Lin Shi (ls2282, ls2282@cornell.edu)

Xiao Yan (yx689, yx689@cornell.edu)

Tongjia Rao (tr426, tr426@cornell.edu)

Subjectivity Always Worse? Evaluating Agent-as-a-Judge, LLM-as-as-Judge, and Unit Tests on Coding tasks

For coding tasks where code solutions are expected, deterministically verifiable unit tests have long been the reliable gold standard for assessing correctness. However, curating correct and comprehensive unit tests is both labor-intensive and difficult. As Large Language Model (LLM) evolves, *LLM-as-a-Judge* (NeurIPS 2023, https://arxiv.org/abs/2306.05685) has emerged as a computationally efficient alternative to human evaluators across various tasks such as coding, writing, and instruction following. More recently, agentic systems have become increasingly capable of reasoning and tool use to manage more complex tasks, motivating researchers to explore *Agent-as-a-Judge* (ICML 2025, https://arxiv.org/abs/2410.10934) as a successor to LLMs, often achieving better evaluation outcomes while introducing new challenges. In 2025, coding and terminal agents (e.g., Claude Code, Codex, Gemini CLI) are emerging as state-of-the-art tools for coding tasks, with access to terminal environments that allow them to write files, execute codes, and interpret outputs. Studies have shown that with extensive tool access, agents are better selectors of debugging patches than pure LLMs (https://arxiv.org/abs/2507.23370). Therefore, we aim to address two key questions:
(1) While objective unit tests are ideal, can subjective judges (both agents and LLMs) act as reliable alternatives?
(2) With terminal access, can domain-specific coding agents become better judges than LLMs for evaluating coding tasks?

Our study aims to fill this research gap with domain-specific datasets, comprehensive benchmarking across various agents, models, and tasks, and trajectory analyses to reveal practical insights. We plan to curate the dataset by selectively aggregating coding tasks from existing benchmarks—such as **LiveCodeBench** (ICML 2025, https://arxiv.org/abs/2403.07974), **EvoEval** (COLM 2024, https://arxiv.org/abs/2403.19114), and **CodeJudge-Eval** (https://arxiv.org/abs/2408.10718)—and optionally complement them with synthetic tasks that are manually reviewed to ensure quality.

We will benchmark coding agents (e.g., Claude Code, Codex CLI, Gemini CLI), general agents (e.g., OpenHands (ICLR 2025, https://arxiv.org/abs/2407.16741)), and foundation models (e.g., GPT-5, Claude-4.5-Haiku, Gemini-2.5-Flash) on our curated dataset, measuring their alignment with unit-test results, biases (e.g., position bias), self-consistency, and computational efficiency (time, cost, etc.).

Lin Shi (ls2282, ls2282@cornell.edu)

Xiao Yan (yx689, yx689@cornell.edu)

Tongjia Rao (tr426, tr426@cornell.edu)

By analyzing agent and model performance with their trajectories, we aim to

- Investigate whether subjective agent and LLM judges can serve as reliable evaluators comparable to unit tests for coding solutions.
- Offer agent + model recommendations for judging different types of coding tasks.
- Explore whether domain-specific agents and models at inference time are also better at judging domain-specific tasks such as coding.
- Reveal practical insights into the causes of empirically superior performance among certain agents and models.

Experiments are estimated to cost a few hundred dollars in API usage, which will be covered primarily through free trial credits (e.g., Gemini) and personal funding.