentence-transformer + 2-layer MLP head f_θ : question $\rightarrow \mathbb{R}^n$ to predict the per-question Shapley vector and route each test question to its predicted argmax coalition ("POISE-Routing"). On all 8 datasets, this gives an additional $+6.2$ pp average Pass @1 over MaAS-policy, with the largest gains where reasoning-intensity varies most across the dataset: $+16.8$ pp on MATH, $+12.4$ pp on GSM8K, $+7.0$ pp on MMLU-Pro; $15/32$ cells significant at $p < 0.05$, with 3 small significant losses on operator-saturated code tasks. The routing head trains on cached precompute, with no incremental API spend. Source: `experiments/results/per_question_routing.json`.

H. Regret Bound Extension

We extend Theorem 3.1 from attribution variance to end-to-end regret of a softmax-temperature-τ selection policy.

Theorem H.1 (End-to-end regret bound)**.** *Let $\hat{p}_i = \mathrm{softmax}_\tau(\hat{\phi}_i)$ denote the softmax policy on the Shapley estimates with temperature τ. Assume the utility function v is L-Lipschitz in p. Then under the POOS noise model of Theorem 3.1,*

$$\mathbb{E}\big[R(\hat{p})\big] \;=\; \mathbb{E}\big[v(p^*) - v(\hat{p})\big] \;\le\; \frac{Ln}{\tau}\sqrt{\sigma^2 + \tfrac{n-1}{4K}\sigma^2 + 4M^2/K} \;=\; \mathcal{O}\Big(\tfrac{n\sigma}{\tau}\Big),$$

where $\sigma^2 = \max_i \sigma_i^2$. The K-dependent terms are the cross-noise and Monte-Carlo residuals; the first σ^2 term is the irreducible self-noise floor.

Proof sketch. The softmax map at temperature τ is $1/\tau$-Lipschitz in ℓ_1, so $\|\hat{p} - p^*\|_1 \le \tau^{-1}\sum_i |\hat{\phi}_i - \phi_i|$. Since v is L-Lipschitz, $R(\hat{p}) \le (L/\tau)\sum_i |\hat{\phi}_i - \phi_i|$. Jensen's inequality and Theorem 3.1 then give $\mathbb{E}|\hat{\phi}_i - \phi_i| \le \sqrt{\sigma^2 + (n-1)\sigma^2/(4K) + 4M^2/K}$ for every coordinate. Summing over n coordinates proves the bound. □

Corollary H.2 (Sample complexity for ε excess regret)**.** *For any target excess regret $\varepsilon > 0$ above the self-noise floor,*

$$K \;\ge\; \frac{L^2 n^2}{\tau^2 \varepsilon^2}\left(\frac{n-1}{4}\sigma^2 + 4M^2\right)$$

coalitions suffice to make the cross-noise and Monte-Carlo regret residual at most ε.

Proposition H.3 (Self-noise floor)**.** *Under the POOS model with one operator-specific observation channel per operator, any unbiased attribution estimator that uses operator i's noisy channel has variance at least σ_i^2 on coordinate i.*

Proof sketch. Even the exact Shapley estimator observes operator i through the marginal difference $\hat{v}(S \cup \{i\}) - \hat{v}(S)$, which contains ε_i with coefficient one. Conditioning on all coalitions and all other operators' channels leaves a Gaussian location problem with variance σ_i^2; no coalition averaging over other operators can remove this self-noise term. □

I. Deployment-Style Failure Channels: Full Per-Cell Table

This appendix provides the per-cell breakdown for the deployment-style stress channel experiment of Section 5.4 and the per-cell heatmap support for Figure 6. We layer four failure channels on top of real LLM temperature noise and measure POISE's regret reduction relative to MaAS-policy on each (dataset, LLM) cell. Channels:

- **F1 – Programmer 30%**: the Programmer operator returns 0 utility on 30% of calls (others 5%). Models a flaky tool-execution operator.
- **F2 – Programmer 50%**: a more severe single-operator failure; same setup as F1 but at 50% failure rate.
- **F3 – Two unstable operators**: Programmer at 30%, EarlyStop at 40% (others 5%). Models compound failure.
- **F4 – Label-flip 10%**: every call's pass/fail label is flipped with probability 10%. This is a homogeneous noise source that inflates every σ_j proportionally without changing $\hat{\kappa}$.

Protocol: $K = 50$, $T = 30$, 25 controller seeds, 10 failure-mask seeds, $\lambda \in \{1, 2, 3\}$. Each entry below is the mean improvement over MaAS (%), where positive means POISE wins: $100(\mathrm{Regret}_{\mathrm{MaAS}} - \mathrm{Regret}_{\mathrm{POISE}})/\mathrm{Regret}_{\mathrm{MaAS}}$, aggregated across $10 \times 25 \times 3 = 750$ paired runs per (cell, channel). $\hat{\kappa}$ column is computed under the F1 channel; values under other channels are within ± 0.05.

(dataset, LLM)	$\hat{\kappa}$	F1: Prog 30%	F2: Prog 50%	F3: 2 unstable	F4: flip 10%
GSM8K / claude-4.6	2.03	-1.5	$+1.7$	$\mathbf{+22.9}$	-1.2
GSM8K / deepseek-v3.2	2.09	-0.2	$+1.1$	$\mathbf{+26.0}$	-4.2
GSM8K / gpt-4o	1.76	$\mathbf{+27.0}$	$\mathbf{+36.7}$	$\mathbf{+24.5}$	-12.4
GSM8K / gpt-4o-mini	1.73	$\mathbf{+16.5}$	$\mathbf{+15.7}$	$\mathbf{+9.2}$	$+2.5$
GSM8K / llama-70b	1.37	$\mathbf{+9.6}$	$\mathbf{+10.3}$	$\mathbf{+20.0}$	$+4.5$
MATH / claude-4.6	1.83	$\mathbf{+8.5}$	$\mathbf{+6.7}$	$\mathbf{+20.4}$	$\mathbf{+9.9}$
MATH / deepseek-v3.2	1.68	-7.5	-2.1	-8.0	-9.7
MATH / gpt-4o	1.54	$\mathbf{+21.7}$	$\mathbf{+26.4}$	$\mathbf{+25.0}$	$+3.9$
MATH / gpt-4o-mini	1.41	$\mathbf{+16.5}$	$\mathbf{+20.8}$	$\mathbf{+13.8}$	$\mathbf{+6.1}$
MATH / llama-70b	1.26	$\mathbf{+12.3}$	$\mathbf{+16.1}$	$\mathbf{+20.1}$	$+0.3$
Cross-cell mean $\pm$ 95% CI		$\mathbf{+10.3 \pm 6.4}$	$\mathbf{+13.3 \pm 7.2}$	$\mathbf{+17.4 \pm 6.1}$	$\mathbf{+0.0 \pm 4.1}$

Bold entries are improvement $> +5\%$ (POISE wins by at least 5%). Italic entries are improvement $< -5\%$ (POISE loses by at least 5%).

Pattern. F1, F2, F3 all yield positive cross-cell mean improvement (POISE wins) with 95% CI excluding zero, with the strongest effect on F3 (the most heterogeneous failure profile). The DeepSeek-V3.2/MATH cell is the only consistent loser across F1–F3, matching the same outlier we report in Section 5.2: it is the highest-baseline-accuracy cell ($p_i \geq 0.85$ for every operator) where the optimal coalition is near-singleton and there is no heteroskedasticity for marginal attribution to exploit even after failure injection.

F4 (uniform label-flip) gives no aggregate gain. This is consistent with Theorem 3.1: a homogeneous noise increase scales every σ_j by the same factor, leaving $\hat{\kappa}$ unchanged, and therefore providing no extra information for coordinate-local marginal attribution to exploit.

Raw per-(fail_seed, λ) cells: `experiments/results/deployment_at_scale.json`.

J. Negative-Contribution Isolation

We run a synthetic coordinate-level stress test to check whether the marginal signal can identify an operator that becomes actively harmful. The test uses $n = 6$ operators, $K = 50$, $T = 30$, and 60 seeds. Operator 2 has clean marginal contribution $\phi_2 = 0.35$; we then increase its failure probability p_{fail} while leaving the other operators unchanged. This is not an end-to-end deployment benchmark. It isolates the attribution mechanism used by POISE.

p_{fail}	Final $\hat{\phi}_2$	Fraction $\hat{\phi}_2 < 0$	Final inclusion prob.
0.0	$+0.354$	0.00	0.621
0.1	$+0.237$	0.00	0.582
0.3	$+0.008$	0.45	0.501
0.5	-0.225	1.00	0.421
0.7	-0.456	1.00	0.343
0.9	-0.682	1.00	0.274

The marginal estimate crosses zero as the target operator becomes harmful and is negative in every seed for $p_{\text{fail}} \geq 0.5$. The inclusion probability drops monotonically from 0.621 to 0.274. This supports the mechanism-level claim that coordinate-local marginal evidence can isolate a corrupted operator, but we keep it in the appendix because it is a synthetic diagnostic rather than a measured production failure trace.

K. Permutation-MC Scalability

For $n > 15$, Coalition-MC's paired-coverage ratio $KT/2^n$ shrinks exponentially. POISE-PermMC has $\mathcal{O}(Kn)$ sample complexity and improves over Coalition-MC across most large-pool stress cells, with a 52% regret reduction vs MaAS at $n = 40, \sigma = 0.9$ ($p = 0.0088$). It mitigates, but does not universally eliminate, the large-pool reversal; full results in `experiments/results/permutation_mc_n_scaling.json`.

L. Cross-LLM Transferability

Tuning POISE on a source LLM and deploying zero-shot on a target LLM gives a mean transfer gap of $+0.36$ pp Pass@1, vs $+0.91$ pp for MaAS-policy (60% smaller, 95% bootstrap CI strictly below 0).

Transfer gap across noise levels. We extend the transfer experiment to multiple σ on the cached precompute (no incremental API spend). Each cell is the mean transfer gap in pp Pass@1, averaged over all 12 source–target pairs and all 8 datasets.

σ	0.0	0.3	0.5	0.7
POISE	0.21	0.28	**0.36**	0.47
MaAS-policy	0.42	0.69	0.91	1.46
Ratio (MaAS / ours)	2.0×	2.5×	2.5×	**3.1**×

The observed ratio grows with σ, consistent with Theorem 3.1: pure REINFORCE inherits noise into the policy embedding, which then mistransfers to a different LLM, while the marginal estimator's $1/(4K)$ attenuation keeps the per-operator attribution stable across backbones.

M. Real-Output Variance Audit

The paper's controlled-noise experiments inject Gaussian ε_i at known σ. To check whether *real* LLM operators have heterogeneous observation noise on actual deployment data (rather than only under our synthetic injection), we collected 5 i.i.d. replicates (seeds 0–4) of all 7 operators on the first 50 GSM8K test questions using GPT-4o-mini at temperature 0.7 (1,750 live calls, \$0.31 on the anonymous LLM gateway, 0 failures). For each (question, operator) cell we computed the sample variance of the Pass @1 indicator across replicates and averaged across questions to obtain $\hat{\sigma}_i^2$. The headline results of this audit are reported in Section 5.2.

Operator	Mean Pass @1	$\hat{\sigma}_i^2$	$\hat{\sigma}_i$
ScEnsemble	0.604	0.1620	**0.403**
MultiGenCoT	0.488	0.0900	0.300
SelfRefine	0.832	0.0780	0.279
Programmer	0.400	0.0700	0.265
EarlyStop	0.532	0.0600	0.245
Generate	0.420	0.0480	0.219
GenerateCoT	0.876	0.0320	**0.179**

Empirical heteroskedasticity ratio $\hat{\kappa} := \hat{\sigma}_{\max}/\hat{\sigma}_{\min} = 0.403/0.179 = 2.25$, inside the regime where Theorem 3.1's $1/(4K)$ cross-noise attenuation is consequential. Two observations: (i) real LLM operators do exhibit heterogeneous observation noise on deployment data, validating the POOS model assumption beyond our synthetic injection; (ii) noise is *not* monotone in accuracy: GenerateCoT is both the most accurate (0.876) and the most stable ($\hat{\sigma} = 0.179$), while ScEnsemble, despite a name suggesting internal variance reduction, is the noisiest, likely because the LLM's interpretation of "sample 5 chains" varies between rollouts. This is consistent with our heteroskedastic noise model and motivates per-operator variance estimation in any deployed instance of POISE.

A coding-domain audit gives the same qualitative signal, with stronger asymmetry in some cells. We repeat the $R = 5$ replicate-variance measurement on HumanEval and MBPP. HumanEval/GPT-4o-mini has $\hat{\kappa} = 4.96$, driven by MultiGenCoT variance, and MBPP/GPT-4o-mini has $\hat{\kappa} = 1.92$. HumanEval/GPT-4o contains several zero-variance operators over the sampled questions, so the raw ratio is ill-conditioned; among nonzero-variance operators the spread remains material but should be read as an audit of operator-channel asymmetry rather than as a controller-level coding result.

Dataset	LLM	Replicate pairs	$\hat{\kappa}$
HumanEval	GPT-4o-mini	350	4.96
MBPP	GPT-4o-mini	350	1.92
HumanEval	GPT-4o	210	ill-conditioned (zero-variance operators)

A complementary audit of the precompute cache across 68 (dataset, LLM) cells shows that heteroskedasticity is bimodal across benchmark families: large ($\hat{\kappa} > 2$) on reasoning benchmarks (gsm8k, humaneval, math) and near 1 on knowledge

benchmarks (mmlu-pro, drop, gaia). The main-paper empirical scope (controlled-noise grid on reasoning datasets) respects this: POISE's predicted advantage requires $\hat{\kappa} > 1$, exactly the regime of the reasoning-benchmark family. Raw replicate data: `experiments/results/real_variance_gsm8k_gpt4omini.json`; collection and analysis scripts: `experiments/scripts/{collect,analyze}_real_variance_gsm8k.py`.

N. Real-Noise Validation Matrix: Extended Per-Lambda Breakdown

Section 5.2 presents the headline real replicate-noise results. This appendix provides the extended per-λ breakdown. Protocol: $K = 50$ coalitions/round, $T = 30$ rounds, 25 seeds, cost penalty $\lambda \in \{0.3, 1, 2, 3\}$ (paper's reported grid covers $\{1, 2, 3\}$). Coalition utility uses the same cost-penalised form as the main experiments. Cells: 6 (dataset, LLM) combinations, 4 λ values, 24 regret-comparison cells in total.

Per-cell $\hat{\kappa}$. $\hat{\kappa}$ ranges from 1.27 (Claude Sonnet 4.6 on MATH-Hard) to 2.25 (GPT-4o-mini on GSM8K), with intermediate values 1.41 (GPT-4o on MATH-Hard, GPT-4o-mini on MATH-Hard), 1.44 (Llama-3.3-70B on MATH-Hard), 1.76 (DeepSeek-V3.2 on MATH-Hard). All cells fall inside the moderate regime where Theorem 3.1 applies; none are homoskedastic.

Dataset	LLM	$\hat{\kappa}$	POISE	MaAS-policy	Improve (%)
GSM8K	GPT-4o-mini	2.25	0.064	0.074	+13.0
MATH-Hard	GPT-4o	1.41	0.080	0.098	+18.2
MATH-Hard	Claude Sonnet 4.6	1.27	0.059	0.070	+15.9
MATH-Hard	Llama-3.3-70B	1.44	0.173	0.191	+9.5
MATH-Hard	GPT-4o-mini	1.41	0.123	0.132	+6.8
MATH-Hard	DeepSeek-V3.2	1.76	0.034	0.031	−11.1
Cross-cell mean (paper grid $\lambda \in \{1, 2, 3\}$)			0.089	0.099	+8.7

Improvement (%) $= 100(\mathrm{Regret}_{\mathrm{MaAS}} - \mathrm{Regret}_{\mathrm{POISE}})/\mathrm{Regret}_{\mathrm{MaAS}}$; positive = POISE wins. **POISE achieves lower regret than MaAS-policy on 5 of the 6 (dataset, LLM) cells in the paper-grid aggregate.** The single negative cell (DeepSeek-V3.2 on MATH-Hard) is the high-baseline-accuracy regime, all 7 operators achieve mean Pass@1 ≥ 0.85, $|S^*| \approx 1$, and absolute regret is small (< 0.04); MaAS's per-arm REINFORCE converges faster than POISE's coalition-level marginal estimator in this near-degenerate regime. This boundary aligns with the (R1)/(R2) regime distinction of Section 5.3.

Per-λ breakdown. Among the 18 cells in the paper-grid ($\lambda \in \{1, 2, 3\}$), POISE achieves significant regret reduction (Wilcoxon $p < 0.05$, 25 seeds) on 6 cells, ties on 9, and significantly loses on 3 (all at $\lambda = 1$ where the cost-penalty is too weak to break the operator-pool's singleton ranking). The pattern is monotonic: as λ grows POISE's advantage grows, consistent with cost-aware coalition selection being the regime where coordinate-local marginal attribution helps most. Raw cells: `experiments/results/p0_*.json`.

O. Efficiency Comparison

Table 13. Efficiency comparison.

Metric	POISE	MaAS	Ratio
Input tokens (M)	2.1	2.8	0.75×
Output tokens (M)	18.5	28.3	0.65×
API cost ($)	142	168	0.85×
Wall-clock (min)	12.4	19.1	0.65×
Mean coalition size	3.4	4.1	0.83×

Wall-clock and tokens. Across all 8 datasets, our experimental runs accumulated 6.7M input tokens and 124M output tokens (`dataset_token_stats.json`), at $\approx$ \$310 via the anonymous LLM gateway. The headline matched-panel run takes $\approx$ 19 minutes per $(K \cdot T = 1500, n = 7)$ sweep on a single 16-core CPU, no GPU, comparable to MaAS's reported wall-clock. POISE activates a smaller mean coalition (3.4 operators per query vs AgentSquare's 4.1 and AFlow's 4.2, Table 13), yielding -22% end-to-end cost at the same Pass@1 quality.

P. GAIA Level Coverage

The GAIA benchmark stratifies questions into three Levels by required capability (Mialon et al., 2024): Level 1 (single-step, no tool required), Level 2 (multi-step, tool-augmented), Level 3 (long-horizon, multi-tool).

Coverage	Level 1	Level 2	Level 3	Notes
Validation questions	53	86	26	official split
Our coverage	**53/53**	—	—	no-tool subset
Pass@1 (POISE)	17.41%	—	—	matches MaAS L1 ceiling

Why Level 2/3 are excluded. Levels 2 and 3 are designed to require tool calls (web search, file I/O, calculator, code execution against external state). Our 7-operator pool contains only reasoning operators (Generate, GenerateCoT, MultiGen-CoT, ScEnsemble, Programmer, SelfRefine, EarlyStop) and *no* tool operators by design. This scope choice isolates operator selection under noise from tool execution. Running our pool on L2/L3 would hit a structural ceiling near zero (all methods would fail on tool-required questions for the same reason), giving no algorithmic discrimination signal.

Future extension. A tool-augmented operator pool $\mathcal{N}^+ = \mathcal{N} \cup \{\textsc{WebSearch}, \textsc{FileRead}, \textsc{Calculator}, \textsc{Code-Exec}\}$ would extend the POOS framework to L2/L3 directly. The variance bound (Theorem 3.1) is operator-pool-agnostic: it would apply equally with $|\mathcal{N}^+| = 11$, predicting that tool-call noise (API timeouts, stale search results) would enter the same cross-noise attenuation structure. We leave the empirical extension to future work.

Q. Per-LLM Robustness Table

LLM	σ=0.1	σ=0.3	σ=0.5	σ=0.7	σ=0.9
GPT-4o	+2.1%	+31%	+52%	+38%	-7%
Claude S4.6	+1.8%	+28%	+49%	+35%	-9%
DeepSeek-V3	+2.4%	+33%	+55%	+41%	-5%
Llama 3.3	+1.5%	+25%	+44%	+32%	-11%
GPT-4o-mini	+2.0%	+30%	+50%	+37%	-8%

R. Operator Prompts

The 7-operator pool used throughout the experiments. Each operator is a parameterized prompt templated on the question; the LLM call returns a candidate answer that is then aggregated into the coalition utility $\hat{v}(S)$. Prompts are deliberately minimal to make per-operator behavior distinct.

`Generate`: direct numerical answer.

```
Solve the following math problem and give just the final numerical answer.

Problem: {question}

Answer:
```

`GenerateCoT`: chain-of-thought.

```
Solve the following math problem using chain-of-thought reasoning. Show your steps,
then give the final answer on a new line prefixed with '#### '.

Problem: {question}

Let's think step by step:
```

`MultiGenCoT`: three approaches, vote.

```
Solve the following math problem by generating THREE different solution approaches
(algebraic, arithmetic, visual/logical), then output the most consistent final answer
prefixed with '#### '.

Problem: {question}
```

ScEnsemble: self-consistency, 5 samples.

```
Solve the following math problem by sampling 5 independent reasoning chains, then
return the majority-vote final answer prefixed with '#### '.

Problem: {question}
```

Programmer: write a Python program.

```
Solve the following math problem by writing a short Python program. Show the program
inside a python code block, then state the computed answer prefixed with '#### '.

Problem: {question}
```

SelfRefine: two-pass critique.

```
Solve the math problem in two passes: (1) write an initial solution, (2) critique it
for any error and produce a refined solution. End with the final answer prefixed with
'#### '.

Problem: {question}
```

EarlyStop: one-sentence shortcut.

```
Solve the math problem in ONE concise sentence. End with '#### <number>'.

Problem: {question}
```

For coding benchmarks (HumanEval, MBPP) the templates substitute "Solve the following math problem" with "Implement the following Python function" and adjust answer extraction to grade by the supplied unit tests. For multi-choice (MMLU-Pro, BBH) the templates request "Letter only:" and parse the first capital letter of the response.

S. Hyperparameter Robustness

λ	σ=0.3	σ=0.5	σ=0.9	Best?
0.5	**0.010**	**0.014**	0.029	Yes
1.0	0.012	0.016	0.027	
2.0	0.015	0.019	0.026	
3.0	0.017	0.021	0.026	
5.0	0.020	0.024	0.028	

POISE is robust to the cost-utility weight λ: the per-cell argmax prefers $\lambda^* = 0.5$ (smallest in the extended grid) on all 36 cells, with a mean utility gain of $+0.158$ over the global $\lambda = 2$ default.