FROM LOCAL ERRORS TO SYSTEM FAILURES: MODELING AND BENCHMARKING ERROR PROPAGATION IN MULTI-AGENT LLM SYSTEMS

Anonymous authors
Paper under double-blind review

ABSTRACT

Errors in LLM-based multi-agent systems (MAS) can propagate through intermediate messages and compromise final answers, yet outcome-based evaluation obscures how agents amplify, attenuate, or correct them. We propose Response-Aware Error Propagation among Agents (REPA), a structural model that connects measured local responses to system-level error propagation. REPA separately estimates input assembly and agent processing across multiple error channels, while theoretical analysis relates local modeling errors to endpoint prediction error. Experiments span three Qwen base models, six MAS workflows, and ten knowledge and reasoning settings in ErrPropBench-10. REPA achieves a mean condition-level Spearman correlation of 0.84 across base models, compared with 0.47 for direct regression. Controlled experiments also demonstrate its utility for targeted recovery. REPA-guided RAG improves accuracy over no retrieval by an average of 19 percentage points with 16% additional tokens. By linking local agent behavior to system susceptibility, REPA provides an interpretable foundation for studying error propagation and developing targeted optimization in MAS.

1 INTRODUCTION

Multi-agent systems (MAS) built on large language models (LLMs) address complex tasks through role specialization and structured communication (Wu et al., 2023; Li et al., 2023; Hong et al., 2024). Within these workflows, downstream agents incorporate intermediate results from upstream agents into their reasoning and decision-making. This dependence, however, also creates pathways through which inaccurate external information or erroneous agent responses can propagate and compromise the final answer (Huang et al., 2025; Jia et al., 2026). Understanding MAS reliability therefore requires examining how individual agents respond to erroneous information and how workflow dependencies link these local responses to the final task outcome.

This local-to-system relationship can not be captured by final-answer correctness alone (Liu et al., 2026; Jia et al., 2026). An agent may reject, revise, or adopt misleading information, and the workflow then determines which downstream agents receive its response (Huang et al., 2025; Shen et al., 2025). Moreover, different patterns of error adoption and correction can produce the same final correctness label, obscuring how erroneous information was handled within the workflow. Therefore, a propagation analysis should examine intermediate inputs and outputs alongside task outcomes, distinguishing changes in message content from changes in task performance.

Recent studies have advanced failure diagnosis and robustness evaluation in MAS (Cemri et al., 2025; Zhang et al., 2025; Jia et al., 2026; Emde et al., 2026b), while others examine how communication topology and repeated interactions shape information propagation and hallucination dynamics (Shen et al., 2025; Jamshidi et al., 2026; Jamshidi, 2026). Some optimization approaches require additional training, whereas post-hoc attribution relies on completed execution traces (Shen et al., 2025; Zhou et al., 2026; Zhang et al., 2025). A complementary question is whether system-level propagation can be predicted from local measurements without fine-tuning the underlying LLMs or observing downstream test outputs. Answering this question requires distinguishing how erroneous information changes an agent's input from how the agent responds across multiple error channels. These local responses must then be composed through downstream processing and aggregation, since local

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.

susceptibility alone does not determine the final system response. The central challenge is therefore to connect separately estimated local responses with independently validated system-level predictions.

To connect measured local responses with system-level predictions, we propose Response-Aware Error Propagation among Agents (REPA), a structural model of error propagation across multiple error channels. Reference and perturbed executions of the same instance are compared under matched conditions, yielding continuous-valued changes at corresponding agent inputs and outputs for separately estimating input-assembly and agent-processing responses. REPA then composes these locally estimated response models along the executed workflow, explicitly accounting for aggregation and finite-round feedback. Once fitted, it predicts system-level propagation from source measurements and execution structure, without fine-tuning the underlying LLMs or observing test-time intermediate states or final outputs. Under the modeling assumptions, theoretical analysis characterizes this composition and relates local modeling errors to endpoint prediction error.

Our evaluation covers six MAS workflows and single-agent baselines using open-source foundation models of varying sizes (Qwen Team, 2024; 2026a). To establish a reliable measurement basis for local response estimation and system-level validation, we first assess observer agreement with independent reference judgments, then evaluate REPA's predictions within fixed conditions. To extend the evaluation across multiple knowledge and reasoning tasks, we construct ErrPropBench-10, covering seven domains and three multi-hop question-answering datasets. Further experiments demonstrate that REPA-derived estimates can guide node selection, targeted prompt restoration, and retrieval-augmented generation (RAG), supporting both agent-level refinement and MAS-level optimization at a reasonable inference cost. Overall, the main contributions are as follows:

- **A local-to-system model of error propagation.** We propose REPA, a structural model of error propagation in MAS based on continuous-valued responses across multiple error channels. REPA characterizes how errors at individual agents are transformed through downstream processing and aggregation. By composing locally estimated agent responses along the executed topology, it predicts system-level susceptibility to error propagation.
- **Local response estimation and compositional analysis.** Motivated by Lipschitz sensitivity analysis, REPA estimates input-construction and agent-processing responses from paired observations. Theoretical analysis derives explicit expressions for topology-dependent error propagation, showing how propagation predictions depend on perturbation placement and output selection, in addition to the communication topology.
- **Guided recovery, enhancement, and a controlled benchmark.** Recovery studies validate prompt refinement and RAG at candidate nodes identified by REPA as effective strategies for improving MAS performance at modest inference cost. We also develop a benchmark for paired evaluation of error propagation across ten knowledge and reasoning settings.

2 Related Work

Failure diagnosis and propagation modeling. MAST and Who&When characterize failures and attribute them from execution traces (Cemri et al., 2025; Zhang et al., 2025), while controlled-fault and system-level evaluations assess robustness (Huang et al., 2025; Jia et al., 2026; Yu et al., 2025; Emde et al., 2026b). Structural studies examine influence-driven topology learning, uncertainty preservation, hallucination dynamics, and preference transmission (Shen et al., 2025; Emde et al., 2026a; Jamshidi et al., 2026; Jamshidi, 2026; Liu, 2026; Niu et al., 2026). These studies establish the importance of interaction structure, although their influence, uncertainty, and preference measures target different quantities from paired semantic-error changes.

Agent refinement and targeted recovery. Self-Refine and Reflexion use feedback to revise outputs, while RAG and Self-RAG supply retrieved evidence (Madaan et al., 2023; Shinn et al., 2023; Lewis et al., 2020; Asai et al., 2024). ResMAS combines learned topology optimization with prompt refinement; MASC detects anomalies through next-execution reconstruction and triggers correction (Zhou et al., 2026; Shen et al., 2026). Self-Refine and Reflexion operate through additional inference rather than parameter updates, whereas the full ResMAS and MASC pipelines include learned components. These approaches improve task outcomes through refinement or correction, but do not directly test whether locally measured responses can predict system-level error propagation.

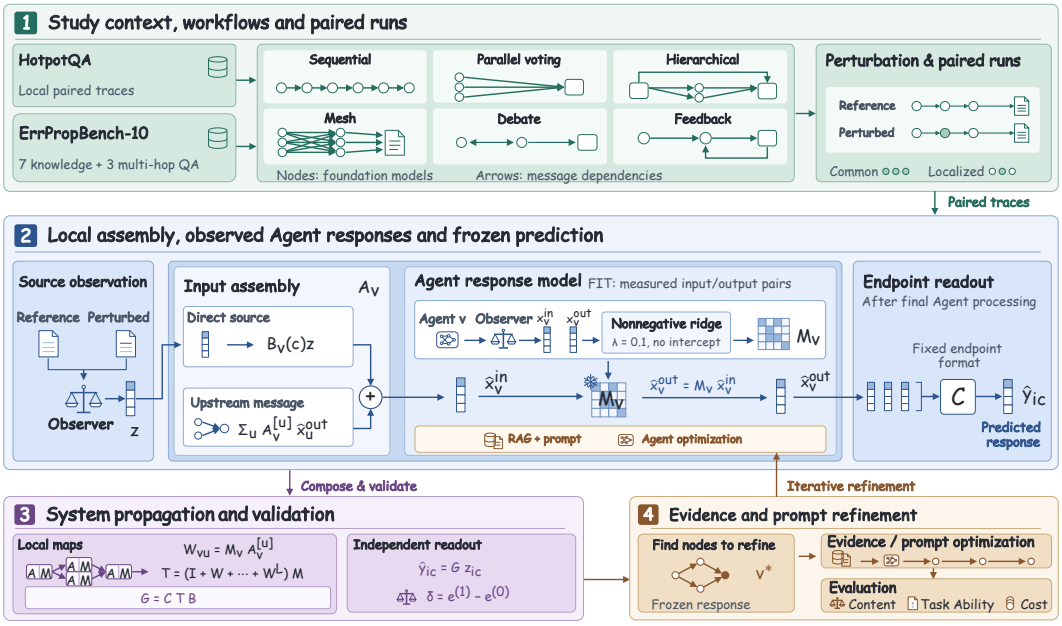


Figure 1: **Overview of REPA and its evaluation workflow.** REPA composes input-assembly and agent-response models along the executed workflow to predict system-level error responses. The diagnostics inform prompt and evidence refinement for improved agent and MAS performance.

REPA connects these research directions through a local-to-system model. Its key distinction is the separate estimation of input-assembly and agent-processing responses across multiple error channels, composed through downstream workflows. Once fitted, the model predicts system responses without fine-tuning the underlying LLMs or observing test-time intermediate states and final outputs. Theoretical analysis links local modeling errors to end-to-end prediction error, while controlled restoration and evidence-placement experiments demonstrate REPA's ability to guide targeted optimization of agents and MAS. Appendix B gives more comparison details.

3 REPA: Formulation and Analysis

As illustrated in Fig. 1, REPA predicts system-level responses to a source perturbation by composing locally estimated input-assembly and agent-response maps. Paired measurements distinguish changes in agent inputs from output responses. The resulting diagnostics inform targeted refinement to improve task performance at reasonable inference cost. Theoretical analysis establishes the finite-execution composition and relates local modeling errors to system-level prediction error. Further details on estimation and theoretical derivations are provided in Appendix A.

3.1 Paired Measurement and Prediction Task

An experimental condition c specifies the model configuration, perturbation and placement, workflow, and observation and readout rules. For each instance i, reference $t_{icv}^{(0),s}$ and perturbed executions $t_{icv}^{(1),s}$ use the same configuration except for the specified intervention, although the reference need not be correct. Each agent invocation forms a separate node v, and finite-round feedback is unfolded in execution order to obtain a directed acyclic graph (DAG) $\mathcal{G}_c$.

Let $s \in \{\text{in}, \text{out}\}$ identify the input or output at node v. A fixed observer ψ uses a paired scoring function $\mathcal{D}_\psi$ to measure content discrepancy across d error channels,

$$\mathbf{x}_{icv}^{s} = \mathcal{D}_\psi\left(t_{icv}^{(0),s}, t_{icv}^{(1),s}\right) \in [0,1]^d. \tag{1}$$

Each coordinate of $\mathbf{x}_{icv}^{s}$ quantifies the discrepancy in one channel. REPA predicts content discrepancies from paired observer measurements without fine-tuning the underlying LLMs. The observer

separately scores error severity in reference and perturbed outputs to distinguish error deterioration from correction, while a task judge evaluates the original final-answer correctness.

The source discrepancy $\mathbf{z}_{ic}$ is the input discrepancy measured at the designated source node, $\mathbf{z}_{ic} := \mathbf{x}^{\text{in}}_{ic,v_{\text{src}}(c)}$. This node is the first invocation under common exposure and the designated recipient under localized exposure. Starting from the measured source discrepancy, REPA recursively predicts downstream discrepancy vectors, using predicted rather than measured parent-output discrepancies. The final prediction is compared with the independently measured output discrepancy.

3.2 Input Assembly and Local Response Estimation

REPA separately models how source perturbations and upstream messages affect an agent's input and how the agent transforms the resulting input discrepancy into an output response. For node v, the assembly vector $\mathbf{f}_{icv} \in \mathbb{R}_+^{h_v}$ collects source and parent-output discrepancies, where h_v is its dimension. Omitting i and c for clarity, input assembly and agent response are modeled as

$$\mathbf{x}^{\text{in}}_v = A_v \mathbf{f}_v + \boldsymbol{\epsilon}^{\text{in}}_v, \qquad \mathbf{x}^{\text{out}}_v = M_v \mathbf{x}^{\text{in}}_v + \boldsymbol{\epsilon}^{\text{out}}_v. \tag{2}$$

The input-assembly map $A_v \in \mathbb{R}_+^{d \times h_v}$ transforms this vector into the d-channel discrepancy. In the agent-response map $M_v \in \mathbb{R}_+^{d \times d}$, rows index output channels, columns index input channels, and off-diagonal entries capture cross-channel predictive associations. The residuals account for variation not explained by assembly and processing (e.g., measurement error). Lipschitz-type sensitivity analysis and Taylor approximation motivate the local linear representation under suitable smoothness assumptions (Khalil, 2002; Nocedal & Wright, 2006). In REPA, the response maps are estimated from paired observations rather than computed as exact Jacobians or certified sensitivity bounds.

The two maps are estimated separately from observed local states on the fitting set:

$$\begin{aligned} A_v &= \arg\min_{A \geq 0} \sum_{(i,c) \in \mathcal{F}_v} \|\mathbf{x}^{\text{in}}_{icv} - A\mathbf{f}_{icv}\|_2^2 + \lambda \|A\|_F^2, \\ M_v &= \arg\min_{M \geq 0} \sum_{(i,c) \in \mathcal{F}_v} \|\mathbf{x}^{\text{out}}_{icv} - M\mathbf{x}^{\text{in}}_{icv}\|_2^2 + \lambda \|M\|_F^2. \end{aligned} \tag{3}$$

The fitting set $\mathcal{F}_v$ contains paired observations across perturbations and placements for a fixed workflow. The response maps are estimated using nonnegative ridge regression, with regularization to stabilize coefficient estimation (Hoerl & Kennard, 1970). Once the maps are frozen, prediction proceeds in workflow order, replacing measured parent outputs in $\mathbf{f}_v$ with their predicted values:

$$\widehat{\mathbf{x}}^{\text{in}}_v = A_v \widehat{\mathbf{f}}_v, \qquad \widehat{\mathbf{x}}^{\text{out}}_v = M_v \widehat{\mathbf{x}}^{\text{in}}_v. \tag{4}$$

3.3 Execution-Aware Composition and Readout

To express the node-wise predictions at the system level, stack the output-discrepancy vectors of all N invocations into $\mathbf{x}^{\text{out}} \in \mathbb{R}^{Nd}$. The block-diagonal matrix $M = \text{diag}(M_1, \ldots, M_N)$ collects their agent-response maps. For the specified perturbation placement, the matrix $B \in \mathbb{R}_+^{Nd \times p}$ collects the active source-input blocks of the assembly maps A_v. For each executed connection $u \to v$, let $A_v^{[u]}$ denote the block that maps node u's output discrepancy into v's input discrepancy. The corresponding propagation block is $W_{vu} = M_v A_v^{[u]}$, with $W_{vu} = 0$ when no such connection exists. Thus, $MB\mathbf{z}$ represents the direct source contribution after agent processing, while W carries parent-output discrepancies through subsequent assembly and processing. The recursive prediction can then be written as

$$\widehat{\mathbf{x}}^{\text{out}} = MB\mathbf{z} + W\widehat{\mathbf{x}}^{\text{out}}, \qquad \widehat{\mathbf{y}} = C\widehat{\mathbf{x}}^{\text{out}}. \tag{5}$$

The execution order makes W block-lower-triangular, enabling sequential prediction. C selects the final output coordinates after the corresponding agent-response models.

Unrolling the recursion maps source perturbations to final output discrepancies. The following proposition shows how this mapping combines the fitted local responses with the execution structure. For a fixed invocation DAG containing L edges,

$$S = \sum_{k=0}^{L} W^k, \qquad T = SM, \qquad G = CTB, \qquad \widehat{\mathbf{y}} = G\mathbf{z}. \tag{6}$$

Each term W^k collects contributions along executed paths of length k. Together with the initial processing factor M, these contributions form T, while $G = CTB$ incorporates source placement and final readout. The system-level map G is constructed by composing the fitted A_v and M_v along the executed workflow, without additional global fitting. This finite-execution identity requires no infinite-horizon stability assumption and does not imply that text generation itself is linear.

To summarize system susceptibility under condition c, we aggregate the observed and predicted response magnitudes over the same set of eligible samples $\mathcal{I}_c$,

$$s_c = \frac{\sum_{i \in \mathcal{I}_c} \|\mathbf{y}_{ic}\|_2}{Z_c}, \qquad \widehat{s}_c = \frac{\sum_{i \in \mathcal{I}_c} \|\widehat{\mathbf{y}}_{ic}\|_2}{Z_c}, \qquad \mathbf{y}_{ic} = C_c \mathbf{x}_{ic}^{\text{out}}. \tag{7}$$

Here s_c and $\widehat{s}_c$ denote measured and predicted susceptibility, respectively. The shared denominator $Z_c > 0$ normalizes for source-perturbation magnitude according to the fixed metric. These scores quantify the response to injected errors rather than its probability of task failure.

3.4 Diagnostics and Targeted Refinement

Suboptimal agent configurations can impair local responses and downstream task performance. REPA distinguishes local deterioration from changes inherited from upstream agents to identify nodes requiring targeted refinement. Let Δh_v denote the observer-scored decline in agent function relative to a reference execution. The diagnostic score and node-selection rule are

$$r_v = \left[\Delta h_v - \sum_{u \in \text{pa}(v)} w_{vu}^{\text{inv}} \Delta h_u \right]_+, \qquad v^\star = \arg\max_{v \in \mathcal{V}_c} r_v. \tag{8}$$

Here w_{vu}^{inv} denotes an entry of the inverse diagnostic operator W_{inv}. The maps A_v, M_v, and W_{inv} are fitted separately in their measurement spaces using the same nonnegative ridge estimation procedure. r_v quantifies local deterioration beyond that explained by upstream changes, prioritizing agents with greater unresolved deficiencies for targeted refinement. The candidate set $\mathcal{V}_c$ includes nodes with valid measurements and observed deterioration. RAG introduces retrieved evidence at the node identified by REPA, providing additional information for further refinement. The evidence is retrieved using the original task query and supplements the agent's input without altering the remaining configurations. Subsequent experiments demonstrate that REPA supports effective node selection, agent refinement and evidence introduction, improving MAS accuracy at reasonable inference cost.

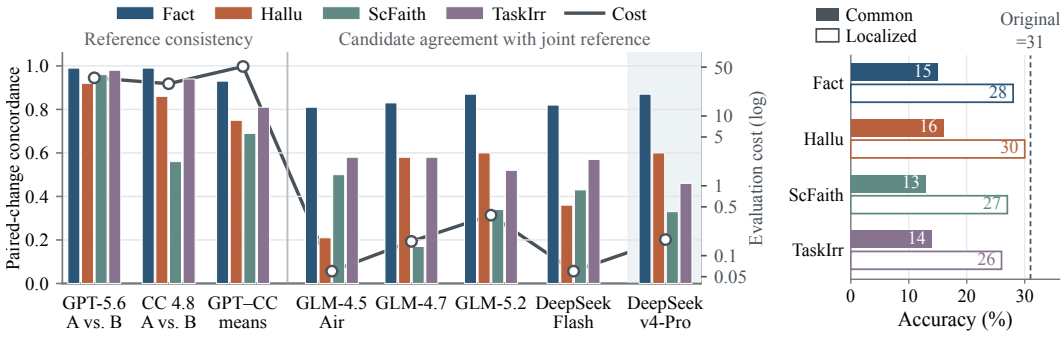
4 Experiments

Our experiments evaluate how multi-channel local errors affect MAS performance and whether REPA can predict system-level susceptibility by composing locally estimated agent responses. Together, these studies connect measurement validation, controlled error propagation, structural prediction, and MAS recovery evaluated through final-answer accuracy and inference cost to inform the development of more trustworthy LLM-based MAS, with ErrPropBench-10 providing broader task coverage. Further protocols, prompts, analyses, and results are provided in Appendices C and D.

4.1 Experimental Setup

Datasets and benchmark construction. The main propagation, prediction, and recovery studies use HotpotQA (Yang et al., 2018), a Wikipedia-based multi-hop question-answering dataset. Its questions require reasoning across multiple documents, with sentence-level supporting-fact annotations identifying the evidence underlying each answer. This structure provides a useful setting for examining how errors in intermediate information affect subsequent reasoning and final answers.

With HotpotQA as one of its three multi-hop question-answering settings, ErrPropBench-10 also includes examples from 2WikiMultiHopQA and MuSiQue-Ans (Ho et al., 2020; Trivedi et al., 2022). The remaining seven domains contain questions constructed from source fact sets covering the Roman Empire (Berger et al., 2016), photosynthesis (Johnson, 2016; Cardona et al., 2018), DNA (National Human Genome Research Institute, 2020), black holes (Reddy, 2020), CRISPR (Zhou & Yao, 2023), blockchain (Yaga et al., 2018), and machine learning (Zhang et al., 2023; Atherton et al., 2023). Each



(a) Observer reliability and cost (b) Error exposure

Figure 2: **Observer reliability and error exposure performance.** (a) Paired-change concordance and estimated API cost (USD) per 100 question pairs; CC denotes Claude. (b) Accuracy under common and localized exposure across four-channel errors.

domain contains 100 instances for method fitting, 20 for validation, and 30 for testing, yielding 1,500 question–answer pairs with supporting contexts overall. No source fact set or complete evidence set is shared across the three splits. Other details appear in Appendix C.1.

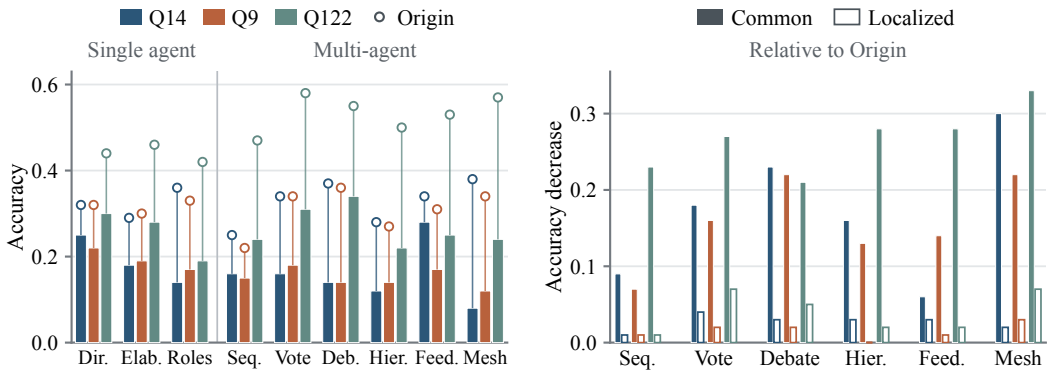
Models and implementation. We use Qwen2.5-14B-Instruct (Q14), Qwen3.5-9B (Q9), and Qwen3.5-122B-A10B (Q122) as the backbone LLMs for instantiating the evaluated MAS workflows (Qwen Team, 2024; 2026a;b). The evaluation covers six MAS workflows (i.e., sequential, parallel voting, debate, hierarchical, finite-round feedback, and mesh) and three single-agent baselines, differing in role prompts, invocation patterns, and communication topologies. Q14 runs on NVIDIA V100 GPUs and the other LLMs on NVIDIA PRO 6000 GPUs.

Measurement, fitting, and evaluation. We compare reference and perturbed executions of the same instance with the model configuration, workflow, and experimental conditions matched. Common exposure supplies the same perturbation to all agents receiving the shared input, whereas localized exposure injects it at a single node. REPA considers factual errors (Fact), hallucination (Hallu), source faithfulness (ScFaith), and task irrelevance (TaskIrr), with Fact and Hallu examined in greater detail in this work. Paired fitting-set observations estimate local responses $\widehat{A}_v$ and $\widehat{M}_v$ using nonnegative ridge regression with $\lambda = 0.1$. These matrices jointly model input assembly and agent processing along the executed workflow, linking local responses to system-level predictions. Fitting, calibration, and testing use disjoint data sets, and the main results are reported on the test sets with the fitted matrices held fixed.

4.2 Observer Reliability and Measurement Validity

Observer accuracy is fundamental to a reliable analysis of error propagation. Classical estimation theory explicitly accounts for measurement uncertainty (Kalman, 1960), while input–output analysis characterizes how disturbances propagate through interconnected systems (Khalil, 2002). In REPA, the observer quantifies local agent responses and system-level changes, making measurement accuracy essential to model fitting and evaluation. We therefore evaluate whether the observer correctly detects increases in error and accurately estimates changes in error scores.

To keep evaluation costs manageable, we use models from the open-source GLM (Z.ai, 2026) and DeepSeek (DeepSeek-AI, 2026) families as candidate observers. Following the LLM-as-a-judge paradigm (Zheng et al., 2023), the proprietary reference models GPT-5.6 (OpenAI, 2026) and Claude Opus 4.8 (Anthropic, 2026) independently evaluate the same paired outputs using evidence-first (A) and role-and-change-first (B) prompts. Reference reliability is assessed through agreement between the two prompts within each model and between the prompt-averaged scores of the two models. The two model means are subsequently averaged to form the joint model-based reference. Candidate-observer reliability is evaluated using CCC (Lin, 1989), which measures agreement with the joint reference in both pattern and magnitude. Evaluation details are provided in the Appendix.



(a) System accuracy

(b) Accuracy decrease

Figure 3: **System performance under error injection.** (a) Origin accuracy (circles) and injected accuracy (bars). (b) Accuracy decrease from Origin under common and localized exposure.

As shown in Fig. 2a, although the evidence-first prompt A and the role-and-change-first prompt B emphasize different aspects of evaluation, they yield high within-model agreement for Fact, Hallu, and TaskIrr (0.99, 0.89, and 0.96), while the two reference-model means remain concordant on the same channels (0.93, 0.75, and 0.81). This two-level agreement reduces the likelihood that the reference measurements are artifacts of a particular prompt formulation or reference model, supporting their use as stable comparison targets for candidate observers. By contrast, ScFaith shows less consistency across two prompts and between the GPT and Claude means, suggesting that these judgments are more sensitive to both prompt framing and model-specific interpretation. Against this joint reference, DeepSeek-v4-Pro (DS) and GLM-5.2 show comparatively high agreement on two channels, with CCCs of 0.87 for Fact and 0.60 for Hallu. Considering both agreement performance and evaluation cost, this paper selects DS as the primary observer and focuses the subsequent analysis on Fact and Hallu. ScFaith and TaskIrr are retained as auxiliary diagnostics because they were not independently perturbed and show weaker consistent candidate-level agreement.

4.3 System Performance under Controlled Error Injection

Following the observer audit, we examine the task-level consequences of error injection on HotpotQA (Yang et al., 2018), covering three single-agent configurations and six MAS workflows with Q14, Q9, and Q122. The factual-error experiments compare the unperturbed reference (Origin) with common exposure through the shared task input and localized exposure at one agent input. We additionally compare common and localized exposure across four error families: Fact, Hallu, ScFaith, and TaskIrr. We report final-answer accuracy and its decrease from the corresponding Origin.

Q122 achieves the highest Origin accuracy in every configuration, yet its MAS accuracy decreases by 21–33% under common exposure (Fig. 3a). Across all combinations, common exposure yields lower accuracy than localized exposure (Fig. 3b). Across models, mean accuracy decreases over the six workflows are 16–27% under common exposure, versus 2–4% under localized exposure. Higher Origin accuracy does not necessarily indicate greater robustness, as Q14's mesh falls from 0.38 to 0.08 under common exposure, whereas feedback declines from 0.34 to 0.28. Together, these observations show that task performance varies with both the workflow configuration and the exposure protocol, motivating the subsequent analysis of how local agent responses combine into system-level changes.

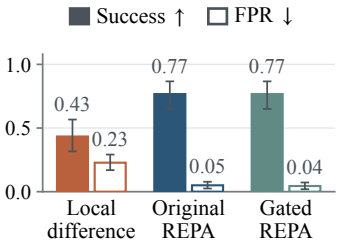
4.4 Predicting System Responses and Exploring Fault Localization

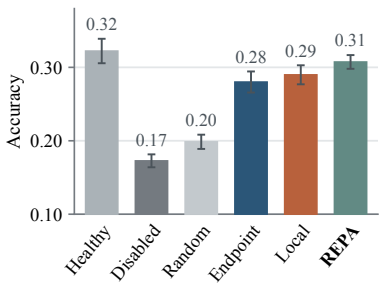
To link local agent behavior to system responses, we need to test whether local measurements predict independently measured endpoint responses on held-out instances. REPA composes fitted maps of input assembly and agent processing along the execution graph, using only the measured source discrepancy at evaluation. We compare REPA with three estimators (scalar, diagonal and direct-matrix) that map source discrepancies to endpoint responses. Predictions are evaluated using Spearman correlation between predicted and observed condition gains and mean absolute error

Table 1: **Agreement between predicted and measured system responses.** Higher ρ and lower MAE are better.

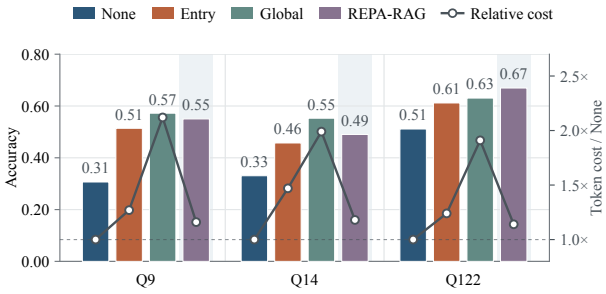
	Spearman ρ			MAE		
Estimator	Q14	Q9	Q122	Q14	Q9	Q122
Scalar	0.25	0.04	0.27	0.29	0.27	0.27
Diagonal	0.44	0.09	0.28	0.28	0.26	0.26
Direct matrix	0.58	0.46	0.36	0.26	0.25	0.25
REPA	0.84	0.85	0.83	0.24	0.23	0.22

Figure 4: **Fault-source localization.** 95% CIs.





(a) Agent refinement.



(b) MAS refinement and relative cost.

Figure 5: **REPA-guided refinement.** (a) Recovery accuracy, with reported bounds. (b) RAG accuracy as a proportion and token cost relative to None.

(MAE) between predicted and observed endpoint vectors. To assess whether REPA can distinguish a fault source from its downstream effects, we compare local differences, original REPA and gated REPA on new instances with a predefined node disabled. Performance is evaluated using localization success rate and false positive rate (FPR). Appendices C.3 and D.2 provide more details.

As shown in Table 1, REPA achieves correlations of 0.84, 0.85 and 0.83 on Q14, Q9 and Q122, respectively, and the lowest MAE among the four estimators. The results show that composing local responses provides predictive information beyond a single source-to-endpoint mapping. In the inverse test, REPA achieves a localization success rate of 0.77, compared with 0.43 for local differences, while lowering the FPR from 0.23 to 0.05 (Fig. 4). This exploratory finding motivates the recovery experiments in Sec. 4.5. These results support REPA's central idea that local measurements of agent inputs and responses can be composed through execution structure to predict system responses.

4.5 Diagnosis-Guided Optimization of MAS

We test whether REPA guides useful optimization in MAS, separately from prediction and fault localization. We disable selected agent functions in six Q14 workflows. Each repair policy can restore the original role prompt at no more than one node, allowing us to compare where it chooses to intervene under the same repair budget. The comparators select a random node, the endpoint or the node with the largest local change. We further evaluate evidence-based optimization through RAG (Lewis et al., 2020), using BM25 retrieval (Robertson & Zaragoza, 2009) followed by MMR reranking (Carbonell & Goldstein, 1998). We compare four correction methods on clean inputs, with None providing no evidence, Entry supplying evidence to root agents, Global supplying it to every node, and REPA-RAG supplying it to one REPA-selected node. Appendix C.4 gives more details.

With the same one-node restoration allowance, REPA brings accuracy closer to Healthy performance than the other repair policies (Fig. 5a). The displayed midpoint rises from 0.17 under disablement to 0.31, compared with 0.32 for Healthy. Its advantage over local-difference selection supports accounting for propagated parent effects when selecting a repair location. These results support

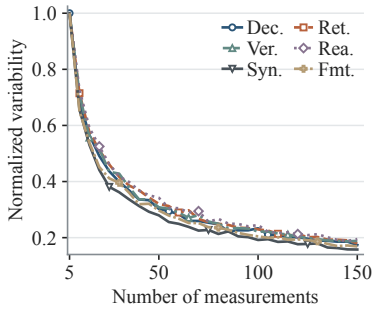


Figure 6: **Measurement stability.** Range width relative to five measurements.

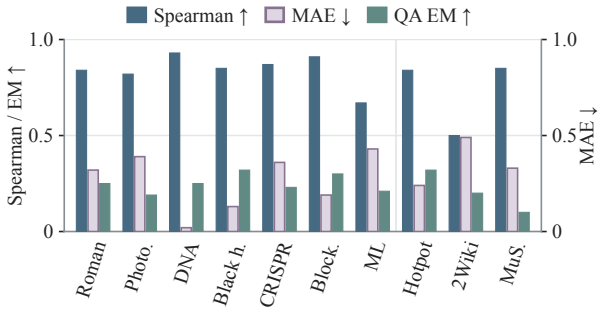


Figure 7: **Cross-domain prediction.** Reported rank correlation and MAE with direct-QA EM as a fraction.

REPA-guided optimization at the agent level, improving final-answer accuracy through targeted prompt restoration under a fixed repair budget.

As shown in Fig. 5b, REPA-RAG uses fewer tokens than Entry while achieving higher accuracy on all three models without changing the underlying workflow structure. REPA-RAG uses 0.55 to 0.60 of Global's token cost, with accuracy 0.02 lower on Q9 and 0.04 higher on Q122. Although Q14 shows a larger accuracy loss, selective delivery retains much of Global's accuracy on Q9 and Q122 at substantially lower token cost. Together with prompt restoration, they support REPA-guided optimization at both the agent and system levels, balancing final-answer accuracy and inference cost.

4.6 Sensitivity Analysis and Extension to ErrPropBench-10

We examine whether estimates for the six Q14 agents become more stable as the number of measurements increases. For each agent, we repeatedly sample 5 to 150 records from the HotpotQA dataset and compare the normalized variability of the resulting estimates. Beyond measurement count, we also examine how propagation prediction varies across task domains and relates to the base model's task performance. Across the ten ErrPropBench-10 domains, we report Spearman rank correlation and MAE for propagation predictions. For context, we also evaluate Q14 on 100 questions per domain with their original evidence, using normalized exact match (EM) to measure direct question-answering performance. Appendices C.5 and C.6 give more details.

In Fig. 6, normalized range widths decrease from 1.00 at five measurements to 0.19 to 0.24 at 100 measurements across all six roles. Increasing the count to 150 further reduces the widths to 0.16 to 0.19 but requires 50 additional measurements. Balancing stability and collection cost, we use 100 samples as a practical default for the main experiments. REPA estimates error propagation across all ten domains, with reported Spearman correlations ranging from 0.50 to 0.93 and MAE values from 0.02 to 0.49 (Fig. 7). The mean rank correlation of 0.81 indicates overall agreement with measured propagation rankings, while MAE shows that accurate ordering does not ensure accurate magnitudes. Domains with higher direct-QA EM tend to have lower prediction error, with Pearson $r = -0.55$. Taken together, the experiments support REPA as an effective framework for predicting system-level error propagation from local agent responses and guiding targeted optimization of MAS.

5 Conclusion

We propose REPA, a structural model that composes measured agent-level error responses to predict system-level propagation. Experiments across models, workflows, and settings support its ability to capture relative propagation strength, while controlled recovery studies illustrate its potential to guide targeted MAS optimization. REPA's strength lies in connecting local behavior with system susceptibility through explicit, interpretable structural composition. It provides the community with a framework for measuring, predicting, and investigating error propagation, establishing a foundation for future research into the reliability and design of multi-agent systems.

AI USE STATEMENT

We used generative AI tools to assist with idea implementation, literature search, and manuscript drafting. We did not use generative AI to draft all sections of the paper. AI assistance was also used in parts of benchmark construction and perturbation generation, and LLM judges were used for automated evaluation, as detailed in the experimental sections and appendices. We reviewed all AI-assisted work and checked AI-generated research ideas for potential plagiarism through a manual literature survey. AI-generated code was verified and tested by the authors, and all manuscript text was reviewed and finalized by two authors. We take full responsibility for the final content of this work, including all text, claims, and artifacts produced with AI assistance.

REFERENCES

Anthropic. Introducing Claude Opus 4.8. Official model announcement, May 2026. URL https://www.anthropic.com/news/claude-opus-4-8. Published May 28, 2026. Accessed September 18, 2026.

Akari Asai, Zeqi Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. Self-RAG: Learning to retrieve, generate, and critique through self-reflection. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/25f7be9694d7b32d5cc670927b8091e1-Abstract-Conference.html.

Daniel Atherton, Reva Schwartz, Peter C. Fontana, and Patrick Hall. The language of trustworthy AI: An in-depth glossary of terms. NIST Artificial Intelligence 100-3, National Institute of Standards and Technology, 2023. URL https://nvlpubs.nist.gov/nistpubs/ai/NIST.AI.100-3.pdf.

Eugene Berger, George L. Israel, Charlotte Miller, Brian Parkinson, Andrew Reeves, and Nadejda Williams. *World History: Cultures, States, and Societies to 1500*. University of North Georgia Press, 2016. ISBN 9781940771106. URL https://oer.galileo.usg.edu/history-textbooks/2/.

Jaime Carbonell and Jade Goldstein. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 335–336. ACM, 1998. doi: 10.1145/290941.291025. URL https://doi.org/10.1145/290941.291025.

Tanai Cardona, Shengxi Shao, and Peter J. Nixon. Enhancing photosynthesis in plants: The light reactions. *Essays in Biochemistry*, 62(1):85–94, 2018. doi: 10.1042/EBC20170015. URL https://pmc.ncbi.nlm.nih.gov/articles/PMC5897789/.

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 Gonzalez, and Ion Stoica. Why do multi-agent LLM systems fail? In *Advances in Neural Information Processing Systems*, volume 38. Curran Associates, Inc., 2025. doi: 10.52202/085713-4082. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/b1041e52d3be19f0a9bc491657488e4a-Abstract-Datasets_and_Benchmarks_Track.html. Datasets and Benchmarks Track.

DeepSeek-AI. DeepSeek-V4: Towards highly efficient million-token context intelligence, 2026. URL https://arxiv.org/abs/2606.19348.

Cornelius Emde, Anmol Goel, Sangdoo Yun, Seong Joon Oh, and Martin Gubri. Lost in communication: Uncertainty propagation in multi-agent systems. In *ICML 2026 Workshop on Statistical Frameworks for Uncertainty in Agentic Systems*, 2026a. URL https://gubri.eu/publication/lost_in_communication/. Non-archival workshop paper.

Cornelius Emde, Alexander Rubinstein, Anmol Goel, Ahmed Heakl, Sangdoo Yun, Seong Joon Oh, and Martin Gubri. MASEval: Extending multi-agent evaluation from models to systems. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume*

3: System Demonstrations), pp. 345–356. Association for Computational Linguistics, 2026b. doi: 10.18653/v1/2026.acl-demo.34. URL https://aclanthology.org/2026.acl-demo.34/.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. Constructing a multi-hop QA dataset for comprehensive evaluation of reasoning steps. In *Proceedings of the 28th International Conference on Computational Linguistics*, pp. 6609–6625. International Committee on Computational Linguistics, 2020. doi: 10.18653/v1/2020.coling-main.580. URL https://aclanthology.org/2020.coling-main.580/.

Arthur E. Hoerl and Robert W. Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67, 1970. doi: 10.1080/00401706.1970.10488634. URL https://doi.org/10.1080/00401706.1970.10488634.

Sirui Hong, Mingchen Zhuge, Jonathan Chen, Xiawu Zheng, Yuheng Cheng, Ceyao Zhang, Jinlin Wang, Zili Wang, Steven Ka Shing Yau, Zijuan Lin, Liyang Zhou, Chenyu Ran, Lingfeng Xiao, Chenglin Wu, and Jürgen Schmidhuber. MetaGPT: Meta programming for a multi-agent collaborative framework. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/6507b115562bb0a305f1958ccc87355a-Abstract-Conference.html.

Roger A. Horn and Charles R. Johnson. *Matrix Analysis*. Cambridge University Press, second edition, 2013. doi: 10.1017/CBO9781139020411. URL https://doi.org/10.1017/CBO9781139020411.

Jen-Tse Huang, Jiaxu Zhou, Tailin Jin, Xuhui Zhou, Zixi Chen, Wenxuan Wang, Youliang Yuan, Michael Lyu, and Maarten Sap. On the resilience of LLM-based multi-agent collaboration with faulty agents. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 26202–26226. PMLR, 2025. URL https://proceedings.mlr.press/v267/huang25ay.html.

Saeid Jamshidi. Collective hallucination in multi-agent LLMs: Modeling and defense, 2026. URL https://arxiv.org/abs/2606.07941v2. Preprint, version 2, 18 August 2026.

Saeid Jamshidi, Arghavan Moradi Dakhel, Kawser Wazed Nafi, and Foutse Khomh. Hallucination cascade: Analyzing error propagation in multi-agent LLM systems, 2026. URL https://arxiv.org/abs/2606.07937v1. Preprint, version 1.

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

Matthew P. Johnson. Photosynthesis. *Essays in Biochemistry*, 60(3):255–273, 2016. doi: 10.1042/EBC20160016. URL https://pmc.ncbi.nlm.nih.gov/articles/PMC5264509/.

R. E. Kalman. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82(1):35–45, 1960. doi: 10.1115/1.3662552. URL https://doi.org/10.1115/1.3662552.

Hassan K. Khalil. *Nonlinear Systems*. Prentice Hall, 3 edition, 2002. ISBN 0-13-067389-7. URL https://www.egr.msu.edu/~khalil/NonlinearSystems/index.html.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems*, volume 33, pp. 9459–9474. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/hash/6b493230205f780e1bc26945df7481e5-Abstract.html.

Guohao Li, Hasan Abed Al Kader Hammoud, Hani Itani, Dmitrii Khizbullin, and Bernard Ghanem. CAMEL: Communicative agents for "mind" exploration of large language model society. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://proceedings.neurips.cc/paper/2023/hash/a3621ee907def47c1b952ade25c67698-Abstract-Conference.html.

L. I. Lin. A concordance correlation coefficient to evaluate reproducibility. *Biometrics*, 45(1):255–268, 1989. doi: 10.2307/2532051. URL https://pubmed.ncbi.nlm.nih.gov/2720055/.

Yiren Liu, Yun Huang, and Shunan Guo. Mosaic: A multi-level framework for observing and evaluating performance of multi-agent systems. In *Proceedings of the Extended Abstracts of the 2026 CHI Conference on Human Factors in Computing Systems*, CHI EA '26, pp. 1–9. Association for Computing Machinery, 2026. doi: 10.1145/3772363.3798830. URL https://doi.org/10.1145/3772363.3798830.

Zewen Liu. Contagion networks: Evaluator preference propagation in multi-agent LLM systems, 2026. URL https://arxiv.org/abs/2606.20493v2. Preprint, version 2, 26 June 2026.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. Self-refine: Iterative refinement with self-feedback. In *Advances in Neural Information Processing Systems*, volume 36, pp. 46534–46594. Curran Associates, Inc., 2023. doi: 10.52202/075280-2019. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/91edff07232fb1b55a505a9e9f6c0ff3-Abstract-Conference.html.

National Human Genome Research Institute. Deoxyribonucleic acid (DNA) fact sheet. National Institutes of Health, 2020. URL https://www.genome.gov/about-genomics/fact-sheets/Deoxyribonucleic-Acid-Fact-Sheet. Updated August 24, 2020.

Ruiwu Niu, Xincheng Shu, and Ying Zhao. Reliability–contagion feasibility in LLM multi-agent networks, 2026. URL https://arxiv.org/abs/2607.21912v1. Preprint, version 1, 24 July 2026.

Jorge Nocedal and Stephen J. Wright. *Numerical Optimization*. Springer, 2 edition, 2006. doi: 10.1007/978-0-387-40065-5. URL https://doi.org/10.1007/978-0-387-40065-5.

OpenAI. GPT-5.6: Frontier intelligence that scales with your ambition. Official model announcement, July 2026. URL https://openai.com/index/gpt-5-6/. Published July 9, 2026. Accessed September 18, 2026.

Qwen Team. Qwen2.5-14B-Instruct: Official model card, 2024. URL https://huggingface.co/Qwen/Qwen2.5-14B-Instruct.

Qwen Team. Qwen3.5-9B: Official model card, 2026a. URL https://huggingface.co/Qwen/Qwen3.5-9B.

Qwen Team. Qwen3.5-122B-A10B: Official model card, 2026b. URL https://huggingface.co/Qwen/Qwen3.5-122B-A10B.

Francis Reddy. What are black holes? NASA, 2020. URL https://www.nasa.gov/universe/what-are-black-holes/. Published September 8, 2020; updated July 26, 2023.

Stephen E. Robertson and Hugo Zaragoza. The probabilistic relevance framework: BM25 and beyond. *Foundations and Trends in Information Retrieval*, 3(4):333–389, 2009. doi: 10.1561/1500000019. URL https://doi.org/10.1561/1500000019.

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 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 12347–12361, Suzhou, China, November 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.emnlp-main.623. URL https://aclanthology.org/2025.emnlp-main.623/.

Xu Shen, Qi Zhang, Song Wang, Zhen Tan, Xinyu Zhao, Laura Yao, Vaishnav Tadiparthi, Hossein Nourkhiz Mahjoub, Ehsan Moradi Pari, Kwonjoon Lee, and Tianlong Chen. Metacognitive self-correction for multi-agent system via prototype-guided next-execution reconstruction. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 23320–23337, San Diego, California, USA, 2026. Association for Computational Linguistics. URL https://aclanthology.org/2026.findings-acl.1168.pdf.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 36, pp. 8634–8652. Curran Associates, Inc., 2023. doi: 10.52202/075280-0377. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/1b44b878bb782e6954cd888628510e90-Abstract-Conference.html.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. MuSiQue: Multihop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics*, 10:539–554, 2022. doi: 10.1162/tacl_a_00475. URL https://aclanthology.org/2022.tacl-1.31/.

Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang Zhu, Li Jiang, Xiaoyun Zhang, Shaokun Zhang, Jiale Liu, Ahmed Hassan Awadallah, Ryen W. White, Doug Burger, and Chi Wang. AutoGen: Enabling next-gen LLM applications via multi-agent conversation, 2023. URL https://arxiv.org/abs/2308.08155.

Dylan Yaga, Peter Mell, Nik Roby, and Karen Scarfone. Blockchain technology overview. NIST Interagency Report 8202, National Institute of Standards and Technology, 2018. URL https://csrc.nist.gov/pubs/ir/8202/final.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 2369–2380. Association for Computational Linguistics, 2018. doi: 10.18653/v1/D18-1259. URL https://aclanthology.org/D18-1259/.

Miao Yu, Shilong Wang, Guibin Zhang, Junyuan Mao, Chenlong Yin, Qijiong Liu, Kun Wang, Qingsong Wen, and Yang Wang. NetSafe: Exploring the topological safety of multi-agent system. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 2905–2938. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.findings-acl.150. URL https://aclanthology.org/2025.findings-acl.150/.

Z.ai. GLM-5.2. Hugging Face model card, 2026. URL https://huggingface.co/zai-org/GLM-5.2. MIT license. Accessed September 18, 2026.

Aston Zhang, Zachary C. Lipton, Mu Li, and Alexander J. Smola. *Dive into Deep Learning*. Cambridge University Press, 2023. URL https://d2l.ai/.

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. PMLR, 2025. URL https://proceedings.mlr.press/v267/zhang25cq.html.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena. In *Advances in Neural Information Processing Systems*, volume 36, pp. 46595–46623, 2023. doi: 10.52202/075280-2020. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/91f18a1287b398d378ef22505bf41832-Abstract-Datasets_and_Benchmarks.html.

Lifang Zhou and Shaohua Yao. Recent advances in therapeutic CRISPR-Cas9 genome editing: Mechanisms and applications. *Molecular Biomedicine*, 4(1):10, 2023. doi: 10.1186/s43556-023-00115-5. URL https://link.springer.com/article/10.1186/s43556-023-00115-5.

Zhilun Zhou, Zihan Liu, Jiahe Liu, Qingyu Shao, Yihan Wang, Kun Shao, Depeng Jin, and Fengli Xu. ResMAS: Resilience optimization in LLM-based multi-agent systems. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 35176–35184. AAAI Press, 2026. doi: 10.1609/aaai.v40i41.40824. URL https://ojs.aaai.org/index.php/AAAI/article/view/40824.

A Estimation and Derivations

A.1 Notation and Measurement

Table S1 collects the dimensions used in the main manuscript. Hats distinguish predicted states from observed states.

Table S1: **Notation.**

Symbol	Dimension	Meaning
N, L	Scalars	Invocation count and longest executed path in edges
d, p, q	Scalars	Local, source and readout dimensions
$\mathbf{z}, \mathbf{f}_v$	$p,\ h_v$	Source discrepancy and assembly features
$\mathbf{x}_v^{\mathrm{in/out}}$	d	Observed local discrepancy
A_v, M_v	$d \times h_v,\ d \times d$	Input assembly and agent processing
B, P	$Nd \times p,\ Nd \times Nd$	Stacked source and parent-message maps
M, W	$Nd \times Nd$	Block-diagonal processing and $W = MP$
S, T	$Nd \times Nd$	Finite propagation sum and $T = SM$
C, G	$q \times Nd,\ q \times p$	Output selection and $G = CTB$
$\boldsymbol{\epsilon}^{\mathrm{in/out}}$	Nd	Stacked local residuals
$\boldsymbol{\beta}^{\mathrm{in/out}}$	Nd	Assumed componentwise residual bounds
$\mathcal{I}_c, Z_c$	Set, scalar	Eligible question pairs and shared positive normalizer

A.2 Local Estimation and Prediction Algorithm

Source and assembly features. For the historical trace-composed predictor,

$$\mathbf{z}_{ic} = \mathbf{x}^{\mathrm{in}}_{ic,v_{\mathrm{inj}}(c)}, \tag{9}$$

where $v_{\mathrm{inj}}(c)$ is the first invocation for common exposure and the designated receiving invocation for localized exposure. For ordered parents $u_1, \ldots, u_k$, write

$$\begin{aligned} \mathbf{f}_{icv} &= \mathrm{col}\big(m_v^{\mathrm{com}}(c)\mathbf{z}_{ic},\, m_v^{\mathrm{loc}}(c)\mathbf{z}_{ic},\, \mathbf{x}^{\mathrm{out}}_{icu_1}, \ldots, \mathbf{x}^{\mathrm{out}}_{icu_k}\big), \\ A_v &= \big[A_v^{\mathrm{com}}\ \ A_v^{\mathrm{loc}}\ \ A_v^{[u_1]}\ \ \cdots\ \ A_v^{[u_k]}\big]. \end{aligned} \tag{10}$$

The full template has $h_v = 2p + kd$ features. The common mask is active only in a common-exposure condition and where external information is directly supplied. The corresponding active source block is

$$B_v(c) = m_v^{\mathrm{com}}(c)A_v^{\mathrm{com}} + m_v^{\mathrm{loc}}(c)A_v^{\mathrm{loc}}. \tag{11}$$

Writing $R_{vu} := A_v^{[u]}$ makes the two contributions explicit:

$$A_v\mathbf{f}_{icv} = B_v(c)\mathbf{z}_{ic} + \sum_{u \in \mathrm{pa}(v)} R_{vu}\mathbf{x}^{\mathrm{out}}_{icu}. \tag{12}$$

A_v is the complete assembly map, $B_v(c)$ is its active source contribution, and R_{vu} retains its parent-message contribution. This is a block decomposition of the same fitted map, not an additional edge-estimation stage.

Fitting. For each model–workflow–invocation, the record set $\mathcal{F}_v$ pools available fitting questions and their registered carrier–placement records. The two supervised targets are the observed assembled input and the observed node output, respectively:

$$\mathbf{x}^{\mathrm{in}}_{icv} = A_v\mathbf{f}_{icv} + \boldsymbol{\epsilon}^{\mathrm{in}}_{icv}, \qquad \mathbf{x}^{\mathrm{out}}_{icv} = M_v\mathbf{x}^{\mathrm{in}}_{icv} + \boldsymbol{\epsilon}^{\mathrm{out}}_{icv}. \tag{13}$$

Equation 3 is solved separately for the two maps, using the same declared observer coordinates. For a row-stacked design matrix X, targets Y, and row-action coefficients Θ, the historical solver minimizes the equivalent objectives

$$\|Y - X\Theta\|_F^2 + \lambda\|\Theta\|_F^2 = \left\| \begin{bmatrix} Y \\ 0 \end{bmatrix} - \begin{bmatrix} X \\ \sqrt{\lambda} I \end{bmatrix} \Theta \right\|_F^2, \qquad \Theta \geq 0. \tag{14}$$

Minimizing either side gives the same augmented nonnegative least-squares problem, not an unconstrained fit followed by coefficient clipping. The discrepancy branch uses $\lambda = 0.1$, no intercept, and no feature centering or standardization.

Implementation correspondence. In the archived implementation, the row-action field `B` stores the *whole* input-assembly map, and `Gamma` stores the processing map. In the column convention here,

$$A_v = \mathtt{B}_v^{\mathsf{T}}, \qquad M_v = \mathtt{Gamma}_v^{\mathsf{T}}. \tag{15}$$

The global B in CTB contains only source subblocks of A_v. An exported output-level source map already includes processing and is MB, not B; applying M again would double-count that step. The magnitude implementation clips negative predictions at zero. On nonnegative source features and nonnegative fitted maps this is algebraically redundant.

Recursive prediction. The following steps define the complete forward predictor.

1. Fix the question partition, measurement definition, invocation order, source masks, endpoint selector, and eligibility/normalization rules.
2. Fit A_v using observed parent outputs and fit M_v using observed local inputs, on fitting questions only. Freeze the maps before evaluation.
3. For an evaluation pair, measure the registered source $\mathbf{z}$. Traverse invocations in topological order, form $\widehat{\mathbf{f}}_v$ using this source and already *predicted* parent outputs, and compute $A_v\widehat{\mathbf{f}}_v$ followed by $M_vA_v\widehat{\mathbf{f}}_v$.
4. Read the completed output using C. Compare predictions with independently measured states and aggregate using the shared eligible question set and score definition.

A.3 Finite-Execution Composition

Stacked construction. Define $P \in \mathbb{R}_+^{Nd\times Nd}$ by $P_{vu} = R_{vu} = A_v^{[u]}$ for executed parents and zero otherwise. Thus $P = [R_{vu}]$ is the stacked parent-message operator, not a replacement that discards those blocks. Set

$$B = \mathrm{col}(B_1(c), \dots, B_N(c)), \qquad M = \mathrm{diag}(M_1, \dots, M_N), \qquad W = MP. \tag{16}$$

Both P and W are strictly block-lower-triangular in invocation order. For the zero-residual predictor,

$$\widehat{\mathbf{x}}^{\mathrm{in}} = B\mathbf{z} + P\widehat{\mathbf{x}}^{\mathrm{out}}, \qquad \widehat{\mathbf{x}}^{\mathrm{out}} = M\widehat{\mathbf{x}}^{\mathrm{in}} = MB\mathbf{z} + W\widehat{\mathbf{x}}^{\mathrm{out}}. \tag{17}$$

A nonzero block of W^k requires a directed path containing k executed edges. There are no paths longer than L, so $W^{L+1} = 0$. With $S = \sum_{k=0}^{L} W^k$, multiplication gives

$$(I - W)S = S(I - W) = I - W^{L+1} = I. \tag{18}$$

Thus the unique recursive solution is $\widehat{\mathbf{x}}^{\mathrm{out}} = SMB\mathbf{z}$. Applying C and defining $T = SM$ yields $\widehat{\mathbf{y}} = CTB\mathbf{z}$.

Order and endpoint processing. For a path $v_0 \to v_1 \to \cdots \to v_k$, its source-to-output contribution is

$$M_{v_k}A_{v_k}^{[v_{k-1}]} \cdots M_{v_1}A_{v_1}^{[v_0]}M_{v_0}B_{v_0}(c). \tag{19}$$

Products follow execution from right to left and include terminal processing. Summing all admitted paths requires no branch-independence assumption.

A.4 Residual Propagation and Score Accuracy

Stacking the observed local relations gives

$$\mathbf{x}^{\mathrm{out}} = MB\mathbf{z} + W\mathbf{x}^{\mathrm{out}} + \boldsymbol{\eta}, \qquad \boldsymbol{\eta} = M\boldsymbol{\epsilon}^{\mathrm{in}} + \boldsymbol{\epsilon}^{\mathrm{out}}. \tag{20}$$

The residuals are defined using *observed* parent outputs and node inputs. Subtracting the predictor with the same source and coefficients gives $(I - W)(\mathbf{x}^{\mathrm{out}} - \widehat{\mathbf{x}}^{\mathrm{out}}) = \boldsymbol{\eta}$. Equation 18 then implies

$$\mathbf{y} - \widehat{\mathbf{y}} = CS\boldsymbol{\eta} = CS(M\boldsymbol{\epsilon}^{\mathrm{in}} + \boldsymbol{\epsilon}^{\mathrm{out}}). \tag{21}$$

For nonnegative C, S, M, the coordinatewise triangle inequality and valid local bounds yield

$$|\mathbf{y} - \widehat{\mathbf{y}}| \preceq CS(M|\boldsymbol{\epsilon}^{\mathrm{in}}| + |\boldsymbol{\epsilon}^{\mathrm{out}}|) \preceq CS(M\boldsymbol{\beta}^{\mathrm{in}} + \boldsymbol{\beta}^{\mathrm{out}}). \tag{22}$$

The identity holds even for biased or dependent residuals. It isolates how unexplained local changes accumulate; it does not assert that they are small. A node-output residual enters after that node's processing, whereas an input residual must first pass through its M_v. Zero measured source discrepancy therefore implies zero prediction for the homogeneous model, but not necessarily zero observed response.

Norm bound. Submultiplicativity also gives

$$\|\mathbf{y} - \widehat{\mathbf{y}}\|_2 \le \|CS\|_2\big(\|M\|_2\|\boldsymbol{\epsilon}^{\mathrm{in}}\|_2 + \|\boldsymbol{\epsilon}^{\mathrm{out}}\|_2\big). \tag{23}$$

This fixed-fit bound follows from standard matrix-norm inequalities (Horn & Johnson, 2013); it excludes refitting uncertainty.

Condition scores and ordering. Suppose $\|\mathbf{y}_{ic} - \widehat{\mathbf{y}}_{ic}\|_2 \le b_{ic}$ on the same eligible pairs used in Equation 7. The reverse triangle inequality gives

$$|s_c - \widehat{s}_c| \le \frac{1}{Z_c}\sum_{i\in\mathcal{I}_c}\big|\|\mathbf{y}_{ic}\|_2 - \|\widehat{\mathbf{y}}_{ic}\|_2\big| \le \frac{\sum_{i\in\mathcal{I}_c} b_{ic}}{Z_c} =: \beta_c. \tag{24}$$

For two conditions on comparable scales and under the same normalization rule,

$$\widehat{s}_c - \widehat{s}_{c'} > \beta_c + \beta_{c'} \quad\Longrightarrow\quad s_c \ge \widehat{s}_c - \beta_c > \widehat{s}_{c'} + \beta_{c'} \ge s_{c'}. \tag{25}$$

B Comparison with Related Methods

B.1 Methodological Attributes

To clarify how REPA relates to prior work, Table S2 summarizes seven methodological attributes of the studies discussed in Section 2. It traces the steps from controlled perturbation and intermediate measurement to system-level prediction and recovery.

Attribute definitions. *Controlled perturbation* specifies changes to inputs, outputs, routing or evaluator instructions. *Graded signals* measure nonbinary output, message or node states, rather than only averaging binary task correctness. *Multi-error state* retains several semantically defined error coordinates in one response state. *Split local fit* separately estimates assembly and processing from observations. *Topology analysis* treats structure, order or depth as a modeling or experimental factor. *Composed prediction* combines measured local relations and checks the resulting system prediction. *Recovery study* evaluates correction, defense or mitigation, without implying optimal selection or uniformly positive benefit.

Table S2 shows that prior work studies controlled faults, graded intermediate signals, topology, and mitigation in different combinations. REPA is the method covering these seven attributes. More specifically, it represents four semantic error channels in one response state, fits input assembly and agent processing separately, and composes these local measurements into a quantitative endpoint prediction. The fitted responses also inform targeted recovery, linking prediction to optimization.

B.2 Comparison Scope and Advantages of REPA

Relation to prior work. Existing studies offer complementary perspectives on failure diagnosis, propagation dynamics and recovery. Attribution and robustness studies characterize observed failures and their consequences (Cemri et al., 2025; Zhang et al., 2025; Jia et al., 2026). Propagation models examine uncertainty, preference transmission or hallucination dynamics (Emde et al., 2026a; Liu, 2026; Jamshidi et al., 2026; Jamshidi, 2026). REPA complements these perspectives by connecting measured local responses to quantitative endpoint predictions across multiple error channels.

Experimental comparison. We did not include these external methods as direct experimental baselines because their targets, available observations or intervention procedures differ from ours. Aligning them would require additional adaptation and, for learned pipelines such as ResMAS and MASC, training (Zhou et al., 2026; Shen et al., 2026). These extensions are outside the present

Table S2: **Reported methodological attributes.** A check marks a reported component under the definitions below. A blank does not imply an inherent limitation.

Study	Controlled perturb.	Graded signals	Multi-error state	Split local fit	Topology analysis	Composed prediction	Recovery study
MAS-FIRE (Jia et al., 2026)	✓				✓		✓
EIB-Learner (Shen et al., 2025)	✓				✓		✓
Lost in Communication (Emde et al., 2026a)		✓					
Hallucination Cascade (Jamshidi et al., 2026)	✓	✓			✓		
Collective Hallucination (Jamshidi, 2026)	✓	✓			✓		✓
Contagion Networks (Liu, 2026)	✓	✓			✓		
ResMAS (Zhou et al., 2026)	✓				✓		✓
MASC (Shen et al., 2026)	✓	✓			✓		✓
REPA	✓	✓	✓	✓	✓	✓	✓

experimental scope. Our prediction controls instead share eligible records and source measurements, while restoration controls share the same one-node repair operation and allowance (Appendices C.3 and C.4). These comparisons isolate the contributions of response composition and intervention selection.

Advantages of REPA. REPA's central advantage is an explicit connection between local agent behavior, system susceptibility and targeted intervention. It separately estimates assembly and processing responses, then composes them across error channels and the executed workflow. The residual analysis links local modeling errors to endpoint prediction error. Once the response maps are fitted, forward prediction uses the paired source measurement and execution structure without downstream test observations. Inverse diagnostics on paired traces guide prompt restoration, while structural response estimates guide evidence placement. This connects interpretable prediction with intervention selection while keeping task-model parameters and workflow structure fixed.

C Experimental Details

C.1 Data and Execution Settings

Benchmark construction. The seven knowledge settings use source-authored questions, with five questions per fact bundle. Sources cover Roman history (Berger et al., 2016), photosynthesis (Johnson, 2016; Cardona et al., 2018), DNA (National Human Genome Research Institute, 2020), black holes (Reddy, 2020), CRISPR (Zhou & Yao, 2023), blockchain (Yaga et al., 2018) and machine learning (Zhang et al., 2023; Atherton et al., 2023). The registered authoring pipeline uses GPT-5.6 and retains reference answers, aliases and source excerpts. Questions sharing a fact bundle stay in one split. The native QA sources are HotpotQA distractor validation, 2WikiMultiHopQA development and MuSiQue-Ans v1.0 development (Yang et al., 2018; Ho et al., 2020; Trivedi et al., 2022). Adapters preserve question text and context order.

The benchmark roster assigns 100 Development, 20 Validation and 30 Test questions per setting. Gold answers and support labels are withheld from task agents. These construction counts describe the benchmark roster, not a denominator automatically assigned to every subsequent analysis.

Runtime and single-agent controls. Table S3 defines the three single-call baselines. They are not token-matched. Schematic prompts appear in Appendix D.1.

The architecture comparison registers temperature 0.7, top-p 0.8, top-k 20 and a 4,096-token output limit. The local-composition runs use temperature zero and a 4,096-token limit. Qwen3.5 models use non-thinking mode. Q122 composition uses tensor parallelism of four, BF16 and a 32,768-token context. Its recorded software versions are Python 3.11.15, PyTorch 2.11.0, Transformers 5.13.0

and vLLM 0.24.0. Fresh Q14 localization and Q122 clean RAG use a 1,024-token output limit per invocation. Checkpoint, graph, prompt and judge differences mean cross-model results describe configurations rather than an isolated parameter-count effect.

Table S3: **Single-agent comparison methods.** Each uses one invocation on the original question, without external tools.

Method	Instruction	Visible output
Direct	Answer concisely and check uncertainty.	Answer, evidence and confidence.
Elaborate prompt	Make a compact plan and check the evidence before answering.	Brief plan and check, followed by the answer fields.
Internal roles	Privately follow decomposition, claim proposal, verification, reasoning, synthesis and formatting.	Final answer and an audit note. No separate role transcript.

A private role checklist creates no separate agent trace. Direct answering generates a task answer. Direct regression predicts an error vector.

Table S4: **Deployment settings for local composition.** Settings are study-specific.

Model	Deployment record	Generation
Q14	Four vLLM endpoints on a registered eight-V100 deployment.	Temperature 0. Output limit 4,096.
Q9	Two RTX PRO 6000 Blackwell Server Edition GPUs with TP4 and BF16. No quantization.	Non-thinking mode. Temperature 0. Output limit 4,096.
Q122	Four RTX PRO 6000 Blackwell Server Edition GPUs with TP4 and BF16. No quantization.	Non-thinking mode. Temperature 0. Output limit 4,096.

C.2 Observer Agreement

The main audit compares signed changes on 200 paired HotpotQA instances. For question i, channel k, evaluator m and prompt $p \in \{A, B\}$, let $s^{(a)}_{ikmp}$ be the grade. The paired change and equally weighted joint reference are

$$\begin{aligned} d_{ikmp} &= s^{(1)}_{ikmp} - s^{(0)}_{ikmp}, \\ \bar{d}_{ikm} &= \tfrac{1}{2}(d_{ikmA} + d_{ikmB}), \qquad d^{\text{joint}}_{ik} = \tfrac{1}{2}(\bar{d}_{ik,\text{GPT}} + \bar{d}_{ik,\text{Claude}}). \end{aligned} \tag{26}$$

Prompt A emphasizes evidence. Prompt B emphasizes role requirements and changed content.

For candidate and reference changes x_i, y_i, Lin's concordance correlation coefficient is (Lin, 1989)

$$\widehat{\rho}_{\text{c}} = \frac{2 s_{xy}}{s_x^2 + s_y^2 + (\bar{x} - \bar{y})^2}, \qquad s_{xy} = \frac{1}{n} \sum_i (x_i - \bar{x})(y_i - \bar{y}). \tag{27}$$

Both variances also use divisor n. Numerical differences assume equally spaced rubric grades. Agreement concerns paired severity changes under this audit.

C.3 Forward Prediction and Fault Localization

Prediction targets and estimators. All four main estimators use the same eligible records. Scalar regression scales the source vector by one coefficient. Diagonal regression estimates one per channel. Direct-matrix regression allows cross-channel terms. The fits use nonnegative ridge regression with $\lambda = 0.1$. Let $\mathcal{F}$ be the fitting records for one execution configuration and let $\mathbf{z}_i, \mathbf{y}_i \in \mathbb{R}^4_+$ denote their source and endpoint vectors. The fit pools the registered error families and placements within that configuration. Diagonal and direct-matrix regression solve

$$\widehat{H}_{\mathcal{H}} = \arg\min_{H \in \mathcal{H}} \sum_{i \in \mathcal{F}} \|\mathbf{y}_i - H\mathbf{z}_i\|_2^2 + 0.1 \|H\|_F^2, \qquad \widehat{\mathbf{y}} = \widehat{H}_{\mathcal{H}} \mathbf{z}, \tag{28}$$

with $\mathcal{H}$ restricted to nonnegative diagonal matrices or allowing all nonnegative 4×4 matrices, respectively. The archived Scalar baseline fits response *magnitudes* rather than the vector objective.

$$\widehat{\alpha} = \frac{\sum_{i \in \mathcal{F}} \|\mathbf{z}_i\|_2 \|\mathbf{y}_i\|_2}{\sum_{i \in \mathcal{F}} \|\mathbf{z}_i\|_2^2 + 0.1}, \qquad \widehat{\mathbf{y}}^{\text{scalar}} = \widehat{\alpha}\mathbf{z}. \tag{29}$$

Its coefficient minimizes the ridge-penalized squared error between endpoint norm and scaled source norm. Diagonal regression gives each channel its own scale, and direct regression additionally permits cross-channel terms.

Localization controls. The Q14 diagnostic uses new HotpotQA questions, excluding 1,721 historical IDs and normalized-text duplicates. The initial Generator is disabled. Its replacement prompt requires an unavailable-answer placeholder. The node observer measures missing substantive role content $q_v \in \{0, 1\}$, independently of factual correctness. It sees the normal role interface, question, input and output, but not fault labels or gold answers. Six earlier development questions with disablement at each invocation fit nonnegative parent weights W_{inv} using a ridge penalty of 0.1 and no intercept. Directly intervened receiver equations are omitted. Each receiver requires at least six defined rows and full parent-input rank. This scalar inverse is distinct from the four-channel forward predictor. Local, original REPA and gated REPA share the paired states. The healthy probe additionally reruns each Disabled input with its Healthy prompt and scores. The placeholder control uses the known artificial fault text. These controls therefore have extra calls or fault-signature information.

C.4 Targeted Recovery and Evidence Placement

One-node prompt restoration. Each evaluation questions receives a single-internal, double-internal or endpoint fault. All one-node restoration candidates are complete reruns acquired before answer judging. Each policy can restore at most one original Healthy role prompt. Other disabled prompts remain unchanged. Random, fixed-endpoint and local-difference selection use the same restoration allowance.

Gated REPA selects the largest gated inverse residual above 0.5, using a paired Healthy trace and frozen node judgments. The fault map is withheld from selection. All inverse fits qualify in this study. The registered no-action rule retains the faulty result when a fit or required state is unavailable. Node selection is fixed before final-answer evaluation.

Disabled makes no intervention. Random spends the same one-node restoration allowance at its registered random location. Endpoint restores the final node. Local selects from observed local change without adjusting for propagated parent effects. REPA uses the inverse residual. These rules separate spending one repair opportunity from selecting where to spend it. All policies restore the same resource, one original Healthy prompt, without using final-answer correctness for selection. Offline acquisition of all candidate reruns is separate from one selected deployment. A double fault can remain partly active, whereas endpoint faults directly match the fixed-endpoint policy.

Evidence and site selection. Question-based retrieval uses a frozen HotpotQA corpus derived from task-provided documents. BM25 relevance and MMR diversification (Robertson & Zaragoza, 2009; Carbonell & Goldstein, 1998) produce question-specific evidence packets without using gold answers. The clean placement studies reuse the same packet across strategies and retain the full workflow. The same retrieved packet is withheld in None, supplied to root agents in Entry, repeated at every invocation in Global, and delivered to one frozen selected invocation in REPA-RAG. Every strategy retains the complete workflow. Sites are frozen before test-answer judging. Global delivers more evidence copies than REPA-RAG. Their comparison measures the trade-off between accuracy and cost, rather than selection at equal budgets. Entry may include several roots.

C.5 Measurement-Count Sensitivity

The diagnostic uses 200 Q14 HotpotQA inputs with three moderate perturbations each. It computes a sentence-overlap distance

$$d(a, b) = 1 - \frac{|F(a) \cap F(b)|}{|F(a) \cup F(b)|}, \qquad \ell_{vj} = \frac{d_{\text{out},vj}}{d_{\text{in},vj}}. \tag{30}$$

F splits text at sentence punctuation or newlines and retains stripped strings longer than ten characters. Distance is zero when both sets are empty. For each role and $k = 5, 10, \ldots, 150$, draw k records with replacement 2,000 times, using seed 42. Let $\bar{\ell}_v^{(b)}(k)$ be the mean in draw b. The normalized width plotted in Fig. 6 is

$$R_v(k) = \frac{W_v(k)}{W_v(5)}, \qquad W_v(k) = Q_{.975}\{\bar{\ell}_v^{(b)}(k)\} - Q_{.025}\{\bar{\ell}_v^{(b)}(k)\}. \tag{31}$$

C.6 Cross-Domain Prediction and Direct QA

Q14 answers 100 questions per domain using the original question and provided evidence. Decoding uses temperature zero, top-p one, and a maximum output length of 768 tokens. We normalize predictions and reference answers by lowercasing, removing ASCII punctuation and English articles, and standardizing whitespace. Exact match (EM) is the fraction of predictions that exactly match their normalized references, with abstentions receiving a score of zero.

Across the ten domains, REPA achieves a macro-average Spearman correlation of 0.81 and an MAE of 0.29. Eight domains achieve correlations of at least 0.82, indicating broadly consistent ordering of propagation strength across the evaluated settings. Absolute prediction accuracy varies more substantially. DNA achieves a correlation of 0.93 and an MAE of 0.02, whereas Photosynthesis achieves a correlation of 0.82 but an MAE of 0.39. These results distinguish the ability to rank propagation strength from the ability to estimate its magnitude accurately.

Direct-QA EM averages 0.24 across domains. Higher EM is associated with lower REPA MAE, with a domain-level Pearson correlation of -0.55. However, Roman empire and DNA both have EM of 0.25 while their MAEs are 0.32 and 0.02, respectively. MuSiQue has the lowest EM, at 0.10, without the largest MAE. Direct-QA performance therefore provides relatively useful task context but does not fully explain variation in propagation-prediction accuracy.

D Essential Prompt Templates

The templates below summarize the essential instructions. Bracketed fields represent supplied data. Documents and messages are treated as evidence, not instructions.

D.1 Task Agent

> Perform [role] for [question] using [upstream messages and evidence]. Preserve supported information, state uncertainty and follow [output interface]. Do not claim unperformed retrieval or verification.

Direct answers immediately. Elaborate adds a plan and self-check. Internal roles adds a private checklist within one invocation. MAS calls use their assigned roles and parent messages.

D.2 Paired Error Measurement

The agreement audit grades each arm separately.

> Using [question, role and references], grade [reference output] and [perturbed output] from 0 to 4 on [Fact, Hallu, ScFaith, TaskIrr] under [rubric and scoring direction]. Do not assume the intervention succeeded. Return grades and supporting reasons.

Prompt A checks evidence first. Prompt B checks role requirements and changed content first. Signed changes are computed from these grades.

The legacy forward observer instead scores discrepancy directly.

> Compare [reference and perturbed inputs/outputs] using [role and references]. Score four-channel severity and input/output discrepancies in $[0, 1]$. Different false content can have equal severity but nonzero discrepancy. Give concise reasons.

Its historical packet included nominal injection metadata with instructions to ignore it as evidence. This protocol was not fully label-blind and differs from the agreement audit.

D.3 Injection and Answer Correctness

The injector edits content before the executor applies the assigned placement.

> Introduce [error family] into [target text] with a minimal fluent edit. Preserve unrelated content. For factual errors, change a concrete entity, relation, date or quantity. Return the edit and its description.

A separate judge evaluates the final answer. Task agents do not receive the gold answer.

> Compare [complete output] with [question, reference answer and supporting material]. Accept equivalent answers that satisfy every requested part. Ignore quoted answers that the output rejects. Score abstention as incorrect. Mark indeterminate only when the reference prevents judgment. Give a concise reason.

D.4 Disablement and Localization

> Disable [role]'s task-processing function for this invocation. Provide no task-specific solution. Emit only [role-compatible unavailable placeholder].

Healthy uses the original prompt, Style changes only presentation, and restoration reinstates Healthy. The node observer receives neither the active fault prompt nor fault labels or gold answers.

> Given [question, normal role interface, input and output], judge whether a substantive role product is present. Placeholders and generic refusals lack content. Incorrect but substantive products count as present. Return present, missing or indeterminate with a reason.

D.5 Evidence Delivery

> Use [frozen packet] as optional evidence for your assigned role. Preserve current claims unless relevant evidence supports correction. Do not invent missing links or change the output interface.

Placement strategies share this packet and wrapper. REPA-RAG changes delivery position while retaining task prompts and workflow, without receiving correctness feedback.

E Limitations and Future Directions

Our evaluation covers three Qwen checkpoints and six workflows, leaving generalization to other model families and dynamically changing execution structures open. Future work could extend the evaluation to heterogeneous agents and adaptive, tool-using workflows under shared tasks and evaluation protocols. Such comparisons would help distinguish model-dependent responses from recurring structural effects and clarify whether REPA retains its predictive accuracy as agent capabilities and interaction patterns change.

ErrPropBench-10 covers a bounded set of knowledge and reasoning domains, leaving broader domain variation to be explored. Adding domains and controlled experimental settings could reveal further propagation patterns and examine how they relate to foundation-model capabilities, pretraining characteristics and MAS configurations, providing a broader basis for understanding when local responses yield reliable system-level predictions.