Can LLMs Reliably Evaluate Themselves? A Probabilistic VC Framework

Jae Oh Woo, Mengdie Flora Wang, Rahul Ghosh, Baishali Chaudhury, Mun Young Kim
AWS GenAI Innovation Center
{jaeohwoo, florawan, rahulgh, baishch, munkim}@amazon.com

Abstract

As large language models (LLMs) find increasing use in critical applications, evaluating their ability to assess their own outputs has become crucial. Our work presents a theoretical and empirical framework that examines whether LLMs can differentiate between correct and incorrect solutions while maintaining properly calibrated confidence. We build upon classical Vapnik-Chervonenkis (VC) dimension theory, adapting it to probabilistic predictors through two new complexity metrics: Probabilistic VC (PVC), which measures a model's ability to confidently classify across problem types, and Calibration-aware PVC (C-PVC), which demands alignment between confidence scores and actual success rates. Unlike traditional metrics such as Expected Calibration Error (ECE) and Actual Error (AE), these measures provide an integrated assessment of self-evaluation expressiveness and calibration, yielding sample complexity bounds and generalization guarantees comparable to traditional VC theory. In our study, we tested eleven models (7-8B parameters) across three diverse benchmarks: 360 mathematical reasoning problems, TruthfulQA for factual accuracy, and CommonsenseQA for commonsense reasoning. Each model had to choose between two of its own generated solutions and report its confidence level—a direct test of self-evaluation capability—with ground-truth determined by a larger model ensemble. The experimental results empirically substantiate a systematic inverse correlation: models exhibiting enhanced self-evaluation expressiveness consistently demonstrate diminished calibration fidelity. Models like s1.1-7B and Qwen2.5-7B-Instruct achieve high PVC-VUS scores, indicating strong discriminative self-assessment capacity, while JiuZhang3.0-7B demonstrates superior calibration with the lowest ECE and smallest PVC-VUS gap. Interestingly, we observe domain-specific variations in self-evaluation abilities, with some models performing better on mathematical reasoning tasks while others excel in factual or commonsense domains. Our analysis suggests complex interactions between training methodologies and self-evaluation performance, indicating that multiple factors beyond training approach influence a model's ability to accurately assess its own outputs. The fundamental trade-off between calibration and expressiveness constitutes a persistent phenomenon transcending architectural variations, training paradigms, and cognitive domains, pointing to a fundamental challenge in developing self-reflective LLMs. The framework we've developed offers practical tools for identifying and addressing these limitations, helping create LLMs that can not only tackle complex problems but also recognize when they might be wrong—an essential capability for safe deployment and meaningful self-improvement in autonomous systems.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

1 Introduction

A key requirement for autonomous reasoning systems is not only the ability to solve complex tasks, but also the ability to evaluate the reliability of their own reasoning processes. This self-evaluation capability—the ability to distinguish between correct and incorrect reasoning paths—is essential for systems that aim to improve without external supervision and ensure safe deployment in critical applications. As large language models (LLMs) are increasingly applied to tasks requiring multi-step reasoning, a natural question arises: to what extent can these models identify which of their reasoning paths are more likely to be correct?

Recent progress in LLMs has led to significant improvements in reasoning performance across domains [32, 70, 13], including mathematical problem solving [38], pattern recognition [48, 76], and abstract reasoning [14]. These advances have been supported by preference-based fine-tuning techniques such as Direct Preference Optimization (DPO) [55] and Group Relative Policy Optimization (GRPO) [59, 22, 72, 74]. However, existing evaluation protocols primarily focus on final answer accuracy, offering limited insight into a model's ability to assess its own reasoning quality.

This motivates our investigation into model self-evaluation capabilities, encompassing *self-reflection*—the internal mechanism through which a model assesses the correctness of its own reasoning—among other aspects of metacognitive assessment. While recent studies [61, 53, 40, 65, 68] have explored various self-evaluation heuristics, a principled framework for analyzing and comparing such capabilities remains lacking.

To address this gap, we propose a formal approach grounded in statistical learning theory. We extend the classical Vapnik–Chervonenkis (VC) dimension [67, 11], which characterizes hypothesis class capacity via binary classification [58]. Standard VC theory is limited for modern LLMs, which generate probabilistic outputs rather than deterministic labels. Our solution is the *Probabilistic VC (PVC)* dimension, designed for probabilistic predictors, maintaining connections to calibration and margin-based generalization.

1.1 Our Contribution

We extend classical VC theory by introducing two new complexity measures. The *Probabilistic VC (PVC)* dimension quantifies a model's ability to make confident predictions across problem categories, while the *Calibration-aware PVC (C-PVC)* dimension additionally requires that confidence levels align with actual correctness probabilities. We establish theoretical guarantees linking these measures to generalization performance and sample complexity, providing a principled basis for analyzing self-evaluation capabilities in language models.

Our framework enables practical assessment by estimating lower bounds on PVC and C-PVC dimensions. Across multiple reasoning domains, we found a consistent inverse relationship between self-evaluation expressiveness and calibration quality in 7-8B parameter models. While models like s1.1-7B and Qwen2.5-7B-Instruct excel at discriminative capacity, others like JiuZhang3.0-7B show better calibration. This fundamental trade-off persists across different architectures, training methods, and domains, suggesting an inherent constraint in probabilistic reasoning systems. These findings highlight that multiple factors beyond training approach affect a model's self-evaluation capabilities—critical for building reliable autonomous systems.

2 Background

2.1 Classical VC Dimension

The Vapnik–Chervonenkis (VC) dimension [67, 11, 58] is a foundational measure of the capacity of a binary hypothesis class $\mathcal{H} \subseteq \{h : \mathcal{X} \to \{0, 1\}\}$. A set $\{x_1, \ldots, x_n\} \subset \mathcal{X}$ is *shattered* by $\mathcal{H}$ if for every labeling $(y_1, \ldots, y_n) \in \{0, 1\}^n$, there exists $h \in \mathcal{H}$ such that $h(x_i) = y_i$ for all i. The VC dimension, $\mathrm{VC}(\mathcal{H})$, is the size of the largest such set, intuitively measuring how many points the hypothesis class can classify in all possible ways.

VC dimension is tightly linked to generalization. If $\text{VC}(\mathcal{H}) = d$, then with high probability:

$$\sup_{h \in \mathcal{H}} |R(h) - \hat{R}(h)| = \mathcal{O}\left(\sqrt{\frac{d + \log(1/\delta)}{n}}\right),$$

where $R(h)$ is true risk (expected error on the population) and $\hat{R}(h)$ is empirical risk (average error on the sample). This implies a sample complexity of $\Omega\left(\frac{d+\log(1/\delta)}{\epsilon^2}\right)$ to achieve accuracy ϵ with confidence $1 - \delta$. However, classical VC theory is limited to binary-valued outputs, making it insufficient for analyzing LLMs that output confidence scores rather than hard decisions.

2.2 Fat-Shattering Dimension

The fat-shattering dimension [6, 1] generalizes VC theory from binary hypothesis classes $\mathcal{H}$ to real-valued function classes $\mathcal{F} \subseteq \{f : \mathcal{X} \to \mathbb{R}\}$, capturing how finely functions can distinguish inputs at a given scale $\alpha > 0$. A set $\{x_1, \ldots, x_n\}$ is α-shattered by $\mathcal{F}$ if there exist thresholds $s_1, \ldots, s_n$ such that for all $\epsilon \in \{\pm 1\}^n$, there exists $f \in \mathcal{F}$ satisfying:

$$\epsilon_t(f(x_t) - s_t) \geq \alpha \quad \text{for all } t \in [n].$$

This condition means f can place each output either α above or below its threshold specified by ϵ_t.

The fat-shattering dimension at scale α, denoted $\text{fat}_\alpha(\mathcal{F})$, is the largest n for which such a set exists. For binary functions, $\text{fat}_\alpha(\mathcal{F}) = \text{VC}(\mathcal{F})$ for all $\alpha \in (0, 1)$, and is zero for $\alpha \geq 1$. Fat-shattering serves as a scale-sensitive extension of VC dimension and provides the foundation for our probabilistic extension to LLMs [1, 7, 6].

3 Methods

We develop a theoretical framework for analyzing LLMs' self-evaluation capabilities through statistical learning theory. We extend classical VC theory to probabilistic predictors via two new complexity metrics: Probabilistic VC (PVC) dimension, which measures a model's ability to make confident classifications, and Calibration-aware PVC (C-PVC), which additionally requires alignment between confidence scores and actual success rates.

3.1 Probabilistic VC Dimension and its Calibration-Aware Extension

In this paper, we use notation $\text{PVC}_\gamma(\mathcal{F})$ and $\text{C-PVC}^\tau_\gamma(\mathcal{F})$ when discussing theoretical definitions and general properties, and the abbreviated notation PVC_γ and C-PVC^τ_γ when the function class is clear from context.

To analyze self-evaluation capabilities in language models, we extend the classical VC framework to accommodate probabilistic predictions. Rather than modeling hypotheses as deterministic binary functions, we consider predictors that output a distribution over possible labels.

Definition 1 (Probabilistic VC Dimension)**.** *Let $\mathcal{F}$ be a class of probabilistic predictors $f : \mathcal{X} \to \Delta(\{0, 1\})$, where $\Delta(\{0, 1\})$ denotes the set of probability distributions over the binary output space. Fix a confidence threshold $\gamma \in (0, 1]$. The probabilistic VC dimension $\text{PVC}_\gamma(\mathcal{F})$ is the largest integer d such that there exists a set $\{x_1, \ldots, x_d\} \subseteq \mathcal{X}$ with the following property: for every labeling $y_1^*, \ldots, y_d^* \in \{0, 1\}$, there exists a predictor $f \in \mathcal{F}$ such that*

$$\mathbb{P}\left(f(x_i) = y_i^*\right) \geq \gamma \quad \textit{for all } i = 1, \ldots, d.$$

For example, if $\text{PVC}_{0.8}(\mathcal{F}) = 3$, then there exists a set of 3 inputs where the model can assign any binary labeling with at least 80% probability, but no set of 4 inputs can be so labeled. This relaxes classical shattering by requiring high-confidence probabilistic support rather than exact realizability. Importantly, this generalization aligns closely with the fat-shattering dimension [1, 7, 6], and can be seen as a probabilistic analogue tailored for confidence-aware reasoning tasks.

While PVC captures confident predictions, it doesn't ensure these confidence scores reflect actual correctness probabilities. For high-stakes applications, calibration is crucial:

Definition 2 (τ-Calibration)**.** *A probabilistic predictor f is τ-calibrated, for some $\tau \in [0, 1)$, if for all $p \in [0, 1]$,*

$$|\mathbb{P}\left(y = y^* \mid f(x) \text{ assigns confidence } p \text{ to } y^*\right) - p| \leq \tau.$$

This captures approximate calibration: a predictor is τ-calibrated if its confidence values deviate from true correctness probabilities by at most τ. We now integrate confident prediction and calibration:

Definition 3 (Calibration-aware Probabilistic VC Dimension)**.** *Let $\mathcal{F}$ be a class of probabilistic predictors $f : \mathcal{X} \to \Delta(\{0, 1\})$ and let $\hat{p} : \mathcal{X} \to [0, 1]$ denote a confidence scoring function associated with each $f \in \mathcal{F}$.*

Fix confidence threshold $\gamma \in (0, 1]$ and calibration error tolerance $\tau \in [0, 1)$. The calibration-aware probabilistic VC dimension $\text{C-PVC}^{\tau}_{\gamma}(\mathcal{F})$ is the largest integer d such that there exists a set $\{x_1, \ldots, x_d\} \subseteq \mathcal{X}$ with the following property:

For every labeling $(y_1^*, \ldots, y_d^*) \in \{0, 1\}^d$, there exists a pair $(f, \hat{p})$ such that:

$$\mathbb{P}(f(x_i) = y_i^*) \geq \gamma, \quad \textit{and} \quad |\hat{p}(x_i) - \mathbb{E}[\mathbf{1}\{f(x_i) = y_i^*\}]| \leq \tau \quad \textit{for all } i = 1, \ldots, d.$$

While PVC captures expressive power alone, C-PVC reflects both expressivity and calibration—essential for reliable self-evaluation.

3.2 LLMs as Function Classes

To apply our framework to LLMs, we must interpret a single model as a function class. We outline three complementary perspectives:

Bayesian view (parameter-level randomization). Treat the parameter vector W as a random variable $W \sim \mathcal{P}$, inducing the function class $\{f_W : W \in \text{supp}(\mathcal{P})\}$. This aligns with PAC-Bayesian analyses of randomized predictors [41, 19], the correspondence between infinitely wide neural networks and Gaussian processes [34, 16], and non-vacuous deep-network generalization bounds via PAC-Bayes [17].

Prompt-based view (input-conditioning as a hypothesis family). With fixed W, varying a prompt p from a constraint set Π yields:

$$\mathcal{H}_{\text{prompt}}(f_W) \;=\; \{\, x \mapsto f_W([p; x]) \;:\; p \in \Pi \,\}.$$

This captures prompt-tuning effects [37, 36] and aligns with in-context learning interpretations [71].

Stochastic decoding view (output-level randomization). Decoding randomness (e.g., temperature sampling or inference-time dropout) induces a distribution over predictors, analogous to Gibbs predictors in PAC-Bayesian theory [41, 19], and the Bayesian interpretation of Monte Carlo dropout pertains to inference-time sampling [18].

In our experiments, we primarily adopt the prompt-based and stochastic-decoding views to define our practical hypothesis class.

3.3 Measuring PVC and C-PVC through Self-Evaluation

To empirically estimate PVC and C-PVC dimensions, we treat dataset categories (e.g., Algebra, Geometry in mathematics) as the units of shattering and employ a three-stage evaluation protocol (Figure 1): 1) The model generates two distinct solutions to each problem, 2) The model evaluates both solutions and reports its confidence, 3) External judge LLMs determine the objectively superior solution.

Our approach is based on two key metrics:

Definition 4 (Self-Evaluation Accuracy)**.** *Let $f \in \mathcal{F}$ be a probabilistic predictor, and C a question category. For each question q, let $(A_1(q), A_2(q))$ be two candidate solutions, $\mathsf{Judge}(q) \in \{1, 2\}$ the index of the superior solution, and $\mathsf{Select}_f(q)$ the model's choice. Then:*

$$\text{SEA}(f, C) = \mathbb{E}_{q \sim C}\left[\mathbf{1}\left\{\mathsf{Select}_f(q) = \mathsf{Judge}(q)\right\}\right].$$

Definition 5 (Calibration Error)**.** *For a category C and a model's reported confidence $\hat{p}(q)$ for each question $q \in C$, we define:*

$$\text{CalibError}(f, C) = \left|\mathbb{E}_{q \sim C}[\hat{p}(q)] - \text{SEA}(f, C)\right|.$$

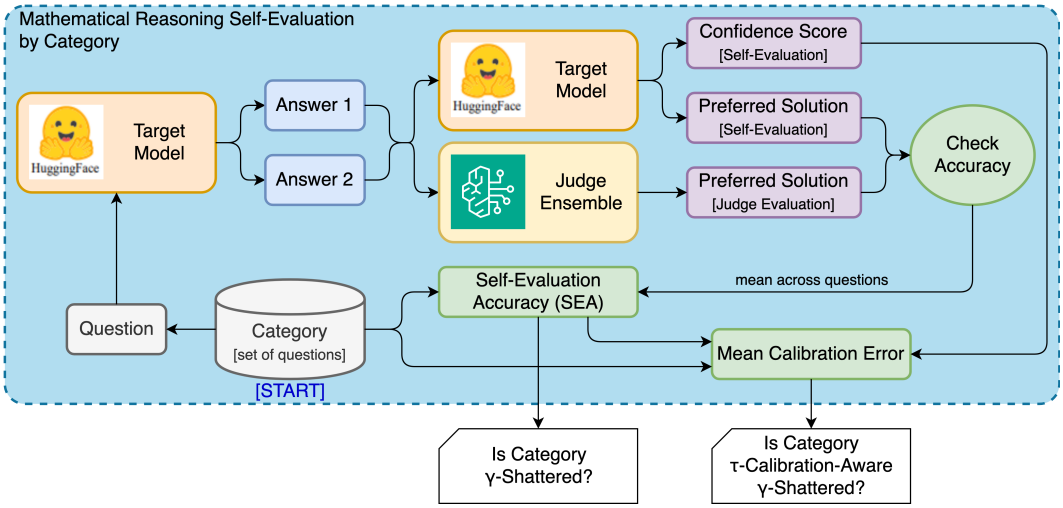


Figure 1: PVC/C-PVC Experiment Flow: Models generate two solutions, perform self-evaluation, and are compared against expert ensemble judgment.

Using these metrics, we say a category C is γ-*shattered* by predictor f if $\mathrm{SEA}(f, C) \geq \gamma$. The empirical PVC dimension is then the largest number of such categories. Similarly, a category counts toward the empirical C-PVC dimension if both $\mathrm{SEA}(f, C) \geq \gamma$ and $\mathrm{CalibError}(f, C) \leq \tau$, reflecting both high accuracy and good calibration.

3.4 Aggregation with Volume Under Surface (VUS)

When evaluating models using PVC, C-PVC, and Sample Complexity metrics, parameter dependence presents a challenge. These measures vary across confidence threshold γ and calibration tolerance τ values, making single-value comparisons difficult. To address this, we adopt the Volume Under Surface (VUS) methodology, extending the AUC concept to three dimensions:

$$M\text{-VUS}(\mathcal{F}) = \iint_{\mathcal{G}} M(\gamma, \tau, \mathcal{F})\, d\gamma\, d\tau \tag{1}$$

where $M \in \{\text{PVC}, \text{C-PVC}, \text{SC}\}$ and $\mathcal{G}$ represents $(\gamma, \tau) \in (0, 1] \times [0, 1)$. This integration captures aggregate performance across the parameter space, providing a robust summary statistic.

A high PVC-VUS indicates strong discriminative self-assessment across confidence thresholds, while high C-PVC-VUS suggests both accurate and well-calibrated self-assessment. SC-VUS summarizes theoretical sample complexity requirements. Although PVC is independent of τ, we treat it within the same framework for consistency, integrating over γ with uniform distribution over τ.

4 Theoretical Results

This section presents the core theoretical foundations of our probabilistic VC (PVC) framework. We establish key relationships between different dimension variants, derive generalization guarantees, and demonstrate how our framework extends classical VC theory to the probabilistic setting of self-evaluating language models.

4.1 Relationships Between Dimension Variants

A natural first question is how different variants of PVC dimension relate to each other and to classical VC dimension. The following proposition establishes these fundamental relationships:

Proposition 1 (Relationships Between Dimension Variants)**.** *For a class of predictors $\mathcal{F}$, the following relationships hold:*

1. *For any $\gamma_1, \gamma_2 \in (0, 1]$ where $\gamma_1 > \gamma_2$:*

$$VC(\mathcal{F}) = PVC_1(\mathcal{F}) \leq PVC_{\gamma_1}(\mathcal{F}) \leq PVC_{\gamma_2}(\mathcal{F}),$$

2. *For a fixed $\gamma \in (0, 1]$ and any $\tau_1, \tau_2 \in [0, 1)$ where $\tau_1 > \tau_2$:*

$$C\text{-}PVC_\gamma^{\tau_2}(\mathcal{F}) \leq C\text{-}PVC_\gamma^{\tau_1}(\mathcal{F}) \leq PVC_\gamma(\mathcal{F}).$$

This proposition reveals several important insights. First, classical VC dimension corresponds to PVC with perfect confidence ($\gamma = 1$), establishing a clear connection to traditional learning theory. Second, as we relax the confidence threshold, the PVC dimension increases monotonically, reflecting the intuition that making lower-confidence predictions allows for greater expressivity.

In our analysis, we distinguish between two key parameters with fundamentally different roles. The parameter γ specifies a minimum confidence threshold that characterizes the expressive power of a hypothesis class. In contrast, δ represents a probabilistic upper bound on error in generalization analysis—the confidence level with which a learned predictor generalizes to unseen data.

A central finding of our work is that despite handling probabilistic outputs, the PVC framework yields generalization bounds that closely mirror those of classical VC theory. This connection becomes clear when we relate PVC dimension to the fat-shattering dimension, a well-established scale-sensitive extension of VC theory [1, 6, 15]. The detailed sample complexity bounds for the basic PVC dimension are provided in Appendix H, which includes the formal proof relating PVC to fat-shattering via a margin parameter $\alpha = \gamma - \frac{1}{2}$.

4.2 Generalization Bounds for Calibration-aware PVC

While PVC captures a model's ability to make confident predictions, the calibration-aware extension provides stronger guarantees about both prediction accuracy and calibration quality:

Theorem 1 (Generalization Bounds for Calibration-aware PVC)**.** *Let $\mathcal{F}$ be a class of binary predictors $f : \mathcal{X} \to \{0, 1\}$, each with an associated confidence function $\hat{p} : \mathcal{X} \to [0, 1]$. For confidence threshold $\gamma \in (0, 1]$ and calibration tolerance $\tau \in [0, 1)$, assume $\text{C-PVC}_\gamma^\tau(\mathcal{F}) = d_{\gamma,\tau} < \infty$.*

Then there exists a universal constant $C > 0$ such that for any $\epsilon, \delta \in (0, 1)$, if the sample size satisfies:

$$m \geq \frac{C}{\epsilon^2}\left(d_{\gamma,\tau} + \log\frac{1}{\delta}\right), \tag{2}$$

the following generalization guarantees hold uniformly across all $f \in \mathcal{F}$ with probability at least $1 - \delta$:

$$\left|\frac{1}{m}\sum_{i=1}^{m}\mathbf{1}\{f(x_i) \neq y_i\} - \mathbb{P}[f(x) \neq y]\right| \leq \epsilon \quad \textit{(prediction error)} \tag{3}$$

$$\left|\frac{1}{m}\sum_{i=1}^{m}\left(\hat{p}(x_i) - \mathbf{1}\{f(x_i) = y_i\}\right) - \mathbb{E}\left[\hat{p}(x) - \mathbf{1}\{f(x) = y\}\right]\right| \leq \epsilon + \tau \quad \textit{(calibration error)} \tag{4}$$

The proof relies on showing that C-PVC dimension bounds the fat-shattering dimension of both the prediction function class and the calibration error function class. This allows us to apply uniform convergence results for real-valued function classes to both prediction accuracy and calibration error simultaneously.

This result mirrors the classical VC generalization bound, highlighting that our PVC framework retains the same asymptotic structure in terms of sample complexity. In particular, the dependence on ϵ^{-2} and $\log(1/\delta)$ remains unchanged, while the VC dimension is replaced by the C-PVC dimension $d_{\gamma,\tau}$, which captures both probabilistic expressivity and calibration quality.

Table 1: Experimental results comparing model performance across three datasets (Math-360, TruthfulQA, and CSQA) and their average. For each metric, we report values as: Math-360 / TruthfulQA / CSQA . PVC-VUS Gap represents the difference between PVC-VUS and C-PVC-VUS. ECE (Expected Calibration Error) measures the alignment between confidence and accuracy (lower is better), while AE (Actual Error) indicates the model's overall error rate in self-evaluation tasks.

Model	PVC-VUS ↑	C-PVC-VUS ↑	PVC-VUS Gap ↓	SC-VUS ↑	ECE ↓	AE ↓
Qwen2.5-7B (Pretrain) [72]	5.39 (4.78 / 5.66 / 5.73)	3.76 (3.27 / 3.93 / 4.09)	1.63 (1.51 / 1.73 / 1.64)	676.5 (627.3 / 693.0 / 709.2)	0.314 (0.320 / 0.318 / 0.303)	0.422 (0.403 / 0.435 / 0.429)
Qwen2.5-7B-Instruct (SFT+RL) [72]	5.61 (4.65 / 6.13 / 6.04)	4.09 (3.13 / **4.73** / 4.40)	1.52 (1.52 / **1.40** / 1.64)	708.3 (612.6 / **772.6** / 739.7)	0.285 (0.337 / **0.247** / 0.290)	0.401 (0.419 / 0.388 / 0.396)
Qwen2.5-Math-7B-Instruct (SFT+RL) [73]	4.70 (4.32 / 4.63 / 5.16)	3.20 (3.10 / 2.95 / 3.56)	1.50 (1.22 / 1.68 / 1.60)	620.3 (610.1 / 595.2 / 655.6)	0.335 (0.289 / 0.388 / 0.329)	0.494 (0.461 / 0.538 / 0.483)
Llama-3.1-8B-Instruct (SFT+DPO) [43]	4.55 (4.84 / 4.13 / 4.68)	2.92 (3.51 / 2.44 / 2.80)	1.63 (1.33 / 1.69 / 1.88)	591.9 (651.1 / 544.4 / 580.2)	0.385 (0.284 / 0.462 / 0.408)	0.505 (0.394 / 0.588 / 0.533)
OpenThinker2-7B (SFT) [49]	5.37 (**5.19** / 5.46 / 5.46)	3.50 (**3.68** / 3.36 / 3.47)	1.87 (1.51 / 2.10 / 1.99)	649.9 (667.5 / 635.6 / 646.6)	0.357 (0.298 / 0.395 / 0.378)	0.420 (**0.351** / 0.456 / 0.454)
DeepSeek-R1-Distill-Qwen-7B (Distill) [22]	5.10 (4.77 / 5.01 / 5.51)	3.49 (3.24 / 3.40 / 3.83)	1.61 (1.53 / 1.61 / 1.68)	648.9 (624.3 / 639.5 / 683.0)	0.339 (0.329 / 0.358 / 0.330)	0.451 (0.404 / 0.500 / 0.450)
Bespoke-Stratos-7B (SFT) [10]	5.33 (4.44 / 5.67 / 5.89)	3.54 (2.73 / 3.95 / 3.94)	1.79 (1.71 / 1.72 / 1.95)	654.0 (573.4 / 694.7 / 693.7)	0.347 (0.388 / 0.315 / 0.338)	0.430 (0.444 / 0.433 / 0.413)
JiuZhang3.0-7B (SFT) [77]	4.79 (4.49 / 4.55 / 5.34)	3.84 (3.74 / 3.22 / 4.57)	**0.95** (**0.75** / 1.33 / **0.77**)	684.4 (**674.1** / 622.1 / 757.1)	**0.209** (**0.169** / 0.309 / **0.149**)	0.484 (0.439 / 0.546 / 0.467)
Ministral-8B-Instruct-2410 (SFT+RL) [44]	4.88 (4.21 / 4.61 / 5.81)	3.00 (2.50 / 2.65 / 3.85)	1.88 (1.71 / 1.96 / 1.96)	600.0 (549.6 / 565.0 / 685.4)	0.402 (0.414 / 0.453 / 0.339)	0.478 (0.472 / 0.542 / 0.421)
Open-Reasoner-Zero-7B (RL) [24]	5.29 (4.52 / 5.47 / 5.88)	3.50 (2.78 / 3.71 / 4.02)	1.79 (1.74 / 1.76 / 1.86)	650.6 (578.2 / 671.3 / 702.4)	0.352 (0.388 / 0.341 / 0.327)	0.434 (0.436 / 0.454 / 0.413)
s1.1-7B (SFT) [46]	**5.83** (4.76 / **6.22** / **6.52**)	**4.27** (3.22 / 4.71 / **4.88**)	1.56 (1.54 / 1.51 / 1.64)	**727.3** (622.5 / 771.4 / **788.0**)	0.285 (0.331 / 0.256 / 0.267)	**0.378** (0.407 / **0.379** / **0.348**)

5 Experiments and Results

5.1 Experimental Setup

To assess reasoning self-evaluation capabilities, we constructed a novel benchmark Math-360 of 360 original math problems across 8 domains, stratified by difficulty and subcategory. This design avoids contamination from pretraining data and captures diverse reasoning patterns. To evaluate whether PVC and C-PVC generalize beyond mathematical reasoning tasks, we extended our empirical analysis to two widely used non-mathematical benchmarks: TruthfulQA [39], which assesses factual correctness under adversarial phrasing, and CommonsenseQA [63], which evaluates everyday commonsense reasoning. To ensure a balanced evaluation, we grouped each of the latter two datasets into 10 broad categories and sampled 240 questions per benchmark.

We evaluate eleven 7–8B parameter models using standardized inference settings and structured parsing. Output confidence scores were judged using an ensemble of three larger LLMs (Claude Sonnet 3.7 [4], Amazon Nova Premier [26], and DeepSeek-R1 [22]) to mitigate individual biases.

For Sample Complexity calculations, we assume fixed values of $C = 1$, $\epsilon = 0.1$ and $\delta = 0.05$. This allows us to compute SC-VUS while maintaining consistent theoretical guarantees across all evaluated models.

5.2 PVC and Calibration Performance

Table 1 presents results comparing model performance across three datasets. We report PVC-VUS, which quantifies a model's expressive self-assessment capacity, and C-PVC-VUS, which additionally incorporates calibration requirements. The PVC-VUS Gap represents the difference between these metrics, with smaller gaps indicating better calibration alignment.

The results demonstrate a consistent pattern across models: a trade-off between expressive power and calibration quality. This empirically validates our theoretical framework, particularly Proposition 1. Models like s1.1-7B and Qwen2.5-7B-Instruct achieve high PVC-VUS scores (5.83 and 5.61 respectively), indicating strong discriminative self-assessment capacity, while JiuZhang3.0-7B demonstrates superior calibration with the lowest ECE (0.209) and smallest PVC-VUS gap (0.95), despite a moderate PVC-VUS (4.79).

Instruction-tuned models consistently outperform their base counterparts in both PVC-VUS and C-PVC-VUS, suggesting that instruction tuning enhances self-evaluation capabilities. This pattern aligns

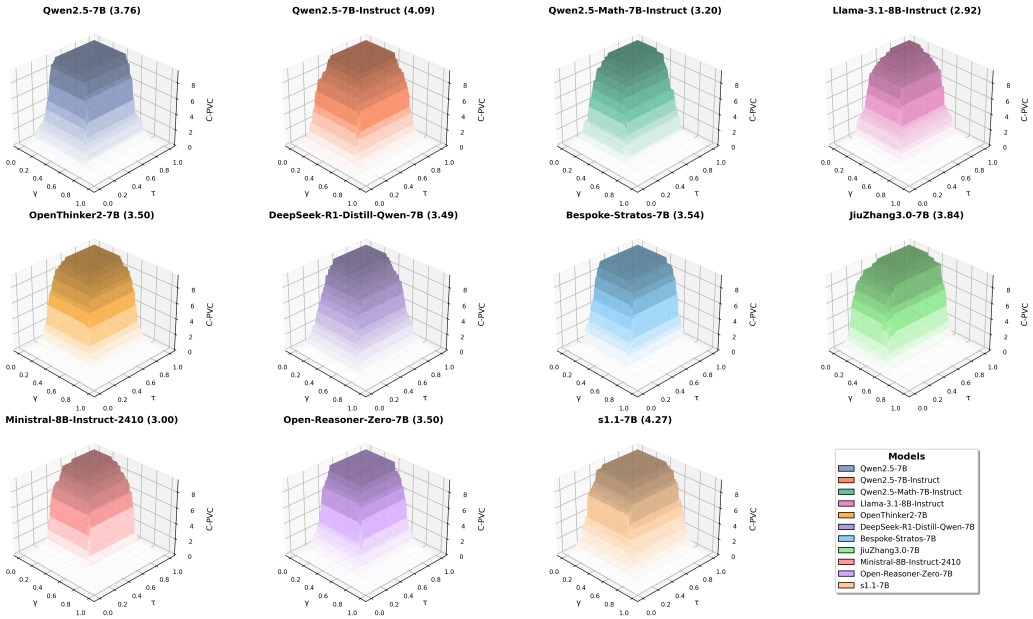


Figure 2: Calibration-aware Probabilistic VC (C-PVC) Dimension: 3D surface plots showing C-PVC dimension values for eleven 7-8B parameter language models. The x-axis (τ) represents calibration error tolerance, y-axis (γ) shows confidence threshold, and z-axis displays C-PVC values. Our framework quantifies self-evaluation capacity through these distinctive topographical signatures. Higher C-PVC-VUS (in parentheses) indicates better self-evaluation capabilities.

with our theoretical understanding that training approaches that explicitly optimize for reasoning quality can improve a model's capacity to discriminate between correct and incorrect solutions.

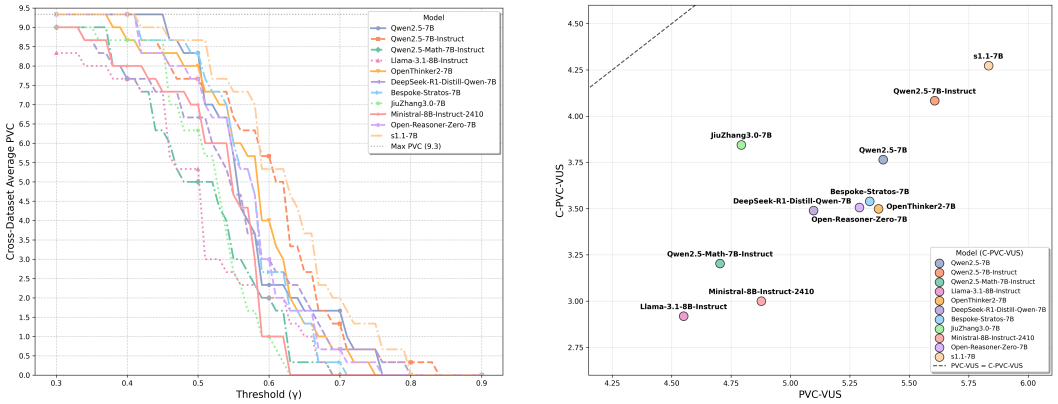
Figure 2 provides a visualization of C-PVC performance across different parameter settings, while Figure 3 illustrates the relationship between expressivity and calibration. The right panel of Figure 3 demonstrates a non-monotonic relationship between discriminative capacity and calibration fidelity.

5.3 Cross-Domain Analysis

Our analysis demonstrates domain-specific patterns in self-evaluation performance. Qwen2.5-7B-Instruct show particularly strong performance in factual domains (TruthfulQA PVC-VUS: 6.13); Llama-3.1-8B-Instruct demonstrates stronger performance in mathematical reasoning (Math-360 PVC-VUS: 4.84), while general-purpose models like s1.1-7B perform more consistently across domains. We also observe that models that are well-calibrated in one domain tend to maintain better calibration across domains. JiuZhang3.0-7B maintains relatively small PVC-VUS Gaps across all datasets, suggesting that calibration quality may transfer more readily across domains than raw discriminative power. Critically, the expressivity-calibration trade-off persists across all three domains, confirming this is a fundamental property of self-evaluation rather than a domain-specific artifact. This consistency supports the generalizability of our PVC framework as predicted by Theorem 1.

6 Related works

Large language models (LLMs) have demonstrated remarkable performance on complex reasoning tasks [32, 70, 13], with standard evaluation benchmarks including MathVista [38], Bongard-Logo [48], Raven [76], and the Abstraction and Reasoning Corpus [14]. Optimization techniques such as Direct Preference Optimization [55] and Group Relative Policy Optimization [59, 22, 72, 74] have further enhanced LLM performance on reasoning tasks. However, as [45] comprehensively document, most evaluation frameworks prioritize final answer correctness over assessing reasoning quality or metacognitive capabilities—a limitation our work addresses directly.



(a) PVC Across Confidence γ

(b) Comparison of PVC-VUS and C-PVC-VUS

Figure 3: Model Self-Evaluation Capabilities and Calibration Performance: (a) PVC scores across confidence thresholds (γ) for eleven language models. Higher curves indicate better ability to maintain expressive reasoning as confidence requirements increase. s1.1-7B and Qwen2.5-7B-Instruct show the strongest performance. (b) Comparison between PVC-VUS (expressiveness) and C-PVC-VUS (calibrated expressiveness). Points closer to the diagonal dashed line indicate better calibration. JiuZhang3.0-7B shows the smallest gap between these metrics, demonstrating superior calibration, while s1.1-7B achieves the highest overall expressiveness.

The Vapnik-Chervonenkis (VC) dimension [67, 11] provides a theoretical foundation for understanding model capacity and generalization in learning systems. Classical VC theory established sample complexity bounds [58] and generalization guarantees that have shaped machine learning theory for decades. Our work expands probabilistic VC theory by introducing calibration-aware extensions that simultaneously capture a model's expressive power and its ability to produce well-calibrated confidence estimates. This builds upon fat-shattering dimension theory [1, 6, 7], which extends VC concepts to real-valued functions.

Calibration—the alignment between confidence scores and actual accuracy—has been extensively studied in deep learning [21, 51] and specifically in LLMs [29]. Our theoretical framework contributes to this literature by establishing direct connections between calibration quality and generalization guarantees, particularly for reasoning tasks. For a more comprehensive review of related research in self-reflection capabilities, probabilistic learning theory, and preference optimization approaches, we refer readers to Appendix M.

7 Conclusion

This paper presents a principled framework for analyzing the self-assessment capabilities of large language models through statistical learning theory. By extending classical VC dimension to probabilistic predictors, we introduce two key complexity measures: the Probabilistic VC (PVC) dimension, capturing confident discriminative capacity, and its calibration-aware extension (C-PVC), evaluating a model's ability to produce reliable confidence estimates. Our theoretical analysis establishes sample complexity bounds and generalization guarantees for both metrics, addressing the unique characteristics of probabilistic predictors.

Our framework enables fine-grained evaluation of introspection and offers practical tools for model development, auditing, and debugging confidence-related failures. The PVC and C-PVC metrics provide integrated assessment extending beyond traditional metrics like ECE and AE. As LLMs are increasingly applied to safety-critical settings, the ability to assess reasoning reliability becomes essential. Our framework lays groundwork for systems that can recognize and communicate uncertainty, fostering transparency and trust. We advocate for future research exploring factors influencing self-assessment capabilities and optimizing both reasoning accuracy and introspective reliability simultaneously.

References

[1] N. Alon, S. Ben-David, N. Cesa-Bianchi, and D. Haussler. Scale-sensitive dimensions, uniform convergence, and learnability. *Journal of the ACM (JACM)*, 44(4):615–631, 1997.

[2] A. N. Angelopoulos and S. Bates. A gentle introduction to conformal prediction and distribution-free uncertainty quantification. *arXiv preprint arXiv:2107.07511*, 2021.

[3] M. Anthony and P. L. Bartlett. *Neural network learning: Theoretical foundations*. cambridge university press, 2009.

[4] Anthropic. Claude 3.7 Sonnet sysem card. `https://assets.anthropic.com/m/785e231869ea8b3b/original/claude-3-7-sonnet-system-card.pdf`, 2025.

[5] P. L. Bartlett, N. Harvey, C. Liaw, and A. Mehrabian. Nearly-tight vc-dimension and pseudodimension bounds for piecewise linear neural networks. *Journal of Machine Learning Research*, 20(63):1–17, 2019.

[6] P. L. Bartlett and P. M. Long. More theorems about scale-sensitive dimensions and learning. In *Proceedings of the eighth annual conference on Computational learning theory*, pages 392–401, 1995.

[7] P. L. Bartlett, P. M. Long, and R. C. Williamson. Fat-shattering and the learnability of real-valued functions. In *Proceedings of the seventh annual conference on Computational learning theory*, pages 299–310, 1994.

[8] P. L. Bartlett and S. Mendelson. Rademacher and gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3(Nov):463–482, 2002.

[9] M. Bentvelzen, P. W. Woźniak, P. S. Herbes, E. Stefanidi, and J. Niess. Revisiting reflection in hci: Four design resources for technologies that support reflection. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 6(1):1–27, 2022.

[10] Bespoke Labs. Bespoke stratos 7b. `https://huggingface.co/bespokelabs/Bespoke-Stratos-7B`, 2023. Accessed: October 19, 2025.

[11] A. Blumer, A. Ehrenfeucht, D. Haussler, and M. K. Warmuth. Learnability and the vapnik-chervonenkis dimension. *Journal of the ACM (JACM)*, 36(4):929–965, 1989.

[12] A. R. Brown. Visualizing digital media interactions: providing feedback on jam2jam AV performances. In *Proceedings of the 22nd Conference of the Computer-Human Interaction Special Interest Group of Australia on Computer-Human Interaction*, pages 196–199, Brisbane Australia, Nov. 2010. ACM.

[13] W. Chen, X. Ma, X. Wang, and W. W. Cohen. Program of thoughts prompting: Disentangling computation from reasoning for numerical reasoning tasks. *arXiv preprint arXiv:2211.12588*, 2022.

[14] F. Chollet. On the measure of intelligence. *arXiv preprint arXiv:1911.01547*, 2019.

[15] R. Colomboni, E. Esposito, and A. Paudice. An improved uniform convergence bound with fat-shattering dimension. *Information Processing Letters*, 188:106539, 2025.

[16] A. G. de G. Matthews, J. Hron, M. Rowland, R. E. Turner, and Z. Ghahramani. Gaussian process behaviour in wide deep neural networks. In *International Conference on Learning Representations*, 2018.

[17] G. K. Dziugaite and D. M. Roy. Computing nonvacuous generalization bounds for deep (stochastic) neural networks with many more parameters than training data. In *Proceedings of the 33rd Annual Conference on Uncertainty in Artificial Intelligence (UAI)*, 2017.

[18] Y. Gal and Z. Ghahramani. Dropout as a bayesian approximation. *arXiv preprint arXiv:1506.02157*, 2015.

[19] P. Germain, F. Bach, A. Lacoste, and S. Lacoste-Julien. Pac-bayesian theory meets bayesian inference. *Advances in Neural Information Processing Systems*, 29, 2016.

[20] S. Govaerts, K. Verbert, E. Duval, and A. Pardo. The student activity meter for awareness and self-reflection. In *CHI '12 Extended Abstracts on Human Factors in Computing Systems*, pages 869–884, Austin Texas USA, May 2012. ACM.

[21] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger. On calibration of modern neural networks. In *International conference on machine learning*, pages 1321–1330. PMLR, 2017.

[22] D. Guo, D. Yang, H. Zhang, J. Song, R. Zhang, R. Xu, Q. Zhu, S. Ma, P. Wang, X. Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

[23] J. Hejna, R. Rafailov, H. Sikchi, C. Finn, S. Niekum, W. B. Knox, and D. Sadigh. Contrastive preference learning: learning from human feedback without rl. *arXiv preprint arXiv:2310.13639*, 2023.

[24] J. Hu, Y. Zhang, Q. Han, D. Jiang, X. Zhang, and H.-Y. Shum. Open-reasoner-zero: An open source approach to scaling up reinforcement learning on the base model, 2025.

[25] HuggingFaceH4. Math-500 dataset. `https://huggingface.co/datasets/HuggingFaceH4/MATH-500/blob/main/README.md`, 2024. Accessed: 2025-05-05.

[26] A. A. G. Intelligence. The amazon nova family of models: Technical report and model card. 2024.

[27] Y. Jiang, V. Nagarajan, C. Baek, and J. Z. Kolter. Assessing generalization of SGD via disagreement. In *International Conference on Learning Representations*, 2022.

[28] A. Johnston, S. Amitani, and E. Edmonds. Amplifying reflective thinking in musical performance. In *Proceedings of the 5th conference on Creativity & cognition - C&C '05*, page 166, London, United Kingdom, 2005. ACM Press.

[29] S. Kadavath, T. Conerly, A. Askell, T. Henighan, D. Drain, E. Perez, N. Schiefer, Z. Hatfield-Dodds, N. DasSarma, E. Tran-Johnson, et al. Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*, 2022.

[30] A. Kirsch and Y. Gal. A note on "assessing generalization of SGD via disagreement". *Transactions on Machine Learning Research*, 2022. Expert Certification.

[31] P. Klesk. Probabilistic estimation of vapnik-chervonenkis dimension. In *ICAART (1)*, pages 262–270, 2012.

[32] T. Kojima, S. S. Gu, M. Reid, Y. Matsuo, and Y. Iwasawa. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213, 2022.

[33] B. Lakshminarayanan, A. Pritzel, and C. Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. *stat*, 1050:5, 2016.

[34] J. Lee, J. Sohl-dickstein, J. Pennington, R. Novak, S. Schoenholz, and Y. Bahri. Deep neural networks as gaussian processes. In *International Conference on Learning Representations*, 2018.

[35] Leijen, I. Lam, L. Wildschut, P. Robert-Jan Simons, and W. Admiraal. Streaming video to enhance students' reflection in dance education. *Computers & Education*, 52(1):169–176, Jan. 2009.

[36] B. Lester, R. Al-Rfou, and N. Constant. The power of scale for parameter-efficient prompt tuning. *arXiv preprint arXiv:2104.08691*, 2021.

[37] X. L. Li and P. Liang. Prefix-tuning: Optimizing continuous prompts for generation. In C. Zong, F. Xia, W. Li, and R. Navigli, editors, *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4582–4597, Online, Aug. 2021. Association for Computational Linguistics.

[38] Z. Li, G. Jiang, H. Xie, L. Song, D. Lian, and Y. Wei. Understanding and patching compositional reasoning in llms. *arXiv preprint arXiv:2402.14328*, 2024.

[39] S. Lin, J. Hilton, and O. Evans. TruthfulQA: Measuring how models mimic human falsehoods. In S. Muresan, P. Nakov, and A. Villavicencio, editors, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3214–3252, Dublin, Ireland, May 2022. Association for Computational Linguistics.

[40] A. Madaan, N. Tandon, P. Gupta, S. Hallinan, L. Gao, S. Wiegreffe, U. Alon, N. Dziri, S. Prabhumoye, Y. Yang, et al. Self-refine: Iterative refinement with self-feedback. *Advances in Neural Information Processing Systems*, 36:46534–46594, 2023.

[41] D. McAllester. Simplified pac-bayesian margin bounds. In *Learning Theory and Kernel Machines: 16th Annual Conference on Learning Theory and 7th Kernel Workshop, COLT/Kernel 2003, Washington, DC, USA, August 24-27, 2003. Proceedings*, pages 203–215. Springer, 2003.

[42] S. Mendelson and R. Vershynin. Entropy and the combinatorial dimension. *Inventiones mathematicae*, 152(1):37–55, 2003.

[43] Meta AI. Llama-3.1-8b-instruct. `https://huggingface.co/meta-llama/Llama-3.1-8B-Instruct`, 2024. Accessed: October 19, 2025.

[44] Mistral AI. Ministral-8b-instruct-2410. https://huggingface.co/mistralai/Ministral-8B-Instruct-2410, 2024. Accessed: October 19, 2025.

[45] P. Mondorf and B. Plank. Beyond accuracy: Evaluating the reasoning behavior of large language models–a survey. *arXiv preprint arXiv:2404.01869*, 2024.

[46] N. Muennighoff, Z. Yang, W. Shi, X. L. Li, L. Fei-Fei, H. Hajishirzi, L. Zettlemoyer, P. Liang, E. Candès, and T. Hashimoto. s1: Simple test-time scaling, 2025.

[47] E. J. Nelson and N. G. Freier. Push-me, pull-me: describing and designing technologies for varying degrees of reflection and invention. In *Proceedings of the 7th international conference on Interaction design and children*, pages 129–132, Chicago Illinois, June 2008. ACM.

[48] W. Nie, Z. Yu, L. Mao, A. B. Patel, Y. Zhu, and A. Anandkumar. Bongard-logo: A new benchmark for human-level concept learning and reasoning. *Advances in Neural Information Processing Systems*, 33:16468–16480, 2020.

[49] Open-Thoughts. Openthinker2-7b. https://huggingface.co/open-thoughts/OpenThinker2-7B, 2023. Accessed: October 19, 2025.

[50] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.

[51] Y. Ovadia, E. Fertig, J. Ren, Z. Nado, D. Sculley, S. Nowozin, J. Dillon, B. Lakshminarayanan, and J. Snoek. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. *Advances in neural information processing systems*, 32, 2019.

[52] A. S. Palinscar and A. L. Brown. Reciprocal Teaching of Comprehension-Fostering and Comprehension-Monitoring Activities. *Cognition and Instruction*, 1(2):117–175, Mar. 1984.

[53] L. Pan, M. Saxon, W. Xu, D. Nathani, X. Wang, and W. Y. Wang. Automatically correcting large language models: Surveying the landscape of diverse self-correction strategies. *arXiv preprint arXiv:2308.03188*, 2023.

[54] J. Platt et al. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in large margin classifiers*, 10(3):61–74, 1999.

[55] R. Rafailov, A. Sharma, E. Mitchell, S. Ermon, C. D. Manning, and C. Finn. Direct preference optimization: Your language model is secretly a reward model. *arXiv preprint arXiv:2305.18290*, 2023.

[56] A. Reznitskaya, M. Glina, B. Carolan, O. Michaud, J. Rogers, and L. Sequeira. Examining transfer effects from dialogic discussions to new tasks and contexts. *Contemporary Educational Psychology*, 37(4):288–306, Oct. 2012.

[57] E. Shabat, L. Cohen, and Y. Mansour. Sample complexity of uniform convergence for multicalibration. *Advances in Neural Information Processing Systems*, 33:13331–13340, 2020.

[58] S. Shalev-Shwartz and S. Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.

[59] Z. Shao, P. Wang, Q. Zhu, R. Xu, J. Song, X. Bi, H. Zhang, M. Zhang, Y. Li, Y. Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

[60] Y. Shi. Enhancing evidence-based argumentation in a Mainland China middle school. *Contemporary Educational Psychology*, 59:101809, Oct. 2019.

[61] N. Shinn, F. Cassano, A. Gopinath, K. Narasimhan, and S. Yao. Reflexion: Language agents with verbal reinforcement learning. *Advances in Neural Information Processing Systems*, 36:8634–8652, 2023.

[62] G. Swamy, S. Choudhury, W. Sun, Z. S. Wu, and J. A. Bagnell. All roads lead to likelihood: The value of reinforcement learning in fine-tuning. *arXiv preprint arXiv:2503.01067*, 2025.

[63] A. Talmor, J. Herzig, N. Lourie, and J. Berant. CommonsenseQA: A question answering challenge targeting commonsense knowledge. In J. Burstein, C. Doran, and T. Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4149–4158, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[64] M. Telgarsky. Neural networks and rational functions. In *International Conference on Machine Learning*, pages 3387–3393. PMLR, 2017.

[65] J. Toy, J. MacAdam, and P. Tabor. Metacognition is all you need? using introspection in generative agents to improve goal-directed behavior. *arXiv preprint arXiv:2401.10910*, 2024.

[66] T. Tseng and C. Bryant. Design, reflect, explore: encouraging children's reflections with mechanix. In *CHI '13 Extended Abstracts on Human Factors in Computing Systems*, pages 619–624, Paris France, Apr. 2013. ACM.

[67] V. N. Vapnik and A. Y. Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability Its Applications*, 16(2):264–280, 1971.

[68] Y. Wang and Y. Zhao. Metacognitive prompting improves understanding in large language models. *arXiv preprint arXiv:2308.05342*, 2023.

[69] A. M. Webb, R. Linder, A. Kerne, N. Lupfer, Y. Qu, B. Poffenberger, and C. Revia. Promoting reflection and interpretation in education: curating rich bookmarks as information composition. In *Proceedings of the 9th ACM Conference on Creativity & Cognition*, pages 53–62, Sydney Australia, June 2013. ACM.

[70] J. Wei, X. Wang, D. Schuurmans, M. Bosma, F. Xia, E. Chi, Q. V. Le, D. Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

[71] S. M. Xie, A. Raghunathan, P. Liang, and T. Ma. An explanation of in-context learning as implicit bayesian inference. In *International Conference on Learning Representations*, 2022.

[72] A. Yang, B. Yang, B. Zhang, B. Hui, B. Zheng, B. Yu, C. Li, D. Liu, F. Huang, H. Wei, et al. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

[73] A. Yang, B. Zhang, B. Hui, B. Gao, B. Yu, C. Li, D. Liu, J. Tu, J. Zhou, J. Lin, K. Lu, M. Xue, R. Lin, T. Liu, X. Ren, and Z. Zhang. Qwen2.5-math technical report: Toward mathematical expert model via self-improvement, 2024.

[74] Q. Yu, Z. Zhang, R. Zhu, Y. Yuan, X. Zuo, Y. Yue, T. Fan, G. Liu, L. Liu, X. Liu, et al. Dapo: An open-source llm reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*, 2025.

[75] W. Yuan, R. Y. Pang, K. Cho, X. Li, S. Sukhbaatar, J. Xu, and J. Weston. Self-rewarding language models. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024.

[76] C. Zhang, F. Gao, B. Jia, Y. Zhu, and S.-C. Zhu. Raven: A dataset for relational and analogical visual reasoning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5317–5327, 2019.

[77] K. Zhou, B. Zhang, J. Wang, Z. Chen, X. Zhao, J. Sha, Z. Sheng, S. Wang, and J.-R. Wen. Jiuzhang3.0: Efficiently improving mathematical reasoning by training small data synthesis models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

Table 2: Notation used throughout the paper.

Symbol	Description
$\mathcal{X}$	Input space
$\mathcal{F}$	Class of probabilistic predictors $f : \mathcal{X} \to \Delta(\{0, 1\})$
$\Delta(\{0, 1\})$	Set of probability distributions over binary output space $\{0, 1\}$
$\gamma \in (0, 1]$	Confidence threshold parameter
$\tau \in [0, 1)$	Calibration error tolerance parameter
$\text{VC}(\mathcal{F})$	Classical Vapnik-Chervonenkis dimension of function class $\mathcal{F}$
$\text{PVC}_\gamma(\mathcal{F})$	Probabilistic VC dimension of $\mathcal{F}$ at confidence threshold γ
$\text{C-PVC}^\tau_\gamma(\mathcal{F})$	Calibration-aware PVC dimension with parameters γ and τ
$\hat{p} : \mathcal{X} \to [0, 1]$	Confidence scoring function associated with predictor f
$\text{fat}_\alpha(\mathcal{F})$	Fat-shattering dimension of function class $\mathcal{F}$ at scale α
$\alpha = \gamma - \frac{1}{2}$	Margin parameter relating PVC to fat-shattering dimension
$\mathbb{P}(f(x) = y)$	Probability that predictor f outputs label y on input x
$\mathbb{E}[\cdot]$	Expectation operator over the specified probability space
$\mathbf{1}\{\cdot\}$	Indicator function: 1 if condition is true, 0 otherwise
$\text{SEA}(f, C)$	Self-Evaluation Accuracy of model f on category C
$\mathsf{Judge}(q)$	Index of objectively superior solution for question q
$\mathsf{Select}_f(q)$	Model f's selected solution index for question q
$\text{CalibError}(f, C)$	Calibration error of model f on category C
$M\text{-VUS}(\mathcal{F})$	Volume Under Surface for metric $M \in \{\text{PVC}, \text{C-PVC}, \text{SC}\}$
$\text{SC}(\gamma, \tau, \mathcal{F})$	Sample complexity bound for function class $\mathcal{F}$ with parameters γ, τ
ϵ	Accuracy parameter in generalization bounds
δ	Confidence parameter in generalization bounds
C	Universal constant in sample complexity bounds
Π	Set of prompts in prompt-based function class definition
W	Parameter vector of a language model
$\mathcal{P}$	Probability distribution over parameter space
$\mathcal{D}$	Data distribution
$\mathcal{D}_C$	Distribution of questions in category C

A Limitations

Despite its theoretical and empirical contributions, this study has several limitations. The proposed PVC and C-PVC frameworks rely on category-level aggregation rather than individual question evaluation, potentially obscuring fine-grained self-assessment patterns. The theoretical guarantees established for PVC and C-PVC dimensions assume certain mathematical conditions that may not strictly hold in practical LLM implementations with complex architectures like Transformers. Furthermore, while the study spans multiple domains (mathematics, factual knowledge, and commonsense reasoning), it may not generalize to other specialized reasoning tasks or to models significantly larger or smaller than the 7-8B parameter range examined. Finally, the analysis of training methodology effects is observational rather than causal, limiting definitive conclusions about which specific training techniques directly improve self-evaluation capabilities.

B LLM Usage Disclosure

In the interest of transparency, we disclose that large language models (LLMs) were utilized during the preparation of this manuscript. LLMs assisted in polishing the writing by suggesting alternative phrasings and improving sentence structure in certain sections, while all content was carefully reviewed and edited by the authors to ensure accuracy. Additionally, LLMs were employed to aid in literature retrieval and discovery, helping to identify potentially relevant papers that were subsequently accessed and critically evaluated by the authors. During early research stages, LLMs also contributed to ideation by exploring potential extensions of VC dimension theory and suggesting experimental designs, though all theoretical formulations and analyses were developed and validated through rigorous scientific investigation by the authors. Throughout this process, LLMs served solely as assistive tools to enhance efficiency and clarity in our research communication, with complete scientific oversight maintained by the authors at all stages of the work.

C Proof of Proposition 1

Proof. We begin by proving the first statement: for any $\gamma_1, \gamma_2 \in (0, 1]$ such that $\gamma_1 > \gamma_2$, we have

$$\mathrm{VC}(\mathcal{F}) = \mathrm{PVC}_1(\mathcal{F}) \leq \mathrm{PVC}_{\gamma_1}(\mathcal{F}) \leq \mathrm{PVC}_{\gamma_2}(\mathcal{F}).$$

By definition, $\mathrm{PVC}_\gamma(\mathcal{F})$ is the largest integer d such that for every labeling $(y_1^*, \ldots, y_d^*)$, there exists $f \in \mathcal{F}$ satisfying

$$\mathbb{P}(f(x_i) = y_i^*) \geq \gamma \quad \text{for all } i = 1, \ldots, d.$$

When $\gamma = 1$, this condition implies that f deterministically assigns the correct label to each input. This is exactly the classical VC shattering criterion, where each binary labeling must be realized exactly. Thus, we have

$$\mathrm{PVC}_1(\mathcal{F}) = \mathrm{VC}(\mathcal{F}).$$

Next, we consider the monotonicity of PVC_γ with respect to γ. Suppose $\gamma_1 > \gamma_2$. Any set that is γ_1-shattered must also be γ_2-shattered, since the confidence requirement γ_2 is weaker. Therefore,

$$\mathrm{PVC}_{\gamma_1}(\mathcal{F}) \leq \mathrm{PVC}_{\gamma_2}(\mathcal{F}),$$

and together with the previous identity, we obtain the desired chain of inequalities.

We now turn to the second statement: for a fixed $\gamma \in (0, 1]$ and any $\tau_1, \tau_2 \in [0, 1)$ where $\tau_1 > \tau_2$:

$$\mathrm{C} - \mathrm{PVC}_\gamma^{\tau_2}(\mathcal{F}) \leq \mathrm{C} - \mathrm{PVC}_\gamma^{\tau_1}(\mathcal{F}) \leq \mathrm{PVC}_\gamma(\mathcal{F}).$$

First, we establish that $\mathrm{C} - \mathrm{PVC}_\gamma^{\tau_1}(\mathcal{F}) \leq \mathrm{PVC}_\gamma(\mathcal{F})$. This follows directly from the definitions. The calibration-aware PVC dimension imposes two simultaneous conditions: for each $i = 1, \ldots, d$,

$$\mathbb{P}(f(x_i) = y_i^*) \geq \gamma \quad \text{and} \quad |\hat{p}_i - \mathbb{E}\mathbf{1}f(x_i) = y_i| \leq \tau_1,$$

where $\hat{p}_i$ is the confidence assigned by f to the label y_i^*. The first condition is identical to that of PVC_γ, but the second introduces an additional constraint that requires approximate calibration. As a result, any set that is $\mathrm{C\text{-}PVC}_\gamma^{\tau_1}$-shattered must also be PVC_γ-shattered, but not necessarily vice versa. Therefore, the inequality holds.

Next, we show that $\text{C-PVC}_\gamma^{\tau_2}(\mathcal{F}) \leq \text{C-PVC}_\gamma^{\tau_1}(\mathcal{F})$ when $\tau_1 > \tau_2$. By definition, for a set to be $\text{C-PVC}_\gamma^{\tau}$-shattered, for every labeling, there must exist a predictor that achieves both the required accuracy γ and calibration error at most τ. Since $\tau_1 > \tau_2$, the calibration requirement for τ_2 is stricter than for τ_1. Specifically, if a predictor satisfies $|\hat{p}_i - \mathbb{E}\mathbf{1}f(x_i) = y_i^*| \leq \tau_2$, then it automatically satisfies $|\hat{p}_i - \mathbb{E}\mathbf{1}f(x_i) = y_i^*| \leq \tau_1$ as well. Therefore, any set that is $\text{C-PVC}_\gamma^{\tau_2}$-shattered must also be $\text{C-PVC}_\gamma^{\tau_1}$-shattered, which implies $\text{C-PVC}_\gamma^{\tau_2}(\mathcal{F}) \leq \text{C-PVC}_\gamma^{\tau_1}(\mathcal{F})$. Combining these two inequalities completes the proof of the second statement. □

D Sample Complexity Bound via Probabilistic VC Dimension

Theorem 2 (Sample Complexity Bound via Probabilistic VC Dimension)**.** *Let $\mathcal{F}$ be a class of probabilistic predictors over $\mathcal{X} \times \mathcal{Y}$ with probabilistic VC dimension $\mathrm{PVC}\gamma(\mathcal{F}) = d_\gamma < \infty$ for some confidence threshold $\gamma \in (0, 1]$. Then there exist universal constants $C, c > 0$ such that the following holds. For any $\epsilon, \delta \in (0, 1)$, if the sample size satisfies*

$$m \geq \frac{C}{\epsilon^2}\left(d_\gamma + \log\frac{1}{\delta}\right),$$

then

$$\mathbb{P}\left(\sup_{f \in \mathcal{F}}\left|\frac{1}{m}\sum_{i=1}^{m}\mathbf{1}\{f(x_i) \neq y_i\} - \mathbb{P}_{(x,y)\sim\mathcal{X}\times\mathcal{Y}}[f(x) \neq y]\right| \leq \epsilon\right) \geq 1 - \delta.$$

Proof. Let each $f \in \mathcal{F}$ be a probabilistic predictor defined as a pair of measurable functions $(f_0, f_1) : \mathcal{X} \to [0, 1]$ such that $f_0(x) + f_1(x) = 1$ for all $x \in \mathcal{X}$. The prediction is interpreted as assigning label 1 when $f_1(x) > f_0(x)$, and label 0 otherwise. Thus, f_1 itself serves as a real-valued function mapping inputs to the confidence assigned to label 1.

Define the function class $\mathcal{G} := \{f_1 : f = (f_0, f_1) \in \mathcal{F}\} \subseteq [0, 1]^{\mathcal{X}}$, and define the margin parameter $\alpha := \gamma - \frac{1}{2} > 0$. We now show that if $\mathrm{PVC}_\gamma(\mathcal{F}) = d_\gamma$, then $\mathcal{G}$ α-fat-shatters a set of size at least d_γ with fixed thresholds $s_i = 1/2$.

Lemma 1 (PVC Implies Fat-Shattering with Fixed Threshold)**.** *Let $\mathcal{F}$, $\mathcal{G}$, and α be as above. Then*

$$\mathrm{PVC}_\gamma(\mathcal{F}) \le \mathrm{fat}_\alpha(\mathcal{G}),$$

where the fat-shattering condition is evaluated with fixed thresholds $s_i = 1/2$.

Proof. Suppose $\{x_1, \dots, x_d\}$ is γ-shattered in the PVC sense. Then for every labeling $(y_1^*, \dots, y_d^*) \in \{0,1\}^d$, there exists a probabilistic predictor $f = (f_0, f_1) \in \mathcal{F}$ such that for all $i = 1, \dots, d$,

$$f_{y_i^*}(x_i) := \mathbb{P}(f(x_i) = y_i^*) \ge \gamma.$$

Consider $f_1 \in \mathcal{G}$ as the function assigning confidence to label 1. For each i, if $y_i^* = 1$, then $f_1(x_i) = f_{y_i^*}(x_i) \ge \gamma = 1/2 + \alpha = s_i + \alpha$. If $y_i^* = 0$, then $f_0(x_i) = f_{y_i^*}(x_i) \ge \gamma$ implies $f_1(x_i) = 1 - f_0(x_i) \le 1 - \gamma = 1/2 - \alpha = s_i - \alpha$. Therefore, the set $\{x_1, \dots, x_d\}$ is α-fat-shattered by $\mathcal{G}$ at fixed thresholds $s_i = 1/2$. □

We now apply the standard uniform convergence result for fat-shattering dimension:

Lemma 2 (Uniform Convergence via Fat-Shattering [15])**.** *Let $\mathcal{G} \subseteq [0,1]^{\mathcal{X}}$ and suppose $\mathrm{fat}_\alpha(\mathcal{G}) = d_\alpha < \infty$ for some $\alpha > 0$. Then there exist universal constants $C, c > 0$ such that for any $\epsilon, \delta \in (0,1)$, if*

$$m \ge \frac{C}{\epsilon^2}\left(d_\alpha + \log\frac{1}{\delta}\right),$$

then with probability at least $1 - \delta$ over i.i.d. samples $x_1, \dots, x_m \sim \mathcal{D}$,

$$\sup_{g \in \mathcal{G}} \left| \frac{1}{m}\sum_{i=1}^m g(x_i) - \mathbb{E}[g(x)] \right| \le \epsilon.$$

By Lemma 1, we have $\mathrm{PVC}_\gamma(\mathcal{F}) \le \mathrm{fat}_\alpha(\mathcal{G})$. Plugging into Lemma 2 yields the claimed result.

We remark that the inequality $\mathrm{PVC}_\gamma(\mathcal{F}) \le \mathrm{fat}_{\gamma - 1/2}(\mathcal{G})$ is a direct consequence of the definition, where $\mathcal{G} = \{f_1 : (f_0, f_1) \in \mathcal{F}\}$. In the binary setting, this upper bound is often tight, since both quantities reflect the ability to encode all binary labelings with high confidence or sufficient separation from the decision boundary. □

E Proof of Theorem 1

Lemma 3 (Fat-Shattering from C-PVC Shattering)**.** *Let $\mathcal{F}$ be as in Theorem 1. Then for any d-tuple $\{x_1, \dots, x_d\}$ that is $\mathrm{C\text{-}PVC}_\gamma^\tau$-shattered, there exists a function $\hat{p} \in \hat{\mathcal{P}} := \{\hat{p} : (f, \hat{p}) \in \mathcal{F}\}$ such that $\{x_1, \dots, x_d\}$ is α-fat-shattered by $\hat{p}$ at thresholds $s_i = 1/2$ with margin $\alpha = \gamma - \tau - 1/2 > 0$.*

Proof. Fix any labeling $(y_1^*, \dots, y_d^*) \in \{0,1\}^d$ that is $\mathrm{C\text{-}PVC}_\gamma^\tau$-shattered. Then by definition, there exists a function pair $(f, \hat{p})$ such that for all $i = 1, \dots, d$,

$$\mathbb{P}(f(x_i) = y_i^*) \ge \gamma, \quad \text{and} \quad |\hat{p}(x_i) - \mathbb{E}[\mathbf{1}\{f(x_i) = y_i^*\}]| \le \tau.$$

Let $s_i := 1/2$ for all i. If $y_i^* = 1$, then $\mathbb{E}[\mathbf{1}\{f(x_i) = 1\}] \ge \gamma$, hence $\hat{p}(x_i) \ge \gamma - \tau \ge s_i + \alpha$. Similarly, if $y_i^* = 0$, then $\mathbb{E}[\mathbf{1}\{f(x_i) = 0\}] \ge \gamma$, and thus $\hat{p}(x_i) \le 1 - \gamma + \tau \le s_i - \alpha$. Therefore, $\hat{p}$ α-shatters the set $\{x_1, \dots, x_d\}$ at thresholds $s_i = 1/2$, establishing the lemma. □

Proof of Theorem 1. Let $\mathcal{H}_1 := \{x \mapsto \mathbb{E}\mathbf{1}\{f(x) = y\} : f \in \mathcal{F}\}$ and $\mathcal{H}_2 := \{x \mapsto \hat{p}(x) - \mathbb{E}\mathbf{1}\{f(x) = y\} : f \in \mathcal{F}\}$. Each function in $\mathcal{H}_1$ and $\mathcal{H}_2$ is bounded in $[0,1]$.

From Lemma 3, we have that $\mathrm{C\text{-}PVC}_\gamma^\tau(\mathcal{F}) \le \mathrm{fat}_\alpha(\hat{\mathcal{P}})$, and hence both $\mathcal{H}_1$ and $\mathcal{H}_2$ have fat-shattering dimension at most $d_{\gamma,\tau}$ at scale α.

Applying the uniform convergence theorem for fat-shattering classes [8, 15] we obtain the desired bounds for both generalization error and calibration deviation. The additional τ in the second inequality is absorbed into the allowable population-level calibration discrepancy. □

F Implications for Self-Evaluating Models

Our theoretical results in Seciton 4 have significant implications for self-evaluating language models. The ability to obtain VC-style bounds is a key strength of our framework, as it enables the application of classical tools from learning theory to the study of confidence-aware predictors, including large language models.

The distinction between PVC and C-PVC becomes particularly important in practical applications. A model with high PVC but poor calibration might confidently discriminate between correct and incorrect solutions, yet provide unreliable confidence estimates. For instance, a model might consistently assign 90% confidence to predictions that are only correct 70% of the time. Conversely, a model with high C-PVC demonstrates both strong discriminative capacity and reliable self-assessment—essential qualities for trustworthy autonomous reasoning [27, 30, 57].

Our framework thus provides theoretical grounding for evaluating and improving self-assessment in language models. The trade-off between expressivity (captured by PVC) and calibration quality (reflected in the gap between PVC and C-PVC) emerges naturally from our theoretical analysis and, as we will see in the experimental section, manifests consistently across different model architectures and training methodologies.

G Expected Calibration Error and Calibration Metrics

In Section 3.3, we introduced the category-level calibration error metric:

$$\text{CalibError}(f, C) = |\mathbb{E}_{q\sim C}[\hat{p}(q)] - \text{SEA}(f, C)| \tag{5}$$

where $\hat{p}(q)$ is the model's reported confidence for question q and $\text{SEA}(f, C)$ is the Self-Evaluation Accuracy on category C. This metric measures the discrepancy between average predicted confidence and actual accuracy at the category level, and is used to determine whether a category satisfies the calibration requirement in our C-PVC framework.

Building on this category-level calibration assessment, we define the Expected Calibration Error (ECE) as the weighted average of calibration errors across all categories:

$$\text{ECE} = \sum_{C\in\mathcal{C}} \frac{|C|}{N} \text{CalibError}(f, C) = \sum_{C\in\mathcal{C}} \frac{|C|}{N} |\mathbb{E}_{q\sim C}[\hat{p}(q)] - \text{SEA}(f, C)| \tag{6}$$

Additionally, we define the Actual Error (AE) as the overall error rate in self-evaluation tasks:

$$\text{AE} = \sum_{C\in\mathcal{C}} \frac{|C|}{N} (1 - \text{SEA}(f, C)) = 1 - \sum_{C\in\mathcal{C}} \frac{|C|}{N} \text{SEA}(f, C) \tag{7}$$

where:

- $\mathcal{C}$ is the set of all categories in our evaluation
- $|C|$ is the number of questions in category C
- N is the total number of questions across all categories
- $\mathbb{E}_{q\sim C}[\hat{p}(q)]$ is the average confidence reported by the model on questions in category C
- $\text{SEA}(f, C)$ is the Self-Evaluation Accuracy on category C

The AE metric represents the weighted average of error rates across all categories, providing a single summary statistic for the model's overall self-evaluation performance. Unlike ECE, which measures calibration quality, AE directly measures the frequency of incorrect self-evaluations regardless of confidence levels.

This category-based ECE differs from the traditional bin-based ECE used in much of the calibration literature. Instead of grouping predictions by confidence level, we group them by problem category, which aligns with our theoretical framework's focus on category-level shattering and calibration. This approach allows us to directly relate ECE to our C-PVC dimension, as both are based on the same underlying CalibError metric.

G.1 Relationships Between Evaluation Metrics

Our experimental results in Table 1 show relationships between the different evaluation metrics we use:

PVC-VUS Gap and ECE. There is a strong correlation between the PVC-VUS Gap (the difference between PVC-VUS and C-PVC-VUS) and ECE across all models. Both metrics capture aspects of calibration quality:

- **PVC-VUS Gap** measures how much expressive power is lost when imposing the calibration constraint (CalibError $\leq \tau$) across different parameter settings.

- **ECE** measures the weighted average of calibration errors across all categories.

As shown in Table 1, models with lower ECE consistently demonstrate smaller PVC-VUS Gaps. For example, JiuZhang3.0-7B has both the lowest ECE (0.209) and the smallest PVC-VUS Gap (0.95), while models with higher ECE values like Ministral-8B-Instruct (ECE: 0.402) also show larger PVC-VUS Gaps (1.88). This correlation is not coincidental—both metrics fundamentally measure how well a model's confidence scores align with its actual performance, though they do so through different mathematical formulations.

SC-VUS and AE. Another notable relationship exists between Sample Complexity (SC-VUS) and Actual Error (AE):

- **SC-VUS** represents the theoretical sample complexity integrated across parameter settings, indicating how many samples are needed to ensure reliable generalization.
- **AE** measures the actual error rate of the model on the evaluation dataset, representing the proportion of incorrect self-evaluations across all categories.

The relationship between these metrics is more complex and multifaceted. While high SC-VUS often correlates with lower AE (as seen with s1.1-7B, which has the highest SC-VUS at 727.3 and the lowest AE at 0.378), this pattern shows variations across different models. This relationship reflects the fundamental trade-off in statistical learning: models with higher expressive power (higher PVC) can achieve lower error rates but require more samples to generalize reliably.

The correlation between SC-VUS and AE can be understood through the lens of model capacity utilization. Models with higher SC-VUS values have greater theoretical capacity (as measured by PVC dimensions across parameter settings), which allows them to achieve lower error rates when this capacity is effectively utilized. However, this relationship is modulated by calibration quality—models with poor calibration may have high SC-VUS but fail to achieve correspondingly low error rates due to overconfident predictions in incorrect answers.

G.2 Relationship Between ECE and C-PVC

While both ECE and our C-PVC framework assess calibration quality, they do so in complementary ways:

- **CalibError** is used at the individual category level to determine whether that category satisfies the calibration requirement (CalibError $\leq \tau$) for computing C-PVC dimension.
- **ECE** aggregates these calibration errors across all categories, providing a single value that summarizes overall calibration quality.

The key difference in how these metrics are used in our framework is:

- **C-PVC** counts the number of categories that simultaneously satisfy both the confidence threshold ($\mathrm{SEA}(f, C) \geq \gamma$) and calibration requirement (CalibError $\leq \tau$).
- **ECE** provides a weighted average of calibration errors across all categories, regardless of whether they meet the confidence threshold.

In well-calibrated models, we expect both low ECE and high C-PVC values (relative to PVC). Indeed, as shown in Table 1, JiuZhang3.0-7B achieves the lowest overall ECE (0.209) and also demonstrates the smallest PVC-VUS Gap (0.95), indicating that both its global calibration (measured by ECE) and category-level calibration (captured by C-PVC through CalibError) are superior to other models.

H Interpretation of Sample Complexity

In this appendix, we provide a detailed interpretation of the sample complexity bounds derived in our PVC framework and discuss their practical implications for evaluating self-assessment capabilities in language models.

H.1 Sample Complexity Bound via Probabilistic VC Dimension

As established in Theorem 2, for a function class $\mathcal{F}$ with probabilistic VC dimension $\mathrm{PVC}_\gamma(\mathcal{F}) = d_\gamma < \infty$, the sample complexity required to achieve generalization error ϵ with confidence $1 - \delta$ is:

$$m \geq \frac{C}{\epsilon^2}\left(d_\gamma + \log\frac{1}{\delta}\right) \tag{8}$$

This bound has several important interpretations:

Relationship to Model Expressivity. The sample complexity scales linearly with the PVC dimension d_γ, which measures the model's capacity to make confident predictions across different problem categories. Models with higher PVC dimensions require more samples to ensure reliable generalization. This reflects the fundamental trade-off between expressivity and sample efficiency: more expressive models (higher d_γ) can represent a wider range of confident prediction patterns, but require more data to avoid overfitting.

Confidence Threshold Dependence. The sample complexity depends on the confidence threshold γ through d_γ. As shown in Proposition 1, lower values of γ lead to higher PVC dimensions, which in turn increase sample complexity. This captures the intuition that making high-confidence predictions (high γ) typically requires less data for generalization than making lower-confidence predictions that cover more complex patterns.

Error-Confidence Trade-off. The bound shows an inverse quadratic relationship between sample complexity and the desired generalization error ϵ. Halving the acceptable error requires approximately four times as many samples. This highlights the diminishing returns in error reduction as sample size increases.

H.2 Sample Complexity for Calibration-aware PVC

For the calibration-aware setting in Theorem 1, the sample complexity bound becomes:

$$m \geq \frac{C}{\epsilon^2}\left(d_{\gamma,\tau} + \log\frac{1}{\delta}\right) \tag{9}$$

where $d_{\gamma,\tau} = \text{C-PVC}^{\tau}_{\gamma}(\mathcal{F})$. This has additional interpretations:

Calibration-Expressivity Balance. The C-PVC dimension $d_{\gamma,\tau}$ captures both the model's expressive power and its calibration quality. As shown in our experimental results, models with similar PVC dimensions may have very different C-PVC dimensions due to varying calibration quality. Well-calibrated models (smaller gap between PVC and C-PVC) generally require fewer samples to achieve reliable generalization of both prediction accuracy and confidence estimation.

Calibration Tolerance Effect. The calibration tolerance parameter τ influences sample complexity through $d_{\gamma,\tau}$. As τ increases (more relaxed calibration requirements), $d_{\gamma,\tau}$ increases as well, potentially increasing sample complexity. However, this is balanced by the fact that larger τ values make the calibration constraint easier to satisfy, which can reduce the effective complexity of the function class.

H.3 Practical Implications for LLM Evaluation

Our sample complexity bounds have several practical implications for evaluating and training self-reflective language models:

Evaluation Efficiency. The SC-VUS metric reported in Table 1 provides a practical measure of how efficiently a model can be evaluated. Higher SC-VUS values indicate that more samples are needed to achieve reliable generalization. Models with better calibration (smaller gap between PVC and C-PVC) like JiuZhang3.0-7B often achieve better generalization efficiency relative to their expressivity, despite having moderately high SC-VUS values.

Training Data Requirements. These bounds inform training data requirements for developing self-reflective capabilities. The SC-VUS values in Table 1 indicate that models with higher values (like s1.1-7B at 727.3) require more training samples to achieve reliable generalization, despite their strong performance on evaluation metrics. This suggests that highly expressive models may need substantially more training data focused on calibration to achieve reliable self-assessment.

Parameter Selection Guidance. The relationship between confidence threshold γ, calibration tolerance τ, and sample complexity provides guidance for selecting appropriate parameter values in practical applications. For high-stakes applications requiring well-calibrated confidence, choosing a model with smaller gap between PVC and C-PVC dimensions may be more important than selecting one with the highest raw PVC dimension.

Cross-domain Generalization. Our cross-domain analysis showed that calibration quality tends to transfer better across domains than raw discriminative power. As seen in Table 1, JiuZhang3.0-7B maintains consistently small PVC-VUS Gaps across all three datasets (0.75/1.33/0.77), while models like OpenThinker2-7B show much greater variability (1.51/2.10/1.99). The sample complexity bounds help explain this phenomenon: well-calibrated models have more efficient generalization properties, requiring fewer domain-specific examples to adapt to new tasks.

In summary, the sample complexity bounds derived from our PVC framework provide not just theoretical guarantees, but also practical insights into model selection, evaluation efficiency, and training strategies for developing reliable self-reflective language models.

I Category-level measurement for PVC and C–PVC

In our analysis, we compute $\mathrm{PVC}_\gamma(\mathcal{F})$ and $\text{C-PVC}_\gamma^\tau(\mathcal{F})$ at the level of problem *categories*, such as *Algebra* or *Geometry*, rather than at the level of individual questions. This design choice is motivated by both theoretical and practical considerations and remains consistent with the assumptions underlying the γ-shattering framework.

Statistical tractability. Estimating VC-style capacity at the question level quickly becomes intractable. Given a candidate shattering set of n elements, the number of binary labelings grows exponentially as 2^n. Aggregating problems into categories reduces the effective hypothesis space while preserving sufficient semantic diversity. In practice, treating each category as a single unit enables efficient computation and still provides meaningful discrimination between models with different generalization behaviors.

Alignment with model organization. Large language models tend to acquire domain-level reasoning skills before mastering fine-grained problem templates. Measuring shattering capacity over category partitions aligns with this inductive structure. Each category can be viewed as a higher-order instance, requiring the model to succeed across a distribution of related problems rather than a single input.

Stability in calibration. Individual prediction confidences are often noisy and variable. Averaging across structurally related problems yields a smoother estimate of empirical correctness, leading to more stable calibration evaluations. This is particularly relevant for computing $\text{C-PVC}_\gamma^\tau(\mathcal{F})$, which involves enforcing the constraint

$$|\hat{p}(x_i) - \mathbb{E}\left[\mathbf{1}\{f(x_i) = y_i^*\}\right]| \leq \tau.$$

Evaluating this quantity at the category level mitigates the effects of variance in single-question estimates.

Theoretical consistency with γ-shattering. Importantly, the category-level formulation remains consistent with the formal definition of γ-shattering. For each element in the shattering set—now treated as a category—the model must satisfy:

$$\mathbb{P}(f(x_i) = y_i^*) \geq \gamma \quad \text{for all } x_i \in \text{category } i.$$

This requirement is more stringent than evaluating the condition on individual inputs, as it demands uniform high-confidence correctness over an entire domain. As such, the category-level perspective yields a conservative but valid estimate of capacity.

J On the finite capacity guarantees for LLMs

According to Anthony and Bartlett [3, See Theorem 14.18 and Theorem 14.19], probabilistic VC capacity remains finite for model classes whose prediction functions are realized by bounded-weight, Lipschitz neural networks. In particular, the finiteness of both PVC_γ and C-PVC_γ^τ follows from Lemma 1 the existence of a finite fat-shattering dimension:

$$\text{C-PVC}_\gamma^\tau(\mathcal{F}) \leq \mathrm{PVC}_\gamma(\mathcal{F}) \leq \mathrm{fat}_\alpha(\mathcal{G}) < +\infty,$$

where $\alpha = \gamma - \frac{1}{2}$ and $\mathcal{G}$ is the associated class of real-valued confidence functions. This implication is powerful in its generality but depends critically on structural conditions—bounded weight norms, controlled depth, and uniformly Lipschitz activations—that are only partially satisfied in practice.

The gap between theoretical assumptions and practical architectures. While the bounded-operator and Lipschitz assumptions hold for simplified multi-layer perceptrons (MLPs), their applicability to realistic large language models (LLMs), such as those built on the Transformer architecture, remains unclear. Modern LLMs incorporate residual pathways, softmax-based attention, layer normalization, and positional encodings, each of which may violate or obscure the assumptions required for a clean fat-shattering analysis.

Although recent efforts attempt to bound the Lipschitz constants of certain attention modules and residual networks, no complete result exists that establishes fat-shattering finiteness for the full Transformer class. i.e.,

the following capacity inequality,

$$\mathrm{fat}_\alpha(\mathcal{G}_{\text{Transformer}}) < \infty,$$

remains a conjecture rather than a provable property under current theoretical tools. This gap complicates our ability to rigorously assert finite probabilistic capacity for the very models used in practice, especially when considering their introspective reliability and calibration behavior.

Gaussian processes and infinite-dimensional predictors. A similar gap arises for Gaussian Process (GP) models, where the predictive function is drawn from a stochastic process defined by a positive-definite kernel. To the best of our knowledge, while each realization of a GP resides in a Reproducing Kernel Hilbert Space (RKHS), and RKHS complexity measures such as covering numbers or Rademacher complexity are known to relate to generalization [8], the fat-shattering dimension of the GP function class is not known in general. In particular, there is no general result guaranteeing that

$$\mathrm{fat}_\alpha(\mathcal{G}_{\text{GP}}) < \infty$$

for any fixed margin $\alpha > 0$, unless additional smoothness or norm constraints are imposed on the sample paths or kernels. This limits the applicability of PVC-style generalization theory to GP-based uncertainty quantification.

Role of category–level shattering. Measuring PVC_γ at the *category* (rather than the *individual–question*) level restricts the effective domain of each predictor to the finite index set $\{\mathcal{D}_1, \ldots, \mathcal{D}_M\}$, where M denotes the number of categories in the benchmark. For any predictor $f = (f_0, f_1) \in \mathcal{F}$, define the aggregated confidence

$$\mathbb{P}\big(f(x) = y^*(x)\big), \qquad c = 1, \ldots, M.$$

Consequently the probabilistic VC dimension satisfies the immediate bound

$$\mathrm{PVC}_\gamma(\mathcal{F}) \;\le\; M,$$

and, by definition, the same upper bound holds for $\text{C-PVC}^\tau_\gamma(\mathcal{F})$. The finiteness of PVC thus follows without relying on global weight–norm or Lipschitz constraints.

Outlook and open directions. Notwithstanding the function–class perspective of Section 3.2, the fat-shattering framework—while principled for bounded, Lipschitz neural networks—has yet to be rigorously extended to modern LLM and kernel-based architectures. At present, finiteness of PVC_γ and C-PVC^τ_γ is provable only under relatively restrictive assumptions, such as

$$\|W^{(\ell)}\|_1 \le B_\ell \quad \text{and} \quad \phi \text{ is } \rho\text{-Lipschitz}$$

for every layer ℓ. In contrast, models used in practice may exceed these bounds or include non-Lipschitz operations.

Developing theoretical tools that bound the probabilistic capacity of richer, structured architectures—particularly Transformers and GVUSsian Processes—therefore remains an important direction for future work. Such advances would narrow the gap between introspection theory and the empirical performance of high-capacity predictors in contemporary systems.

K Evaluation on Math-500 Benchmark

To assess the robustness and generality of our framework, we replicate the full evaluation procedure on the MATH-500 benchmark [25], a well-established diagnostic suite that spans seven mathematical sub-domains.

The results largely mirror the trends observed in our main evaluation, reinforcing the robustness of our proposed metrics. OpenThinker2-7B continues to demonstrate strong introspective performance, achieving the highest PVC-VUS and C-PVC-VUS on Math-360 and ranking second on MATH-500. This consistency underscores the model's reliable self-assessment capabilities across diverse mathematical reasoning tasks. JiuZhang3.0-7B exhibits similarly stable behavior, consistently achieving the lowest PVC-VUS Gap and ECE while attaining the highest SC-VUS on both datasets. This superior calibration translates to higher C-PVC scores than would be expected from raw reasoning performance alone. Unlike models that systematically overestimate their correctness, JiuZhang produces conservative yet well-aligned self-assessments.

However, several ranking deviations reveal how benchmark characteristics subtly influence calibration behavior. Most notably, Llama-3.1-8B-Instruct shows degraded performance on MATH-500, with substantial drops in C-PVC-VUS, ECE, and AE rankings compared to Math-360, despite maintaining moderate overall PVC-VUS scores.

The joint calibration plot in Figure 5 shows all models falling below the diagonal, reflecting decreased C-PVC relative to PVC when calibration constraints are applied. Models such as Open-Reasoner-Zero-7B and

Table 3: Experimental results comparing model performance on Math 500 dataset. PVC-VUS Gap represents the difference between PVC-VUS and C-PVC-VUS. SC-VUS is Sample Complexity - VUS.

Model	PVC-VUS ↑	C-PVC-VUS ↑	PVC-VUS Gap ↓	SC-VUS ↑	ECE ↓	AE ↓
Qwen2.5-7B (Pretrain) [72]	3.93	2.56	1.37	555.6	0.356	0.439
Qwen2.5-7B-Instruct (SFT+RL) [72]	3.94	2.54	1.39	554.3	0.358	0.438
Qwen2.5-Math-7B-Instruct (SFT+RL) [73]	3.68	2.57	1.11	556.8	0.304	0.476
Llama-3.1-8B-Instruct (SFT+DPO) [43]	3.97	2.67	1.30	566.7	0.331	0.435
OpenThinker2-7B (SFT) [49]	4.36	3.09	1.26	609.4	0.294	0.378
DeepSeek-R1-Distill-Qwen-7B (Distill) [22]	3.95	2.52	1.43	551.6	0.369	0.437
Bespoke-Stratos-7B (SFT) [10]	4.19	2.88	1.30	588.4	0.330	0.403
JiuZhang3.0-7B (SFT) [77]	3.74	**3.11**	**0.63**	**610.9**	**0.169**	0.466
Ministral-8B-Instruct-2410 (SFT+RL) [44]	3.97	2.50	1.46	550.3	0.377	0.434
Open-Reasoner-Zero-7B (RL) [24]	**4.39**	2.99	1.40	598.9	0.322	**0.374**
s1.1-7B (SFT) [46]	4.26	2.93	1.32	593.2	0.317	0.393

OpenThinker2-7B position closest to the upper-right frontier, achieving optimal balance between reasoning coverage and confidence alignment. These cross-dataset results validate the stability of our evaluation framework while illuminating benchmark-specific effects.

The MATH-500 dataset, being more extensively studied, may exhibit greater overlap with certain models' training data. Nevertheless, the consistent ranking patterns in PVC and C-PVC across datasets demonstrate that these metrics provide a meaningful and generalizable approach to assessing model self-evaluation capabilities. This cross-dataset consistency further validates our framework's utility for understanding language model introspective performance.

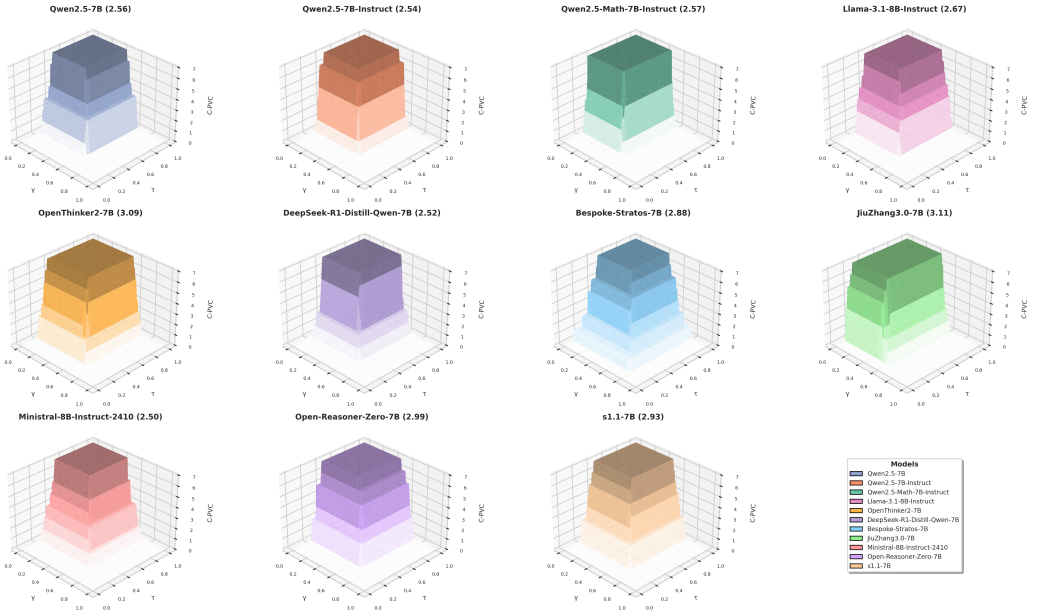


Figure 4: Calibration-aware Probabilistic VC (C-PVC) Dimension Across Different Models on MATH 500: 3D surface plots showing C-PVC dimension values for eleven 7-8B parameter language models. The x-axis (τ) represents calibration error tolerance, y-axis (γ) shows confidence threshold, and z-axis displays the C-PVC values. Higher C-PVC-VUS in parentheses beside each model name indicates better self-evaluation capabilities.

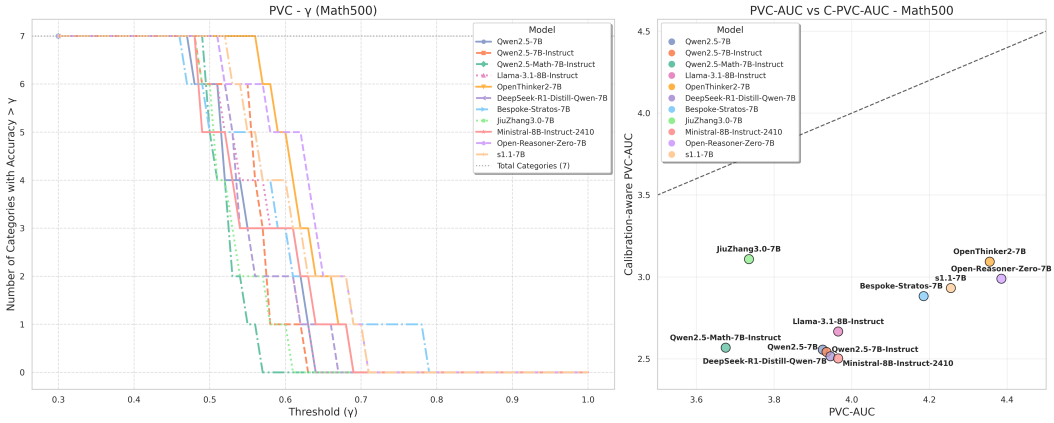


Figure 5: Model Self-Evaluation Capabilities and Calibration Performance: (a) PVC scores across confidence thresholds (γ) for eleven language models. Higher curves indicate better ability to maintain expressive reasoning as confidence requirements increase. Open-Reasoner-Zero-7B and OpenThinker2-7B show the strongest performance. (b) Comparison between PVC-VUS (expressiveness) and C-PVC-VUS (calibrated expressiveness). Points closer to the diagonal dashed line indicate better calibration. JiuZhang3.0-7B shows the smallest gap between these metrics, demonstrating superior calibration, while Open-Reasoner-Zero-7B achieves the highest overall expressiveness.

L Judge Correlation Study

To systematically assess the quality and consistency of the language models' self-evaluation capabilities, we conducted a detailed correlation analysis comparing model-generated self-assessments against judgments provided by three reference LLM judges: Claude 3.7 Sonnet, Amazon Nova Premier, and DeepSeek-R1. This analysis focused on the Math-360 dataset to evaluate mathematical reasoning self-assessment. Figure 6 presents two complementary analyses: the Pearson correlation coefficients indicating the strength and direction of linear relationships between model self-evaluation scores and judge scores, and the direct agreement percentages reflecting how often model selections matched those of each judge.

As illustrated in the left heatmap of Figure 6, the correlation between models and judges exhibits notable variability. Specifically, Amazon Nova Premier consistently shows higher correlations (average correlation = 0.18; see Table 4), particularly pronounced with models such as Open-Reasoner-Zero-7B (correlation = 0.52) and Qwen2.5-7B-Instruct (correlation = 0.31). These findings suggest Amazon Nova Premier employs evaluation criteria broadly aligned with model-generated assessments, serving as a reliable benchmark for assessing self-reflection in various models.

In contrast, Claude 3.7 Sonnet presents considerably weaker correlations overall (average correlation = -0.09). Notably, negative correlations were observed with specific models, such as Qwen2.5-7B-Instruct (-0.17) and Open-Reasoner-Zero-7B (-0.27). This indicates a potential divergence in evaluation strategies, where Claude adopts either stricter or fundamentally distinct criteria compared to other judges.

DeepSeek-R1 maintains moderate correlations (average correlation = 0.01) with models, displaying intermediate consistency. The relatively balanced correlation patterns indicate DeepSeek-R1's evaluation criteria align moderately well across models without extreme bias towards specific methodologies.

The agreement rate heatmap (right side of Figure 6) provides additional clarity regarding model-judge alignment at the outcome-level (correct versus incorrect decisions). Here, Amazon Nova Premier again demonstrates the highest agreement rate on average (0.61; Table 4). A high agreement rate of 0.75 is observed with Open-Reasoner-Zero-7B, reinforcing its robust alignment in judgment criteria. Conversely, Claude 3.7 Sonnet exhibits lower overall agreement (0.53), with the lowest recorded agreement rate of 0.43 (with Open-Reasoner-Zero-7B). This further supports the interpretation that Claude employs significantly more stringent or distinct assessment standards. DeepSeek-R1, with an overall moderate agreement rate (0.55), reflects consistent yet cautious alignment.

An interesting phenomenon observed involves overall accuracy surpassing individual judge accuracy rates. For instance, OpenThinker2 shows an overall accuracy of approximately 0.65, exceeding judge-specific accuracy metrics. This can occur due to partial overlaps and complementary patterns in judge-model agreement, emphasizing the importance of integrating multiple evaluation perspectives when assessing LLM self-reflection capabilities.

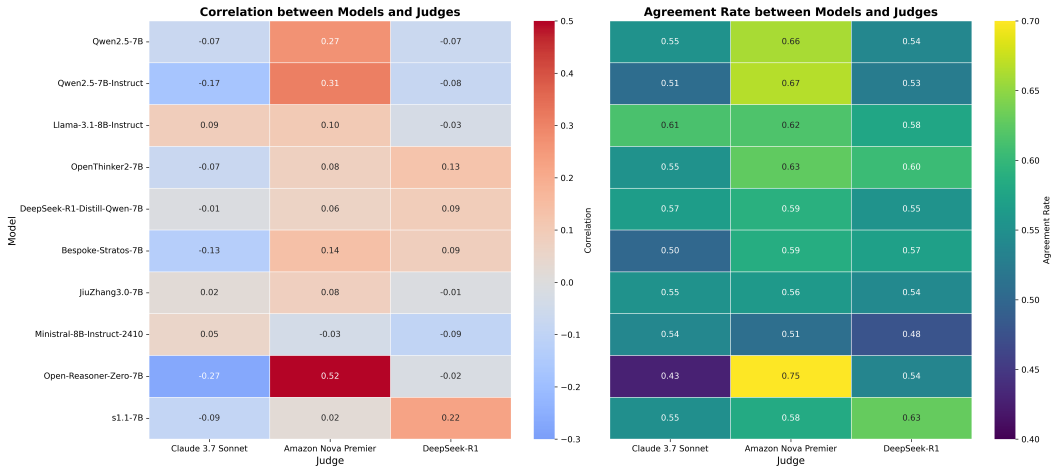


Figure 6: Left: Heatmap showing Pearson correlation between correct model self-evaluation answers and correct judge answers; Right: Heatmap showing direct agreement percentage between model self-evaluation answers and judge answers

In summary, our correlation and agreement analyses highlight meaningful differences among LLM judges regarding their evaluation criteria and model alignment. Amazon Nova Premier emerges as the most broadly applicable reference judge, suitable for general benchmarking. Conversely, Claude 3.7 Sonnet represents a valuable benchmark for stringent evaluation contexts, potentially capturing nuanced deficiencies in model self-assessment. These findings underscore the necessity of incorporating diverse evaluation perspectives to thoroughly gauge and enhance LLM self-evaluation accuracy and reliability.

LLM Judge	**Correlation**	**Agreement rate**
Claude 3.7 Sonnet	-0.09	0.53
Amazon Nova Premier	0.18	0.61
DeepSeek - R1	0.01	0.55

Table 4: Overall pearson correlation and agreement rate between judge and model

M Extended Related Works

Research into LLM self-reflection capabilities has evolved from educational psychology frameworks [12, 20, 28, 35, 47, 52, 56, 66, 69] to LLM-specific methods [60, 9]. In this context, self-reflection refers to a model's internal mechanism for examining its own reasoning processes, while self-evaluation focuses on the quantitative assessment of output quality and confidence estimation. Recent approaches include Reflexion [61], which implements iterative self-improvement through reflection, Self-Refine [40], which enables progressive output refinement, and CRITIC [53], which separates generation from evaluation. Theoretical frameworks like Metacognitive Prompting [65] and metacognitive language models [68] attempt to formalize these self-evaluation processes.

While extensions to probabilistic predictors have been explored in limited contexts [31], they have not been fully developed for analyzing modern neural language models. Our work builds upon fat-shattering dimension theory [42, 3, 64, 5, 15], providing a theoretical bridge between classical learning theory and modern probabilistic models.

Post-hoc calibration methods such as temperature scaling [54] and conformal prediction [2] have been proposed to improve calibration quality. Uncertainty quantification techniques including ensemble methods [33] and Bayesian approaches [18] have been adapted for language models.

Recent advances in language model training have leveraged various forms of preference optimization [62]. Methods such as Reinforcement Learning from Human Feedback [50], Direct Preference Optimization [55], Contrastive Preference Learning [23], and self-rewarding approaches [75] have significantly improved LLM capabilities through incorporating human or AI preferences into training processes. While these approaches often focus on optimizing general helpfulness and harmlessness, specialized variants for reasoning tasks have emerged, such as Group Relative Policy Optimization for mathematics [59].

Collectively, these research areas converge in our work: we leverage classical VC theory to develop a principled framework for evaluating self-reflection capabilities in LLMs, while addressing the critical challenge of calibration that has been identified across deep learning systems. Our approach provides a theoretical foundation for understanding how different training methodologies, particularly preference optimization techniques, influence a model's ability to reliably assess its own reasoning—bridging the gap between theoretical learning guarantees and practical LLM development.

N Detailed experimental methodology

This section elaborates on our empirical approach to measuring PVC dimensions in language models, providing implementation details for the framework introduced in Section 3.3.

N.1 Measurement Protocol

Our empirical evaluation follows a three-stage protocol designed to measure a model's capacity for reasoning self-assessment while avoiding information leakage. The process is illustrated in Figure **??**.

Stage 1: Answer Generation. For each problem instance, the model generates two *independent* solutions using distinct decoding trajectories:

- A solution that follows standard, step-by-step reasoning patterns, and
- A solution that explores a non-standard, potentially more intuitive or compact reasoning path.

These variants ensure diversity in reasoning quality while remaining model-generated.

Stage 2: Self-Evaluation. The model is then presented with both candidate solutions in randomized order. It must choose the answer it believes to be more correct and assign a confidence score to its choice. Notably, the model is *not* given access to the ground-truth answer at any point, ensuring that this choice reflects genuine internal evaluation, not supervised feedback.

Stage 3: External Judging. To determine the ground-truth correctness, a separate ensemble of larger language models—acting as external judges—is given access to the same pair of solutions, along with **the correct final answer**. The ensemble selects the objectively superior solution, serving as the gold label for evaluating self-reflection accuracy. This design ensures that the evaluation signal is (i) grounded in correctness, (ii) independent of the model being evaluated, and (iii) robust to individual model biases.

N.2 Hyperparameters

The decoding and evaluation configuration reported below was applied identically to all eleven models examined in this study—QWEN2.5-7B [72], QWEN2.5-7B-INSTRUCT [72], QWEN2.5-MATH-7B-INSTRUCT [73], DEEPSEEK-R1-DISTILL-QWEN-7B [22], LLAMA-3.1-8B-INSTRUCT [43], OPENTHINKER2-7B [49], BESPOKE-STRATOS-7B [10], JIUZHANG3.0-7B [77], MINISTRAL-8B-INSTRUCT-2410 [44], OPEN-REASONER-ZERO-7B [24], and S1.1-7B [46]. The same parameter settings were used for every query sent to the judge ensemble.

Table 5: Decoding and evaluation settings

Parameter	Temperature	Top-P	Max tokens	Judge ensemble
Value	0.7	0.9	4096	Claude 3.7 Sonnet [4]; Amazon Nova Premier [26]; DeepSeek-R1 [22]

N.3 Benchmark dataset construction

To evaluate mathematical reasoning capabilities more effectively, we developed a new benchmark that addresses one potential limitation of some existing datasets: the possibility of data contamination. Certain prior benchmarks—particularly those derived from widely available competition problems, textbooks, or online resources—may have some degree of overlap with pretraining corpora of large language models, which could potentially influence performance measurements. Our benchmark was developed with attention to originality and domain diversity, aiming to provide a complementary evaluation resource that helps assess reasoning abilities while reducing the likelihood of familiarity effects.

Taxonomic Coverage. Table 6 presents the taxonomy of mathematical reasoning categories and subcategories included in our benchmark. The dataset spans eight core domains—Arithmetic, Algebra, Calculus, Geometry, Number Theory, Combinatorics, Statistics, and Linear Algebra—each subdivided into five foundational subtopics. This taxonomy reflects the structure of standard mathematics curricula and facilitates interpretable analysis across well-defined conceptual boundaries.

Table 6: Taxonomy of mathematical reasoning categories in our benchmark.

Category	Subcategories
Arithmetic	Basic Operations, Fractions, Percentages, Numerical Approximation, Order of Operations
Algebra	Equations, Inequalities, Polynomials, Functions, Systems of Equations
Calculus	Differentiation, Integration, Limits, Series, Applications
Geometry	Plane Geometry, Coordinate Geometry, Transformations, Mensuration, Trigonometry
Number Theory	Divisibility, Modular Arithmetic, Primes, Diophantine Equations, Number Sequences
Combinatorics	Counting Principles, Permutations, Combinations, Probability, Recursion
Statistics	Descriptive Statistics, Distributions, Hypothesis Testing, Regression, Bayesian Inference
Linear Algebra	Matrices, Determinants, Vector Spaces, Eigenvalues, Linear Transformations

Problem Distribution. Each subcategory contains five problems distributed across varying difficulty levels (easy, medium, hard), yielding a total of 360 problems. This balanced structure prevents overrepresentation of specific skills and ensures a fair and comprehensive assessment of mathematical reasoning.

Representative Examples. Table 7 provides one representative problem per subcategory, highlighting the range and depth of reasoning skills required. Problems are designed to elicit multi-step thinking, abstraction, and formal manipulation, rather than direct recall or pattern matching. For instance, problems in combinatorics test combinatorial enumeration and recurrence, while those in number theory evaluate modular reasoning and structural properties of integers.

Taken together, the benchmark offers a principled and contamination-resistant testbed for evaluating mathematical reasoning in language models. It supports both overall capability assessment and fine-grained diagnosis of strengths and weaknesses across mathematical domains.

N.4 Experimental prompts

Our measurement procedure operationalizes the theoretical concept of PVC dimension through a three-stage process.

N.4.1 First solution generation

The target model generates an expert-level solution using the following prompt:

System: You are an expert mathematics educator with deep knowledge across all mathematical domains. Your task is to solve a math problem with exceptional clarity and accuracy.

User: Please solve the following mathematical problem step by step, ensuring complete accuracy:
PROBLEM: {problem}
Provide a detailed solution that demonstrates expert-level understanding.
Include:
1. Clear explanation of your approach
2. Each step fully worked out
3. Proper mathematical notation
4. Verification of your answer
Make sure your final answer is clearly indicated and correct.

N.4.2 Second solution generation

The model generates an alternative solution using:

Table 7: Representative problems from each subcategory in our mathematical reasoning benchmark 360.

Category	Subcategory	Example Problem
Arithmetic	Basic Operations	Calculate 238 + 149.
	Fractions	What is 2/5 + 1/5?
	Percentages	What is 25% of 80?
	Numerical Approximation	Round 47.68 to the nearest whole number.
	Order of Operations	Calculate 3 + 4 × 2.
Algebra	Equations	Solve for x: x + 5 = 12.
	Inequalities	Solve: x + 3 > 7.
	Polynomials	Simplify: $(3x^2 + 2x) + (4x^2 - 5x + 1)$.
	Functions	If f(x) = 2x + 3, find f(4).
	Systems of Equations	Solve the system: x + y = 5, x - y = 1.
Calculus	Differentiation	Find the derivative of $f(x) = 3x^2 + 2x - 5$.
	Integration	Find $(3x^2 + 2)\,dx$.
	Limits	Evaluate $\lim(x \to 3)\ (x^2 - 4)$.
	Series	Find the sum of the first 10 terms of the arithmetic sequence with a = 3 and d = 4.
	Applications	Find the maximum value of $f(x) = -x^2 + 6x - 5$ on the interval [0, 5].
Geometry	Plane Geometry	Find the area of a rectangle with length 8 cm and width 5 cm.
	Coordinate Geometry	Find the distance between the points (3, 4) and (6, 8).
	Transformations	Reflect the point (3, 5) across the x-axis.
	Mensuration	Find the circumference of a circle with radius 5 cm.
	Trigonometry	Find sin(30°).
Number Theory	Divisibility	Determine whether 156 is divisible by 4.
	Modular Arithmetic	Calculate 17 mod 5.
	Primes	List all prime numbers less than 20.
	Diophantine Equations	Find all integer solutions to x + y = 10.
	Number Sequences	Find the next number in the sequence: 3, 7, 11, 15, ...
Combinatorics	Counting Principles	How many different 3-digit numbers can be formed using the digits 1, 2, 3, 4, 5 without repetition?
	Permutations	How many permutations can be formed using all the letters of the word 'MATH'?
	Combinations	How many ways are there to select 3 books from a shelf containing 7 different books?
	Probability	A fair die is rolled. What is the probability of getting a number greater than 4?
	Recursion	Find the 6th term in the Fibonacci sequence, where F = 1, F = 1, and F = F + F for n 3.
Statistics	Descriptive Statistics	Find the mean of the numbers 4, 7, 9, 3, and 2.
	Distributions	If X follows a normal distribution with mean 10 and standard deviation 2, what is P(X > 12)?
	Hypothesis Testing	In a hypothesis test, what is the meaning of a Type I error?
	Regression	In a linear regression model, what does the slope coefficient represent?
	Bayesian Inference	State Bayes' theorem.
Linear Algebra	Matrices	Calculate the sum of the matrices A = [[3, 5], [1, 2]] and B = [[4, 0], [6, 3]].
	Determinants	Calculate the determinant of the matrix A = [[3, 4], [1, 2]].
	Vector Spaces	Determine whether the set of vectors (1, 0, 0), (0, 1, 0), (0, 0, 1) forms a basis for R^3.
	Eigenvalues	Find the eigenvalues of the matrix A = [[2, 0], [0, 3]].
	Linear Transformations	Let T: $R^2 \to R^2$ be defined by T(x, y) = (2x, 3y). Find T(1, 2).

System: You are a brilliant mathematics educator known for your creative problem-solving approaches. Your task is to solve a math problem using your new methods.

User: Please solve the following mathematical problem:
PROBLEM: {problem}
Provide an extremely detailed solution that demonstrates understanding.
Include:
1. A clear explanation of your approach using an alternative and creative solution path
2. Each step fully worked out with mathematical reasoning
3. Proper mathematical notation and formalism
4. If applicable, connect this problem to broader mathematical concepts or real-world applications
Make your solution as thorough and educational as possible, showing multiple ways to think about and verify the solution. Focus on approaches that demonstrate mathematical insight and elegance.

This design choice—requesting alternative approaches rather than explicitly introducing errors—allows for natural quality variation while avoiding artificially degraded solutions that would be too obviously flawed.

N.4.3 Self-evaluation for Math dataset

The model evaluates both solutions (presented in randomized order), identifies which it believes is superior, and provides a confidence score (0-100%). The evaluation prompt follows:

System: You are an expert mathematics evaluator tasked with comparing two solutions to the same problem.

User: Below are two different solutions to the same mathematical problem. Your task is to carefully evaluate both solutions and determine which one is superior in terms of correctness, clarity, and mathematical insight.
PROBLEM: {problem}
SOLUTION A: {solution_a}
SOLUTION B: {solution_b}
Please compare these solutions carefully and respond with:
1. Which solution is better (A or B)
2. Your confidence in this judgment (0-100%)
3. A brief explanation of why one solution is superior to the other
Format your response as:
Better solution: [A or B]
Confidence: [0-100%]
Explanation: [Your analysis]

O Examples

The following examples are provided for illustrative purposes only, to give readers a concrete understanding of how models perform self-evaluation in our framework. While our quantitative results in Section 5 comprehensively evaluate model performance through PVC and C-PVC metrics, these examples offer qualitative insights into the self-evaluation process. These examples illustrate when the model exhibits accurate introspective judgment, and when it fails due to overconfidence or conceptual misunderstanding.

O.1 Examples of Self-Evaluation Behavior

We present two representative cases—one demonstrating successful self-evaluation and another showing flawed assessment—to highlight the different patterns of reasoning that emerge during self-reflection. These examples are anonymized and selected to illustrate typical behaviors observed across multiple models, rather than to evaluate any specific model's performance.

Case 1: Correct Self-Evaluation. In this example, the model is asked to compute the inverse of the function $f(x) = \frac{3x-2}{x+1}$. It generates two answers: Solution A follows a rigorous, textbook-style derivation, while Solution B uses a creative algebraic reformulation. Both arrive at algebraically equivalent forms, though Solution B makes minor verification errors. The model correctly selects Solution A and assigns a confidence score of 90. This choice aligns with the external judge's label, and the model's justification reflects meaningful understanding of correctness. Thus, both its selection and confidence are well-calibrated, yielding high values in PVC and C-PVC.

Case 2: Flawed Self-Evaluation. Here, the model is asked whether $f(x) = |x^2 - 4|$ is one-to-one, and if not, to restrict the domain and compute its inverse. Solution A correctly identifies the need for domain restriction (e.g., $x \geq 2$) and derives the inverse. In contrast, Solution B makes multiple conceptual errors, such as misinterpreting the absolute value function as a square root and incorrectly restricting the domain. Despite these flaws, the model selects Solution B with a confidence score of 80, justifying its choice based on surface-level clarity. This leads to a self-reflection error: the model is confident in a mathematically invalid solution. While PVC remains high (a choice was made), C-PVC is low due to confidence miscalibration.

Table 8: Comparison of two model self-evaluation behaviors.

Aspect	**Case 1: Rational Function Inversion**	**Case 2: One-to-One Determination**
Task Type	Inverse function derivation via algebraic manipulation	One-to-one check and domain-restricted inverse
Judge Ensemble Correct Answer	Solution A	Solution A
Model's Selected Answer	Solution A	Solution B
Confidence Score	90	80
Model's Justification	Emphasized rigor and step-by-step reasoning	Praised perceived clarity despite flawed math
Actual Reasoning Errors	None; correct and calibrated	Misinterpreted absolute value; incorrect domain

Discussion. These examples highlight the need for introspective evaluation beyond accuracy. While both cases involve the model choosing between two self-generated solutions, only one reflects proper alignment between correctness and confidence. Our proposed PVC and C-PVC metrics allow us to distinguish not just whether a model chooses the right answer, but whether it *knows* it is right. Such analysis reveals that some models (e.g., RLHF-finetuned) may appear confident while being systematically miscalibrated, whereas others (e.g., distilled or conservative models) exhibit more cautious but reliable introspection.

O.2 Case 1: Correct Self-Evaluation

User: Find the inverse function of $f(x) = \frac{3x-2}{x+1}$.

Solution A: Step 1: Understanding the Problem
To find the inverse function of $f(x)$, we need to express the original function in a way that allows us to easily swap the roles of x and y. The function given is $f(x) = \frac{3x-2}{x+1}$.
Step 2: Setting Up the Inverse Function
To find the inverse function, we'll start by letting $y = f(x)$, which gives us the equation:
$y = \frac{3x-2}{x+1}$
Next, we'll switch the roles of x and y to begin solving for the inverse function. This means replacing y with x and replacing x with y in the original equation.
Step 3: Swapping Variables
Swapping the variables gives us:
$x = \frac{3y-2}{y+1}$
Step 4: Solving for y
To solve for y, we'll first multiply both sides of the equation by $(y + 1)$ to get rid of the fraction:
$x(y + 1) = 3y - 2$
Step 5: Expanding and Simplifying
Now, we'll expand the left side of the equation and simplify:
$xy + x = 3y - 2$
Next, we'll move all terms involving y to one side of the equation and all other terms to the other side:
$xy - 3y = -x - 2$
Step 6: Factoring and Solving
We can factor out y from the terms on the left side:
$y(x - 3) = -x - 2$
To isolate y, we'll divide both sides of the equation by $(x - 3)$:
$y = \frac{-x-2}{x-3}$
Step 7: Writing the Inverse Function
Now that we've solved for y, we can write the inverse function of $f(x)$:
$f^{(-1)}(x) = \frac{-x-2}{x-3}$
Verification
To verify that this is the correct inverse function, we can check that $f(f^{(-1)}(x)) = x$ and $f^{(-1)}(f(x)) = x$.
Let's start by checking that $f(f^{(-1)}(x)) = x$:
$f(f^{(-1)}(x)) = f(\frac{-x-2}{x-3})$
Substituting this into the original function $f(x) = \frac{3x-2}{x+1}$, we get:
$f(f^{(-1)}(x)) = \frac{3(\frac{-x-2}{x-3})-2}{(\frac{-x-2}{x-3})+1}$
Simplifying this expression will show that $f(f^{(-1)}(x)) = x$, which verifies that the inverse function is correct.
Final Answer
The inverse function of $f(x) = \frac{3x-2}{x+1}$ is:
$f^{(-1)}(x) = \frac{-x-2}{x-3}$
This answer is verified through a detailed and accurate solution process.

Solution B: Inverse Function Solution: A Creative Approach
Step 1: Set up the function equation
We are given the function $f(x) = \frac{3x-2}{x+1}$. To find the inverse function, we'll use a substitution method. Let's start by setting $y = f(x)$:
$y = \frac{3x-2}{x+1}$
Step 2: Switch x and y
To find the inverse, we need to switch the roles of x and y. This means that we'll replace x with y and y with x:
$x = \frac{3y-2}{y+1}$
Step 3: Solve for y
Now, we need to solve this new equation for y. Our goal is to isolate y on one side of the equation. We can do this by first multiplying both sides by $y + 1$ to eliminate the fraction:
$x(y + 1) = 3y - 2$
Step 4: Distribute x and simplify
Distribute x to the terms inside the parentheses:
$xy + x = 3y - 2$
Now, let's move all terms involving y to one side and the constant terms to the other side:
$xy - 3y = -x - 2$
Step 5: Factor out y and simplify
Factor out y from the terms on the left-hand side:
$y(x - 3) = -x - 2$
Step 6: Solve for y
To solve for y, divide both sides by $x - 3$. However, we must consider two cases to avoid division by zero:
Case 1: $x \neq 3$
$y = \frac{-x-2}{x-3}$
Case 2: $x = 3$ (this is a special case that we'll revisit later)
For the general case $x \neq 3$, we have found the inverse function.
Step 7: Simplify the inverse function
Let's rewrite the inverse function in a simpler form by multiplying the numerator and denominator by -1 to make the fraction more elegant:
$y = \frac{x+2}{3-x}$
This is our inverse function.
Step 8: Verify the inverse function
To verify that our inverse function is correct, we can plug it back into the original function:
$f(f^{-1}(x)) = f\left(\frac{x+2}{3-x}\right)$
$f(f^{-1}(x)) = \frac{3\left(\frac{x+2}{3-x}\right)-2}{\left(\frac{x+2}{3-x}\right)+1}$
Step 9: Simplify the expression
To simplify this expression, multiply the numerator and denominator by $(3 - x)$ to eliminate the fraction:
$f(f^{-1}(x)) = \frac{3(x+2)-2(3-x)}{(x+2)+(3-x)}$
$f(f^{-1}(x)) = \frac{3x+6-6+2x}{3}$
$f(f^{-1}(x)) = \frac{5x}{3}$
Step 10: Conclusion
We started with the function $f(x) = \frac{3x-2}{x+1}$ and found its inverse function using a creative substitution method. The inverse function is $f^{-1}(x) = \frac{x+2}{3-x}$. We verified the inverse function by plugging it back into the original function and simplifying the expression.
Broader Mathematical Concepts and Real-World Applications
Inverse functions have numerous applications in various fields, including physics, engineering, and computer science. For example, in physics, the inverse function of a displacement-time function can be used to find the velocity and acceleration of an object. In engineering, the inverse function of a stress-strain function can be used to design materials with specific properties. In computer science, inverse functions are used in algorithms for solving systems of equations and in data analysis.
Conclusion
In this solution, we demonstrated a creative approach to finding the inverse function of $f(x) = \frac{3x-2}{x+1}$ using a substitution method. We worked through each step with mathematical reasoning and proper notation, and we verified the inverse function by plugging it back into the original function. We also connected this problem to broader mathematical concepts and real-world applications, highlighting the importance of inverse functions in various fields.

Self-Evaluation:
Selected Solution: A
Confidence Score: 90
Upon reviewing both solutions, I noticed that Solution A is more thorough and accurate in its step-by-step process. It starts by properly setting up the inverse function, swapping the variables, and solving for y. In contrast, Solution B uses a more creative approach but lacks the necessary rigor and clarity in its solution process.
One major issue with Solution B is its failure to properly verify the inverse function. While it attempts to simplify the expression by multiplying the numerator and denominator by (3 - x), it does not actually demonstrate that this process is valid or that it correctly solves for y. Additionally, Solution B's approach is more prone to errors due to its reliance on a single substitution, whereas Solution A uses a more systematic approach that involves multiple steps.
Another issue with Solution B is its use of the special case x = 3, which is not properly addressed. While it mentions this case in passing, it does not provide a clear solution or explanation for this scenario.
In terms of broader mathematical concepts and real-world applications, both solutions could benefit from more connections to these areas. However, Solution A provides a more solid foundation in algebraic manipulation and inverse function theory, making it a more reliable choice for students and educators.
Overall, I would rate Solution A as the better of the two solutions due to its thoroughness, accuracy, and clarity. While Solution B attempts to be more creative, its flaws and omissions make it less reliable.

Analysis of Self-Evaluation:
This is an example of a flawed self-evaluation. The evaluator correctly selected Solution A as the better answer, but their reasoning contains significant errors and misunderstandings:
The evaluator claims Solution B "fails to properly verify the inverse function," but in reality, Solution B attempts verification but makes a computational error in Step 9. When simplifying $(x + 2) + (3 - x)$, the result should be 5, not 3. This leads to the incorrect final verification result of $\frac{5x}{3}$ instead of x.
The evaluator praises Solution A's verification process, but Solution A only outlines the verification without actually completing it. It merely states "Simplifying this expression will show that $f(f^{(-1)}(x)) = x$ without performing the calculations.
The evaluator mentions that Solution B has issues with addressing the special case x = 3, but this is actually a strength of Solution B, as it acknowledges domain restrictions that Solution A completely ignores.
The evaluator fails to recognize that Solution A provides its final answer as $\frac{-x-2}{x-3}$ while Solution B gives $\frac{x+2}{3-x}$. These are equivalent after multiplying numerator and denominator by -1, but the evaluator doesn't acknowledge this.
A better self-evaluation would have recognized that both solutions follow similar algebraic approaches but Solution B actually attempts (though with errors) the verification that Solution A merely mentions. It should also have noted that both solutions arrive at equivalent forms of the inverse function, with Solution B taking the extra step to present it in a slightly more elegant form.

O.3 Case 2: Flawed Self-Evaluation

User: Determine whether the function $f(x) = |x^2 - 4|$ is one-to-one. If not, restrict the domain to make it one-to-one and find its inverse.

Solution A: To determine whether the function $f(x) = |x^2 - 4|$ is one-to-one, we need to understand what it means for a function to be one-to-one. A function f is one-to-one if it assigns distinct outputs to distinct inputs, meaning that no two different inputs produce the same output.
Let's start by analyzing the given function $f(x) = |x^2 - 4|$. This is an absolute value function composed with a quadratic function.
Step 1: Understand the nature of the quadratic function The quadratic function $x^2 - 4$ has its vertex at (0, -4) and its roots at $x = -2$ and $x = 2$. This means that the graph of this function is a parabola that opens upwards, with x-intercepts at (-2, 0) and (2, 0).
Step 2: Understand the absolute value function The absolute value function |u| returns the non-negative value of u, where u is any real number. This means that $|x^2 - 4|$ will always produce non-negative values, except when $x^2 - 4$ is negative, which occurs when $-4 < x^2 - 4 < 0$.
Step 3: Determine the behavior of the function $f(x) = |x^2 - 4|$ The graph of $f(x) = |x^2 - 4|$ consists of two branches: one above the x-axis for x < -2 and one below the x-axis for x > 2, and both are symmetric about the y-axis. This is becVUSe the absolute value function $|x^2 - 4|$ takes the absolute value of $x^2 - 4$, resulting in non-negative values for both branches.
Step 4: Check if the function is one-to-one To check if $f(x) = |x^2 - 4|$ is one-to-one, we need to verify if it satisfies the condition that no two different inputs produce the same output. Since the graph of $f(x) = |x^2 - 4|$ consists of two branches, one above the x-axis and one below, we can see that this function does not satisfy the one-to-one condition.
Step 5: Restrict the domain to make the function one-to-one To make $f(x) = |x^2 - 4|$ one-to-one, we need to restrict its domain to either $x \geq 2$ or $x \leq -2$. This is becVUSe restricting the domain to either of these intervals will ensure that the function has an inverse.
Step 6: Find the inverse of the function $f(x) = |x^2 - 4|$ Let's find the inverse of $f(x) = |x^2 - 4|$, restricted to $x \geq 2$.
We can start by writing $y = |x^2 - 4|$ and solving for x in terms of y. Since $y \geq 0$, we have:
$y = x^2 - 4\ y + 4 = x^2\ x^2 = y + 4\ x = \pm\sqrt{(y + 4)}$
However, since $x \geq 2$, we only take the positive square root:
$x = \sqrt{(y + 4)}$
Substituting $y = |x^2 - 4|$ back into the equation, we get:
$x = \sqrt{(|x^2 - 4| + 4)}$
This can be simplified to:
$f^{-1}(x) = \sqrt{(x + 4)}$
Step 7: Verify the inverse To verify that $f^{-1}(x) = \sqrt{(x + 4)}$ is the inverse of $f(x) = |x^2 - 4|$, we need to check that $f(f^{-1}(x)) = x$ and $f^{-1}(f(x)) = x$.
Substituting $f^{-1}(x) = \sqrt{(x + 4)}$ into $f(x) = |x^2 - 4|$, we get:
$f(f^{-1}(x)) = f(\sqrt{(x + 4)}) = |(\sqrt{(x + 4)})^2 - 4| = |x + 4 - 4| = |x| = x$
Similarly, substituting $f(x) = |x^2 - 4|$ into $f^{-1}(x) = \sqrt{(x + 4)}$, we get:
$f^{-1}(f(x)) = \sqrt{(|x^2 - 4| + 4)}$
Since the domain of f is restricted to $x \geq 2$, we know that $|x^2 - 4| \geq 0$. Therefore:
$f^{-1}(f(x)) = \sqrt{(|x^2 - 4| + 4)} = \sqrt{(x^2 - 4 + 4)} = \sqrt{x^2} = x$
This verifies that $f^{-1}(x) = \sqrt{(x + 4)}$ is indeed the inverse of $f(x) = |x^2 - 4|$.
The final answer is: $f^{-1}(x) = \sqrt{(x + 4)}$ for $x \geq 2$.

Solution B: To solve this problem, I'll employ a combination of algebraic manipulations, graphical analysis, and domain restriction techniques. This approach will allow us to investigate the function's behavior and determine whether it's one-to-one.
Step 1: Algebraic Manipulation
Let's rewrite the function $f(x) = |x^2 - 4|$ using a substitution. We can let $u = x^2 - 4$, which implies that u is non-negative (since the absolute value of a quantity is non-negative). This allows us to rewrite the original function as:
$f(x) = |u| = \sqrt{u}$, since the square root of a non-negative quantity is equivalent to the absolute value.
Step 2: Graphical Analysis
Now, let's analyze the graph of the function $f(x) = \sqrt{u} = \sqrt{x^2 - 4}$. To do this, we need to understand the graph of the related function $y = x^2 - 4$. Since this is a quadratic function with a negative leading coefficient, its graph is a parabola that opens downwards, with a vertex at (0, -4).
The graph of $f(x) = \sqrt{x^2 - 4}$ can be obtained by taking the square root of the y-values from the graph of $y = x^2 - 4$, which results in a half-parabola that opens upwards, with the vertex at (0, 0).
Step 3: Domain Restriction
Notice that the graph of $f(x) = \sqrt{x^2 - 4}$ is not one-to-one on the entire real number line. To find a domain restriction that would make the function one-to-one, we can analyze the behavior of the function in different intervals.
Since the graph of $f(x) = \sqrt{x^2 - 4}$ has a minimum point at (0, 0), we can restrict the domain to either $(, 0]$ or $[0,)$. Let's choose the domain $[0,)$ for this example.
Step 4: Finding the Inverse
To find the inverse of $f(x) = \sqrt{x^2 - 4}$ on the domain $[0,)$, we'll use the following steps:
Write $y = \sqrt{x^2 - 4}$. Square both sides to get $y^2 = x^2 - 4$. Add 4 to both sides to get $y^2 + 4 = x^2$. Take the square root of both sides to get $y = \pm\sqrt{x^2 + 4}$. Since $y = \sqrt{x^2 - 4}$ on the domain $[0,)$, we have $y = \sqrt{x^2 + 4}$ as the inverse function. The final inverse function is $f^{-1}(x) = \sqrt{x^2 + 4}$.
Conclusion
The original function $f(x) = |x^2 - 4|$ is not one-to-one on the entire real number line. However, by restricting the domain to $[0,)$ and using the algebraic manipulations and graphical analysis techniques presented above, we found the inverse function $f^{-1}(x) = \sqrt{x^2 + 4}$.
Broader Mathematical Concepts
This problem demonstrates several key mathematical concepts:
The use of substitution to simplify expressions and make algebraic manipulations more manageable. The importance of graphical analysis in understanding the behavior of functions and making informed domain restriction decisions. The application of the concept of the inverse of a function and its relevance to the problem at hand. Real-World Applications
The problem-solving techniques employed in this solution have real-world applications in:
Signal processing and filtering: The domain restriction technique can be applied to filter out unwanted signals in audio or image processing applications. Computer graphics: The use of substitution and algebraic manipulation can be applied to create more efficient and effective algorithms for rendering graphics. Optimization problems: The technique of finding the inverse of a function can be applied to solve optimization problems in various fields, such as economics, engineering, and physics. By employing a combination of algebraic manipulations, graphical analysis, and domain restriction techniques, we've been able to determine whether the function $f(x) = |x^2 - 4|$ is one-to-one, find its inverse, and explore broader mathematical concepts and real-world applications.

Self-Evaluation:
Selected Solution: B
Confidence Score: 80
Both solutions A and B aim to determine whether the function $f(x) = |x^2 - 4|$ is one-to-one, and if not, find its inverse by restricting the domain.
Solution A starts by analyzing the nature of the quadratic function $x^2 - 4$, followed by an understanding of the absolute value function. It correctly determines the behavior of the function $f(x) = |x^2 - 4|$, concludes that it is not one-to-one on the entire real number line, and restricts the domain to either $x \geq 2$ or $x \leq -2$ to make it one-to-one.
However, when finding the inverse of $f(x) = |x^2 - 4|$, solution A takes the square root of $|x^2 - 4| + 4$, which results in $\sqrt{(x + 4)}$ for $x \geq 2$. This inverse function is correct, but the reasoning for obtaining it is not as clear as in solution B.
Solution B uses a combination of algebraic manipulations and graphical analysis to determine the nature of the function $f(x) = |x^2 - 4|$. It correctly identifies the function as not one-to-one on the entire real number line and restricts the domain to $[0, \infty)$ to make it one-to-one.
When finding the inverse of $f(x) = |x^2 - 4|$ on the domain $[0, \infty)$, solution B takes a clear and logical approach by first rewriting the function as $\sqrt{(x^2 - 4)}$, then squaring both sides to get $y^2 = x^2 - 4$, adding 4 to both sides to get $y^2 + 4 = x^2$, and finally taking the square root of both sides to get $y = \pm\sqrt{(x^2 + 4)}$.
However, since the original function $f(x) = |x^2 - 4|$ is only defined on the domain $[0, \infty)$, solution B correctly determines the inverse as $f^{-1}(x) = \sqrt{(x^2 + 4)}$, which is also correct.
Both solutions A and B are mathematically correct and provide a clear explanation of the process for determining whether the function $f(x) = |x^2 - 4|$ is one-to-one and finding its inverse. However, solution B is slightly more detailed and provides a clearer explanation of the reasoning behind obtaining the inverse function.
Therefore, solution B is of higher quality overall due to its clearer explanations and more logical approach to finding the inverse function.

Analysis of Self-Evaluation:
This is an example of a problematic self-evaluation with critical errors in judgment. The evaluator incorrectly selects Solution B as superior, despite Solution B containing multiple fundamental mathematical errors.
Key issues with the self-evaluation:
The evaluator fails to recognize that Solution B makes a fundamental error by equating $|x^2 - 4|$ with $\sqrt{(x^2 - 4)}$, which is mathematically incorrect. The absolute value is not equivalent to the square root function.
The evaluator doesn't notice that Solution B incorrectly describes the quadratic function $x^2 - 4$ as having a negative leading coefficient and opening downward, when it actually has a positive leading coefficient and opens upward.
The evaluator accepts Solution B's incorrect domain restriction to $[0, \infty)$, which doesn't make the function one-to-one. The correct restriction would be $x \geq 2$ (or $x \leq -2$) as correctly identified in Solution A.
The evaluator erroneously claims both solutions are "mathematically correct" when Solution B contains several critical mathematical errors that lead to an incorrect inverse function.
The evaluator praises Solution B for being "more detailed" and providing "clearer explanations," when in fact Solution B's explanations are built on incorrect mathematical foundations.
A proper self-evaluation would have recognized that Solution A correctly analyzes the function, identifies the proper domain restriction to make it one-to-one, and derives the correct inverse function with appropriate verification. It would have identified the multiple fundamental errors in Solution B's approach and reasoning.