# Verifier Reliability Under Budgeted Code Generation: Research Status and Simulation Limits

Cisco Caceres · 17 October 2026 (Pacific) · Working Paper / Research Status

## Abstract

This working paper records an open research question: under a fixed inference budget, can modeling dependence among verifier errors improve code-candidate selection or stopping decisions? The evidence must be separated by source. A completed 120-task HumanEval engineering pilot is reported in a separate, linked report; it does not test the proposed joint-verifier and task-family-shift hypothesis. The prospective 576-slot H1 collection remains unrun.

**Correction to the earlier Pillar 1 report:** the reported 504-task/4,032-candidate sweep was not an empirical benchmark evaluation. The checked-in generator assigns candidate correctness, public-test outcomes, verifier scores, costs and latencies using seeded random draws. It does not generate or load candidate programs, invoke models, or execute public or hidden benchmark tests. The task identifiers and metadata are inputs to a synthetic simulation only. The reported covariance and policy rates are properties of that simulation and are withdrawn as measurements of verifier behavior, code-generation accuracy, generalization or VEC-SCR performance.

The three simulation split labels reuse the same fourteen domain labels; they do not establish a held-out-family evaluation. The earlier claim of early stopping and token conservation is also withdrawn: the configured threshold of 0.85 is above the maximum score of 0.75 reachable on the public-pass branch, so the stopping condition never fires. The simulation remains available as implementation/debugging material, clearly labeled below.

The information-ceiling derivation and missing-outcome bounds in Sections 2–3 are authored mathematical analyses, not model results or claims of novelty.

## 1. Research question and evidence boundaries

Inference-time methods can spend additional resources generating candidates, running tests, querying judges or deciding whether to stop. A legitimate study must specify what information each policy sees and charge all relevant costs. Agreement among judges is not by itself evidence that their errors are independent or that an ensemble improves selected-output correctness. The proposed question is narrower: after matching candidate pools, available information, verifier quality and total cost, does modeling cross-verifier dependence improve task-level correctness under a defined shift?

The current evidence does not answer that question. The separate 120-task HumanEval engineering pilot has limited candidate-pool headroom and is descriptive. The 504-task seeded simulation is not an empirical model study. The prospective MBPP/BigCodeBench H1 collection and confirmatory analysis have not run. No frontier-model transfer or new verifier method is claimed.

## 2. Information-Theoretic Stopping Bounds

We formalize candidate selection and verifier acquisition as a finite-action decision problem under partial observation.

### 2.1 FORMAL DECISION FRAMEWORK

Let $W \in \mathcal{W}$ denote a finite latent outcome state representing the joint correctness vector of candidate solutions $W = (Y_1, \ldots, Y_K) \in \{0,1\}^K$. Let $X \in \mathcal{X}$ denote the initial observation available to the agent, including the task prompt, problem specification, and public unit test results. Let $Z \in \mathcal{Z}$ denote an additional verifier observation, such as execution logs from supplementary test suites or scores from neural judges.

A finite action set $\mathcal{A}$ is available to the decision maker, consisting of selecting candidate $j \in \{1, \ldots, K\}$ or abstaining ($a = \varnothing$). For each action $a \in \mathcal{A}$, let $u(a, x, w) \in [0,1]$ represent the utility of action $a$ given observation $x$ and latent state $w$. For candidate selection, $u(j, x, w) = Y_j$ corresponds to binary correctness, while abstention yields $u(\varnothing, x, w) = 0$.

Let $p_X(w) = P(W = w \mid X = x)$ denote the prior belief over latent states given initial observation $X$. Let $q_{XZ}(w) = P(W = w \mid X = x, Z = z)$ denote the posterior belief updated after observing verifier signal $Z$.

Define the prior Bayes-optimal value $V_0$ and posterior Bayes-optimal value $V_1$:

$$V_0 = \mathbb{E}_X[\max_{a \in \mathcal{A}} \Sigma_{w \in \mathcal{W}} p_X(w)\, u(a, X, w)]$$

$$V_1 = \mathbb{E}_{XZ}[\max_{a \in \mathcal{A}} \Sigma_{w \in \mathcal{W}} q_{XZ}(w)\, u(a, X, w)]$$

The net value of information provided by verifier signal $Z$ is defined as $\Delta = V_1 - V_0$.

### 2.2 DERIVATION OF THE INFORMATION CEILING

We define the expected Total Variation (TV) distance between posterior and prior distributions:

$$T = \mathbb{E}_{XZ}\,[\, TV(q_{XZ}, p_X)\,] = \tfrac{1}{2} \cdot \mathbb{E}_{XZ}\,[\, \Sigma_{w \in \mathcal{W}} \left| q_{XZ}(w) - p_X(w) \right|\,]$$

and the conditional mutual information $I(W; Z \mid X)$ measured in nats:

$$I = I(W; Z \mid X) = \mathbb{E}_{XZ}\,[\, D_{KL}(q_{XZ} \,\|\, p_X)\,] = \mathbb{E}_{XZ}\,[\, \Sigma_{w \in \mathcal{W}}\, q_{XZ}(w)\, \ln(\, q_{XZ}(w) \,/\, p_X(w)\,)\,]$$

**Theorem 1 (Information-Theoretic Ceiling on Verifier Gain).** For any finite action set $\mathcal{A}$, bounded utility $u \in [0,1]$, and joint distribution $P(W, X, Z)$:

$$\mathbf{0 \le \Delta \le \min\{\, 1 - V_o,\; T,\; \sqrt{(\, I \,/\, 2\,)}\,\}}$$

Proof. Since an agent observing $(X, Z)$ can always ignore $Z$ and deploy the Bayes-optimal decision for $X$, Jensen's inequality on the convex maximum operator guarantees $V_1 \ge V_0$, establishing $\Delta \ge 0$. Because $u \le 1$, $V_1 \le 1$, implying $\Delta \le 1 - V_0$.

For each realization $(x, z)$, let $a_z^* = \arg\max_a \Sigma_w q_{xz}(w) u(a, x, w)$. The prior value is at least the expected utility of $a_z^*$ under $p_x$:

$$\max_a \mathbb{E}_q[u(a, x, W)] - \max_a \mathbb{E}_p[u(a, x, W)] \le \mathbb{E}_q[u(a_z^*, x, W)] - \mathbb{E}_p[u(a_z^*, x, W)]$$

$$= \sum_w (q_{xz}(w) - p_x(w))\, u(a_z^*, x, w) \le \sum_{w\,:\,q > p} (q_{xz}(w) - p_x(w)) = TV(q_{xz}, p_x)$$

Taking expectations over $(X, Z)$ yields $\Delta \le T$.

By Pinsker's inequality with natural logarithms, $\mathrm{TV}(q, p) \le \sqrt{\frac{1}{2} D_{\mathrm{KL}}(q \parallel p)}$. By Jensen's inequality applied to the concave square root function:

$$T = \mathbb{E}_{XZ}[\, TV(q, p)\,] \le \mathbb{E}_{XZ}[\, \sqrt{(\,\tfrac{1}{2} D_{KL}(q \parallel p)\,)}\,] \le \sqrt{(\,\tfrac{1}{2}\, \mathbb{E}_{XZ}[\, D_{KL}(q \parallel p)\,]\,)} = \sqrt{(\, I / 2\,)}$$

This completes the proof. ■

### 2.3 ECONOMIC IMPLICATIONS FOR PAID VERIFICATION

In production code generation environments, querying a secondary model judge or running containerized execution sandboxes incurs billed dollar cost or token latency. Let $c > 0$ denote the acquisition cost of signal $Z$ expressed in equivalent utility units.

The net decision value of paid verification is $\Delta_{net} = \Delta - c$. An immediate corollary of Theorem 1 is:

$$\text{If } \sqrt{(\, I(W; Z \mid X) / 2\,)} < c, \text{ then } \Delta_{net} < 0$$

Whenever the mutual information between the verifier signal and latent program correctness is strictly below $2c^2$ nats, purchasing verification reduces expected utility under these assumptions. At equality, the bound establishes only nonpositive net value.

## 3. Partial Identification Bounds with Missing Outcomes

In real-world benchmarking, execution outcomes are frequently incomplete: long-running tasks timeout, container sandboxes hit memory limits, or test fixtures return ambiguous exit codes. Rather than imputing missing executions as zero or dropping failed runs, we derive sharp partial identification bounds.

For each task $t \in \{1, \ldots, N\}$ and candidate $j \in \{1, \ldots, K\}$, let $Y_{tj} \in \{0, 1\}$ denote latent suite correctness. Let observed executions partition candidate labels into known values and unknown sets $\mathcal{U}_t$. Let $\mathcal{A}_t = \{0, 1\}^{|\mathcal{U}_t|}$ denote the set of all binary completions of unknown labels.

For any two fixed candidate selection policies $S$ and $B$, let $s_t, b_t \in \{1, \ldots, K\}$ denote their respective selected candidate indices. For assignment $a \in \mathcal{A}_t$, the task-level difference is:

$$D_t(a) = Y_{t, s_t}(a) - Y_{t, b_t}(a)$$

We define task-level lower and upper bounds:

$$\ell_t = \min_{a \in \mathcal{A}_t} D_t(a), \qquad u_t = \max_{a \in \mathcal{A}_t} D_t(a)$$

The population difference is bounded sharply by:

$$(1/N) \sum_{t=1}^{N} \ell_t \le \bar{D} \le (1/N) \sum_{t=1}^{N} u_t$$

### EXACT CANCELLATION UNDER SHARED SELECTION

A critical property of candidate selection under partial observation is shared candidate cancellation:

$$\text{If } s_t = b_t, \text{ then } D_t(a) = Y_{t,\,s_t}(a) - Y_{t,\,s_t}(a) = 0 \quad \forall a \in \mathcal{A}_t$$

Treating policies S and B independently under missing data would assign each an interval of $[0, 1]$, resulting in a loose difference bound of $[-1, 1]$. Recognizing shared candidate choices collapses the difference to exactly 0, preventing missing data from artificially inflating uncertainty.

Similarly, we define the bank oracle $O_t(a) = \max_j Y_{tj}(a)$ and oracle headroom $H_t(a) = O_t(a) - Y_{t,\,b_t}(a)$.

Headroom must be evaluated jointly over assignments: if baseline candidate $b_t$ is known to pass, headroom $H_t$ is strictly zero, even if all other candidate outcomes are missing.

## 1. Earlier Pillar 1 artifact: deterministic synthetic simulation

The retained manifest contains 504 rows of task metadata labeled with LiveCodeBench and BigCodeBench names. The accompanying script does not load the task prompts or solutions. With NumPy seed 42, it samples generator labels, token counts, costs, latency, candidate correctness, public-test outcomes and verifier scores from hand-coded random distributions. Candidate “code hashes” hash synthetic placeholder strings. No candidate source code is produced, no model or judge is called, and no benchmark test suite is executed.

The resulting 4,032 records can be used to exercise analysis code and demonstrate how a chosen synthetic data-generating process flows through policy code. They cannot estimate real verifier error covariance, code pass rates, accuracy, monetary costs, task-family transfer or selection utility. All previously reported aggregate values from that sweep are therefore retained only as simulation output and withdrawn as empirical findings.

The manifest labels three groups as `dev_source`, `calib_source` and `eval_target_shift`, but inspection shows all three contain the same fourteen domain labels. They are not disjoint task-family splits, so comparisons between them do not test natural family shift.

A source-level reachability check also found the configured default stopping threshold `0.85` cannot be attained: on the branch that permits stopping, the score formula is bounded above by `0.75`. The early-stop branch never executes at that threshold. Any earlier statement that the sweep demonstrated sequential stopping or token conservation is withdrawn. The threshold and controller require a separately specified, calibrated design before they can support empirical claims.

## 5. What remains to test

Before the proposed H1 study can produce evidence, the serving executable, loaded model, tokenizer and prompt template must be qualified; the frozen 576-slot collection must be run against actual task prompts with retained candidate outputs, verifier responses, test results and costs; task-family separation must be verified from source identities; and policy baselines, outcome handling, stopping rules and analysis must be frozen before confirmatory evaluation. The distinct 120-task engineering pilot remains descriptive and should not be pooled with this prospective study.

Negative, inconclusive and null results remain acceptable outcomes. Any future result should publish task-level denominators, paired uncertainty, candidate-pool headroom, verifier overlap, complete inference costs, failures and the exact runnable artifacts.

---

**Evidence boundary:** This page contains authored mathematics, the separately described 120-task engineering pilot reference, and a correction of a synthetic simulation. It contains no empirical result from the 504-task simulation and no result from the proposed H1 study.