NeuroSym-Cal: Bridging the Reasoning-Execution Gap in Code Generation via Hierarchical Calibration

Peiyang Liu[1,2], **Yining Wang**[3], **Youru Li**[4], **Long Li**[5], **Zhi Cai**[4] **and Wei Ye**[1,*]
[1] National Engineering Research Center for Software Engineering, Peking University, Beijing, China,
[2] School of Software and Microelectronics, Peking University, Beijing, China,
[3] Electrical and Computer Engineering, University of Toronto, Canada,
[4] College of Computer Science, Beijing University of Technology, Beijing, China,
[5] Infly Tech, Shanghai, China.
liupeiyang@pku.edu.cn

Abstract

While Chain-of-Thought (CoT) reasoning enhances code generation in Large Language Models (LLMs), it introduces a critical challenge in uncertainty estimation: *Confidence Saturation*. Existing calibration methods, such as Self-Consistency, rely on the assumption that consensus implies correctness. This assumption fails under systematic errors, where models confidently repeat flawed logic, leading to miscalibrated high-confidence predictions. To address this, we introduce NEUROSYM-CAL, a hierarchical calibration framework. We posit that reliable confidence requires interrogating the model at two complementary levels: the *extrinsic consensus* of its symbolic outputs and the *intrinsic self-assessment* of its generated logic. Specifically, we propose *Self-Verification Analysis*, which prompts the model to holistically re-evaluate its completed candidate, exploiting the cognitive asymmetry between autoregressive generation and post-hoc reviewing. This provides a fine-grained continuous signal that persists even when output consensus saturates. These orthogonal features, augmented by code-level descriptors, are fused by a Contextual Calibration Network to predict correctness. Experiments across state-of-the-art reasoning models (e.g., DeepSeek-R1) demonstrate that NEUROSYM-CAL effectively **desaturates** overconfident errors, achieving state-of-the-art Expected Calibration Error (ECE) and superior selective generation performance on Out-Of-Domain (OOD) benchmarks.

1 Introduction

The advent of Large Language Models (LLMs) with "System 2" reasoning capabilities has fundamentally transformed automated software engineering and broad AI applications (Li et al., 2025b; Dong et al., 2026; Chen et al., 2026; Hu et al., 2026; Zhang et al., 2026a; Li et al., 2024, 2026; Liu et al., 2025a, 2026). Models such as DeepSeek-R1 and OpenAI's o-series (Guo et al., 2025; Yan et al., 2025) engage in extended CoT (Zhang et al., 2025b) processes to decompose complex algorithmic problems. While these models achieve state-of-the-art performance on benchmarks like LiveCodeBench (Jain et al., 2024), their deployment in critical workflows is hindered by a persistent challenge: the lack of reliable *confidence calibration* (Liu et al., 2025c).

Ideally, a code generation model should be well-calibrated: a solution predicted with 90% confidence should be correct 90% of the time. However, current reasoning models are prone to *overconfidence*, particularly when they exhibit systematic errors. We argue that prevailing calibration paradigms are ill-suited for this new generation of models due to the problem of **Confidence Saturation**.

Standard calibration methods, such as Self-Consistency (SC) (Wang et al., 2022), operate on the "wisdom of the crowd" assumption: if the majority of sampled outputs are identical, the solution is likely correct. While effective for stochastic errors, this assumption collapses under *Entrenched Hallucinations* (Jiang et al., 2026). When a model falls into a reasoning trap, it may confidently reproduce the same flawed logic across all samples. In such cases, SC assigns a confidence score of 1.0, rendering the metric indistinguishable from a correct solution. This saturation destroys the granularity required for effective risk management.

To address this, we posit that reliable calibration requires interrogating the model at two orthogonal levels, **corresponding to the horizontal and vertical axes in Figure 1**: the *extrinsic* consensus of its symbolic outputs (what it says) and the *intrinsic* self-assessment of its own code (how reliable it judges itself to be). We introduce NEUROSYM-CAL, a hierarchical framework that fuses these complementary signals.

* Corresponding author

Findings of the Association for Computational Linguistics: ACL 2026, pages 6132–6143
July 2-7, 2026

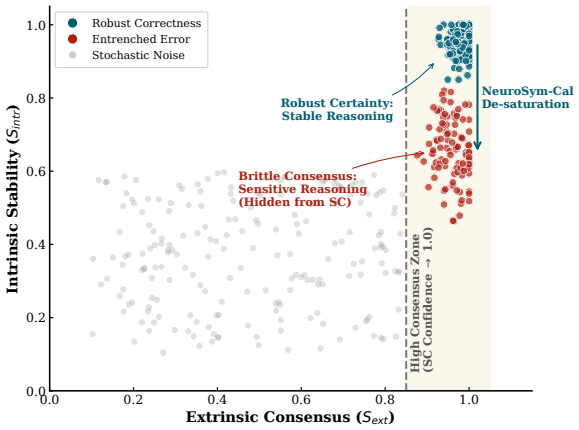

Figure 1: **The Intuition behind NEUROSYM-CAL.** Traditional methods rely solely on Extrinsic Semantic Consensus (x-axis), suffering from *Confidence Saturation*: both robust correctness (Blue) and entrenched errors (Red) cluster at 1.0. By introducing a second orthogonal dimension, Intrinsic Self-Verification (y-axis), we recover a critical signal for calibration. While not perfectly separable, errors often exhibit lower self-assessment scores than ground truths. Our framework exploits this statistical difference to "de-saturate" the confidence of high-consensus but unreliable solutions.

NEUROSYM-CAL operates on a neuro-symbolic principle. At the symbolic level, we employ *Semantic Equivalence Clustering* based on Abstract Syntax Trees (ASTs) (Zhang et al., 2019; Liu et al., 2025b) to measure consensus, filtering out syntactic noise. At the intrinsic level, we introduce *Self-Verification Analysis*. We prompt the LLM to re-evaluate its own generated code, exploiting the cognitive asymmetry between myopic autoregressive generation and holistic post-hoc reviewing. Our key observation is that while entrenched errors may be behaviorally consistent across samples, the model's intrinsic self-assessment of such errors is statistically lower than that of correct solutions, providing a continuous calibration signal that generation-time token probabilities fail to capture.

These two signals are synthesized by a *Contextual Calibration Network (CCN)*. Rather than treating calibration as a simple voting mechanism, the CCN learns a non-linear mapping that softens predicted probabilities in high-consensus regimes when self-verification disagrees.

We evaluate NEUROSYM-CAL across three state-of-the-art reasoning models and three diverse benchmarks. Our experiments demonstrate that by leveraging the "vertical axis" of self-verification, our method significantly outperforms both probability-based and consistency-based baselines. In summary, our contributions are as follows:

- We identify *Confidence Saturation* as a primary failure mode of Self-Consistency in reasoning models, where systematic errors mimic the statistical signature of correct solutions.
- We propose NEUROSYM-CAL, a framework that combines AST-based consensus (Extrinsic) with self-verification analysis (Intrinsic), augmented by lightweight code-level descriptors (length, cyclomatic complexity), to capture a holistic view of epistemic uncertainty.
- We demonstrate that fusing these orthogonal signals allows for effective *confidence de-saturation*, achieving state-of-the-art Expected Calibration Error (ECE) (Posocco and Bonnefoy, 2021) and superior selective generation performance on OOD tasks.

2 Related Work

2.1 Reasoning and Alignment in Code Generation

The paradigm of code generation has shifted from statistical sequence completion (Chen, 2021; Roziere et al., 2023) to "System 2" reasoning (Zhang et al., 2025a). Models like DeepSeek-R1 and OpenAI's o-series (Guo et al., 2025) employ CoT (Wei et al., 2022; Li and Ma, 2025; Zhang et al., 2026b) to decompose complex specifications before implementation. While CoT significantly enhances performance on algorithmic benchmarks (Jain et al., 2024), it introduces the *alignment* challenge known as the "Reasoning-Execution Gap" (Turpin et al., 2023; Lanham et al., 2023; Li et al., 2025a; Fang et al., 2026b,a; Fu et al., 2026). Prior works have largely treated reasoning traces as static context. In contrast, we treat the model's own assessment of its generated code as a dynamic signal of reliability.

2.2 Uncertainty Estimation and Calibration

Reliable confidence estimation is critical for deploying LLMs. Standard methods fall into two categories: logit-based and consistency-based. Logit-based methods (e.g., Platt Scaling) (Guo et al., 2017; Platt et al., 1999) rely on the model's raw token probabilities. However, LLMs are notoriously overconfident (Tian et al., 2025), and raw probabilities often fail to reflect functional correctness. Consistency-based approaches, such as SC (Wang

et al., 2022) and Semantic Entropy (Kuhn et al., 2023), aggregate multiple sampled outputs. While SC generally outperforms logit methods, it relies on the assumption that the majority vote converges to the truth. This assumption fails under *systematic errors* or "entrenched hallucinations" (Zhang et al., 2025b), where a model consistently generates the same flawed solution. In these high-consensus regimes, SC suffers from *Confidence Saturation*, assigning near-perfect scores to incorrect code. Our work addresses this by introducing an orthogonal signal, intrinsic self-verification, to regularize these saturated estimates.

2.3 Neuro-Symbolic Analysis

Our framework bridges two distinct analytical traditions: symbolic structure and neural self-assessment. On the *symbolic* side, metrics like CodeBERTScore (Zhou et al., 2023) utilize ASTs primarily for offline evaluation. We repurpose AST analysis for online calibration, using canonicalization to de-noise the variance in generated code and obtain a cleaner consensus signal. On the *neural* side, recent works have explored using LLMs as evaluators for continuous $P(\text{True})$ estimates (Kadavath et al., 2022). However, existing methods typically deploy self-verification as a standalone, scalar confidence estimator. NEUROSYM-CAL unifies these directions: rather than treating self-evaluation as an isolated metric, we treat it as an intrinsic feature to *de-saturate* overconfident symbolic consensus. This effectively fuses the strict logical grouping of AST with the semantic intuition of neural self-assessment.

3 Methodology

We propose NEUROSYM-CAL, a hierarchical calibration framework illustrated in Figure 2, designed to address the limitations of single-modal uncertainty estimation in CoT code generation.

Current state-of-the-art methods, particularly SC, suffer from *Confidence Saturation*. When a model generates the same solution repeatedly (whether correct or consistently wrong), SC assigns a confidence score approaching 1.0. This lack of granularity leads to severe miscalibration in high-confidence regimes. To resolve this, we posit that uncertainty must be interrogated from two complementary perspectives:

- **Extrinsic Consensus (S_{ext}):** A coarse-grained measure that filters out stochastic noise (Aleatoric Uncertainty) by observing output agreement.
- **Intrinsic Self-Verification (S_{intr}):** A fine-grained measure that probes the model's post-hoc assessment of a completed candidate, measuring how reliably the model judges its own code when explicitly prompted to re-evaluate.

Our framework fuses these signals to learn a calibrated probability function, effectively using intrinsic self-verification to regularize overconfident consensus.

3.1 Problem Formulation

Let $\mathcal{M}$ be a generative code model taking a prompt x. The model generates a reasoning chain r and code y, denoted as $(r, y) \sim \mathcal{M}(x)$. The functional correctness is given by an oracle $\mathcal{O}(y, x) \in \{0, 1\}$. Our goal is to learn a calibration function $f_\theta(x, r, y) \to \hat{p} \in [0, 1]$ such that $\hat{p}$ accurately reflects the true probability $P(\mathcal{O}(y, x) = 1)$. Specifically, we aim to minimize the ECE by distinguishing between *robust certainty* (high self-verification, high consensus) and *brittle consensus* (low self-verification, high consensus).

3.2 Intrinsic Uncertainty: Self-Verification Analysis

While sample-based consistency captures output variance, it treats the generation process as a black box and suffers from saturation when errors are entrenched. To capture fine-grained epistemic uncertainty without requiring white-box access to latent states, we implement a **Post-hoc Self-Verification Analysis** that leverages the neural model's intrinsic capacity to critically evaluate its own outputs.

Holistic Re-evaluation. During autoregressive generation, the model commits to each token sequentially, often falling into localized reasoning traps. However, when presented with a complete solution in evaluation mode, the model can assess the code holistically. We exploit this cognitive asymmetry by formulating a verification prompt $p_{ver}(x, y)$, which instructs the model $\mathcal{M}$ to act as an impartial judge and rate the functional correctness of its own candidate y for problem x on a continuous scale of 0 to 100.

Let $v \in [0, 100]$ be the scalar score extracted from the model's textual response. To account

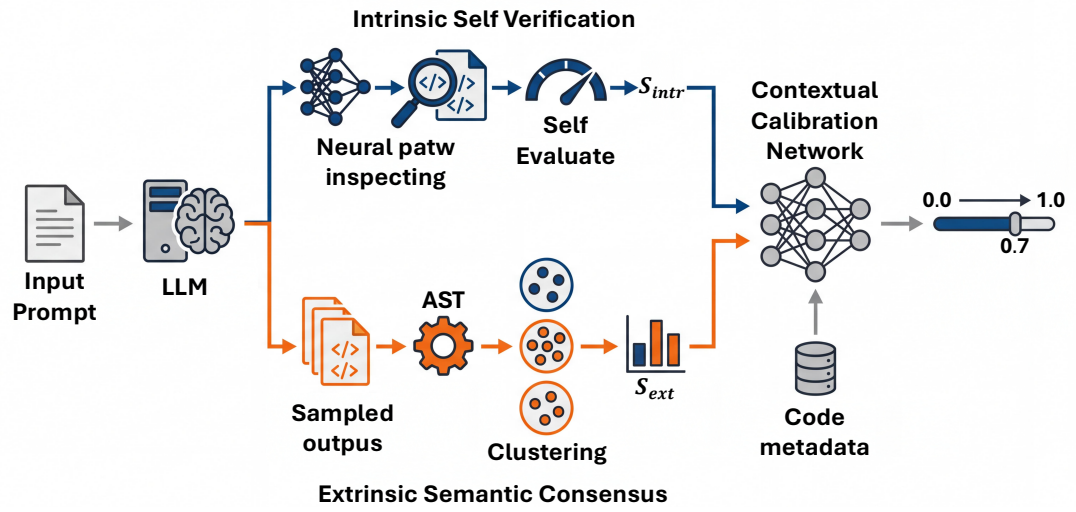

Figure 2: **The NEUROSYM-CAL Architecture.** The framework addresses the "Confidence Saturation" problem in reasoning models. **Top (Intrinsic):** We perform *Self-Verification* by prompting the model to re-evaluate its own generated code, producing a confidence score that captures the model's post-hoc assessment. **Bottom (Extrinsic):** We compute *Semantic Consensus* based on canonicalized ASTs. **Fusion:** The *CCN* synthesizes these orthogonal signals. Crucially, when Extrinsic Consensus saturates (approaches 1.0), the network uses Intrinsic self-verification to "de-saturate" the confidence, providing a more granular and honest probability estimate.

for parsing anomalies, we define the *Intrinsic Self-Verification* score $S_{intr}(y)$ as a normalized expectation:

$$S_{intr}(y) = \begin{cases} \frac{v}{100} & \text{if } v \text{ is a valid parsed integer,} \\ \gamma & \text{otherwise,} \end{cases} \tag{1}$$

where $\gamma = 0.5$ serves as a conservative fallback penalty for unparsable or evasive responses, reflecting high uncertainty.

Rationale. Generation confidence (derived from token-level log-probabilities) often misaligns with correctness, as models can confidently hallucinate flawed logic. In contrast, S_{intr} conditions on the *completed* logical structure. A lower S_{intr} score on a high-consensus output indicates a discrepancy between generation-time determinism and post-hoc structural validity, serving as a powerful proxy for *implicit self-doubt*.

3.3 Extrinsic Uncertainty: AST-based Semantic Consensus

To quantify the extrinsic uncertainty, we estimate the probability mass associated with the semantic logic of a generated solution. Given a set of K sampled outputs $\mathcal{Y} = \{y_k\}_{k=1}^{K}$ from the model $\mathcal{M}(x)$, relying on exact string matching yields a fragmented distribution due to syntactic variance (e.g., variable naming, formatting). We mitigate this by projecting solutions into a canonical semantic space (Liu et al., 2020, 2021c,a).

Equivalence Partitioning. We define a canonicalization mapping $\mathcal{T} : \mathcal{Y} \to \mathcal{Z}$, which parses code into an Abstract Syntax Tree (AST) and applies structural normalization rules. This induces an equivalence relation $\sim$ over the sampled set, where two solutions are deemed semantically equivalent if their canonical forms are identical:

$$y_i \sim y_j \iff \mathcal{T}(y_i) = \mathcal{T}(y_j). \tag{2}$$

Based on this relation, we partition the sample space $\mathcal{Y}$ into M disjoint equivalence classes (clusters) $\mathbf{C} = \{C_1, C_2, \ldots, C_M\}$, such that $\bigcup_{m=1}^{M} C_m = \mathcal{Y}$ and $\forall y_a, y_b \in C_m, y_a \sim y_b$.

Consensus Estimation. For any candidate solution $y \in \mathcal{Y}$, let $C(y) \in \mathbf{C}$ denote the unique equivalence class containing y. We define the *Extrinsic Consensus* score $S_{ext}(y)$ as the empirical probability of the semantic cluster to which y belongs:

$$S_{ext}(y) = \hat{P}(y \mid x) = \frac{|C(y)|}{K}. \tag{3}$$

Unlike standard Self-Consistency which selects the mode of the distribution ($\arg\max |C_m|$), we assign this density score to *every* sample in the cluster. This provides the subsequent calibration network with a continuous measure of the model's surface-level confidence in that specific logical path.

3.4 Contextual Calibration Network (CCN)

The core innovation of NEUROSYM-CAL lies in the fusion of these signals. We employ a lightweight MLP (Popescu et al., 2009; Dong et al., 2025; Liu et al., 2021b), the Contextual Calibration Network (CCN), to map the feature vector $\mathbf{v} = [S_{intr}, S_{ext}, L_{code}, L_{reason}, C_{cyc}]$ to a calibrated probability $\hat{p}$, where L_{code} and L_{reason} are lengths of code and reasoning chains, and C_{cyc} is the cyclomatic complexity (Ebert et al., 2016) of the generated code.

De-saturation Mechanism. The CCN learns a non-linear interaction between the features. We hypothesize that the network acts as a *conditional gate*:

- When S_{ext} is low (high disagreement), the model is clearly uncertain; S_{ext} dominates the prediction.
- When S_{ext} is high (saturation), the network relies on S_{intr} to determine the *quality* of that consensus.

If a solution has High Consensus ($S_{ext} \approx 1.0$) but Low Self-Verification ($S_{intr} \ll 1.0$), the CCN learns to penalize the confidence score, effectively "de-saturating" it to reflect the true empirical risk.

Objective Function. We train the CCN using a hybrid objective to handle class imbalance and directly optimize calibration:

$$\mathcal{L} = \mathcal{L}_{FL}(\hat{p}, y) + \beta \mathcal{L}_{Soft-ECE}(\hat{p}, y). \quad (4)$$

where $\mathcal{L}_{FL}$ is Focal Loss (Lin et al., 2017). To ensure differentiability for $\mathcal{L}_{Soft-ECE}$, we employ a kernel-based soft binning approach (Karandikar et al., 2021; Liu et al., 2022). Let $\{c_m\}_{m=1}^{M}$ be the fixed centers of M bins. For a sample i with prediction $\hat{p}_i$, we compute a soft assignment weight $w_{i,m}$ using a softmax over the Euclidean distance to bin centers:

$$w_{i,m} = \frac{\exp\left(-(\hat{p}_i - c_m)^2/\sigma^2\right)}{\sum_{k=1}^{M} \exp\left(-(\hat{p}_i - c_k)^2/\sigma^2\right)}. \quad (5)$$

The Soft-ECE is then calculated as the weighted absolute difference between average confidence and accuracy per bin:

$$\mathcal{L}_{Soft-ECE} = \sum_{m=1}^{M} \frac{S_m}{N} \Bigg| \underbrace{\frac{\sum_{i=1}^{N} w_{i,m}\hat{p}_i}{S_m}}_{\text{Soft-Conf}_m} - \underbrace{\frac{\sum_{i=1}^{N} w_{i,m} y_i}{S_m}}_{\text{Soft-Acc}_m} \Bigg|, \quad (6)$$

where $S_m = \sum_{i=1}^{N} w_{i,m}$ is the effective size of bin m, and σ controls the smoothness of the assignment.

4 Experimental Setup

We design our experiments to rigorously evaluate the NEUROSYM-CAL framework against diverse baselines on reasoning-intensive code generation tasks. Our goal is to validate whether fusing intrinsic and extrinsic signals effectively mitigates confidence saturation. We frame our analysis around three research questions:

RQ1 (Calibration Efficacy): Does incorporating intrinsic self-verification yield better calibrated confidence scores than methods relying solely on output consistency, particularly in de-saturating overconfident errors?

RQ2 (Selective Generation): Can NEUROSYM-CAL effectively filter out incorrect code to improve the reliability of the system in deployment scenarios (Risk-Coverage analysis)?

RQ3 (Signal Complementarity): Are both the symbolic consensus and neural self-verification necessary for optimal calibration, and what are their respective roles?

4.1 Benchmarks and Models

To ensure our evaluation reflects the capabilities of modern reasoning models and avoids training data contamination, we employ three challenging, execution-based benchmarks:

LiveCodeBench (LCB) (Jain et al., 2024). We utilize the *Test* split of LCB (v2406–v2506), consisting of 400 LeetCode-style contest problems. Because the three evaluated models have different knowledge cutoffs (Qwen2.5-Coder-32B $\approx$ 2024-07, DeepSeek-R1-Distill-Llama-70B $\approx$ 2024-07, Llama-3.3-70B $\approx$ 2024-12), only problems released strictly after a given model's cutoff are treated as out-of-distribution for that model. We report the held-out subset per model and follow

the LCB convention of treating the full v2406–v2506 window as a harder-than-training testbed rather than a universally OOD set.

HumanEval-Pro (Yu et al., 2025). An extension of the classic HumanEval dataset that includes additional, more rigorous hidden test cases. This mitigates the issue of "false positives" where models overfit to simple public tests, allowing for a more accurate ground-truth correctness label.

DS-1000 (Lai et al., 2023). A data science coding benchmark comprising 1,000 real-world Python problems requiring the use of libraries like Pandas and NumPy. This dataset tests the model's ability to handle complex API logic and multi-step data manipulation, providing a different reasoning flavor than algorithmic puzzles.

For all datasets, we use the provided canonical solutions and test suites to determine the binary correctness label $y \in \{0, 1\}$.

Training Data and CCN Implementation. To strictly prevent data leakage and rigorously evaluate Out-Of-Domain (OOD) generalization, the Contextual Calibration Network (CCN) is trained on a completely disjoint corpus. Specifically, we construct our training set using the *APPS* (Introductory split) (Hendrycks et al., 2021) combined with an older, historical split of *LiveCodeBench*. To capture a realistic distribution of "high-consensus but incorrect" negative samples (i.e., entrenched hallucinations) (Liu et al., 2023; Liu, 2024) for the CCN to effectively learn the de-saturation mechanism, we sample the training trajectories at the same temperature used for inference ($T = 0.6$). This ensures strict consistency in the consensus distribution between training and evaluation. **None of the evaluation benchmarks** overlap with this training set. For reproducibility, the CCN is implemented as a lightweight 3-layer MLP with hidden dimensions $[64, 32]$. We train the network for 50 epochs using the AdamW optimizer with a learning rate of 1e-3 and a batch size of 128. The soft-binning parameter is set to $\sigma = 0.05$ and the loss weighting factor to $\beta = 0.5$.

We evaluate our method on three state-of-the-art open-weight models that exhibit strong reasoning capabilities:

- **DeepSeek-R1-Distill-Llama-70B (Guo et al., 2025):** A distilled reasoning model that explicitly generates CoT traces. This is our primary subject for analyzing the *Reasoning-Execution Gap*.
- **Qwen2.5-Coder-32B-Instruct (Hui et al., 2024):** Currently one of the strongest open-source code-specific models.
- **Llama-3.3-70B-Instruct (Grattafiori et al., 2024):** A general-purpose strong baseline to test the universality of our method.

All models are run in half-precision (BF16) using the vLLM inference engine. We set the sampling temperature to $T = 0.6$ and top-$p = 0.95$ to encourage diversity for our extrinsic uncertainty estimation, generating $K = 10$ samples per problem. For intrinsic self-verification, we prompt each model to rate its own solutions with temperature $T = 0$.

4.2 Baselines

We compare NEUROSYM-CAL against a comprehensive set of calibration strategies, categorized into three groups to highlight different levels of sophistication:

I. Standard Baselines (Logit-based). These methods rely solely on the output token probabilities, representing the default uncertainty estimation capability of the LLM.

- **AvgLogit:** The exponential of the average log-probability. We report *AvgLogit-Code* (probabilities averaged over code tokens only) as it typically outperforms full-sequence averaging for correctness prediction.
- **Platt Scaling (Platt et al., 1999):** A parametric approach that fits a logistic regression model on the *AvgLogit-Code* scores to correct global miscalibration.
- **P-True (Kadavath et al., 2022):** We prompt the model to verbally estimate the probability that its generated solution is correct.

II. Strong Baselines (Consistency-based). These methods utilize sampling diversity to estimate uncertainty, currently considered the gold standard for reasoning tasks. **SC (Wang et al., 2022):** We sample $K = 10$ solutions and define confidence as the percentage of samples that are strictly identical (string match) to the majority vote.

III. State-of-the-Art Baseline (Multicalibration). We implement the most effective multicalibration approach identified in recent literature (Campos et al., 2025) to serve as our primary competitor. **Iterative Grouped Linear Binning (IGLB):** This method iteratively refines calibration errors across intersecting groups. Following prior work on code generation, we define groups based on *cyclomatic complexity* and *code length*. IGLB represents the best-performing metadata-driven calibration technique, allowing us to directly compare our neuro-symbolic features against explicit metadata grouping.

4.3 Metrics

We assess performance using three complementary metrics:

- **Expected Calibration Error (ECE):** We use the unbiased estimator with $M = 10$ bins to measure the average discrepancy between confidence and accuracy (Posocco and Bonnefoy, 2021).
- **Brier Score (BS):** The mean squared error between predicted probabilities and binary outcomes. Lower is better (Rufibach, 2010).
- **AUROC:** The Area Under the Receiver Operating Characteristic curve. This measures the *discriminative* power of the confidence score, i.e., how well it separates correct from incorrect code, independent of the calibration threshold (McDermott et al., 2024).

5 Experimental Results

5.1 Main Results: Calibration Performance (RQ1)

Table 1 presents the calibration performance across three benchmarks. NEUROSYM-CAL consistently achieves superior calibration compared to standard baselines, validating the efficacy of fusing extrinsic and intrinsic uncertainty signals.

Mitigation of Confidence Saturation. On the OOD **LiveCodeBench**, our method yields the most significant gains, reducing Expected Calibration Error (ECE) by **37.4%** compared to Self-Consistency (0.072 vs. 0.115) for DeepSeek-R1. The improvement in Brier Score (0.182 vs. 0.198) further confirms that our framework penalizes overconfident errors more effectively than consensus alone. This confirms that the CCN acts as a conditional gate, successfully de-saturating confidence when self-verification (S_{intr}) contradicts surface-level consensus (S_{ext}).

Generalization vs. Statistical Baselines. A nuanced trend emerges when comparing against the multicalibration baseline IGLB. On in-distribution tasks (**HumanEval-Pro**), IGLB remains highly competitive (ECE 0.038 vs. 0.035), suggesting that statistical binning based on code metadata (length, complexity) is sufficient for standard problems. However, on the complex OOD logic of LiveCodeBench, NEUROSYM-CAL regains the advantage (ECE 0.072 vs. 0.078). This indicates that while metadata captures aleatoric noise in simple distributions, *intrinsic self-verification* is essential for capturing epistemic uncertainty in harder, unseen reasoning trajectories.

Discriminative Power. In terms of ranking (AUROC), our method attains the best score on LCB (0.805), though the margin over IGLB (0.798) is modest. This indicates that while *Extrinsic Consensus* drives the primary ranking of solutions, the *Intrinsic* module is most useful for the probability scaling required to minimize ECE rather than for re-ordering candidates.

5.2 Selective Generation (RQ2)

To assess practical utility, we evaluate selective generation, where the system abstains from predictions below a confidence threshold. Figure 3 illustrates the Risk-Coverage (RC) curves on LiveCodeBench. NEUROSYM-CAL demonstrates a superior risk profile, minimizing the error rate at equivalent coverage levels. Quantitatively, our framework reduces the Area Under Risk-Coverage (AURC) by **6.7%** relative to the strongest baseline (IGLB) and **35.7%** relative to SC.

The performance advantage is most pronounced in the operational coverage interval (20%–80%). In this regime, standard SC suffers from saturation, assigning near-perfect confidence to "entrenched hallucinations" which prevents effective filtration. In contrast, NEUROSYM-CAL leverages self-verification (S_{intr}) to detect unreliable solutions within these high-consensus errors. By pulling these samples down from near-saturated confidence to intermediate values, our framework re-ranks the generation queue so that the system prioritizes genuinely robust solutions over confidently wrong ones, acting as a safety filter for deployment.

Model	Method	LiveCodeBench (OOD)			HumanEval-Pro			DS-1000 (Data Sci)		
		ECE ↓	BS ↓	AUC ↑	ECE ↓	BS ↓	AUC ↑	ECE ↓	BS ↓	AUC ↑
DeepSeek-R1 (Distill-70B)	AvgLogit	0.155	0.231	0.710	0.120	0.215	0.735	0.145	0.228	0.698
	Platt Scaling	0.085	0.218	0.710	0.075	0.185	0.735	0.091	0.208	0.698
	P-True	0.132	0.225	0.738	0.105	0.205	0.755	0.125	0.215	0.722
	Self-Consistency	0.115	0.198	0.792	0.095	0.168	0.810	0.105	0.188	0.765
	IGLB (SOTA)	0.078	0.195	0.798	0.038	0.150	0.825	0.062	0.180	0.780
	NS-CAL (Ours)	**0.072**	**0.182**	**0.805**	**0.035**	**0.148**	**0.828**	**0.055**	**0.172**	**0.792**
Qwen2.5 (Coder-32B)	AvgLogit	0.138	0.228	0.725	0.115	0.212	0.748	0.132	0.225	0.715
	Platt Scaling	0.092	0.215	0.725	0.068	0.182	0.748	0.088	0.205	0.715
	P-True	0.118	0.222	0.752	0.098	0.198	0.768	0.115	0.212	0.738
	Self-Consistency	0.108	0.195	0.801	0.088	0.162	0.822	0.098	0.185	0.778
	IGLB (SOTA)	0.075	0.192	0.805	0.040	0.145	0.838	0.065	0.175	0.795
	NS-CAL (Ours)	**0.051**	**0.178**	**0.812**	**0.038**	**0.140**	**0.840**	**0.058**	**0.168**	**0.805**
Llama-3.3 (70B-Instruct)	AvgLogit	0.142	0.230	0.705	0.125	0.218	0.728	0.135	0.226	0.692
	Platt Scaling	0.095	0.216	0.705	0.072	0.188	0.728	0.092	0.210	0.692
	P-True	0.128	0.224	0.730	0.108	0.208	0.745	0.122	0.218	0.715
	Self-Consistency	0.112	0.202	0.785	0.092	0.172	0.805	0.102	0.192	0.758
	IGLB (SOTA)	0.082	0.196	0.790	0.045	0.155	0.818	0.075	0.184	0.772
	NS-CAL (Ours)	**0.075**	**0.185**	**0.798**	**0.042**	**0.150**	**0.820**	**0.065**	**0.176**	**0.785**

Table 1: **Main Calibration Results.** Best results are **bolded**, second-best are underlined.

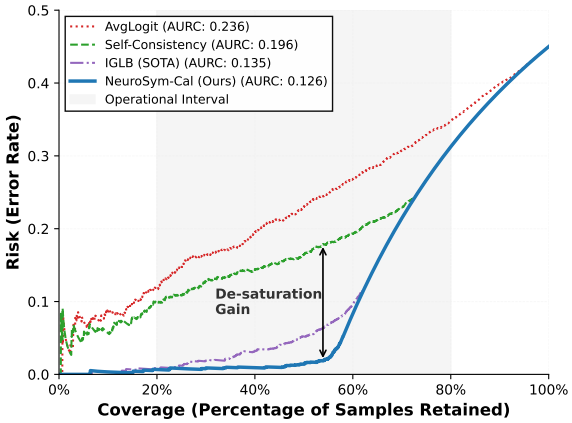

Figure 3: Risk-Coverage Curves on LiveCodeBench.

Configuration	ECE ↓	BS ↓	AUC ↑	Δ ECE
NEUROSYM-CAL (Full)	**0.072**	**0.182**	**0.805**	-
w/o Intrinsic (S_{intr})	0.082	0.192	0.799	+13.9%
w/o Extrinsic (S_{ext})	0.125	0.220	0.745	+73.6%
w/o AST (String Match)	0.089	0.188	0.798	+23.6%

Table 2: Ablation Results on LiveCodeBench.

5.3 Ablation Study (RQ3)

To validate the complementarity of our neuro-symbolic components, we conduct an ablation study on LCB using DeepSeek-R1. Table 2 summarizes the performance impact of removing or simplifying specific modules.

Orthogonality of Signals. The results highlight distinct roles for extrinsic and intrinsic features. Removing Extrinsic Consensus (S_{ext}) causes the sharpest decline in AUROC ($0.805 \rightarrow 0.745$), confirming that consensus remains the primary *discriminator* for ranking solutions. Conversely, removing Intrinsic Self-Verification (S_{intr}) degrades ECE (+13.9%) and Brier Score (+5.5%) while only mildly affecting AUROC. This is consistent with our hypothesis that self-verification functions mainly as a *calibration regularizer*: it re-ranks solutions only weakly, but scales probabilities by detecting unreliable code in high-consensus errors.

Necessity of Symbolic Abstraction. Replacing AST canonicalization with naive string matching (*w/o AST*) increases ECE ($0.072 \rightarrow 0.089$). This indicates that syntactic noise (e.g., formatting, variable naming) dilutes the consensus signal. Structural parsing is essential to filter aleatoric noise and recover the true epistemic distribution.

6 Conclusion

In this work, we address the challenge of *Confidence Saturation* in reasoning-enhanced code generation, where standard consistency methods fail to detect entrenched hallucinations. We introduce NEUROSYM-CAL, a hierarchical framework that fuses extrinsic AST-based symbolic consensus with

a novel intrinsic self-verification analysis. By leveraging the model's own post-hoc re-evaluation of its generated code, our method effectively identifies unreliable solutions even when output consensus saturates. Empirical results across multiple benchmarks demonstrate that NEUROSYM-CAL achieves state-of-the-art calibration error and superior selective generation performance, particularly in out-of-distribution scenarios. Our findings underscore that reliable uncertainty estimation in code generation requires moving beyond output agreement to probe the model's own assessment of its solutions. Future work will explore leveraging these calibrated signals to guide iterative self-correction in complex reasoning domains (Lin et al., 2025).

7 Limitations

While NEUROSYM-CAL demonstrates significant advancements in calibrating reasoning-oriented code generation models, we acknowledge several limitations that define the boundaries of our current work and point towards future research directions.

Dependency on Self-Evaluation Capability. Unlike our initial explorations into white-box latent probing, the finalized *Intrinsic* module relies on the model's ability to critically evaluate its own generated code via self-verification. While this elegantly removes the need for white-box access to internal states (gradients or hidden representations) and broadens applicability to closed-source APIs, it heavily assumes that the model possesses sufficient meta-cognitive ability to assess code correctness. For smaller or less capable models (e.g., $<$ 7B parameters), self-verification scores may collapse or become uninformative. Future work could investigate decoupling this by using a separate, specialized verifier model.

Inference Latency and Computational Overhead. The overall framework incurs higher computational costs compared to greedy decoding. The *Extrinsic* module requires sampling K solutions (where $K = 10$ in our experiments) and parsing them into ASTs. The *Intrinsic* module requires an additional inference pass for self-verification. While both can be parallelized, they increase the token consumption. This trade-off between calibration reliability and inference latency may restrict deployment in strictly real-time, low-latency coding assistants, though it remains highly viable for asynchronous tasks like automated code review or repository-level synthesis.

Language-Specific Symbolic Constraints. The *Semantic Equivalence Clustering* relies on robust AST parsers and manually defined canonicalization rules (e.g., normalizing variable names and loops). In this work, we implemented rules for high-resource languages (Python). Extending NEUROSYM-CAL to low-resource languages, domain-specific languages (DSLs), or mixed-language scenarios requires engineering effort to develop corresponding parsers and transformation rules. A purely neural or language-agnostic approach to semantic equivalence remains an open challenge.

References

Viola Campos, Robin Kuschnereit, and Adrian Ulges. 2025. Multicalibration for llm-based code generation. *arXiv preprint arXiv:2512.08810*.

Mark Chen. 2021. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*.

Zhiwei Chen, Yupeng Hu, Zhiheng Fu, Zixu Li, Jiale Huang, Qinlei Huang, and Yinwei Wei. 2026. Intent: Invariance and discrimination-aware noise mitigation for robust composed image retrieval. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 20463–20471.

Haonan Dong, Kehan Jiang, Haoran Ye, Wenhao Zhu, Zhaolu Kang, and Guojie Song. 2026. Neureasoner: Towards explainable, controllable, and unified reasoning via mixture-of-neurons. *arXiv preprint arXiv:2604.02972*.

Haonan Dong, Wenhao Zhu, Guojie Song, and Liang Wang. 2025. Aurora: Breaking low-rank bottleneck of lora with nonlinear mapping. *arXiv preprint arXiv:2505.18738*.

Christof Ebert, James Cain, Giuliano Antoniol, Steve Counsell, and Phillip Laplante. 2016. Cyclomatic complexity. *IEEE software*, 33(6):27–29.

Yangyi Fang, Jiaye Lin, Xiaoliang Fu, Cong Qin, Haolin Shi, Chaowen Hu, Lu Pan, Ke Zeng, and Xunliang Cai. 2026a. How to allocate, how to learn? dynamic rollout allocation and advantage modulation for policy optimization. *arXiv preprint arXiv:2602.19208*.

Yangyi Fang, Jiaye Lin, Xiaoliang Fu, Cong Qin, Haolin Shi, Chang Liu, and Peilin Zhao. 2026b. Proximity-based multi-turn optimization: Practical credit assignment for llm agent training. *arXiv preprint arXiv:2602.19225*.

Xiaoliang Fu, Jiaye Lin, Yangyi Fang, Chaowen Hu, Cong Qin, Zekai Shao, Binbin Zheng, Lu Pan, and

Ke Zeng. 2026. From $\log \pi$ to π: Taming divergence in soft clipping via bilateral decoupled decay of probability gradient weight. *arXiv preprint arXiv:2603.14389*.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. 2017. On calibration of modern neural networks. In *International conference on machine learning*, pages 1321–1330. PMLR.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, and 1 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Dan Hendrycks, Steven Basart, Saurav Kadavath, Mantas Mazeika, Akul Arora, Ethan Guo, Collin Burns, Samir Puranik, Horace He, Dawn Song, and Jacob Steinhardt. 2021. Measuring coding challenge competence with apps. *Preprint*, arXiv:2105.09938.

Yupeng Hu, Zixu Li, Zhiwei Chen, Qinlei Huang, Zhiheng Fu, Mingzhu Xu, and Liqiang Nie. 2026. Refine: Composed video retrieval via shared and differential semantics enhancement. *ACM Transactions on Multimedia Computing, Communications and Applications*.

Binyuan Hui, Jian Yang, Zeyu Cui, Jiaxi Yang, Dayiheng Liu, Lei Zhang, Tianyu Liu, Jiajun Zhang, Bowen Yu, Kai Dang, and 1 others. 2024. Qwen2. 5-coder technical report. *arXiv preprint arXiv:2409.12186*.

Naman Jain, King Han, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. 2024. Livecodebench: Holistic and contamination free evaluation of large language models for code. *arXiv preprint arXiv:2403.07974*.

Kehan Jiang, Haonan Dong, Zhaolu Kang, Zhengzhou Zhu, and Guojie Song. 2026. Foe: Forest of errors makes the first solution the best in large reasoning models. *arXiv preprint arXiv:2604.02967*.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, and 1 others. 2022. Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*.

Archit Karandikar, Nicholas Cain, Dustin Tran, Balaji Lakshminarayanan, Jonathon Shlens, Michael C Mozer, and Becca Roelofs. 2021. Soft calibration objectives for neural networks. *Advances in Neural Information Processing Systems*, 34:29768–29779.

Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. 2023. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. *arXiv preprint arXiv:2302.09664*.

Yuhang Lai, Chengxi Li, Yiming Wang, Tianyi Zhang, Ruiqi Zhong, Luke Zettlemoyer, Wen-tau Yih, Daniel Fried, Sida Wang, and Tao Yu. 2023. Ds-1000: A natural and reliable benchmark for data science code generation. In *International Conference on Machine Learning*, pages 18319–18345. PMLR.

Tamera Lanham, Anna Chen, Ansh Radhakrishnan, Benoit Steiner, Carson Denison, Danny Hernandez, Dustin Li, Esin Durmus, Evan Hubinger, Jackson Kernion, and 1 others. 2023. Measuring faithfulness in chain-of-thought reasoning. *arXiv preprint arXiv:2307.13702*.

Mengdi Li, Jiaye Lin, Xufeng Zhao, Wenhao Lu, Peilin Zhao, Stefan Wermter, and Di Wang. 2025a. Curriculum-rlaif: Curriculum alignment with reinforcement learning from ai feedback. *arXiv preprint arXiv:2505.20075*.

Xiping Li and Jianghong Ma. 2025. Aimcot: Active information-driven multimodal chain-of-thought for vision-language reasoning. *arXiv preprint arXiv:2509.25699*.

Xiping Li, Jianghong Ma, Kangzhe Liu, Shanshan Feng, Haijun Zhang, and Yutong Wang. 2024. Category-based and popularity-guided video game recommendation: a balance-oriented framework. In *Proceedings of the ACM Web Conference 2024*, pages 3734–3744.

Xiping Li, Aier Yang, Jianghong Ma, Kangzhe Liu, Shanshan Feng, Haijun Zhang, and Yi Zhao. 2026. Cpgrec+: A balance-oriented framework for personalized video game recommendations. *ACM Transactions on Information Systems*, 44(3):1–44.

Zhong-Zhi Li, Duzhen Zhang, Ming-Liang Zhang, Jiaxin Zhang, Zengyan Liu, Yuxuan Yao, Haotian Xu, Junhao Zheng, Pei-Jie Wang, Xiuyi Chen, and 1 others. 2025b. From system 1 to system 2: A survey of reasoning large language models. *arXiv preprint arXiv:2502.17419*.

Jiaye Lin, Yifu Guo, Yuzhen Han, Sen Hu, Ziyi Ni, Licheng Wang, Mingguang Chen, Hongzhang Liu, Ronghao Chen, Yangfan He, and 1 others. 2025. Seagent: Self-evolution trajectory optimization in multi-step reasoning with llm-based agents. *arXiv preprint arXiv:2508.02085*.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. 2017. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988.

Peiyang Liu. 2024. Unsupervised corrupt data detection for text training. *Expert Systems with Applications*, 248:123335.

Peiyang Liu, Zhirui Chen, Xi Wang, Di Liang, Youru Li, Zhi Cai, and Wei Ye. 2026. Learning from contrasts: Synthesizing reasoning paths from diverse search trajectories. *Preprint*, arXiv:2604.11365.

Peiyang Liu, Ziqiang Cui, Di Liang, and Wei Ye. 2025a. Who stole your data? a method for detecting unauthorized rag theft. *arXiv preprint arXiv:2510.07728*.

Peiyang Liu, Sen Wang, Xi Wang, Wei Ye, and Shikun Zhang. 2021a. Quadrupletbert: An efficient model for embedding-based large-scale retrieval. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3734–3739.

Peiyang Liu, Xi Wang, Ziqiang Cui, and Wei Ye. 2025b. Queries are not alone: Clustering text embeddings for video search. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 874–883.

Peiyang Liu, Xi Wang, Lin Wang, Wei Ye, Xiangyu Xi, and Shikun Zhang. 2021b. Distilling knowledge from bert into simple fully connected neural networks for efficient vertical retrieval. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pages 3965–3975.

Peiyang Liu, Xi Wang, Sen Wang, Wei Ye, Xiangyu Xi, and Shikun Zhang. 2021c. Improving embedding-based large-scale retrieval via label enhancement. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 133–142.

Peiyang Liu, Xiangyu Xi, Wei Ye, and Shikun Zhang. 2022. Label smoothing for text mining. In *Proceedings of the 29th international conference on computational linguistics*, pages 2210–2219.

Peiyang Liu, Jinyu Yang, Lin Wang, Sen Wang, Yunlai Hao, and Huihui Bai. 2023. Retrieval-based unsupervised noisy label detection on text data. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, pages 4099–4104.

Peiyang Liu, Wei Ye, Xiangyu Xi, Tong Wang, Jinglei Zhang, and Shikun Zhang. 2020. Not all synonyms are created equal: Incorporating similarity of synonyms to enhance word embeddings. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE.

Xiaoou Liu, Tiejin Chen, Longchao Da, Chacha Chen, Zhen Lin, and Hua Wei. 2025c. Uncertainty quantification and confidence calibration in large language models: A survey. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pages 6107–6117.

Matthew McDermott, Haoran Zhang, Lasse Hansen, Giovanni Angelotti, and Jack Gallifant. 2024. A closer look at auroc and auprc under class imbalance. *Advances in Neural Information Processing Systems*, 37:44102–44163.

John Platt and 1 others. 1999. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in large margin classifiers*, 10(3):61–74.

Marius-Constantin Popescu, Valentina E Balas, Liliana Perescu-Popescu, and Nikos Mastorakis. 2009. Multilayer perceptron and neural networks. *WSEAS Transactions on Circuits and Systems*, 8(7):579–588.

Nicolas Posocco and Antoine Bonnefoy. 2021. Estimating expected calibration errors. In *International conference on artificial neural networks*, pages 139–150. Springer.

Baptiste Roziere, Jonas Gehring, Fabian Gloeckle, Sten Sootla, Itai Gat, Xiaoqing Ellen Tan, Yossi Adi, Jingyu Liu, Romain Sauvestre, Tal Remez, and 1 others. 2023. Code llama: Open foundation models for code. *arXiv preprint arXiv:2308.12950*.

Kaspar Rufibach. 2010. Use of brier score to assess binary predictions. *Journal of clinical epidemiology*, 63(8):938–939.

Zailong Tian, Zhuoheng Han, Yanzhe Chen, Haozhe Xu, Xi Yang, Richeng Xuan, Houfeng Wang, and Lizi Liao. 2025. Overconfidence in llm-as-a-judge: Diagnosis and confidence-driven solution. *arXiv preprint arXiv:2508.06225*.

Miles Turpin, Julian Michael, Ethan Perez, and Samuel Bowman. 2023. Language models don't always say what they think: Unfaithful explanations in chain-of-thought prompting. *Advances in Neural Information Processing Systems*, 36:74952–74965.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2022. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, and 1 others. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Zijun Yan, Ke-qin Fan, Qi Zhang, Xinyan Wu, Yuquan Chen, Xinyu Wu, Ting Yu, Ning Su, Yan Zou, Hao Chi, and 1 others. 2025. Comparative analysis of the performance of the large language models deepseek-v3, deepseek-r1, open ai-o3 mini and open ai-o3 mini high in urology. *World Journal of Urology*, 43(1):416.

Zhaojian Yu, Yilun Zhao, Arman Cohan, and Xiao-Ping Zhang. 2025. Humaneval pro and mbpp pro: Evaluating large language models on self-invoking code generation task. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 13253–13279.

Jian Zhang, Xu Wang, Hongyu Zhang, Hailong Sun, Kaixuan Wang, and Xudong Liu. 2019. A novel neural source code representation based on abstract syntax tree. In *2019 IEEE/ACM 41st International Conference on Software Engineering (ICSE)*, pages 783–794. IEEE.

Mingyu Zhang, Zixu Li, Zhiwei Chen, Zhiheng Fu, Xiaowei Zhu, Jiajia Nie, Yinwei Wei, and Yupeng Hu. 2026a. Hint: Composed image retrieval with dual-path compositional contextualized network. *arXiv preprint arXiv:2603.26341*.

Qianchi Zhang, Hainan Zhang, Liang Pang, Yongxin Tong, Hongwei Zheng, and Zhiming Zheng. 2025a. Less is more: Compact clue selection for efficient retrieval-augmented generation reasoning. *arXiv preprint arXiv:2502.11811*.

Yuanjun Zhang, Fuzel Ahamed Shaik, Suvojit Acharjee, Fahad Khalid, and Mourad Oussalah. 2026b. Towards reliable multimodal disaster severity assessment through preference optimization and explainable vision-language reasoning. *Reliability Engineering & System Safety*, page 112674.

Zhuosheng Zhang, Yao Yao, Aston Zhang, Xiangru Tang, Xinbei Ma, Zhiwei He, Yiming Wang, Mark Gerstein, Rui Wang, Gongshen Liu, and 1 others. 2025b. Igniting language intelligence: The hitchhiker's guide from chain-of-thought reasoning to language agents. *ACM Computing Surveys*, 57(8):1–39.

Shuyan Zhou, Uri Alon, Sumit Agarwal, and Graham Neubig. 2023. Codebertscore: Evaluating code generation with pretrained models of code. *arXiv preprint arXiv:2302.05527*.