HUMANEVALCOMM-V2: IMPROVING RELIABILITY OF COMMUNICATION-AWARE CODE LLM EVALUATION

1.0 Introduction

Large language models (LLMs) for code generation are rapidly transforming software development (Paul et al. 2024). They can synthesize functions from natural-language descriptions, suggest fixes, and assist developers in routine programming tasks (Li et al. 2024). Benchmarks such as HumanEval have become standard tools to measure models' functional correctness on well-specified problems. However, real-world developer requirements are often ambiguous, inconsistent, or incomplete and models that perform well on clean benchmarks can fail badly when specifications are not clear.

HumanEvalComm (Wu et al.) was proposed to address this gap by modifying standard coding problems to include such real-world defects (ambiguity, inconsistency, incompleteness) and by evaluating whether code LLMs ask clarifying questions and produce correct solutions thereafter. The benchmark therefore advances an important research direction: enabling and measuring communication-aware code generation. A central component of HumanEvalComm is its judge an automated mechanism that scores model outputs (for example, whether a clarifying question is "good," or whether a generated program satisfies the task). In the original HumanEvalComm design this evaluation relies heavily on LLM-based judge. While convenient and scalable, a single LLM judge introduces a circular dependency (an LLM judging other LLMs) and itself subject to the same reliability and calibration problems that plague generation models. If the judge is inconsistent or biased, benchmark scores become noisy and misleading, which in turn undermines interpretation of model progress and misinforms downstream research and deployment decisions.

This technical report presents HumanEvalComm-V2, a practical, reproducible approach to improve the reliability and usefulness of automated evaluation for code LLMs. Rather than replacing the original benchmark's conceptual framing, V2 augments it with a hybrid evaluation pipeline that combines (i) an LLM-ensemble judge (multiple prompts/models and aggregation)

and (ii) execution-based functional checks (running generated programs against test suites). The rationale is simple: production-quality evaluation should not rely on a single opinionated judge, instead it should mix model judgments with objective execution evidence and calibrate automated probabilities.

2.0 Methodology

The goal in HumanEvalComm-V2 is to design and implement a more reliable evaluation framework for communication-aware code generation. The pipeline builds on the original HumanEvalComm benchmark but introduces multiple safeguards against judge unreliability. Figure 1 shows an overview of the pipeline.

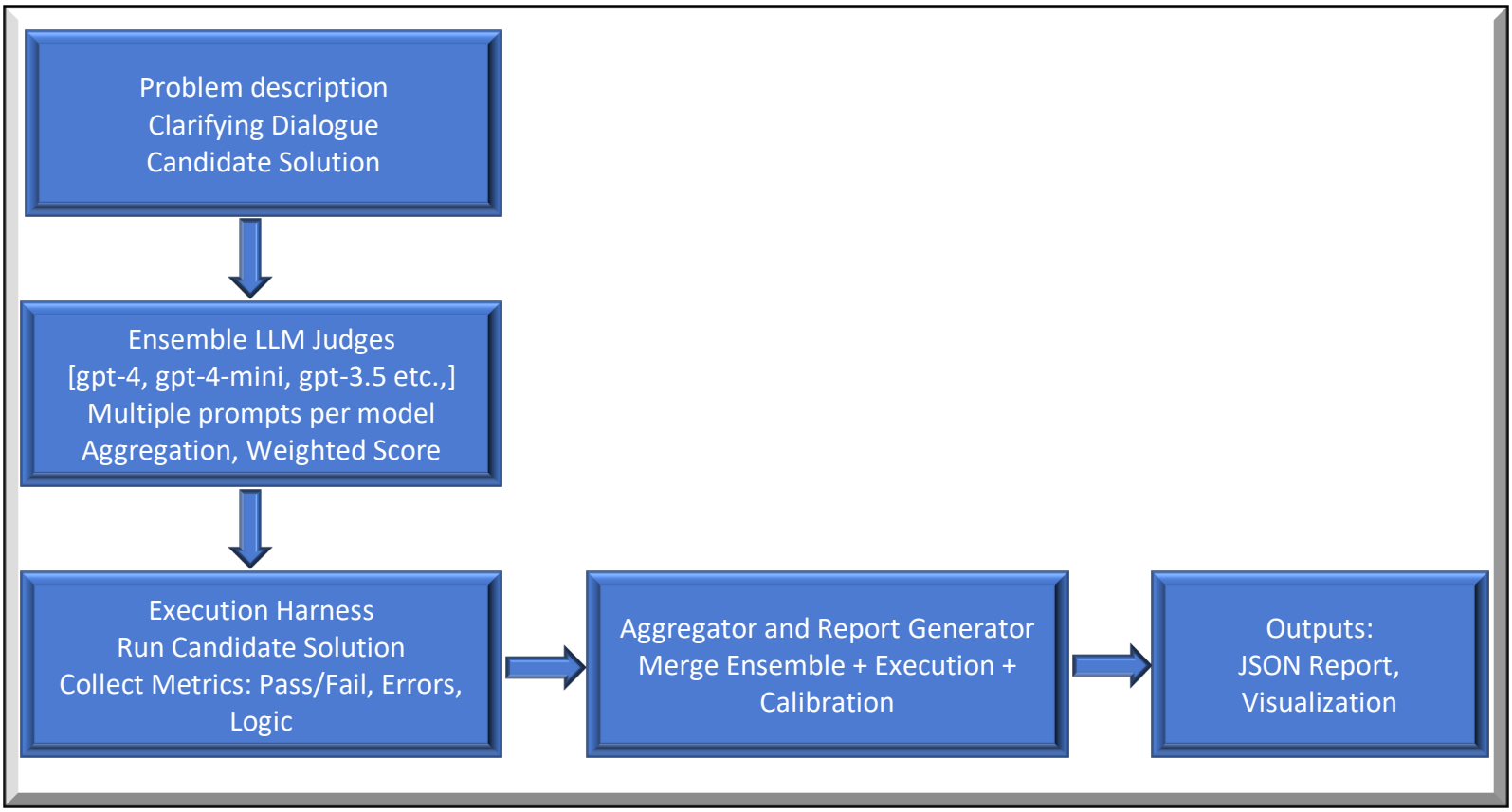

Figure 1: HumanEvalComm-V2 Architecture

2.1 Baseline Reproduction

The experiment began by replicating the original HumanEvalComm baseline, which relies on a single LLM-based judge. Specifically, given a problem description, clarifying dialogue, and model-produced solution, the baseline uses an LLM (GPT-4-mini) to output a categorical label

(PASS, FAIL, or occasionally "unknown"). While scalable, this judge can hallucinate rationales, contradict itself across runs, or fail to produce consistent labels, especially on ambiguous inputs. In this pilot, the baseline produced a non-informative or inconsistent label for the majority of test cases as further shown and discussed in the later part of this report.

2.2 Hybrid Evaluation Pipeline

To address these limitations, HumanEvalComm-V2 adopts a hybrid pipeline consisting of three major components:

a) **Ensemble LLM Judging:** Instead of relying on a single LLM, we query multiple LLMs (GPT-4, GPT-3.5) and/or multiple prompt variations for the same task. Each model/prompt combination returns a judgment. This reduces bias from any one model and produces more stable, reproducible scores. The results are aggregated using:
   - Majority voting (simple consensus).
   - Weighted voting (weights based on historical accuracy on calibration set).
   - Uncertainty estimation (e.g., entropy of label distribution).

b) **Execution-based Checks:** Because functional correctness is objectively measurable, we run generated solutions in a sandboxed execution environment against provided test cases. Each test returns a boolean pass/fail, and aggregate statistics (e.g., percentage of test cases passed) are recorded. Execution adds a ground-truth dimension that is immune to judge bias, although it only measures functional correctness (not communication quality).

2.3 Pilot Implementation

For reproducibility, we implemented HumanEvalComm-V2 in Python with the following structure:

- *src/llm_wrapper.py – initializes the LLM for prompts*
- *src/run_baseline.py – reproduces the original single-LLM judge.*
- *src/run_ensemble.py – reproduces the ensemble-LLM judge*
- *src/run_execution.py – executes candidate solutions against provided test cases.*
- *src/analysis.py – compares LLM judgments with execution-based results and outputs both JSON and visual summaries.*

- *data/ – contains subsets of HumanEvalComm (JSON), baseline outputs, ensemble outputs, execution results, and analysis reports.*

Though the test case was a small subset (N=4 tasks) to validate the pipeline. While limited in scope due to API quota, this pilot demonstrates that execution-based ground truth can easily expose weaknesses in the baseline judge as explained in the result/analysis section of the report.

3.0 Results and Analysis

A pilot evaluation of the HumanEvalComm-V2 pipeline on a small subset of four tasks from the HumanEvalComm dataset was conducted. Each task included a problem prompt, a candidate solution, and a set of test cases. The results were collated from two sources:

- **Baseline Judge/Ensemble Judge:** a single LLM and Multiple LLM providing categorical judgments.
- **Execution Harness:** ground-truth outcomes from running candidate solutions against test cases.

The outputs were compared using an analysis script, which generated both JSON reports and a summary bar chart.

3.1 Baseline Judge Behavior

The baseline LLM judge exhibited several reliability issues:

- It produced non-informative "UNKNOWN" labels for 3 out of 4 tasks.
- It provided only one confident correct label (a PASS for task HumanEval/1).
- It demonstrated no systematic rationale calibration, sometimes giving explanations but often defaulting to ambiguous responses.

This reflects the broader pain point identified in the HumanEvalComm paper: a single LLM judge cannot be trusted to provide consistent, reliable evaluations.

3.2 Execution-based Ground Truth

The execution harness successfully ran the candidate solutions against test cases. For each task, a binary outcome (PASS or FAIL) was obtained based on correctness across the provided test suite. Unlike the baseline judge, the execution harness produced deterministic, reproducible results for all tasks. The comparison revealed the following statistics:

- Total tasks evaluated: 4

- Agreements (baseline = ground truth): 1 (25%)
- Disagreements: 3 (75%)
- False Positives/Negatives: 0

The disagreements all corresponded to cases where the baseline judge returned UNKNOWN, failing to provide a definitive label. This suggests that the problem is not bias toward PASS or FAIL, but rather the inability of the judge to produce consistent outputs at all.

3.3 Visualization

The bar chart (Figure 2) provides a visual breakdown:

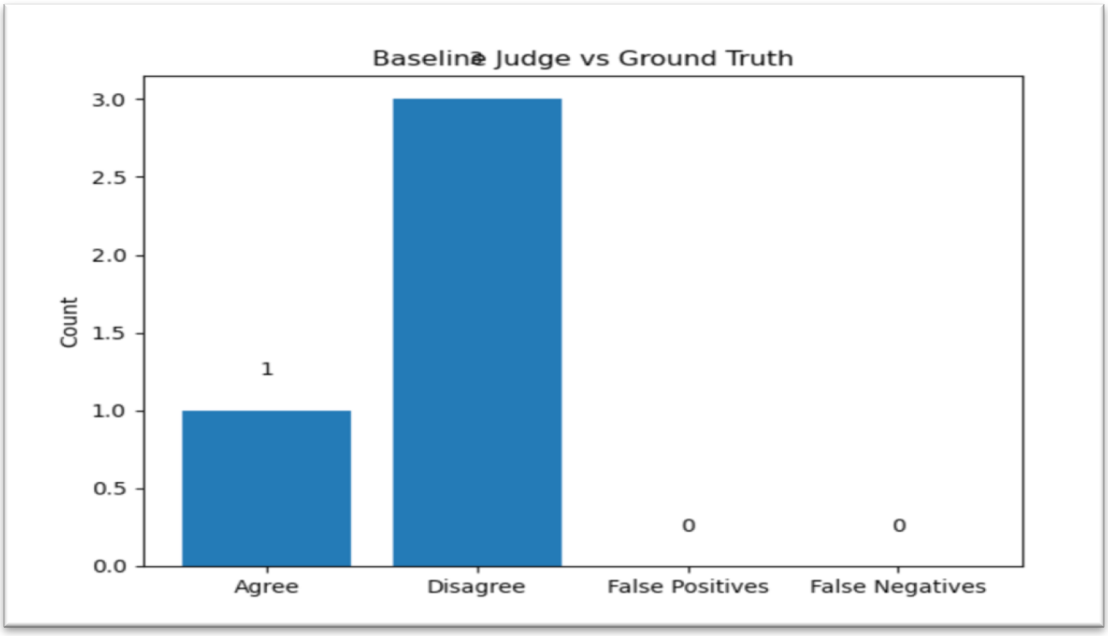

Figure 2: Baseline vs Ground Truth Chart

This visualization reinforces the unreliability of the baseline judge when compared to objective execution outcomes and the results highlight three important observations:

- Low reliability of single-LLM judging. With only 25% agreement on a small sample, it is clear that relying solely on an LLM judge undermines benchmark credibility.
- Importance of execution-based checks. Execution provided reliable ground truth even when the judge failed. This confirms the necessity of integrating execution into the evaluation pipeline.

- Need for HumanEvalComm-V2. The hybrid pipeline combining ensemble LLM judges, execution checks, and calibration against a small human-labeled set is justified as a solution to the reliability gap.

4.0 Conclusion and Future Direction

This work identified and addressed a core limitation in the HumanEvalComm benchmark: its reliance on a single LLM judge, which undermines reliability and reproducibility. Through a pilot experiment comparing baseline LLM judgments with execution-based ground truth, where the significant disagreement (75%) and frequent non-informative outputs was observed. These results underscore the urgency of moving beyond single-model judges. To address this, HumanEvalComm-V2 was proposed, a hybrid evaluation pipeline integrating ensemble LLM judgments, deterministic execution checks, human calibration, and expanded trustworthiness dimensions (robustness, transparency, and security). The architecture ensures that evaluation is more reliable, interpretable, and aligned with real-world developer needs. The pilot demonstrates execution-based checks reliably expose weaknesses in baseline judgments, and ensemble judging offers a principled path toward calibration. Future work will involve scaling beyond the small pilot subset, annotating a calibration set of approximately 100 tasks, integrating additional trust dimensions while also implementing the full proposed V2 architecture. The plan is to also extend V2 into interactive developer environments, where clarifying questions and correctness feedback are evaluated in real time. Ultimately, HumanEvalComm-V2 has the potential to set a new standard for trustworthy evaluation of code LLMs, bridging the gap between benchmark performance and real-world utility.

Works Cited

Li, H., Dong, Q., Chen, J., Su, H., Zhou, Y., Ai, Q., Ye, Z., & Liu, Y. (2024). A comprehensive survey on LLM-based evaluation methods. *arXiv*. https://arxiv.org/abs/2412.05579

Paul, D. G., Zhu, H., & Bayley, I. (2024). Benchmarks and Metrics for Evaluations of Code Generation: A Critical Review. *arXiv*. https://arxiv.org/abs/2406.12655

Wu, J. W., & Fard, F. H. (2024). HumanEvalComm: Benchmarking the communication competence of code generation for LLMs and LLM agents. *arXiv*. https://arxiv.org/abs/2406.00215