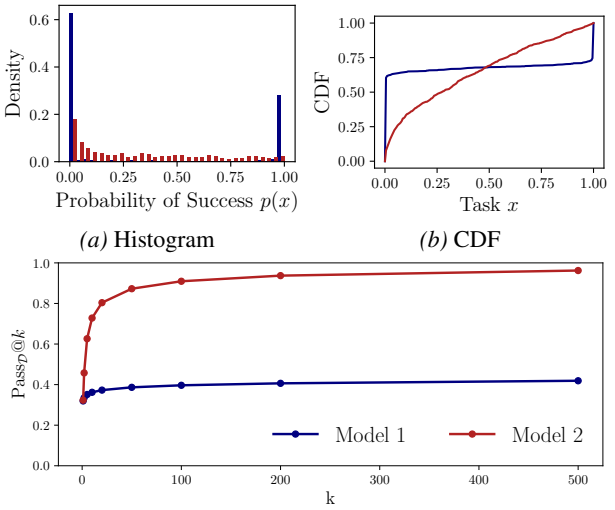
Estimating Pass@k from Fewer Samples with Hierarchical Bayesian Priors

Alexandre Vérine [1] **Florian Le Bronnec** [2] **Alexandre Allauzen** [3 4] **Benjamin Negrevergne** [3]

Abstract

Large Language Models are commonly evaluated on coding tasks using sampling-based metrics such as Pass@k, the probability of generating at least one correct solution after k independent generations. Estimating Pass@k curves from limited evaluation samples is important for benchmark design and stress testing, but can require many generations per task when per-sample success probabilities are small. We study this low-evaluation-budget regime using standard empirical-Bayes hierarchical priors over task-level success probabilities. The resulting posterior-predictive estimators pool information across tasks to estimate dataset-level Pass@k curves and to diagnose when additional sampling is likely to help. We also study a Beta–Binomial improvability diagnostic, ΔPass, whose interpretation is tied to the fitted-prior approximation. Across CodeContests, MPBB, and HumanEval, the experiments show complementary regimes: low-pass@1 tradeoffs, high-pass@1 Pareto frontiers, and a near-zero boundary-mass setting where explicit zero inflation is particularly informative.



(a) Histogram *(b)* CDF

(c) Pass@k curves

Figure 1. Empirical distribution of per-instance success probabilities on two toy dataset examples. (a) Histograms of per-instance success probabilities estimated with 1000 samples per instance. (b) Corresponding empirical CDFs. (c) Pass@k curves for each model. Despite having the same pass@1, Model (blue) sees its pass@k plateau quickly, while Model (red) continues to improve as the generation budget k increases. This difference is partly explained by the underlying distributions of per-instance success probabilities.

1. Introduction

Pass@k is an important metric for benchmark evaluation of large language models on coding and reasoning tasks. For a given task, Pass@k is the probability that at least one out of k independent candidate solutions generated by the model is correct (Chen et al., 2021; Guo et al., 2025). Given a dataset of tasks, Pass@k is usually estimated on each task individually, then averaged over the dataset. Throughout the paper, we distinguish two budgets: the generation budget k (how many generations are allowed before declaring success) and the evaluation budget m (how many generations we draw per task to estimate Pass@k).

Although many deployed systems use only a small number of generations, benchmark designers and model evaluators often want to understand how performance changes as the generation budget k increases. Estimating this curve can require a much larger evaluation budget m, because the standard combinatorial estimator needs at least $m \geq k$ generations per task and lower-variance estimates often use substantially more. Thus, the measured cost in this paper is sample efficiency: reducing the number of generations needed to estimate dataset-level Pass@k to a given accuracy. Actual wall-clock or monetary savings depend on generation cost, test execution cost, batching, and implementation overhead. Moreover, as we will discuss later in this paper, additional candidate solutions can have very different returns across models and datasets. A useful benchmark should therefore estimate not only a single score but also

[1]DI ENS, École normale supérieure, Université PSL, CNRS, Paris, France [2]Institute of Science Tokyo, Tokyo, Japan [3]Miles, LAMSADE, Université Paris-Dauphine-PSL, CNRS, Paris, France [4]ESPCI Paris, Université PSL, Paris, France. Correspondence to: Alexandre Vérine <alexandre.verine@ens.fr>.

Accepted to the 1st Workshop on Combining Theory and Benchmarks, CTB@ICML 2026, Seoul, South Korea. Copyright 2026 by the author(s).

the shape of the Pass@k curve.

We address this problem by adapting standard empirical-Bayes hierarchical priors to dataset-level Pass@k estimation. The model pools information across tasks through a prior over per-task success probabilities. Fitting this dataset-level prior extracts global structure (e.g., the prevalence of near-impossible and near-trivial tasks), which in turn (i) regularizes Pass@k estimation under tight budgets and (ii) provides diagnostics for how Pass@k evolves as the generation budget k increases. The toy example in Figure 1c illustrates why this dataset-level information matters: two models can have identical pass@1 yet exhibit markedly different Pass@k trajectories, implying very different returns to additional generations (larger k). Our experiments include HumanEval as a boundary-heavy regime: for most model and temperature configurations, pass@1 is essentially zero, so the main question becomes whether any configuration escapes the zero-success region and whether explicit boundary mass improves the fitted prior.

In this paper, we make the following contributions.

- We study two empirical-Bayes estimators for dataset-level Pass@k: a tractable BB model and a Zero–One Inflated BB extension (ZOIBB) that captures boundary mass. We also evaluate *LinMix*, a simple fixed linear mixture of both estimators used as a heuristic compromise across evaluation budgets m.
- We show that the distribution of task difficulty varies across datasets and can change how model choices should be compared as k increases. More specifically, under the BB approximation, we derive an improvability diagnostic $\Delta\text{Pass}(P)$ that is independent of the actual value of k and admits an ordering result at fixed Pass@1. This diagnostic is useful for within-regime model comparison, but its raw scale should be interpreted carefully across datasets, especially in near-zero regimes such as HumanEval.
- We evaluate several code-generation models and temperatures on CodeContests, MPBB, and HumanEval. In the low-evaluation-budget regime, the hierarchical estimators often reduce absolute error relative to the naive plug-in estimator; this sample-efficiency can translate into lower evaluation cost when generation or test execution is expensive.Finally

First we recall challenges of evaluating models using Pass@k and the Bayesian priors in Section 2, then we discuss related work in Section 3, we present our hierarchical Bayesian estimators in Section 4, and finally report experimental results in Section 5.

2. Background

2.1. Pass@k metric

In this paper, we model a large language model using a probability distribution denoted P which we can sample from. With this notation, the conditional probability $P(\boldsymbol{y} \mid \boldsymbol{x})$ models the probability of sampling candidate solution $\boldsymbol{y}$ for a given task $\boldsymbol{x}$ (a.k.a. *a prompt*). We also assume an oracle (or test) function $c(\boldsymbol{y}, \boldsymbol{x}) \in \{0, 1\}$ which indicates whether a candidate solution $\boldsymbol{y}$ is an actual correct solution to the task $\boldsymbol{x}$.

Task-wise Pass@k. Given an task $\boldsymbol{x}$ and an integer $k \in \mathbb{N}^*$, let $\boldsymbol{y}_{1:k} \coloneqq (\boldsymbol{y}_1, \ldots, \boldsymbol{y}_k)$ denote k i.i.d. samples from $P(\cdot \mid \boldsymbol{x})$. The task-wise Pass@$k$ metric is the probability that *at least one* of these samples is correct:

$$\text{Pass@}k(P, \boldsymbol{x}) \;\coloneqq\; \mathbb{P}_{\boldsymbol{y}_{1:k}}[\exists\, i \le k :\; c(\boldsymbol{y}_i, \boldsymbol{x}) = 1] \tag{1}$$

$$\;=\; \mathbb{E}_{\boldsymbol{y}_{1:k}}\left[\mathbb{1}_{\{\exists\, i \le k:\, c(\boldsymbol{y}_i, \boldsymbol{x}) = 1\}}\right]. \tag{2}$$

Equivalently, if we define the per-sample probability of success, as follows:

$$p(\boldsymbol{x}) \;\coloneqq\; \mathbb{P}_{\boldsymbol{y} \sim P(\cdot \mid \boldsymbol{x})}\big[c(\boldsymbol{y}, \boldsymbol{x}) = 1\big] \;=\; \mathbb{E}_{\boldsymbol{y} \sim P(\cdot \mid \boldsymbol{x})}[c(\boldsymbol{y}, \boldsymbol{x})], \tag{3}$$

then each indicator $c(\boldsymbol{y}_i, \boldsymbol{x})$ is a Bernoulli random variable with parameter $p(\boldsymbol{x})$. Under the usual independence assumption across samples, the probability that *all* k samples fail is $(1 - p(\boldsymbol{x}))^k$, and therefore

$$\text{Pass@}k(P, \boldsymbol{x}) \;=\; 1 - (1 - p(\boldsymbol{x}))^k. \tag{4}$$

This independence assumption is part of the standard Pass@k formalization; in practice, correlated decoding procedures or shared prompting artifacts can violate it.

Estimating Pass@k for a given task. To estimate Pass@k for a given task $\boldsymbol{x}$ we draw $m_{\boldsymbol{x}} \ge k$ independent samples $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_{m_{\boldsymbol{x}}} \sim P(\cdot \mid \boldsymbol{x})$ and count the number of successes $s_{\boldsymbol{x}}$:

$$s_{\boldsymbol{x}} \;\coloneqq\; \sum_{j=1}^{m_{\boldsymbol{x}}} c(\boldsymbol{y}_j, \boldsymbol{x}). \tag{5}$$

A common starting point is to estimate the per-sample success probability by the empirical success rate $\widehat{p}_{\text{naive}}(\boldsymbol{x}) \coloneqq s_{\boldsymbol{x}}/m_{\boldsymbol{x}}$. Plugging this estimate into the closed-form expression $\text{Pass@}k(P, \boldsymbol{x}) = 1 - (1 - p(\boldsymbol{x}))^k$ yields the corresponding plug-in estimator

$$\widehat{\text{Pass@}k}^{\,\text{naive}}(m_{\boldsymbol{x}}, s_{\boldsymbol{x}}) \;\coloneqq\; 1 - \left(1 - \frac{s_{\boldsymbol{x}}}{m_{\boldsymbol{x}}}\right)^k. \tag{6}$$

While simple, this estimator is biased because it applies a nonlinear transformation to the random quantity $s_{\boldsymbol{x}}/m_{\boldsymbol{x}}$. In particular, for finite $m_{\boldsymbol{x}}$ it typically *underestimates* $\text{Pass@}k(P, \boldsymbol{x})$, especially when $p(\boldsymbol{x})$ is small or when k is moderate to large.

We therefore adopt the standard unbiased combinatorial estimator of $\mathrm{Pass@}k(P, \boldsymbol{x})$ under sampling without replacement from the $m_{\boldsymbol{x}}$ draws (used, e.g., in Chen et al. (2021)):

$$\widehat{\mathrm{Pass@}k}^{\mathrm{comb}}(m_{\boldsymbol{x}}, s_{\boldsymbol{x}}) \;:=\; 1 - \frac{\binom{m_{\boldsymbol{x}} - s_{\boldsymbol{x}}}{k}}{\binom{m_{\boldsymbol{x}}}{k}}. \tag{7}$$

Unlike the naive plug-in estimator, the combinatorial estimator directly targets $\mathrm{Pass@}k(P, \boldsymbol{x})$ at budget k, and therefore requires at least k samples per instance to be well-defined. In contrast, $\widehat{\mathrm{Pass@}k}^{\mathrm{naive}}$ can be computed even when $m_{\boldsymbol{x}} < k$, since it only relies on an estimate of $p(\boldsymbol{x})$. This estimator makes explicit the evaluation burden (large $m_{\boldsymbol{x}}$): one must draw at least k samples per instance, and in practice reliable estimates of $\mathrm{Pass@}k$ for moderate or large k often require $m_{\boldsymbol{x}}$ substantially larger than k.

Dataset-wise Pass@k. In practice, evaluation is performed on a dataset $\mathcal{D}$ of tasks rather than on a single task. The dataset-wise metric is the simply the expected task-wise $\mathrm{Pass@}k$ over the entire dataset:

$$\mathrm{Pass@}k_{\mathcal{D}}(P) \;:=\; \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}}\big[\mathrm{Pass@}k(P, \boldsymbol{x})\big]. \tag{8}$$

Given a finite evaluation set $\{\boldsymbol{x}_i\}_{i=1}^N$ and per-instance counts (m_i, s_i), the corresponding combinatorial estimator is

$$\widehat{\mathrm{Pass@}k}_{\mathcal{D}}^{\mathrm{comb}}(P) \;:=\; \frac{1}{N}\sum_{i=1}^{N} \widehat{\mathrm{Pass@}k}^{\mathrm{comb}}(m_i, s_i). \tag{9}$$

Computationally, this corresponds to drawing $\sum_{i=1}^N m_i$ generations in total, with the constraint $m_i \geq k$ for every instance. In many empirical settings, m_i is chosen several times larger than k to reduce estimator variance, which can make evaluating $\mathrm{Pass@}k$ at large k prohibitively expensive.

2.2. Prior over per-instance success probabilities

In the low-sampling regime, estimating $\mathrm{Pass@}k$ benefits from pooling statistical strength across tasks via a prior on the latent per-instance success probabilities $p(\boldsymbol{x})$. In this work we consider two priors. The Beta distribution provides a simple and expressive baseline, and is conjugate to the Bernoulli/Binomial likelihood. The zero-one inflated Beta (ZOIB) extends this baseline by allowing non-zero probability mass at 0 and 1, which is natural on code benchmarks where a model may render some tasks effectively unsolvable (near 0) while solving others almost always (near 1). Using a single dataset-level prior is an exchangeability approximation: if a benchmark mixes distinct task families with different difficulty profiles, stratified fitting or richer mixture priors may be more appropriate.

The Beta Distribution: The Beta distribution is the canonical conjugate prior for the Bernoulli likelihood. For parameters $a, b > 0$, a random variable $p \in (0, 1)$ follows a Beta distribution, denoted $p \sim \mathrm{Beta}(a, b)$, if it has density

$$f(p \mid a, b) \;=\; \frac{p^{a-1}(1-p)^{b-1}}{\beta(a, b)}, \qquad p \in (0, 1), \tag{10}$$

where $\beta(\cdot, \cdot)$ is the Beta function with $\beta(a, b) := \frac{\Gamma(a)\,\Gamma(b)}{\Gamma(a+b)}$. Here $\Gamma(\cdot)$ denotes the Gamma function, defined for $t > 0$ by $\Gamma(t) := \int_0^\infty x^{t-1}e^{-x}\,dx$. The mean of $\mathrm{Beta}(a, b)$ is $\mathbb{E}[p] \;=\; a/(a+b)$, and the parameters (a, b) control both the location and concentration of the mass. Informally, increasing a effectively increases the probability mass allocated near 1, while increasing b increases the mass allocated near 0. The sum $a + b$ controls concentration: larger $a + b$ yields a distribution more tightly concentrated around its mean. To build intuition, consider four representative regimes: (i) $\mathrm{Beta}(1, 1)$ is uniform on $(0, 1)$; (ii) $\mathrm{Beta}(1, 5)$ concentrates mass near 0 (many hard instances); (iii) $\mathrm{Beta}(5, 1)$ concentrates mass near 1 (many easy instances); (iv) $\mathrm{Beta}(2, 2)$ is unimodal with a central mode around $1/2$ (most instances have intermediate difficulty).

The Zero-One Inflated Beta (ZOIB) distribution: The ZOIB distribution (Ospina & Ferrari, 2010; 2012) extends the Beta distribution by placing explicit probability mass at the boundary points 0 and 1. It is parameterized by (a, b, π_0, π_1), where $a, b > 0$ are the Beta parameters and $\pi_0, \pi_1 \in [0, 1]$ are the masses at 0 and 1, respectively, with $\pi_0 + \pi_1 < 1$. A random variable $p \in [0, 1]$ follows a ZOIB distribution, denoted $p \sim \mathrm{ZOIB}(a, b, \pi_0, \pi_1)$, if it can be written as the mixture

$$p \;\sim\; \begin{cases} 0, & \text{w.p. } \pi_0, \\ 1, & \text{w.p. } \pi_1, \\ \tilde{p}, & \text{w.p. } 1 - \pi_0 - \pi_1, \;\text{ where } \tilde{p} \sim \mathrm{Beta}(a, b). \end{cases} \tag{11}$$

This representation makes the interpretation explicit: π_0 and π_1 capture the prevalence of (near-) impossible and (near-) trivial instances.

3. Related Work

Code generation with LLMs. Large language models achieve strong performance on code-generation benchmarks evaluated with executable tests, including HumanEval, MBPP, APPS, and CodeContests (Chen et al., 2021; Austin et al., 2021; Hendrycks et al., 2021; Li et al., 2022; Grattafiori et al., 2024; OpenAI et al., 2024; Guo et al., 2025). We use HumanEval as an additional benchmark regime; in our logged generations it is substantially more boundary-heavy than MPBB and CodeContests.

Generation-based evaluation and Pass@k. Code-generation performance is often reported via $\mathrm{Pass@}k$, the probability that at least one out of k independent generations solves a task (Chen et al., 2021; Tang et al., 2025; Zheng

et al., 2025). Even when downstream systems use only a small number of generations, estimating the Pass@k curve is useful for benchmark design, stress testing, and comparing sampling policies. Our work addresses the low-m regime by fitting a dataset-level prior over per-task success probabilities and using it to estimate the Pass@k curve and expected gains as k increases.

Diversity and generation gains. Recent work highlights that models may generate redundant correct solutions and that diversity can affect gains as k increases (Le Bronnec et al., 2024; Lee et al., 2025; Wang et al., 2025; Yao et al., 2025; Le Bronnec et al., 2026). We treat diversity as a secondary, correlational diagnostic and relate generation gains to a code-similarity measure over successful outputs when enough successful generations are available.

Bayesian approaches to evaluation. Bayesian methods provide principled regularization and uncertainty quantification for evaluation in low-evaluation-budget regimes (Luettgau et al., 2025; Hariri et al., 2025). Kazdan et al. (2025) develop efficient BB-based evaluation with an emphasis on adaptive allocation of evaluation generations. The BB and ZOIBB components we use are standard statistical tools; our focus is their use as simple posterior-predictive estimators and diagnostics for Pass@k curves. Direct empirical comparison with concurrent Bayesian evaluation frameworks is left to future work.

4. Hierarchical diagnostics for Pass@k curves

In the low-evaluation-budget regime (small m), estimating how Pass@k scales with respect to the generation budget k requires more information than is available in the individual per-task counts alone. It is also important to understand how task-wise success probabilities are distributed across the dataset. To model that information, we adopt a Bayesian hierarchical formulation that pools statistical strength across tasks by introducing a dataset-level prior over task-wise success probabilities. Once fitted, this model yields a posterior-predictive estimate of the dataset-level Pass@k curve from limited evaluation generations.

In this section, we consider two hierarchical models: the classical Beta − Binomial (BB) model, and its Zero–One Inflated Beta − Binomial (ZOIBB). We then introduce *LinMix*, a simple fixed linear mixture of BB and ZOIBB used as a heuristic compromise across sampling regimes. Finally, we present fit diagnostics illustrating how these models approximate the empirical distribution of success probabilities on real benchmarks. All the derivations are deferred to Appendix A.

4.1. BB estimator of dataset-level Pass@k

Hierarchical model. For each task $i \in \{1, \dots, N\}$, let $p_i \in [0, 1]$ denote the (latent) probability that a single sample from $P(\cdot \mid \boldsymbol{x}_i)$ solves the task. We model task heterogeneity by a Beta prior,

$$p_i \overset{\text{i.i.d.}}{\sim} \text{Beta}(a, b), \qquad s_i \mid p_i \sim \text{Binomial}(m_i, p_i),$$

where (m_i, s_i) are the observed evaluation budget and number of successes. The hyperparameters (a, b) describe the dataset-level distribution of task difficulties. This exchangeability assumption is an approximation: if a benchmark contains structured task clusters, the fitted prior should be interpreted as an aggregate summary rather than a claim that tasks are homogeneous.

Empirical Bayes fitting. Integrating out p_i in (4.1) yields the BB marginal likelihood

$$p(s_i \mid m_i, a, b) \;=\; \binom{m_i}{s_i} \frac{\beta(a + s_i,\, b + m_i - s_i)}{\beta(a, b)}. \quad (12)$$

We estimate (a, b) by maximizing the total log-evidence $\sum_{i=1}^{N} \log p(s_i \mid m_i, a, b)$ over $a, b > 0$. In practice, we optimize in $(\log a, \log b)$.

Posterior-predictive Pass@k. Under the BB model, the unconditional expectation

$$\mathbb{E}_{p \sim \text{Beta}(a,b)}\big[1 - (1 - p)^k\big] \;=\; 1 - \frac{\beta(a,\, b + k)}{\beta(a, b)} \quad (13)$$

characterizes the expected Pass@k of a *new* task whose success probability is drawn from the fitted Beta law. In this work, however, we estimate Pass@k on a *fixed finite dataset* $\mathcal{D} = \{\boldsymbol{x}_i\}_{i=1}^{N}$ for which we observe per-task counts (m_i, s_i). We therefore use the posterior predictive pass@k for each task and average across tasks. By conjugacy, the posterior of p_i under (4.1) is

$$p_i \mid (m_i, s_i) \sim \text{Beta}(a + s_i,\, b + m_i - s_i). \quad (14)$$

The corresponding posterior predictive pass@k for task i is

$$\widehat{\text{Pass@}k}^{\,\text{BB}}(s_i, m_i) \;\coloneqq\; \mathbb{E}\big[1 - (1 - p_i)^k \mid m_i, s_i\big] \;=\; 1 - \frac{\beta(a + s_i,\, b + m_i - s_i + k)}{\beta(a + s_i,\, b + m_i - s_i)}.$$

We then define the dataset-level BB estimator as

$$\widehat{\text{Pass@}k}_{\mathcal{D}}^{\,\text{BB}}(P) \;\coloneqq\; \frac{1}{N} \sum_{i=1}^{N} \widehat{\text{Pass@}k}^{\,\text{BB}}(s_i, m_i). \quad (15)$$

We prefer (15) over the unconditional quantity (13) because it conditions on the observed per-task evidence (m_i, s_i):

when m_i is small it shrinks each task toward the dataset-level prior, while for large m_i it reflects strong task-specific evidence.

A k-independent improvement metric. Within the Beta family, $\mathrm{Pass@1} = a/(a+b)$ fixes the mean, leaving the concentration $a+b$ as the remaining degree of freedom. We therefore define

$$\Delta\mathrm{Pass}(P) \;:=\; a+b, \tag{16}$$

which quantifies how concentrated the dataset-level distribution of task success probabilities is around its mean. Larger $\Delta\mathrm{Pass}(P)$ corresponds to less heterogeneity under the fitted BB approximation, which translates into larger $\mathrm{Pass@}k$ values for $k \geq 2$ at fixed pass@1 in the theorem below.

Theorem 4.1 (Pass@k ordering at fixed pass@1)**.** *Let P_1 and P_2 be two models whose per-task success probabilities follow $\mathrm{Beta}(a_1, b_1)$ and $\mathrm{Beta}(a_2, b_2)$, respectively. Assume $\frac{a_1}{a_1+b_1} = \frac{a_2}{a_2+b_2}$ (same pass@1) and $a_1, b_1, a_2, b_2 > 0$. If $\Delta\mathrm{Pass}(P_1) = a_1 + b_1 > a_2 + b_2 = \Delta\mathrm{Pass}(P_2)$, then for all $k \geq 2$,*

$$\mathrm{Pass@}k_{\mathcal{D}}(P_1) > \mathrm{Pass@}k_{\mathcal{D}}(P_2). \tag{17}$$

Theorem 4.1 formalizes a key evaluation phenomenon: two models can have similar Pass@1 yet exhibit different gains as the generation budget k increases. Under the Beta idealization, Pass@1 fixes the mean, while $\Delta\mathrm{Pass}(P) = a+b$ controls how unevenly success probability is distributed across tasks. Outside this approximation, we treat ΔPass as a diagnostic rather than a model-free guarantee.

4.2. ZOIBB estimator of dataset-level Pass@k

The Beta prior can be misspecified when a dataset contains a substantial fraction of tasks that are effectively unsolvable ($p_i \approx 0$) or almost always solved ($p_i \approx 1$). To capture such boundary effects, we extend BB using the zero–one inflated Beta prior introduced in Section 2. This extension is especially relevant on boundary-heavy settings such as HumanEval in our experiments, where many model/temperature configurations produce no successful samples and the fitted prior must represent mass at zero. Concretely, we assume

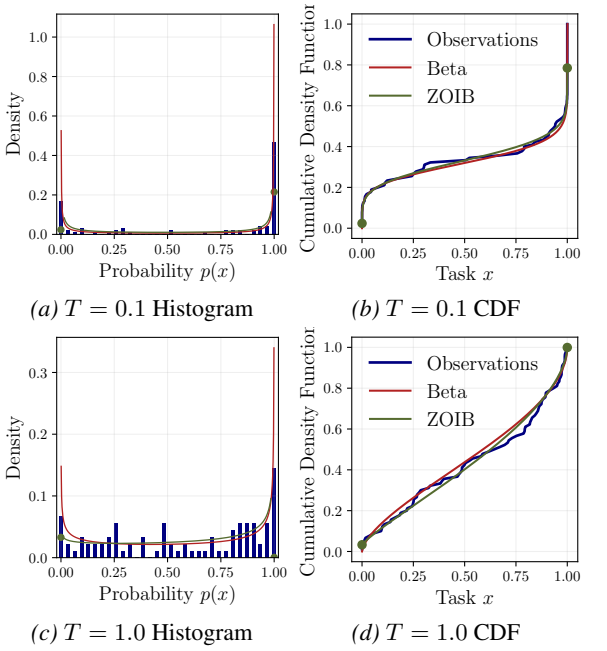
$$p_i \sim \mathrm{ZOIB}(a, b, \pi_0, \pi_1), \qquad s_i \mid p_i \sim \mathrm{Binomial}(m_i, p_i),$$

and fit (a, b, π_0, π_1) by evidence maximization. As in the BB case, our objective is to predict posterior dataset-level $\mathrm{Pass@}k$ from fitted hyperparameters. We refer to the resulting marginal-count model and induced $\mathrm{Pass@}k$ predictor as the ZOIBB estimator.

Posterior-predictive Pass@k. A useful simplification is that the ZOIB boundary masses only influence the posterior when the observed count is itself at the boundary. If $0 < s_i < m_i$, the point-mass components at 0 and 1 are incompatible with the data and therefore receive zero posterior weight. In that interior case, the posterior reduces to the Beta update and the ZOIBB per-task predictor is *exactly* the same as the BB posterior-predictive $\mathrm{Pass@}k$.

When $s_i = 0$, the observation is consistent both with a structural zero ($p_i = 0$) and with the continuous Beta component producing zero successes. The posterior is therefore a two-component mixture between a point mass at 0 and a Beta posterior on $(0, 1)$. Consequently, the posterior-predictive $\mathrm{Pass@}k$ is a convex combination of 0 (from the structural-zero component) and the BB posterior-predictive $\mathrm{Pass@}k$ (from the Beta component), with weights given by the posterior probability of being a structural zero. Similarly, when $s_i = m_i$, the posterior is a two-component mixture between a point mass at 1 and a Beta posterior, and the posterior-predictive $\mathrm{Pass@}k$ is a convex combination of 1 and the BB posterior-predictive $\mathrm{Pass@}k$, weighted by the posterior probability of being a structural one.

For completeness, Appendix A.5 provides the closed-form marginal likelihood, the posterior mixture weights, and the resulting closed-form expression for $\widehat{\mathrm{Pass@}k}^{\mathrm{ZOIBB}}(s_i, m_i)$



(a) $T = 0.1$ Histogram *(b)* $T = 0.1$ CDF

(c) $T = 1.0$ Histogram *(d)* $T = 1.0$ CDF

Figure 2. Illustration of the goodness-of-fit diagnostics for the BB and ZOIBB models on the MPBB dataset for ministral-8B Chat (Liu et al., 2026) at temperatures $T = 0.1$ (top) and $T = 1.0$ (bottom). Left: histogram of empirical success rates $\widehat{p}_i = s_i/m_i$ (blue) overlaid with the fitted Beta density (red) and the ZOIB density (green). Right: empirical CDF of $\widehat{p}_i$ overlaid with the fitted Beta CDF (red) and the ZOIB CDF (green).

in each case:

$$\widehat{\text{Pass@}k}_{\mathcal{D}}^{\text{ZOIBB}}(P) \coloneqq \frac{1}{N}\sum_{i=1}^{N}\widehat{\text{Pass@}k}^{\text{ZOIBB}}(s_i, m_i).$$

4.3. LinMix: a linear mixture of BB and ZOIBB

BB provides a low-variance baseline in the low-evaluation-budget regime whereas ZOIBB can better capture boundary mass (near 0 and 1) when the evaluation budget m is sufficiently large. More generally, the dominant source of error can change with m: for small m, estimation variance (and overfitting by overly flexible priors) tends to dominate, while for larger m model misspecification—especially in the tails—becomes increasingly important. To obtain a single estimator for comparison, we introduce *LinMix*, a deterministic two-regime interpolation between the BB and ZOIBB predictors using a monotone weight that depends only on the evaluation budget m:

$$\begin{aligned}\widehat{\text{Pass@}k}^{\text{LinMix}}(s_i, m_i) \coloneqq\ & w(m_i)\,\widehat{\text{Pass@}k}^{\text{ZOIBB}}(s_i, m_i) \\ & + (1 - w(m_i))\,\widehat{\text{Pass@}k}^{\text{BB}}(s_i, m_i)\end{aligned}$$

where $w(m) \in [0,1]$ is a fixed linear schedule $w(m) = \text{clip}((m - m_{\text{low}})/(m_{\text{high}} - m_{\text{low}}),\, 0,\, 1)$. Throughout, m_{low} and m_{high} are fixed a priori. The linear form is deliberately simple: it is monotone, interpretable, and introduces no task-specific adaptation. We do not claim that this schedule is theoretically optimal; it is a fixed heuristic used to study whether interpolation can reduce the complementary biases of BB and ZOIBB. In Section 5, we study how predictive accuracy varies with m and evaluate whether this schedule is near-best in our experiments.

4.4. Model fit diagnostics

To assess whether the proposed priors provide a reasonable approximation of the dataset-level distribution of success probabilities, we use a large-sample regime in which $\widehat{p}_i = s_i/m_i$ is a good proxy for p_i. We compare the BB and ZOIBB fits using 10-fold cross-validated expected log predictive density (CV-ELPD) computed from the corresponding marginal likelihoods. The full comparison across models, temperatures, and datasets is reported in Appendix Table 3: some settings prefer BB while others prefer ZOIBB. Across many fitted configurations, we observe boundary-heavy regimes consistent with the presence of near-impossible and near-trivial tasks. HumanEval is the most extreme case: many configurations have all or almost all observed mass at zero, while the few nonzero high-temperature configurations can produce very large fitted concentration values.

Figure 2 illustrates these diagnostics on MPBB for ministral-8B Chat (Liu et al., 2026) at two temperatures. We choose this model specifically because its behavior changes markedly with temperature, making it easy to visualize how the fitted prior shifts across evaluation conditions (via temperature and the induced difficulty profile). If these diagnostics show poor predictive fit for both priors, the appropriate fallback is not to trust the aggregate prior blindly but to stratify the benchmark or use a richer mixture model.

5. Experiments

Benchmarks. We evaluate on MPBB (Austin et al., 2021), CodeContests (Li et al., 2022), and HumanEval (Chen et al., 2021). MPBB has many tasks and supports detailed m- and N-dependence analyses; HumanEval is smaller and boundary-heavy in our experiments, with most model/temperature configurations at zero or near-zero pass@1.

Models and decoding. We consider multiple code generation models of varying sizes, Qwen-2.5 3B & 7B Chat (Qwen et al., 2025), Qwen-3.0 30B (A3b) Chat (Yang et al., 2025), Code Llama 7B & 13B Chat (Grattafiori et al., 2024) and ministral-8B Chat (Liu et al., 2026). All models are evaluated using temperature decoding with $T \in \{0.1, 1.0, 1.2\}$.

Sampling protocol and large-budget reference. For each task $\boldsymbol{x}_i$ we draw m_i independent generations and record the number of successes s_i. For a target $k < 500$, the reference Pass@k is the combinatorial estimator computed from the large evaluation budget ($m = 1000$) when available. Results at a given (m, k) come from 10 subsampling runs from this pool; appendix tables report standard deviations.

Estimators compared. We compare the naive plug-in estimator, the unbiased combinatorial estimator, BB, ZOIBB, and *LinMix* (Section 4.3). The combinatorial estimator is included only when $m \geq k$; LinMix is treated as a fixed heuristic rather than an optimized model-selection rule.

Error metrics. For each configuration and target k, we report the absolute error $|\widehat{\text{Pass@}k} - \text{Pass@}k^{\star}|$ against the reference Pass@$k^{\star}$. When aggregating results across tasks or models, we average errors over runs and report either per-setting tables or curves as a function of m or N.

Similarity / diversity score (JPlag). For tasks with at least two successful outputs, we run JPlag (Maisch, 2023) and average pairwise similarity scores. This exploratory diagnostic is not used by any estimator; HumanEval is excluded because most models have too few successful samples.

5.1. Experimental setup

5.2. Predicting the Pass@k curve from few samples

We first provide a representative example of Pass@k extrapolation from limited evaluation generations. Figure 3 shows

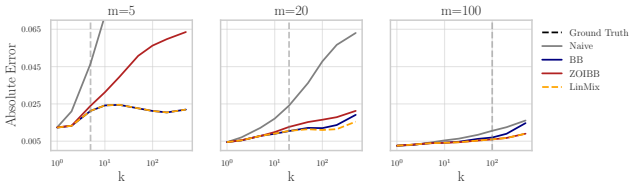


Figure 3. Absolute error of Pass@k estimation on MPBB for ministral-8B Chat at $T = 1.0$, compared with a large-budget reference estimate for several evaluation budgets m. The grey line shows the maximum k for which the combinatorial estimator is defined at each m. This is one representative setting; results across models, temperatures, and datasets are summarized in Table 1 and Appendix B.

absolute errors on MPBB for ministral-8B Chat at $T = 1.0$ under several evaluation budgets m. In this representative low-evaluation-budget setting, hierarchical pooling reduces error relative to the naive plug-in baseline, especially at larger generation budgets k where naive estimates are most biased.

5.3. Generality across models, temperatures, and datasets

We next test whether the accuracy gains persist across models, temperatures, and datasets. Table 1 reports low-budget absolute errors, averaged over models and temperatures. Bayesian estimators improve upon naive estimation in most settings. On HumanEval, many entries are tied or small because reference Pass@k values are near zero; the boundary-mass behavior is discussed below and in Appendix B. LinMix is typically a near-best empirical compromise, reflecting complementary behavior of BB and ZOIBB. Appendix B reports the full grid, standard deviations, and low-m/high-k pairwise win rates.

As a sample-efficiency example, on MPBB at $k = 100$, *LinMix* with $m = 5$ has lower average absolute error than the naive estimator with $m = 20$ (0.023 versus 0.034). Such comparisons support the claim that fewer evaluation generations can be sufficient in some regimes, without directly measuring wall-clock or monetary cost.

5.4. Dependency on the number of evaluation generations per task

We now isolate the dependency on the per-task evaluation budget m. Figure 4a reports MPBB error as a function of m. Bayesian estimators help most at low m; naive and combinatorial estimates catch up as m grows. The plot also motivates *LinMix*: BB is often more stable at very small m, whereas ZOIBB can benefit from greater tail flexibility at intermediate budgets.

5.5. Dependency on the number of tasks

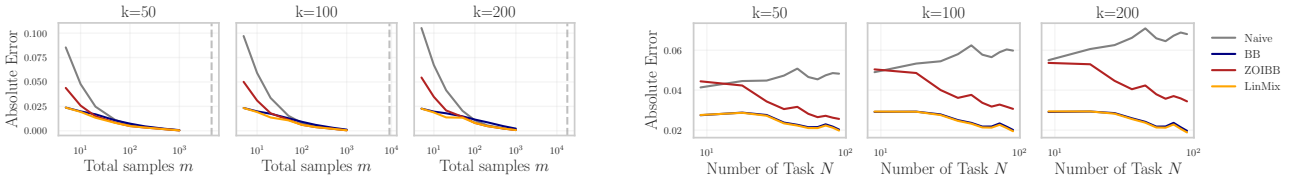
Since the improvement of the Bayesian estimators stems from pooling statistical strength across tasks, we study how estimation accuracy scales with the number of tasks N used to compute the dataset average. Figure 4b fixes $m = 10$ and varies the number of MPBB tasks. All estimators improve with larger N, and Bayesian pooling helps when the fitted shared prior is informative; under stronger heterogeneity, this advantage may require stratified fitting.

5.6. Improvability, generation gains, and diversity

We conclude with an analysis of the improvability diagnostic $\Delta\text{Pass}(P)$ introduced in Section 4.1. Figure 5 shows three dataset-specific model-selection stories in the $(\text{Pass@1}, \Delta\text{Pass}(P))$ plane. CodeContests trades off Pass@1 against improvability; MPBB has several Pareto-dominated configurations and a top-right frontier; HumanEval is mostly near zero, with only a few configurations escaping the zero-success region. To quantify the relationship between $\Delta\text{Pass}(P)$ and gains as k increases, we compute correlations between $\Delta\text{Pass}(P)$ and the relative improvement $(\text{Pass@200} - \text{Pass@1})/\text{Pass@1}$ when this quantity is defined. Table 2 shows a positive correlation between $\Delta\text{Pass}(P)$ and relative improvement on the configurations for which the relative improvement is defined. Configurations with zero unrounded Pass@1 are excluded from this ratio-based correlation.

Table 1. Absolute error of Pass@k estimation on CodeContests, HumanEval, and MPBB for several values of m (evaluation generations per task) and k (generation budget). We compare the naive estimator, the combinatorial estimator, BB, ZOIBB, and *LinMix*. Results are averaged over models and temperatures (10 runs). Best results are in **bold**.

Data	m	k	Naive	BB	ZOIBB	LinMix
CodeContest	1	50	0.134	0.105	**0.066**	0.105
		100	0.158	0.123	**0.077**	0.123
		200	0.181	0.138	**0.086**	0.138
	20	50	0.035	0.017	0.019	**0.016**
		100	0.056	0.022	0.026	**0.021**
		200	0.079	0.027	0.035	**0.026**
	100	50	0.009	**0.007**	**0.007**	**0.007**
		100	0.016	**0.010**	**0.010**	**0.010**
		200	0.028	**0.013**	0.014	0.014
HumanEval	1	50	**0.006**	**0.006**	**0.006**	**0.006**
		100	**0.011**	**0.011**	**0.011**	**0.011**
		200	**0.021**	**0.021**	**0.021**	**0.021**
	20	50	**0.004**	**0.004**	**0.004**	**0.004**
		100	0.009	**0.007**	**0.007**	**0.007**
		200	0.019	**0.014**	**0.014**	**0.014**
	100	50	0.002	**0.001**	**0.001**	**0.001**
		100	0.005	**0.003**	**0.003**	**0.003**
		200	0.012	**0.005**	**0.005**	**0.005**
MPBB	1	50	0.265	0.076	**0.060**	0.076
		100	0.276	0.068	**0.061**	0.068
		200	0.285	0.061	**0.060**	0.061
	20	50	0.025	0.017	0.015	**0.013**
		100	0.034	0.017	0.018	**0.014**
		200	0.041	0.018	0.021	**0.014**
	100	50	0.006	0.007	**0.005**	**0.005**
		100	0.008	0.009	**0.006**	**0.006**
		200	0.011	0.011	**0.008**	**0.008**



(a) Dependency on m *(b)* Dependency on N

Figure 4. (a) Absolute error of Pass@k estimation on MPBB as a function of the number of evaluation generations per task m (averaged over all models and temperatures), shown for $k \in \{50, 100, 200\}$. (b) Absolute error of Pass@k estimation on MPBB for a fixed $m = 10$ as a function of the number of tasks N used to compute the dataset average (averaged over all models and temperatures). We focus these detailed m and N curves on MPBB because it has many tasks and richer nonzero success structure; HumanEval and CodeContests are summarized in tables.

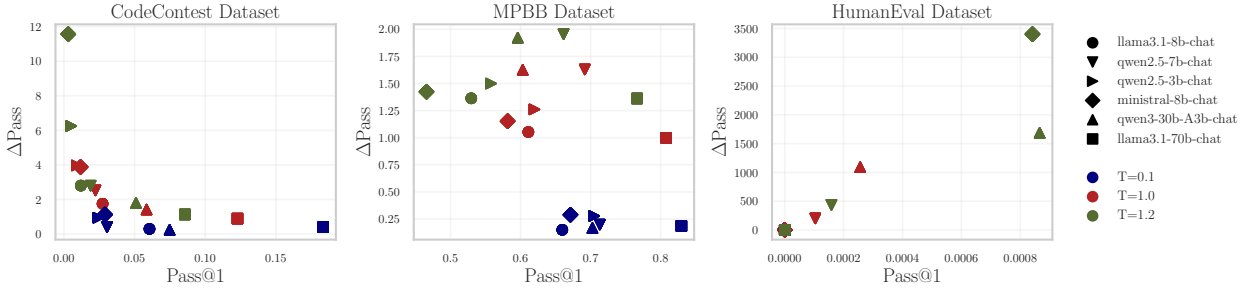


Figure 5. Pass@1 versus improvability. Each panel corresponds to one benchmark dataset, and each point is a (model, temperature) configuration. The x-axis is Pass@1 and the y-axis is $\Delta\text{Pass}(P)$, plotted on the natural scale for each dataset. Markers indicate model and colors indicate decoding temperature.

Finally, we explore the association between diversity and gains. JPlag similarity is negatively correlated with generation gains on MPBB and CodeContests, in , where enough successful samples are available. Table 2 reports these correlations.

Table 2. Correlation between the BB improvability diagnostic $\Delta\text{Pass}(P)$, relative gain $(\text{Pass@200} - \text{Pass@1})/\text{Pass@1}$, and JPlag similarity (higher means more similar solutions). Relative-gain correlations exclude configurations with zero unrounded Pass@1. Similarity correlations are computed only on MPBB and CodeContests, where enough successful samples are available.

	ΔPass	$\frac{\text{Pass@200}-\text{Pass@1}}{\text{Pass@1}}$	Similarity
ΔPass	1.000	0.783	-0.656
$\frac{\text{Pass@200}-\text{Pass@1}}{\text{Pass@1}}$	0.783	1.000	-0.613
Similarity	-0.656	-0.613	1.000

6. Discussion

Why does a low-dimensional prior work? We conjecture that a 2- to 4-parameter prior captures dataset-level difficulty well because of a selection effect on benchmarks: datasets that fail to discriminate among contemporary models—all tasks trivially solved or uniformly unsolved—tend to be abandoned, while surviving benchmarks span a structured range of difficulties relative to the models evaluated. The Beta family is simple but expressive enough to cover the U-shaped, right-skewed, and unimodal regimes that arise from this curation. Our fits are consistent with this view: right-skewed on CodeContests, closer to unimodal on MPBB, boundary-degenerate on HumanEval. A corollary is that the approach should degrade when a benchmark transitions out of its discriminative regime, e.g., when it saturates against newer models—consistent with the boundary-heavy HumanEval results.

Decision regimes and ΔPass. The three benchmarks illustrate complementary regimes. CodeContests exposes a Pass@1–improvability tradeoff; MPBB shows Pareto-dominated and Pareto-favorable configurations; HumanEval is boundary-heavy, with most configurations near zero Pass@k. Reporting only Pass@1 conflates these regimes. The improvability diagnostic $\Delta\text{Pass}(P)$ is exact under the BB approximation (Theorem 4.1) and outside it should be treated as a diagnostic rather than a guarantee. Its raw scale is not comparable across datasets—HumanEval's fitted concentration values reflect prior degeneracy more than improvability—but within a comparable regime it correlates positively with relative gain.

Limitations. The shared prior assumes exchangeability across tasks and may be misspecified on heterogeneous benchmarks, where stratified fitting or richer mixture priors would be more appropriate. Independence between generations can fail under correlated decoding pipelines. *LinMix* is a fixed heuristic rather than a principled selection rule.

7. Conclusion

Pass@k is central to evaluating code-generation models but can require many samples per task when per-sample success probabilities are small. We addressed the low-evaluation-budget regime with empirical-Bayes hierarchical priors—BB, ZOIBB, and their mixture *LinMix*—which improve Pass@k estimation in many low-m settings across CodeContests, MPBB, and the boundary-heavy HumanEval. The BB approximation also yields a k-independent improvability diagnostic $\Delta\text{Pass}(P)$ that orders models at fixed Pass@1 within a comparable regime. Future work includes principled mixture or selection rules, stratified priors for heterogeneous benchmarks, and a clearer link between output diversity and sampling gains.

Acknowledgements

It was also supported by the French government under management of Agence Nationale de la Recherche as part of the "Investissements d'avenir" program, PR[AI]RIE reference ANR-19-P3IA-0001, PR[AI]RIE-PSAI reference 23-IACL-0008 and SHARP reference SHARP ANR-23-PEIA-0008.,This work was granted access to the HPC resources of IDRIS under the allocations 2025-AD011014053R2 and 2025-A0181016159 made by GENCI.

References

Austin, J., Odena, A., Nye, M., Bosma, M., Michalewski, H., Dohan, D., Jiang, E., Cai, C., Terry, M., Le, Q., et al. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*, 2021.

Chen, M., Tworek, J., Jun, H., Yuan, Q., de Oliveira Pinto, H. P., Kaplan, J., Edwards, H., Burda, Y., Joseph, N., Brockman, G., Ray, A., Puri, R., Krueger, G., Petrov, M., Khlaaf, H., Sastry, G., Mishkin, P., Chan, B., Gray, S., Ryder, N., Pavlov, M., Power, A., Kaiser, L., Bavarian, M., Winter, C., Tillet, P., Such, F. P., Cummings, D., Plappert, M., Chantzis, F., Barnes, E., Herbert-Voss, A., Guss, W. H., Nichol, A., Paino, A., Tezak, N., Tang, J., Babuschkin, I., Balaji, S., Jain, S., Saunders, W., Hesse, C., Carr, A. N., Leike, J., Achiam, J., Misra, V., Morikawa, E., Radford, A., Knight, M., Brundage, M., Murati, M., Mayer, K., Welinder, P., McGrew, B., Amodei, D., McCandlish, S., Sutskever, I., and Zaremba, W. Evaluating large language models trained on code, 2021. URL https://arxiv.org/abs/2107.03374.

Grattafiori, A., Dubey, A., Jauhri, A., Pandey, A., and al. The llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

Guo, D., Yang, D., Zhang, H., and Song, J. a. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645(8081):633–638, September 2025. ISSN 1476-4687. doi: 10.1038/s41586-025-09422-z. URL http://dx.doi.org/10.1038/s41586-025-09422-z.

Hariri, M., Samandar, A., Hinczewski, M., and Chaudhary, V. Don't Pass@k: A Bayesian Framework for Large Language Model Evaluation. October 2025. URL https://openreview.net/forum?id=PTXi3Ef4sT.

Hendrycks, D., Basart, S., Kadavath, S., Mazeika, M., Arora, A., Guo, E., Burns, C., Puranik, S., He, H., Song, D., and Steinhardt, J. Measuring coding challenge competence with apps. In Vanschoren, J. and Yeung, S. (eds.), *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1, 2021.

Kazdan, J., Schaeffer, R., Allouah, Y., Sullivan, C., Yu, K., Levi, N., and Koyejo, S. Efficient Prediction of Pass@k Scaling in Large Language Models, October 2025. URL http://arxiv.org/abs/2510.05197. arXiv:2510.05197 [cs].

Le Bronnec, F., Verine, A., Negrevergne, B., Chevaleyre, Y., and Allauzen, A. Exploring precision and recall to assess the quality and diversity of LLMs. In Ku, L.-W., Martins, A., and Srikumar, V. (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 11418–11441, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.616. URL https://aclanthology.org/2024.acl-long.616/.

Le Bronnec, F., Verine, A., Negrevergne, B., and Yokota, R. Beyond pass@k: Redundancy-Aware RLVR for Multi-Sample Code Generation, May 2026. URL http://arxiv.org/abs/2605.28022. arXiv:2605.28022 [cs.CL].

Lee, S., Chon, H., Jang, J., Lee, D., and Yu, H. How diversely can language models solve problems? exploring the algorithmic diversity of model-generated code. In Christodoulopoulos, C., Chakraborty, T., Rose, C., and Peng, V. (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 152–167, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-335-7. doi: 10.18653/v1/2025.findings-emnlp.10. URL https://aclanthology.org/2025.findings-emnlp.10/.

Li, Y., Choi, D., Chung, J., Kushman, N., Schrittwieser, J., Leblond, R., Eccles, T., Keeling, J., Gimeno, F., Lago, A. D., Hubert, T., Choy, P., de Masson d'Autume, C., Babuschkin, I., Chen, X., Huang,

P.-S., Welbl, J., Gowal, S., Cherepanov, A., Molloy, J., Mankowitz, D. J., Robson, E. S., Kohli, P., de Freitas, N., Kavukcuoglu, K., and Vinyals, O. Competition-level code generation with alphacode. *Science*, 378(6624):1092–1097, 2022. doi: 10.1126/science.abq1158. URL https://www.science.org/doi/abs/10.1126/science.abq1158.

Liu, A. H., Khandelwal, K., Subramanian, S., Jouault, V., Rastogi, A., Sadé, A., Jeffares, A., Jiang, A., Cahill, A., Gavaudan, A., Sablayrolles, A., Héliou, A., You, A., Ehrenberg, A., Lo, A., Eliseev, A., Calvi, A., Sooriyarachchi, A., Bout, B., Rozière, B., Monicault, B. D., Lanfranchi, C., Barreau, C., Courtot, C., Grattarola, D., Dabert, D., Casas, D. d. l., Chane-Sane, E., Ahmed, F., Berrada, G., Ecrepont, G., Guinet, G., Novikov, G., Kunsch, G., Lample, G., Martin, G., Gupta, G., Ludziejewski, J., Rute, J., Studnia, J., Amar, J., Delas, J., Roberts, J. S., Yadav, K., Chandu, K., Jain, K., Aitchison, L., Fainsin, L., Blier, L., Zhao, L., Martin, L., Saulnier, L., Gao, L., Buyl, M., Jennings, M., Pellat, M., Prins, M., Poirée, M., Guillaumin, M., Dinot, M., Futeral, M., Darrin, M., Augustin, M., Chiquier, M., Schimpf, M., Grinsztajn, N., Gupta, N., Raghuraman, N., Bousquet, O., Duchenne, O., Wang, P., Platen, P. v., Jacob, P., Wambergue, P., Kurylowicz, P., Muddireddy, P. R., Chagniot, P., Stock, P., Agrawal, P., Torroba, Q., Sauvestre, R., Soletskyi, R., Menneer, R., Vaze, S., Barry, S., Gandhi, S., Waghjale, S., Gandhi, S., Ghosh, S., Mishra, S., Aithal, S., Antoniak, S., Scao, T. L., Cachet, T., Sorg, T. S., Lavril, T., Saada, T. N., Chabal, T., Foubert, T., Robert, T., Wang, T., Lawson, T., Bewley, T., Bewley, T., Edwards, T., Jamil, U., Tomasini, U., Nemychnikova, V., Phung, V., Maladière, V., Richard, V., Bouaziz, W., Li, W.-D., Marshall, W., Li, X., Yang, X., Ouahidi, Y. E., Wang, Y., Tang, Y., and Ramzi, Z. Ministral 3, January 2026. URL http://arxiv.org/abs/2601.08584. arXiv:2601.08584 [cs].

Luettgau, L., Coppock, H., Dubois, M., Summerfield, C., and Ududec, C. Hibayes: A hierarchical bayesian modeling framework for ai evaluation statistics, 2025. URL https://arxiv.org/abs/2505.05602.

Maisch, R. JPlag. October 2023. URL https://helmholtz.software/software/jplag.

OpenAI, Achiam, J., Adler, S., and al. Gpt-4 technical report, 2024. URL https://arxiv.org/abs/2303.08774.

Ospina, R. and Ferrari, S. L. P. Inflated beta distributions. *Statistical Papers*, 51(1):111–126, January 2010. ISSN 1613-9798. doi: 10.1007/s00362-008-0125-4. URL https://doi.org/10.1007/s00362-008-0125-4.

Ospina, R. and Ferrari, S. L. P. A general class of zero-or-one inflated beta regression models. *Computational Statistics & Data Analysis*, 56(6):1609–1623, June 2012. ISSN 0167-9473. doi: 10.1016/j.csda.2011.10.005. URL https://www.sciencedirect.com/science/article/pii/S0167947311003628.

Qwen, Yang, A., Yang, B., Zhang, B., Hui, B., Zheng, B., Yu, B., Li, C., Liu, D., Huang, F., Wei, H., Lin, H., Yang, J., Tu, J., Zhang, J., Yang, J., Yang, J., Zhou, J., Lin, J., Dang, K., Lu, K., Bao, K., Yang, K., Yu, L., Li, M., Xue, M., Zhang, P., Zhu, Q., Men, R., Lin, R., Li, T., Tang, T., Xia, T., Ren, X., Ren, X., Fan, Y., Su, Y., Zhang, Y., Wan, Y., Liu, Y., Cui, Z., Zhang, Z., and Qiu, Z. Qwen2.5 Technical Report, January 2025. URL http://arxiv.org/abs/2412.15115. arXiv:2412.15115 [cs].

Tang, Y., Zheng, K., Synnaeve, G., and Munos, R. Optimizing language models for inference time objectives using reinforcement learning. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=ZVWJO5YTz4.

Wang, T., Liu, Z., Chen, Y., Light, J., Liu, W., Chen, H., Zhang, X., and Cheng, W. On the effect of sampling diversity in scaling llm inference, 2025. URL https://arxiv.org/abs/2502.11027.

Yang, A., Li, A., Yang, B., Zhang, B., Hui, B., Zheng, B., Yu, B., Gao, C., Huang, C., Lv, C., Zheng, C., Liu, D., Zhou, F., Huang, F., Hu, F., Ge, H., Wei, H., Lin, H., Tang, J., Yang, J., Tu, J., Zhang, J., Yang, J., Yang, J., Zhou, J., Zhou, J., Lin, J., Dang, K., Bao, K., Yang, K., Yu, L., Deng, L., Li, M., Xue, M., Li, M., Zhang, P., Wang, P., Zhu, Q., Men, R., Gao, R., Liu, S., Luo, S., Li, T., Tang, T., Yin, W., Ren, X., Wang, X., Zhang, X., Ren, X., Fan, Y., Su, Y., Zhang, Y., Zhang, Y., Wan, Y., Liu, Y., Wang, Z., Cui, Z., Zhang, Z., Zhou, Z., and Qiu, Z. Qwen3 Technical Report, May 2025. URL http://arxiv.org/abs/2505.09388. arXiv:2505.09388 [cs].

Yao, J., Cheng, R., Wu, X., Wu, J., and Tan, K. Diversity-aware policy optimization for large language model reasoning. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=5eZ0iykpDU.

Zheng, K., Decugis, J., Gehring, J., Cohen, T., benjamin negrevergne, and Synnaeve, G. What makes large language models reason in (multi-turn) code generation? In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=Zk9guOl9NS.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

A. Mathematical Supplement

This appendix collects the mathematical details that are used in Section 4 but not derived there. Throughout, for $a, b > 0$ we use the Beta function $\beta(a, b) = \int_0^1 t^{a-1}(1-t)^{b-1}\, dt$ and the Gamma function $\Gamma(\cdot)$, related by

$$\beta(a, b) = \frac{\Gamma(a)\,\Gamma(b)}{\Gamma(a+b)}. \tag{18}$$

A.1. Beta − Binomial marginal likelihood

For a fixed task with latent success probability $p \in [0, 1]$, the Binomial likelihood of observing $s \in \{0, \dots, m\}$ successes out of $m \in \mathbb{N}$ independent trials is

$$p(s \mid m, p) = \binom{m}{s} p^s (1-p)^{m-s}. \tag{19}$$

Under the Beta prior $p \sim \mathrm{Beta}(a, b)$ with density

$$p(p \mid a, b) = \frac{1}{\beta(a, b)}\, p^{a-1}(1-p)^{b-1}, \tag{20}$$

integrating out p yields the Beta–Binomial marginal likelihood

$$\begin{aligned} p(s \mid m, a, b) &= \int_0^1 p(s \mid m, p)\, p(p \mid a, b)\, dp && (21)\\ &= \binom{m}{s} \frac{1}{\beta(a, b)} \int_0^1 p^{a+s-1}(1-p)^{b+m-s-1}\, dp && (22)\\ &= \binom{m}{s} \frac{\beta(a+s,\, b+m-s)}{\beta(a, b)}, && (23) \end{aligned}$$

which is the expression used in (12).

A.2. Posterior conjugacy and posterior-predictive Pass@k

Posterior of p given (m, s). Combining (19) and (20), the posterior density satisfies

$$\begin{aligned} p(p \mid m, s, a, b) &\propto p(s \mid m, p)\, p(p \mid a, b) \propto p^s(1-p)^{m-s}\, p^{a-1}(1-p)^{b-1} \\ &\propto p^{(a+s)-1}(1-p)^{(b+m-s)-1}, \end{aligned}$$

so by inspection

$$p \mid (m, s) \sim \mathrm{Beta}(a+s,\, b+m-s), \tag{24}$$

which is the conjugacy claim used in Section 4.1.

A Beta moment identity. For $p \sim \mathrm{Beta}(a, b)$ and integer $k \geq 0$,

$$\mathbb{E}[(1-p)^k] = \frac{\beta(a,\, b+k)}{\beta(a, b)}. \tag{25}$$

This follows directly from the definition of the Beta function:

$$\begin{aligned} \mathbb{E}[(1-p)^k] &= \int_0^1 (1-p)^k\, \frac{1}{\beta(a, b)} p^{a-1}(1-p)^{b-1}\, dp \\ &= \frac{1}{\beta(a, b)} \int_0^1 p^{a-1}(1-p)^{b+k-1}\, dp = \frac{\beta(a, b+k)}{\beta(a, b)}. \end{aligned}$$

Posterior-predictive $\mathrm{Pass@}k$ **for one task.** Recall $\mathrm{Pass@}k = 1-(1-p)^k$ under the independence assumption across k samples. Using (24) and (25), the posterior-predictive pass@k for a task with counts (m, s) is

$$\mathbb{E}[\mathrm{Pass@}k \mid m, s] = 1 - \mathbb{E}[(1-p)^k \mid m, s] = 1 - \frac{\beta(a+s,\, b+m-s+k)}{\beta(a+s,\, b+m-s)}, \tag{26}$$

which matches the definition of $\widehat{\mathrm{Pass@}k}^{\mathrm{BB}}(\boldsymbol{x})$ in Section 4.1.

A.3. Closed form for unconditional $\mathrm{Pass@}k$ under a Beta law

This subsection derives the closed form used in (13) and in the proof of Theorem 4.1.

Lemma A.1 (Beta ratio as a finite product)**.** *Let $a, b > 0$ and let $k \in \mathbb{N}$. Then*

$$\frac{\beta(a,\, b+k)}{\beta(a,b)} = \prod_{j=0}^{k-1} \frac{b+j}{a+b+j}. \tag{27}$$

Proof. Using (18),

$$\frac{\beta(a,b+k)}{\beta(a,b)} = \frac{\Gamma(a)\Gamma(b+k)}{\Gamma(a+b+k)} \cdot \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)} = \frac{\Gamma(b+k)}{\Gamma(b)} \cdot \frac{\Gamma(a+b)}{\Gamma(a+b+k)}.$$

For integer $k \geq 1$, the Gamma recursion $\Gamma(x+1) = x\Gamma(x)$ yields $\Gamma(b+k) = \Gamma(b)\prod_{j=0}^{k-1}(b+j)$ and $\Gamma(a+b+k) = \Gamma(a+b)\prod_{j=0}^{k-1}(a+b+j)$. Substituting proves (27). □

Unconditional pass@k. If $p \sim \mathrm{Beta}(a,b)$, then by (25) and Lemma A.1,

$$\mathbb{E}[\mathrm{Pass@}k] = 1 - \frac{\beta(a,\, b+k)}{\beta(a,b)} = 1 - \prod_{j=0}^{k-1} \frac{b+j}{a+b+j}. \tag{28}$$

This is the formula referenced as (13) in the main text.

A.4. Full proof of Theorem 4.1

We restate the theorem for completeness.

Theorem A.2 (Pass@k ordering at fixed pass@1)**.** *Let P_1 and P_2 be two models whose per-task success probabilities follow $\mathrm{Beta}(a_1, b_1)$ and $\mathrm{Beta}(a_2, b_2)$, respectively, with $a_i, b_i > 0$. Assume they have the same pass@1, i.e.*

$$\frac{a_1}{a_1+b_1} = \frac{a_2}{a_2+b_2} =: \pi \in (0,1). \tag{29}$$

Define $\Delta\mathrm{Pass}(P_i) := a_i + b_i$. If $\Delta\mathrm{Pass}(P_1) > \Delta\mathrm{Pass}(P_2)$, then for all integers $k \geq 2$,

$$\mathrm{Pass@}k_{\mathcal{D}}(P_1) > \mathrm{Pass@}k_{\mathcal{D}}(P_2). \tag{30}$$

Proof. Let $\Delta_i := a_i + b_i > 0$. Under (29), we can reparameterize

$$a_i = \pi\Delta_i, \qquad b_i = (1-\pi)\Delta_i, \qquad i \in \{1,2\}. \tag{31}$$

For fixed $k \geq 1$, define the unconditional pass@k function under a Beta law by $\mathrm{Pass@}k(\Delta) := \mathbb{E}_{p\sim\mathrm{Beta}(\pi\Delta,(1-\pi)\Delta)}[1-(1-p)^k]$. By (28) and substituting (31),

$$\mathrm{Pass@}k(\Delta) = 1 - \prod_{j=0}^{k-1} \frac{(1-\pi)\Delta + j}{\Delta + j}. \tag{32}$$

The $j=0$ factor equals $\frac{(1-\pi)\Delta}{\Delta} = 1-\pi$, independent of Δ. For each $j \geq 1$, define

$$f_j(\Delta) := \frac{(1-\pi)\Delta + j}{\Delta + j}. \tag{33}$$

A direct differentiation gives, for all $\Delta > 0$,

$$f_j'(\Delta) = \frac{(1-\pi)(\Delta+j) - ((1-\pi)\Delta + j)}{(\Delta+j)^2} = -\frac{\pi j}{(\Delta+j)^2} < 0, \tag{34}$$

so each f_j is strictly decreasing in Δ when $j \geq 1$. For $k \geq 2$, the product in (32) contains at least one such strictly decreasing factor (e.g. $j = 1$), while all factors are positive. Therefore the product $\prod_{j=0}^{k-1} f_j(\Delta)$ is strictly decreasing in Δ for $k \geq 2$, and thus $\mathrm{Pass@}k(\Delta) = 1 - \prod_{j=0}^{k-1} f_j(\Delta)$ is strictly increasing in Δ for $k \geq 2$. Consequently, if $\Delta_1 > \Delta_2$ then $\mathrm{Pass@}k(\Delta_1) > \mathrm{Pass@}k(\Delta_2)$. Applying this with $\Delta_i = a_i + b_i$ yields (30). □

A.5. ZOIBB marginal likelihood and posterior-predictive Pass@k

This subsection provides the derivations underlying the ZOIBB estimator used in Section 4.2.

ZOIB **prior as a mixture.** Let $p \in [0,1]$ denote a latent success probability. Under the zero–one inflated Beta prior $p \sim \mathrm{ZOIB}(a, b, \pi_0, \pi_1)$, we have the mixture representation

$$p \sim \begin{cases} 0, & \text{with probability } \pi_0, \\ 1, & \text{with probability } \pi_1, \\ \tilde{p}, & \text{with probability } 1-\pi_0-\pi_1, \text{ where } \tilde{p} \sim \mathrm{Beta}(a,b). \end{cases} \tag{35}$$

ZOIBB **marginal likelihood.** Given $m \in \mathbb{N}$ and $s \in \{0, \ldots, m\}$, the Binomial likelihood is $p(s \mid m, p) = \binom{m}{s} p^s (1-p)^{m-s}$. Integrating this likelihood against the mixture prior (35) yields the ZOIBB marginal likelihood

$$p(s \mid m, a, b, \pi_0, \pi_1) \;=\; \pi_0 \, \mathbb{1}_{\{s=0\}} + \pi_1 \, \mathbb{1}_{\{s=m\}} + (1-\pi_0-\pi_1) \binom{m}{s} \frac{\beta(a+s,\, b+m-s)}{\beta(a,b)}. \tag{36}$$

Posterior decomposition given (m, s). The posterior $p(p \mid m, s)$ is again a three-component mixture. Define the unnormalized component weights

$$\tilde{w}_0 \;=\; \pi_0 \, \mathbb{1}_{\{s=0\}}, \tag{37}$$

$$\tilde{w}_1 \;=\; \pi_1 \, \mathbb{1}_{\{s=m\}}, \tag{38}$$

$$\tilde{w}_{\mathrm{B}} \;=\; (1-\pi_0-\pi_1) \binom{m}{s} \frac{\beta(a+s,\, b+m-s)}{\beta(a,b)}, \tag{39}$$

and let $Z = \tilde{w}_0 + \tilde{w}_1 + \tilde{w}_{\mathrm{B}}$ with normalized weights $w_0 = \tilde{w}_0/Z$, $w_1 = \tilde{w}_1/Z$, and $w_{\mathrm{B}} = \tilde{w}_{\mathrm{B}}/Z$. Conditioned on the Beta-interior component, conjugacy gives

$$p \mid (m, s, \text{interior}) \sim \mathrm{Beta}(a+s,\, b+m-s). \tag{40}$$

Thus the full posterior is the mixture: point mass at 0 with weight w_0, point mass at 1 with weight w_1, and a $\mathrm{Beta}(a+s, b+m-s)$ density on $(0,1)$ with weight w_{B}.

In particular, if $0 < s < m$ then the boundary components are impossible under the data and $w_0 = w_1 = 0$, so the posterior reduces to the conjugate Beta update. If $s = 0$ (resp. $s = m$), only the spike at 0 (resp. 1) competes with the continuous component.

Posterior-predictive pass@k (case decomposition). Recall $\mathrm{Pass@}k = 1 - (1-p)^k$. Using the posterior mixture weights defined above, the posterior-predictive pass@k is

$$\mathbb{E}[\mathrm{Pass@}k \mid m, s] \;=\; w_1 + w_{\mathrm{B}} \left(1 - \frac{\beta(a+s,\, b+m-s+k)}{\beta(a+s,\, b+m-s)}\right), \tag{41}$$

where w_1 is the posterior weight of the spike at 1 and w_{B} is the posterior weight of the Beta-interior component. This expression simplifies substantially by cases.

For convenience, define the (unnormalized) Beta-interior evidence term

$$r(m,s) \;\coloneqq\; (1-\pi_0-\pi_1)\binom{m}{s}\frac{\beta(a+s,\,b+m-s)}{\beta(a,b)}. \tag{42}$$

Interior counts ($0 < s < m$). In this case the spikes at 0 and 1 are incompatible with the data, hence $w_0 = w_1 = 0$ and $w_{\mathrm{B}} = 1$. Therefore

$$\mathbb{E}[\mathrm{Pass@}k \mid m,s] \;=\; 1-\frac{\beta(a+s,\,b+m-s+k)}{\beta(a+s,\,b+m-s)}, \qquad (0<s<m), \tag{43}$$

which coincides exactly with the Beta–Binomial posterior-predictive expression.

All failures ($s = 0$). Here the posterior is a two-component mixture between a structural zero ($p = 0$) and the Beta-interior component. The unnormalized weights are π_0 for the spike at 0 and $r(m, 0)$ for the Beta interior, hence

$$w_{\mathrm{B}} = \frac{r(m,0)}{\pi_0 + r(m,0)}, \qquad w_1 = 0. \tag{44}$$

Consequently,

$$\mathbb{E}[\mathrm{Pass@}k \mid m,0] \;=\; \frac{r(m,0)}{\pi_0 + r(m,0)}\left(1-\frac{\beta(a,\,b+m+k)}{\beta(a,\,b+m)}\right). \tag{45}$$

All successes ($s = m$). Here the posterior is a two-component mixture between a structural one ($p = 1$) and the Beta-interior component. The unnormalized weights are π_1 for the spike at 1 and $r(m, m)$ for the Beta interior, hence

$$w_1 = \frac{\pi_1}{\pi_1 + r(m,m)}, \qquad w_{\mathrm{B}} = \frac{r(m,m)}{\pi_1 + r(m,m)}. \tag{46}$$

Consequently,

$$\mathbb{E}[\mathrm{Pass@}k \mid m,m] \;=\; \frac{\pi_1}{\pi_1 + r(m,m)} + \frac{r(m,m)}{\pi_1 + r(m,m)}\left(1-\frac{\beta(a+m,\,b+k)}{\beta(a+m,\,b)}\right). \tag{47}$$

Equations (43)-(45)-(47) are the closed-form expressions used for $\widehat{\mathrm{Pass@}k}^{\mathrm{ZOIBB}}(\boldsymbol{x})$ in Section 4.2.

B. Additional experimental results

This appendix reports the complete tables underlying the summary results in Section 5. Unless stated otherwise, results are aggregated over all (model, temperature) configurations used in the main experiments, and uncertainty is reported over repeated runs of the evaluation procedure.

Goodness-of-fit and model selection. Table 3 reports the fitted hyperparameters for the BB and ZOIBB hierarchical models, as well as their comparative predictive fit. For each (dataset, model, temperature) setting, we fit hyperparameters by empirical Bayes (evidence maximization) and compare the two likelihood families using 10-fold cross-validated expected log predictive density (CV-ELPD) computed from the corresponding marginal likelihoods. A higher CV-ELPD indicates better out-of-fold predictive performance. Two qualitative patterns are worth noting. First, some configurations prefer the simpler BB fit, while others prefer ZOIBB, consistent with the fact that boundary mass at 0 and 1 is not uniformly present across models and decoding temperatures. Second, HumanEval is qualitatively different from MBPP and CodeContests: for many settings all sampled attempts fail, so the fitted models place nearly all mass near zero and ZOIBB is often selected because its explicit zero atom matches this boundary behavior. Third, the fitted Beta-shape parameters often indicate boundary-heavy regimes that match the empirical prevalence of near-impossible and near-trivial tasks.

Table 3. Fitted parameters for the $\mathrm{Beta} - \mathrm{Binomial}$ model and the ZOIBB model on CodeContests, HumanEval, and MBPP for various models and temperatures. HumanEval includes several near-degenerate zero-success rows; rounded values such as $a = 0.00$ should be read as very small fitted parameters.

Dataset	Model	Temperature	a_{BB}	b_{BB}	a_{ZOIBB}	b_{ZOIBB}	π_0	π_1	Better Fit	ELPD BB	ELPD ZOIBB
CodeContest	llama3.1-70b-chat	0.1	0.06	0.34	0.12	0.37	0.35	0.00	BB	-15289.73	-15292.21
		1.0	0.11	0.78	0.12	0.79	0.04	0.00	BB	-19428.37	-19428.75
		1.2	0.10	1.04	0.19	1.23	0.32	0.00	BB	-16036.09	-16036.59
	llama3.1-8b-chat	0.1	0.02	0.27	0.03	0.57	0.20	0.02	ZOIBB	-5995.77	-5994.36
		1.0	0.05	1.71	0.05	1.72	0.06	0.00	BB	-6479.82	-6479.89
		1.2	0.03	2.77	0.15	3.76	0.71	0.00	BB	-3980.45	-3981.85
	ministral-8b-chat	0.1	0.03	1.10	0.09	1.24	0.58	0.00	BB	-5976.73	-5977.66
		1.0	0.04	3.84	0.21	5.69	0.67	0.00	BB	-4309.52	-4310.25
		1.2	0.04	11.53	0.39	23.24	0.81	0.00	BB	-1565.75	-1568.16
	qwen2.5-3b-chat	0.1	0.03	0.91	0.03	0.90	0.07	0.00	BB	-4393.94	-4394.02
		1.0	0.04	3.93	0.07	4.22	0.36	0.00	BB	-3051.32	-3058.21
		1.2	0.03	6.22	0.03	6.12	0.02	0.00	BB	-1923.07	-1926.70
	qwen2.5-7b-chat	0.1	0.02	0.39	0.02	0.92	0.18	0.01	BB	-4307.60	-4310.69
		1.0	0.06	2.46	0.06	2.45	0.01	0.00	BB	-6361.16	-6361.29
		1.2	0.05	2.74	0.08	2.96	0.30	0.00	BB	-5875.90	-5876.83
	qwen3-30b-A3b-chat	0.1	0.02	0.22	0.03	0.35	0.04	0.02	BB	-4927.78	-4928.29
		1.0	0.08	1.34	0.18	1.66	0.42	0.00	BB	-13273.32	-13274.00
		1.2	0.08	1.71	0.13	1.86	0.24	0.00	BB	-12815.33	-12816.35
HumanEval	llama3.1-70b-chat	0.1	0.00	2.02	0.00	2.00	0.52	0.00	ZOIBB	-0.00	-0.00
		1.0	0.00	2.02	0.00	2.00	0.52	0.00	ZOIBB	-0.00	-0.00
		1.2	0.00	2.02	0.00	2.00	0.52	0.00	ZOIBB	-0.00	-0.00
	llama3.1-8b-chat	0.1	0.00	2.02	0.00	2.00	0.52	0.00	ZOIBB	-0.00	-0.00
		1.0	0.00	2.02	0.00	2.00	0.52	0.00	ZOIBB	-0.00	-0.00
		1.2	0.00	2.02	0.00	2.00	0.52	0.00	ZOIBB	-0.00	-0.00
	ministral-8b-chat	0.1	0.00	2.02	0.00	2.00	0.52	0.00	ZOIBB	-0.00	-0.00
		1.0	0.00	2.02	0.00	2.00	0.52	0.00	ZOIBB	-0.00	-0.00
		1.2	2.87	3399.65	251.58	212152.28	0.27	0.00	ZOIBB	-1114.49	-1113.03
	qwen2.5-3b-chat	0.1	0.00	2.02	0.00	2.00	0.52	0.00	ZOIBB	-0.00	-0.00
		1.0	0.00	2.02	0.00	2.00	0.52	0.00	ZOIBB	-0.00	-0.00
		1.2	0.00	2.02	0.00	2.00	0.52	0.00	ZOIBB	-0.00	-0.00
	qwen2.5-7b-chat	0.1	0.00	2.02	0.00	2.00	0.52	0.00	ZOIBB	-0.00	-0.00
		1.0	0.02	200.22	0.05	209.18	0.53	0.00	BB	-141.78	-142.00
		1.2	0.07	435.71	0.09	453.37	0.20	0.00	BB	-232.25	-232.53
	qwen3-30b-A3b-chat	0.1	0.00	2.02	0.00	2.00	0.52	0.00	ZOIBB	-0.00	-0.00
		1.0	0.28	1090.56	5.22	6153.54	0.70	0.00	BB	-379.91	-379.91
		1.2	1.46	1685.18	1.47	1698.90	0.00	0.00	ZOIBB	-1140.57	-1135.00
MPBB	llama3.1-70b-chat	0.1	0.15	0.03	0.15	0.03	0.00	0.01	BB	-4309.05	-4310.80
		1.0	0.76	0.24	1.14	0.27	0.02	0.00	ZOIBB	-22103.22	-22098.18
		1.2	0.96	0.40	1.44	0.46	0.02	0.00	ZOIBB	-28861.85	-28854.25
	llama3.1-8b-chat	0.1	0.10	0.05	0.19	0.14	0.10	0.37	BB	-9618.17	-9619.46
		1.0	0.60	0.45	0.79	0.50	0.03	0.00	ZOIBB	-33884.55	-33881.81
		1.2	0.68	0.68	0.79	0.73	0.02	0.00	ZOIBB	-39351.54	-39351.20
	ministral-8b-chat	0.1	0.20	0.09	0.24	0.17	0.02	0.21	BB	-15063.55	-15066.16
		1.0	0.64	0.52	0.85	0.58	0.03	0.00	ZOIBB	-36372.46	-36369.46
		1.2	0.64	0.79	0.79	0.88	0.03	0.00	BB	-39580.29	-39581.50
	qwen2.5-3b-chat	0.1	0.19	0.09	0.34	0.19	0.07	0.26	BB	-14769.23	-14770.01
		1.0	0.73	0.53	0.80	0.55	0.01	0.00	BB	-36161.61	-36162.66
		1.2	0.79	0.71	0.86	0.74	0.01	0.00	BB	-40288.94	-40289.90
	qwen2.5-7b-chat	0.1	0.14	0.06	0.15	0.07	0.01	0.03	BB	-9621.19	-9623.13
		1.0	1.02	0.61	1.18	0.66	0.01	0.00	BB	-36754.21	-36754.74
		1.2	1.18	0.78	1.39	0.85	0.01	0.00	ZOIBB	-40995.93	-40995.67
	qwen3-30b-A3b-chat	0.1	0.12	0.05	0.11	0.05	0.00	0.01	BB	-6819.11	-6819.25
		1.0	0.87	0.76	1.32	0.94	0.03	0.00	ZOIBB	-43662.96	-43655.90
		1.2	1.02	0.90	1.65	1.19	0.03	0.00	ZOIBB	-46447.16	-46436.16

Absolute error across (m, k) on CodeContests. The next table reports absolute errors $|\widehat{\mathrm{Pass@}k} - \mathrm{Pass@}k^{\star}|$ for several values of m (samples per task) and target budgets k. The reference value $\mathrm{Pass@}k^{\star}$ is computed using the large-sample combinatorial estimator (Section 2) on the full sampling budget available for each task. Each entry is then obtained by subsampling m generations per task, computing each estimator from the resulting counts (m_i, s_i), and comparing to $\mathrm{Pass@}k^{\star}$. Numbers are averaged over all models and temperatures and over 10 independent subsampling runs; parentheses report the standard deviation across these runs. In these logged experiments, Bayesian estimators often reduce error in the low-sampling regime, especially for larger k, where naive plug-in estimates tend to underestimate the large-budget reference.

Table 4. Absolute error of Pass@k estimation on CodeContests for various values of m (number of samples per problem) and k (sampling budget). We compare the naive estimator, the combinatorial estimator, the BB estimator, the ZOIBB estimator, and *LinMix*. Results are averaged over all models and temperatures over 10 runs. Standard deviations are in parentheses.

Data	m	k	Naive	Combinatorial	BB	ZOIBB	LinMix
CodeContest	1	10	0.076 (0.050)	-	0.058 (0.061)	**0.038 (0.031)**	0.058 (0.061)
		20	0.101 (0.061)	-	0.078 (0.079)	**0.050 (0.041)**	0.078 (0.079)
		50	0.134 (0.073)	-	0.105 (0.099)	**0.066 (0.053)**	0.105 (0.099)
		100	0.158 (0.079)	-	0.123 (0.110)	**0.077 (0.060)**	0.123 (0.110)
		200	0.181 (0.085)	-	0.138 (0.118)	**0.086 (0.067)**	0.138 (0.118)
		500	0.212 (0.094)	-	0.155 (0.125)	**0.097 (0.073)**	0.155 (0.125)
	5	10	0.031 (0.023)	-	**0.019 (0.014)**	0.020 (0.016)	**0.019 (0.014)**
		20	0.053 (0.033)	-	**0.028 (0.022)**	0.033 (0.027)	**0.028 (0.022)**
		50	0.085 (0.045)	-	**0.045 (0.041)**	0.058 (0.049)	**0.045 (0.041)**
		100	0.108 (0.051)	-	**0.063 (0.068)**	0.083 (0.078)	**0.063 (0.068)**
		200	0.132 (0.057)	-	**0.083 (0.107)**	0.111 (0.117)	**0.083 (0.107)**
		500	0.163 (0.066)	-	**0.111 (0.160)**	0.148 (0.161)	**0.111 (0.160)**
	10	10	0.018 (0.015)	0.012 (0.011)	0.012 (0.010)	**0.011 (0.011)**	0.012 (0.010)
		20	0.031 (0.022)	-	**0.017 (0.014)**	0.018 (0.016)	**0.017 (0.014)**
		50	0.058 (0.032)	-	**0.026 (0.023)**	0.030 (0.025)	**0.026 (0.023)**
		100	0.082 (0.039)	-	**0.035 (0.035)**	0.042 (0.036)	**0.035 (0.034)**
		200	0.105 (0.045)	-	0.045 (0.052)	0.054 (0.053)	**0.044 (0.052)**
		500	0.137 (0.053)	-	0.057 (0.080)	0.072 (0.076)	**0.056 (0.080)**
	20	10	0.010 (0.008)	**0.007 (0.006)**	**0.007 (0.006)**	**0.007 (0.006)**	**0.007 (0.006)**
		20	0.017 (0.013)	0.012 (0.010)	**0.010 (0.009)**	0.011 (0.009)	**0.010 (0.009)**
		50	0.035 (0.020)	-	0.017 (0.014)	0.019 (0.015)	**0.016 (0.014)**
		100	0.056 (0.026)	-	0.022 (0.019)	0.026 (0.020)	**0.021 (0.018)**
		200	0.079 (0.031)	-	0.027 (0.024)	0.035 (0.026)	**0.026 (0.022)**
		500	0.110 (0.039)	-	0.034 (0.032)	0.047 (0.035)	**0.033 (0.028)**
	50	10	0.005 (0.004)	**0.004 (0.004)**	**0.004 (0.004)**	**0.004 (0.004)**	**0.004 (0.004)**
		20	0.008 (0.007)	**0.007 (0.005)**	**0.007 (0.005)**	**0.007 (0.005)**	**0.007 (0.005)**
		50	0.017 (0.012)	0.011 (0.009)	**0.010 (0.009)**	**0.010 (0.009)**	**0.010 (0.009)**
		100	0.029 (0.016)	-	**0.014 (0.012)**	0.015 (0.012)	0.015 (0.012)
		200	0.047 (0.021)	-	**0.018 (0.015)**	0.021 (0.017)	0.021 (0.017)
		500	0.078 (0.029)	-	**0.024 (0.020)**	0.032 (0.026)	0.031 (0.025)
	100	10	**0.003 (0.002)**	**0.003 (0.002)**	**0.003 (0.002)**	**0.003 (0.002)**	**0.003 (0.002)**
		20	0.005 (0.004)	**0.004 (0.003)**	**0.004 (0.003)**	**0.004 (0.003)**	**0.004 (0.003)**
		50	0.009 (0.007)	**0.007 (0.005)**	**0.007 (0.005)**	**0.007 (0.005)**	**0.007 (0.005)**
		100	0.016 (0.010)	0.011 (0.008)	**0.010 (0.007)**	**0.010 (0.008)**	**0.010 (0.008)**
		200	0.028 (0.014)	-	**0.013 (0.009)**	0.014 (0.011)	0.014 (0.011)
		500	0.055 (0.021)	-	**0.018 (0.013)**	0.023 (0.018)	0.023 (0.018)
	200	10	**0.002 (0.002)**	**0.002 (0.002)**	**0.002 (0.002)**	**0.002 (0.002)**	**0.002 (0.002)**
		20	**0.003 (0.003)**	**0.003 (0.002)**	**0.003 (0.002)**	**0.003 (0.002)**	**0.003 (0.002)**
		50	0.006 (0.004)	**0.004 (0.003)**	**0.004 (0.003)**	**0.004 (0.003)**	**0.004 (0.003)**
		100	0.009 (0.006)	**0.006 (0.005)**	**0.006 (0.005)**	**0.006 (0.005)**	**0.006 (0.005)**
		200	0.016 (0.010)	0.011 (0.008)	**0.008 (0.006)**	0.009 (0.007)	0.009 (0.007)
		500	0.035 (0.017)	-	**0.013 (0.010)**	0.016 (0.012)	0.016 (0.012)
	500	10	**0.001 (0.001)**	**0.001 (0.001)**	**0.001 (0.001)**	**0.001 (0.001)**	**0.001 (0.001)**
		20	**0.001 (0.001)**	**0.001 (0.001)**	**0.001 (0.001)**	**0.001 (0.001)**	**0.001 (0.001)**
		50	0.003 (0.002)	**0.002 (0.002)**	**0.002 (0.002)**	**0.002 (0.002)**	**0.002 (0.002)**
		100	0.004 (0.003)	**0.003 (0.003)**	**0.003 (0.003)**	**0.003 (0.003)**	**0.003 (0.003)**
		200	0.007 (0.005)	**0.005 (0.004)**	**0.005 (0.004)**	**0.005 (0.004)**	**0.005 (0.004)**
		500	0.016 (0.010)	0.009 (0.007)	**0.008 (0.006)**	**0.008 (0.007)**	**0.008 (0.007)**

Regime behavior across m. A noteworthy pattern in these full tables is the existence of distinct sampling regimes. At $m = 1$, the ZOIBB estimator often achieves the lowest absolute error, consistent with the utility of explicit boundary mass when only a single sample is observed. For intermediate budgets (roughly $m \in \{5, 10\}$), BB is frequently best, suggesting that the simpler prior can be more stable when the evidence is still sparse but no longer purely boundary-driven. Across many reported (m, k) pairs, ***LinMix*** is at (or close to) the best method, providing an empirical compromise. In particular, for $m \in [20, 50]$ we often observe ***LinMix*** outperforming both BB and ZOIBB, which suggests that one method tends to overestimate while the other underestimates in this range; the linear interpolation can therefore reduce bias.

Absolute error across (m, k) on MBPP. This table repeats the same analysis on MBPP. MBPP typically contains many

tasks and a rich range of nonzero success probabilities, so it is the most useful dataset for the detailed m- and N-dependence analyses in the main text. As in CodeContests, the Bayesian estimators provide regularization when successes are rare. In many (m, k) settings, BB and ZOIBB exhibit complementary strengths; ***LinMix*** is designed as a heuristic compromise across sampling regimes, while remaining task-agnostic (the mixture weight depends only on m).

Table 5. Absolute error of Pass@k estimation on MBPP for various values of m (number of samples per problem) and k (sampling budget). We compare the naive estimator, the combinatorial estimator, the BB estimator, the ZOIBB estimator, and *LinMix*. Results are averaged over all models and temperatures over 10 runs. Standard deviations are in parentheses.

Data	m	k	Naive	Combinatorial	BB	ZOIBB	LinMix
MPBB	1	10	0.217 (0.111)	-	0.090 (0.065)	**0.056 (0.035)**	0.090 (0.065)
		20	0.242 (0.119)	-	0.086 (0.063)	**0.059 (0.027)**	0.086 (0.063)
		50	0.265 (0.124)	-	0.076 (0.060)	**0.060 (0.022)**	0.076 (0.060)
		100	0.276 (0.124)	-	0.068 (0.056)	**0.061 (0.023)**	0.068 (0.056)
		200	0.285 (0.123)	-	0.061 (0.052)	**0.060 (0.025)**	0.061 (0.052)
		500	0.292 (0.120)	-	**0.055 (0.047)**	0.060 (0.027)	**0.055 (0.047)**
	5	10	0.047 (0.030)	-	**0.020 (0.015)**	0.023 (0.020)	**0.020 (0.015)**
		20	0.064 (0.038)	-	**0.023 (0.015)**	0.033 (0.028)	**0.023 (0.015)**
		50	0.085 (0.045)	-	**0.024 (0.015)**	0.044 (0.036)	**0.024 (0.015)**
		100	0.097 (0.047)	-	**0.023 (0.015)**	0.050 (0.040)	**0.023 (0.015)**
		200	0.105 (0.047)	-	**0.023 (0.014)**	0.055 (0.042)	**0.023 (0.014)**
		500	0.113 (0.047)	-	**0.022 (0.013)**	0.059 (0.044)	**0.022 (0.013)**
	10	10	0.022 (0.015)	0.013 (0.011)	0.015 (0.011)	**0.012 (0.010)**	0.015 (0.010)
		20	0.031 (0.021)	-	0.018 (0.012)	**0.017 (0.014)**	0.018 (0.011)
		50	0.048 (0.027)	-	0.020 (0.012)	0.026 (0.020)	**0.019 (0.012)**
		100	0.059 (0.029)	-	0.020 (0.012)	0.031 (0.023)	**0.019 (0.012)**
		200	0.067 (0.030)	-	0.020 (0.012)	0.035 (0.026)	**0.019 (0.012)**
		500	0.075 (0.031)	-	0.020 (0.011)	0.038 (0.028)	**0.019 (0.011)**
	20	10	0.012 (0.010)	**0.008 (0.007)**	0.011 (0.008)	**0.008 (0.007)**	0.009 (0.007)
		20	0.016 (0.013)	0.011 (0.009)	0.014 (0.009)	**0.010 (0.009)**	0.012 (0.008)
		50	0.025 (0.018)	-	0.017 (0.011)	0.015 (0.013)	**0.013 (0.009)**
		100	0.034 (0.020)	-	0.017 (0.012)	0.018 (0.016)	**0.014 (0.009)**
		200	0.041 (0.022)	-	0.018 (0.012)	0.021 (0.018)	**0.014 (0.009)**
		500	0.049 (0.023)	-	0.019 (0.011)	0.025 (0.020)	**0.014 (0.009)**
	50	10	0.006 (0.005)	0.004 (0.003)	0.006 (0.004)	**0.005 (0.003)**	**0.005 (0.003)**
		20	0.007 (0.006)	**0.006 (0.004)**	0.008 (0.005)	**0.006 (0.004)**	**0.006 (0.004)**
		50	0.011 (0.009)	0.009 (0.007)	0.011 (0.008)	**0.008 (0.006)**	**0.008 (0.006)**
		100	0.015 (0.011)	-	0.013 (0.009)	**0.011 (0.009)**	**0.011 (0.008)**
		200	0.020 (0.014)	-	0.015 (0.010)	0.014 (0.010)	**0.013 (0.010)**
		500	0.027 (0.017)	-	**0.017 (0.010)**	0.018 (0.013)	**0.017 (0.012)**
	100	10	**0.003 (0.003)**	**0.003 (0.003)**	0.004 (0.003)	**0.003 (0.003)**	**0.003 (0.003)**
		20	**0.004 (0.004)**	**0.004 (0.003)**	0.005 (0.004)	**0.004 (0.003)**	**0.004 (0.003)**
		50	0.006 (0.005)	**0.005 (0.004)**	0.007 (0.005)	**0.005 (0.004)**	**0.005 (0.004)**
		100	0.008 (0.007)	0.007 (0.005)	0.009 (0.007)	**0.006 (0.005)**	**0.006 (0.005)**
		200	0.011 (0.009)	-	0.011 (0.009)	**0.008 (0.007)**	**0.008 (0.007)**
		500	0.016 (0.013)	-	0.014 (0.009)	**0.011 (0.009)**	**0.011 (0.009)**
	200	10	**0.002 (0.002)**	**0.002 (0.002)**	**0.002 (0.002)**	**0.002 (0.002)**	**0.002 (0.002)**
		20	**0.003 (0.002)**	**0.003 (0.002)**	**0.003 (0.002)**	**0.003 (0.002)**	**0.003 (0.002)**
		50	0.004 (0.003)	**0.003 (0.003)**	0.005 (0.003)	**0.003 (0.003)**	**0.003 (0.003)**
		100	0.005 (0.004)	**0.004 (0.003)**	0.006 (0.005)	**0.004 (0.003)**	**0.004 (0.003)**
		200	0.006 (0.006)	0.006 (0.006)	0.009 (0.006)	**0.005 (0.005)**	**0.005 (0.005)**
		500	0.008 (0.009)	-	0.012 (0.007)	**0.007 (0.007)**	**0.007 (0.007)**
	500	10	**0.001 (0.001)**	**0.001 (0.001)**	**0.001 (0.001)**	**0.001 (0.001)**	**0.001 (0.001)**
		20	**0.001 (0.001)**	**0.001 (0.001)**	**0.001 (0.001)**	**0.001 (0.001)**	**0.001 (0.001)**
		50	**0.002 (0.001)**	**0.002 (0.001)**	**0.002 (0.002)**	**0.002 (0.001)**	**0.002 (0.001)**
		100	**0.002 (0.002)**	**0.002 (0.002)**	0.003 (0.002)	**0.002 (0.002)**	**0.002 (0.002)**
		200	**0.002 (0.003)**	**0.002 (0.002)**	0.005 (0.003)	**0.002 (0.002)**	**0.002 (0.002)**
		500	0.004 (0.004)	**0.003 (0.004)**	0.008 (0.005)	**0.003 (0.003)**	**0.003 (0.003)**

Absolute error across (m, k) on HumanEval. HumanEval behaves differently because most reference Pass@k values are zero or near zero for most configurations. Many estimators therefore tie at very small absolute error. The most informative HumanEval signal is not the absolute-error table alone, but the boundary-mass behavior in Table 3 and the few

high-temperature configurations with nonzero pass@200 in Table 10.

Table 6. Absolute error of Pass@k estimation on HumanEval for various values of m (number of samples per problem) and k (sampling budget). We compare the naive estimator, the combinatorial estimator, the BB estimator, the ZOIBB estimator, and *LinMix*. Results are averaged over all models and temperatures over 10 runs. Standard deviations are in parentheses.

Data	m	k	Naive	Combinatorial	BB	ZOIBB	LinMix
HumanEval	1	10	**0.001 (0.003)**	-	**0.001 (0.003)**	**0.001 (0.003)**	**0.001 (0.003)**
		20	**0.002 (0.005)**	-	0.003 (0.006)	0.003 (0.006)	0.003 (0.006)
		50	**0.006 (0.013)**	-	**0.006 (0.013)**	**0.006 (0.013)**	**0.006 (0.013)**
		100	**0.011 (0.025)**	-	**0.011 (0.024)**	**0.011 (0.024)**	**0.011 (0.024)**
		200	**0.021 (0.047)**	-	**0.021 (0.046)**	**0.021 (0.046)**	**0.021 (0.046)**
		500	0.045 (0.100)	-	**0.044 (0.099)**	**0.044 (0.099)**	**0.044 (0.099)**
	5	10	**0.001 (0.002)**	-	0.002 (0.004)	0.002 (0.003)	0.002 (0.004)
		20	**0.002 (0.005)**	-	0.003 (0.007)	0.003 (0.007)	0.003 (0.007)
		50	**0.005 (0.012)**	-	0.007 (0.017)	0.007 (0.016)	0.007 (0.017)
		100	**0.011 (0.024)**	-	0.014 (0.032)	0.013 (0.030)	0.014 (0.032)
		200	**0.021 (0.046)**	-	0.025 (0.057)	0.024 (0.053)	0.025 (0.057)
		500	**0.045 (0.100)**	-	0.050 (0.111)	**0.045 (0.099)**	0.050 (0.111)
	10	10	**0.001 (0.002)**	**0.001 (0.003)**	**0.001 (0.003)**	**0.001 (0.002)**	**0.001 (0.003)**
		20	**0.002 (0.003)**	-	**0.002 (0.005)**	**0.002 (0.005)**	**0.002 (0.005)**
		50	**0.005 (0.010)**	-	**0.005 (0.012)**	**0.005 (0.012)**	**0.005 (0.012)**
		100	**0.010 (0.022)**	-	**0.010 (0.023)**	**0.010 (0.022)**	**0.010 (0.023)**
		200	0.020 (0.044)	-	0.018 (0.041)	**0.017 (0.039)**	0.018 (0.041)
		500	0.044 (0.098)	-	0.037 (0.081)	**0.032 (0.072)**	0.037 (0.081)
	20	10	**0.001 (0.002)**	**0.001 (0.002)**	**0.001 (0.002)**	**0.001 (0.002)**	**0.001 (0.002)**
		20	**0.001 (0.003)**	0.002 (0.004)	0.002 (0.004)	0.002 (0.004)	0.002 (0.004)
		50	**0.004 (0.009)**	-	**0.004 (0.009)**	**0.004 (0.009)**	**0.004 (0.009)**
		100	0.009 (0.020)	-	**0.007 (0.017)**	**0.007 (0.016)**	**0.007 (0.017)**
		200	0.019 (0.042)	-	**0.014 (0.031)**	**0.014 (0.030)**	**0.014 (0.031)**
		500	0.043 (0.096)	-	0.030 (0.063)	**0.029 (0.063)**	**0.029 (0.063)**
	50	10	**0.000 (0.001)**	**0.000 (0.001)**	**0.000 (0.001)**	**0.000 (0.001)**	**0.000 (0.001)**
		20	**0.001 (0.002)**	**0.001 (0.002)**	**0.001 (0.002)**	**0.001 (0.002)**	**0.001 (0.002)**
		50	0.003 (0.007)	**0.002 (0.005)**	**0.002 (0.005)**	**0.002 (0.005)**	**0.002 (0.005)**
		100	0.007 (0.016)	-	**0.004 (0.009)**	**0.004 (0.009)**	**0.004 (0.009)**
		200	0.016 (0.037)	-	**0.007 (0.017)**	**0.007 (0.018)**	**0.007 (0.018)**
		500	0.040 (0.090)	-	**0.015 (0.035)**	**0.015 (0.038)**	**0.015 (0.038)**
	100	10	**0.000 (0.001)**	**0.000 (0.001)**	**0.000 (0.001)**	**0.000 (0.001)**	**0.000 (0.001)**
		20	**0.001 (0.001)**	**0.001 (0.001)**	**0.001 (0.001)**	**0.001 (0.001)**	**0.001 (0.001)**
		50	0.002 (0.004)	**0.001 (0.003)**	**0.001 (0.003)**	**0.001 (0.003)**	**0.001 (0.003)**
		100	0.005 (0.011)	0.002 (0.006)	**0.003 (0.006)**	**0.003 (0.006)**	**0.003 (0.006)**
		200	0.012 (0.027)	-	**0.005 (0.012)**	**0.005 (0.012)**	**0.005 (0.012)**
		500	0.035 (0.078)	-	**0.011 (0.027)**	0.012 (0.029)	0.012 (0.029)
	200	10	**0.000 (0.000)**	**0.000 (0.000)**	**0.000 (0.000)**	**0.000 (0.000)**	**0.000 (0.000)**
		20	**0.000 (0.001)**	**0.000 (0.001)**	**0.000 (0.001)**	**0.000 (0.001)**	**0.000 (0.001)**
		50	**0.001 (0.002)**	**0.001 (0.002)**	**0.001 (0.002)**	**0.001 (0.002)**	**0.001 (0.002)**
		100	0.003 (0.006)	**0.002 (0.004)**	**0.002 (0.004)**	**0.002 (0.004)**	**0.002 (0.004)**
		200	0.007 (0.017)	**0.003 (0.007)**	**0.003 (0.007)**	**0.003 (0.007)**	**0.003 (0.007)**
		500	0.025 (0.057)	-	**0.006 (0.015)**	0.007 (0.016)	0.007 (0.016)
	500	10	**0.000 (0.000)**	**0.000 (0.000)**	**0.000 (0.000)**	**0.000 (0.000)**	**0.000 (0.000)**
		20	**0.000 (0.001)**	**0.000 (0.001)**	**0.000 (0.001)**	**0.000 (0.001)**	**0.000 (0.001)**
		50	**0.001 (0.002)**	**0.001 (0.001)**	**0.001 (0.001)**	**0.001 (0.001)**	**0.001 (0.001)**
		100	**0.001 (0.003)**	**0.001 (0.003)**	**0.001 (0.003)**	**0.001 (0.003)**	**0.001 (0.003)**
		200	0.004 (0.009)	**0.002 (0.005)**	**0.002 (0.005)**	**0.002 (0.005)**	**0.002 (0.005)**
		500	0.014 (0.032)	**0.004 (0.009)**	**0.004 (0.010)**	**0.004 (0.009)**	**0.004 (0.009)**

Pairwise win rates. The next table summarizes how often each estimator achieves lower absolute error than another estimator in the low-m, high-k regime.

Table 7. Proportion of times method A (row) achieves lower absolute error than method B (column) across all datasets, models, temperatures, and low m (1, 2, 5, 10), high k (50, 100, 200, 500) regimes. Best results are in **bold**.

A \ B	Naive	BB	ZOIBB	LinMix
Naive	–	18.1	11.5	15.4
BB	81.9	–	55.6	32.5
ZOIBB	**88.5**	44.4	–	40.6
LinMix	84.6	**67.5**	**59.4**	–

Table 8. Pass@K improvement on CodeContests evaluated over all models and temperatures. The markers indicate the symbols used in Figure 6.

Model	T	Pass@1	Pass@200	Δ Pass	Marker
llama3.1-8b-chat	0.1	0.061	0.156	0.294	○
llama3.1-8b-chat	1.0	0.027	0.194	1.752	○
llama3.1-8b-chat	1.2	0.012	0.137	2.801	○
qwen2.5-7b-chat	0.1	0.031	0.119	0.401	▽
qwen2.5-7b-chat	1.0	0.022	0.220	2.522	▽
qwen2.5-7b-chat	1.2	0.019	0.208	2.796	▽
qwen2.5-3b-chat	0.1	0.024	0.159	0.943	▷
qwen2.5-3b-chat	1.0	0.009	0.155	3.967	▷
qwen2.5-3b-chat	1.2	0.005	0.110	6.251	▷
ministral-8b-chat	0.1	0.029	0.159	1.127	◇
ministral-8b-chat	1.0	0.012	0.178	3.880	◇
ministral-8b-chat	1.2	0.003	0.111	11.570	◇
qwen3-30b-A3b-chat	0.1	0.075	0.189	0.242	△
qwen3-30b-A3b-chat	1.0	0.059	0.337	1.418	△
qwen3-30b-A3b-chat	1.2	0.051	0.341	1.791	△
llama3.1-70b-chat	0.1	0.183	0.372	0.401	□
llama3.1-70b-chat	1.0	0.123	0.494	0.893	□
llama3.1-70b-chat	1.2	0.086	0.431	1.132	□

Sampling gains by model on CodeContests. Table 8 summarizes sampling gains for each (model, temperature) configuration on CodeContests. For each configuration we report pass@1 and higher-k reference values, and we also report the fitted improvability statistic $\Delta\mathrm{Pass}(P)$ computed under the BB approximation. These quantities support the main-text analysis: models with similar pass@1 can exhibit markedly different gains as k increases.

Table 9. Pass@K improvement on MBPP evaluated over all models and temperatures. The markers indicate the symbols used in Figure 6.

Model	T	Pass@1	Pass@200	Δ Pass	Marker
llama3.1-8b-chat	0.1	0.660	0.834	0.150	○
llama3.1-8b-chat	1.0	0.611	0.964	1.052	○
llama3.1-8b-chat	1.2	0.530	0.969	1.363	○
qwen2.5-7b-chat	0.1	0.713	0.846	0.198	▽
qwen2.5-7b-chat	1.0	0.692	0.979	1.629	▽
qwen2.5-7b-chat	1.2	0.661	0.986	1.955	▽
qwen2.5-3b-chat	0.1	0.705	0.887	0.278	▷
qwen2.5-3b-chat	1.0	0.620	0.972	1.261	▷
qwen2.5-3b-chat	1.2	0.558	0.979	1.500	▷
ministral-8b-chat	0.1	0.671	0.892	0.289	◇
ministral-8b-chat	1.0	0.581	0.964	1.153	◇
ministral-8b-chat	1.2	0.466	0.966	1.424	◇
qwen3-30b-A3b-chat	0.1	0.703	0.838	0.167	△
qwen3-30b-A3b-chat	1.0	0.603	0.949	1.626	△
qwen3-30b-A3b-chat	1.2	0.596	0.958	1.922	△
llama3.1-70b-chat	0.1	0.829	0.929	0.186	□
llama3.1-70b-chat	1.0	0.808	0.977	0.999	□
llama3.1-70b-chat	1.2	0.766	0.977	1.362	□

Table 10. Pass@K improvement on HumanEval evaluated over all models and temperatures. Most configurations have zero pass@1 and pass@200; the few high-temperature configurations with nonzero pass@200 also show very large ΔPass values. The markers indicate the symbols used in Figure 6.

Model	T	Pass@1	Pass@200	Δ Pass	Marker
llama3.1-8b-chat	0.1	0.000	0.000	2.024	○
llama3.1-8b-chat	1.0	0.000	0.000	2.024	○
llama3.1-8b-chat	1.2	0.000	0.000	2.024	○
qwen2.5-7b-chat	0.1	0.000	0.000	2.024	▽
qwen2.5-7b-chat	1.0	0.000	0.015	200.244	▽
qwen2.5-7b-chat	1.2	0.000	0.025	435.781	▽
qwen2.5-3b-chat	0.1	0.000	0.000	2.024	▷
qwen2.5-3b-chat	1.0	0.000	0.000	2.024	▷
qwen2.5-3b-chat	1.2	0.000	0.000	2.024	▷
ministral-8b-chat	0.1	0.000	0.000	2.024	◇
ministral-8b-chat	1.0	0.000	0.000	2.024	◇
ministral-8b-chat	1.2	0.001	0.152	3402.521	◇
qwen3-30b-A3b-chat	0.1	0.000	0.000	2.024	△
qwen3-30b-A3b-chat	1.0	0.000	0.046	1090.838	△
qwen3-30b-A3b-chat	1.2	0.001	0.149	1686.639	△
llama3.1-70b-chat	0.1	0.000	0.000	2.024	□
llama3.1-70b-chat	1.0	0.000	0.000	2.024	□
llama3.1-70b-chat	1.2	0.000	0.000	2.024	□

Sampling gains by model on MBPP and HumanEval. Table 9 provides the same summary on MBPP, and Table 10 provides the boundary-heavy HumanEval case. On MBPP and CodeContests, $\Delta\text{Pass}(P)$ is useful for comparing configurations with nontrivial pass@1 values. On HumanEval, absolute gains are often exactly zero and relative gains are unstable or undefined when pass@1 is zero, so we interpret the table mainly as evidence that only a few configurations escape the near-zero regime.

Structure in model space. Figure 6 visualizes all (model, temperature) configurations in the $(\text{Pass@1}, \Delta\text{Pass}(P))$ plane. Each dataset has its own scale and interpretation: CodeContests shows a tradeoff between pass@1 and improvability, MBPP shows a Pareto frontier, and HumanEval shows a near-degenerate regime where almost all configurations are close to zero on both axes.

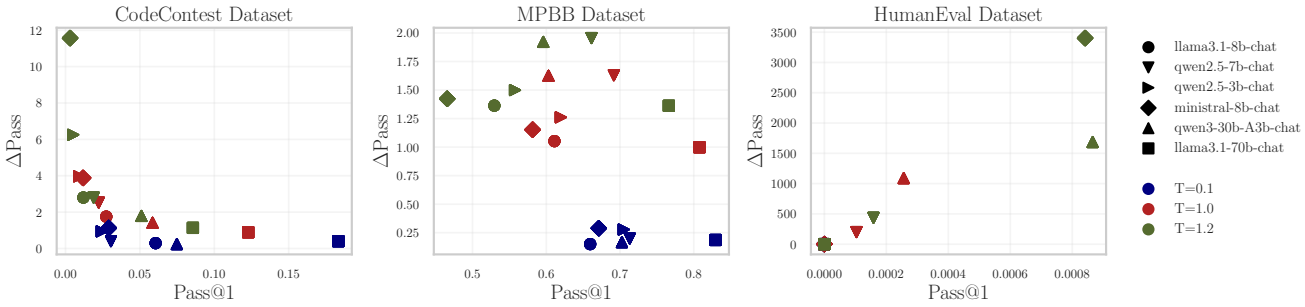


Figure 6. Pass@1 vs $\Delta\text{Pass}(P)$ on CodeContests, MBPP, and HumanEval for various models and temperatures. Each dot corresponds to a model and temperature combination; markers indicate model type and colors indicate temperature.

Diversity analysis. The diversity analysis excludes HumanEval, which has too few successful samples for stable JPlag similarities, and relative-improvement summaries are undefined when Pass@1 is zero.