NeurIPS 2026

Efficient Forecasting of Task Failures in LLM Agents through Adaptive Fault Injection

Vartika Sengar*[1], Parth Thakkar*[1], **Pranoy Panda***[1], Shrey Satapara[1], Emmy Liu[2], Vijay Viswanathan[2], Sho Takemori[1], Graham Neubig[2], Chaitanya Devaguptapu[1]

[1]Fujitsu Research [2]Carnegie Mellon University *Equal contribution

Problem: Forecasting Task Failure from Partial Traces

- Post-hoc attribution answers only analyzes agent failure after the run ends
- PRMs typically detect local errors, which could still be recoverable by the agent

Task Failure Forecasting

$$f_\theta(\tau_{\le t}) := P(Y(\tau) = 1 \mid \tau_{\le t}, \pi)$$

$\tau_{\le t}$: executed prefix up to step t
π: the deployed agent policy
$Y(\tau) = 1$: the task ends in failure



Challenge: Distinguishing Recoverable from Hard-to-recover Errors

53.9%

Human experts

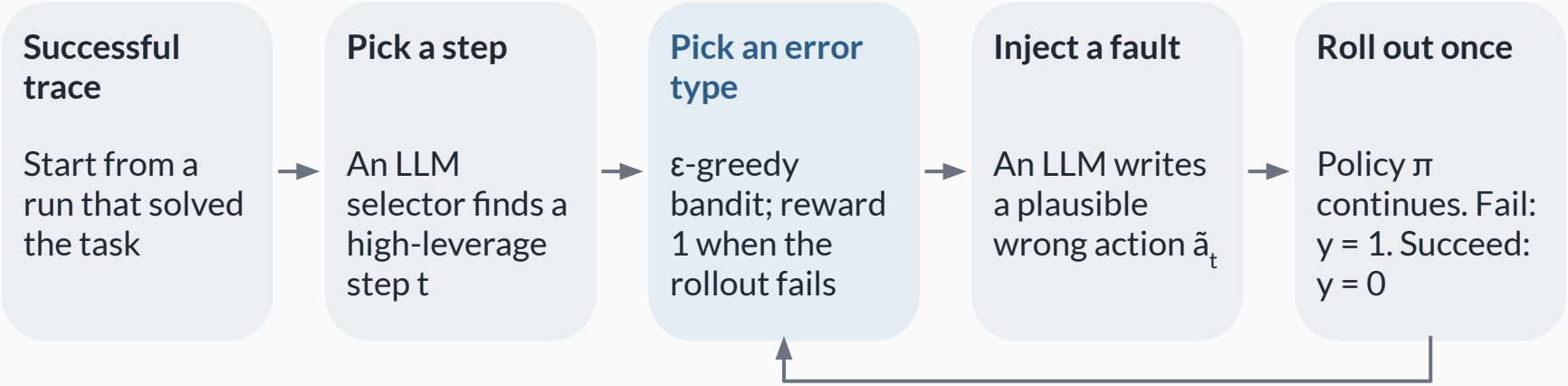
accuracy, recoverable vs. high-risk errors (chance: 50%)

Why step-level supervision is hard to get

- Failed traces mix benign and fatal mistakes, hiding which step mattered
- Monte Carlo rollouts from every prefix of a long trace cost too much

Human study: 2 annotators, 100 partial traces (Table 13). Model scores: Who&When benchmark, Zhang et al., ICML 2025 (Table 2).

Method: Adaptive Fault Injection for Step-Level Supervision



One rollout gives an unbiased label

$$E[y_{\text{roll}} \mid \tau_{\le t}] = P(Y(\tau) = 1 \mid \tau_{\le t}, \pi)$$

Class-weighted focal loss

$$l_{\text{focal}} = -w_i(1-p_t)^{\gamma} \log p_t$$

Analysis: Single-Rollout Labeling under a Fixed Budget

Proposition 1: risk decomposition

S distinct states, N rollouts per state, budget B = S · N. With high probability:

$$E(f_\theta) \le R_{emp}(f_\theta) + O(1/\sqrt{S}) + O(1/\sqrt{N})$$

O(1/√S): coverage and generalization
O(1/√N): rollout label noise

Substituting S = B/N, the gap scales as O(√(N/B)) + O(1/√N). In a large state space coverage dominates, so we use N = 1.

78.0 vs 52.6

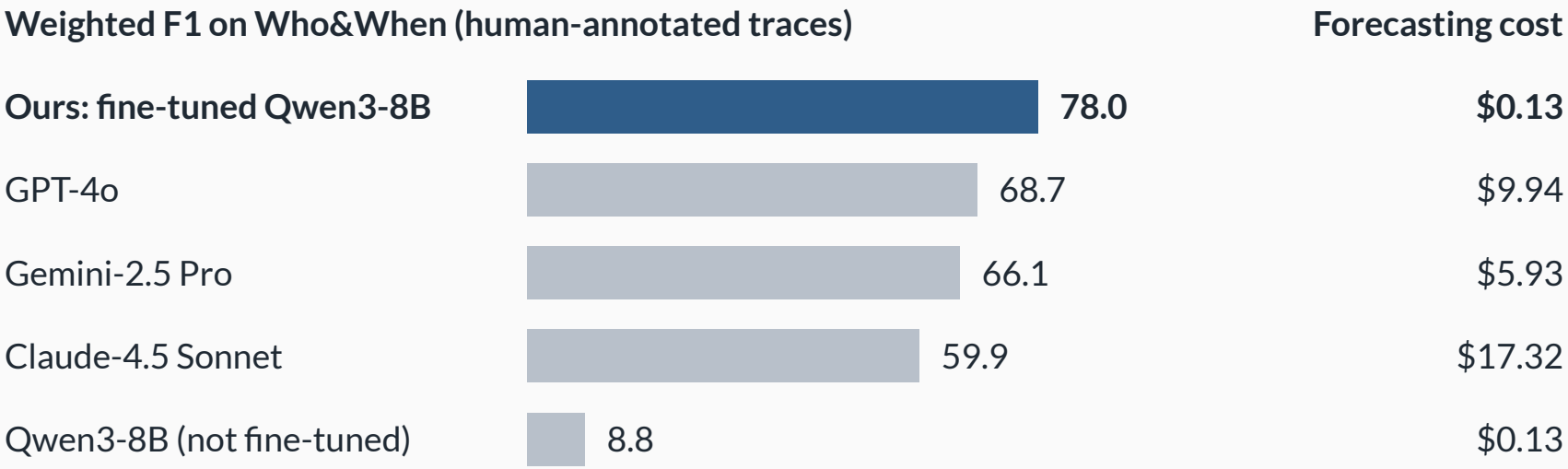
weighted F1 on Who&When: one rollout per state vs three, same total rollouts

35 vs 8

unique tools covered at B = 100: 100 states × 1 rollout vs 5 states × 20

Assumptions: bounded, Lipschitz surrogate loss (Appendix 12)

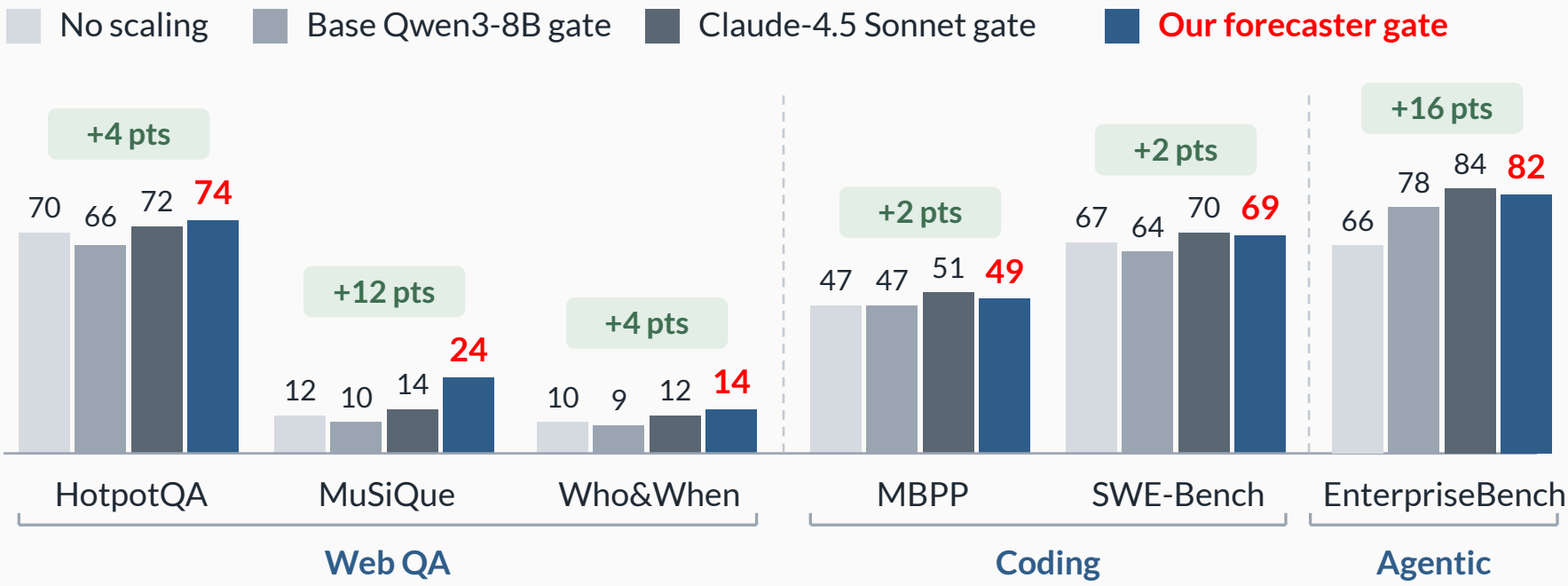
Results: Task Failure Forecasting on Who&When



Trained only on synthetic data. Bandit-selected faults reach 78.0; uniformly random faults reach 57.3 at the same perturbation budget.

Cost: USD for the full test set (937 partial traces) at OpenRouter prices. High-risk-class F1 is low for every model (ours 16.2, Gemini-2.5 Pro 17.7); our gain comes from fewer false alarms on safe steps. Tables 2 and 4.

Results: Forecaster-Gated Inference-Time Scaling (up to +16 Points)



Additional inference cost over no scaling (across all six benchmarks): **$17** our gate vs. $266 Claude gate

Task success rate (%); green: our gate minus no scaling, in points. Who&When: the 58 GAIA and AssistantBench failure traces. MuSiQue is unseen in training.

Conclusion

1. **Task Failure Forecasting**

   Predict online, from a partial trace, whether the current policy will fail

2. **Adaptive fault injection**

   A bandit over error types plus one rollout gives cheap step-level labels

3. **A small forecaster is enough**

   78.0 weighted F1 at about 100× lower cost, and 2 to 16 points more task success