Calibrating LLM Judges: Linear Probes for Fast and Reliable Uncertainty Estimation

Bhaktipriya Radharapu[1,2], Eshika Saxena[1,2], Kenneth Li[2], Chenxi Whitehouse[1,2], Adina Williams[1,2], Nicola Cancedda[1,2]
[1]FAIR at Meta, [2]Meta Superintelligence Labs
Correspondence: bhakti@meta.com

Abstract

As LLM-based judges become integral to industry applications, obtaining well-calibrated uncertainty estimates efficiently has become critical for production deployment. However, existing techniques, such as verbalized confidence and multi-generation methods, are often either poorly calibrated or computationally expensive. We introduce linear probes trained with a Brier score-based loss to provide calibrated uncertainty estimates from reasoning judges' hidden states, requiring no additional model training. We evaluate our approach on both objective tasks (reasoning, mathematics, factuality, coding) and subjective human preference judgments. Our results demonstrate that probes achieve superior calibration compared to existing methods with $\approx$ 10x computational savings, generalize robustly to unseen evaluation domains, and deliver higher accuracy on high-confidence predictions. However, probes produce conservative estimates that underperform on easier datasets but may benefit safety-critical deployments prioritizing low false-positive rates. Overall, our work demonstrates that interpretability-based uncertainty estimation provides a practical and scalable plug-and-play solution for LLM judges in production.

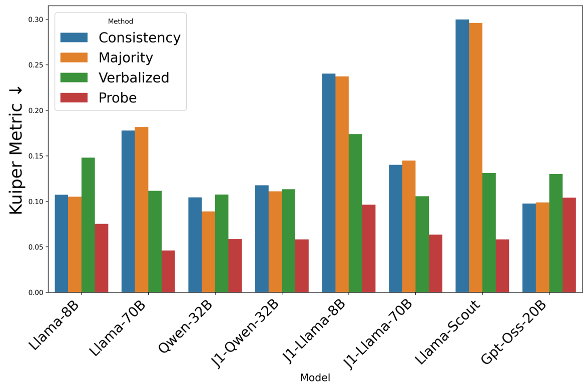


Figure 1: Calibration of models across model architectures, datasets, and uncertainty estimation methods as measured by the Kuiper metric. We compare four approaches across dense prompted judges (LLaMA 8B/70B, Qwen 32B), dense fine-tuned judges (J1 family), and MoE prompted judges (GPT-OSS 20B, LLaMA Scout). Our probe-based method outperforms baseline approaches across all architectures and training paradigms. Results averaged across all evaluation datasets. **Lower values indicate better calibration.**

1 Introduction

LLM-as-judge paradigm (Zheng et al., 2023; Dubois et al., 2023; Touvron et al., 2023) has become ubiquitous in modern AI development, providing reward signals for alignment training (RLAIF) (Bai et al., 2022; Lee et al., 2023; Kirk et al., 2024; Zeng et al., 2024; Zhang et al., 2025b) and ranking models for deployment decisions (Chiang et al., 2024). With the rapid advancement of reasoning-capable models (OpenAI, 2024; DeepSeek-AI, 2025), LLM judges that generate reasoning traces before rendering verdicts are becoming increasingly prevalent due to their higher accuracy and transparency (Whitehouse et al., 2025; Chen et al., 2025).

Yet, current practice treats all judgments as equally reliable. Without calibrated confidence estimates, we cannot distinguish high-confidence judgments from cases where the LLM is essentially guessing. Moreover, LLM judges are known to be overconfident, systematically expressing higher confidence than their empirical accuracy supports (Tian et al., 2025; Jung et al., 2024). This leaves practitioners to either blindly trust all judgments or manually review everything (defeating the purpose of automation). This lack of uncertainty awareness is particularly problematic across judgment types: for correctness evaluation, we cannot identify objectively wrong judgments to exclude (Zhou et al., 2024b; Ross et al., 2024; Wang et al., 2023); for preference evaluation, we cannot detect genuinely ambiguous cases where humans may dis-

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 6: Industry Track), pages 193–216
July 2-7, 2026 ©2026 Association for Computational Linguistics

agree or multiple valid answers exist (Aroyo and Welty, 2015; Pavlick and Kwiatkowski, 2019; Nie et al., 2020; Talat et al., 2022; Radharapu et al., 2025).

Calibrated LLM judges, whose expressed confidence matches their empirical accuracy, enable systems to route straightforward cases to efficient models while reserving expensive judges or human reviewers for uncertain decisions (Jung et al., 2024; Chen et al., 2023), dramatically reducing computational and labor costs. Training processes become more efficient by down-weighting uncertain judgments, preventing noisy labels from causing reward hacking (Gao et al., 2022) and model collapse (Zhang et al., 2024). Across these applications, calibration serves complementary purposes: flagging likely errors in correctness tasks while preserving valid diversity in preference tasks. Calibration is thus essential for building reliable and trustworthy AI systems with LLM judges.

In this work, our main contributions are: (1) A Brier-score–trained linear probe that produces calibrated uncertainty estimates from reasoning judges' hidden states, requiring no additional model training or multi-sample generation. (2) Our probe achieves substantially better calibration than verbalized and multi-generation baselines across multiple model families (dense and MoE) and judging styles (prompted and fine-tuned), while requiring an order of magnitude less compute. (3) Strong out-of-distribution generalization to unseen benchmarks, with analysis of key trade-offs in probe's performance.

2 Related Work

2.1 LLM Judges and Reasoning-Based Evaluation

Large language model judges have become ubiquitous for evaluating model outputs across tasks ranging from pairwise preference ranking for RLHF (Christiano et al., 2017; Ouyang et al., 2022) to judging for correctness in verifiable tasks. These judges span a spectrum from prompted general-purpose models like GPT-4 (Zheng et al., 2023; Dubois et al., 2023) to specialized fine-tuned judges (Whitehouse et al., 2025; Zhu et al., 2025; Kim et al., 2024a,b; Li et al., 2024a) that reason before giving their verdict. While these judges achieve high accuracy, LLM judges are known to suffer from systematic overconfidence (Jung et al., 2024; Tian et al., 2025; Xiong et al., 2024).

2.2 Uncertainty Estimation in Large Language Models

Uncertainty estimation methods for LLMs fall into several categories, each with distinct trade-offs:

Logit-Based Methods. Token-level uncertainty estimation methods like Perplexity and Temperature Scaling (Guo et al., 2017) assume uniform calibration across tokens, ignoring the semantic and contextual nuances crucial for reasoning tasks (Xie et al., 2024a). Maximum Softmax Probability (MSP) (Plaut et al., 2024) has been shown to be consistently miscalibrated in reasoning and multiple-choice QA tasks. Additionally, approaches such as contextual calibration (Zhao et al., 2021) and batch calibration (Zhou et al., 2024a) operate on single-token output logits, making them inapplicable to reasoning judges that generate multi-token responses.

Verbalized Confidence. Asking models directly for confidence scores (Tian et al., 2023; Kadavath et al., 2022; Lin et al., 2022) is straightforward, and has been argued to be better than logit-based methods. However, verbalized confidence produces overconfident estimates (Tian et al., 2025; Xiong et al., 2024; Tao et al., 2025; Lyu et al., 2025).

Consistency-Based Methods. Self-consistency (Wang et al., 2022; Manakul et al., 2023), semantic entropy (Kuhn et al., 2023), and related approaches (Chen and Mueller, 2024) achieve strong calibration by aggregating uncertainty across multiple generations. Methods such as prompt ensembles or model ensembles (Huang et al., 2023; Jung et al., 2024; Tian et al., 2025) involve perturbing prompts or aggregating uncertainty from different models. However, they incur substantial computational costs (typically 10–20$\times$ inference overhead), limiting practical deployment.

Training-Based Approaches. Methods like fine-tuning the model for improved verbalized confidence (Damani et al., 2025; Li et al., 2025a,b) or specialized architectures (Huang et al., 2023; Kapoor et al., 2024; Xie et al., 2024a) require significant computational overhead through additional training or architectural modifications.

Interpretability-Based Approaches. Recent work has extracted uncertainty signals from internal representations (Azaria and Mitchell, 2023; Zou et al., 2023; Li et al., 2024b; Kossen et al., 2024; Burns et al., 2023; Sriramanan et al., 2024), but focuses predominantly on hallucination detection in factual question-answering where models recall memorized knowledge—a fundamentally different setting from reasoning. Other methods (Zhang et al., 2025a; Bi et al., 2025; Wang et al., 2024; Ji et al., 2025; Xie et al., 2024b) targetted at reasoning, store hidden states per layer and per intermediate reasoning step/token during inference, incurring significant computational overhead with longer reasoning chains. Moreover, these methods target *uncertainty-based selective classification* (distinguishing correct from incorrect predictions) rather than *uncertainty calibration* (aligning predicted uncertainty with actual correctness). These tasks are complementary, and performance on one does not imply the other (Tao et al., 2025).

Our work addresses these limitations with a plug-and-play approach: linear probes that achieve strong calibration for reasoning judges without requiring additional model training, multi-sample generation, or per-token state storage during inference, offering a practical and computationally efficient solution for deployment at scale.

3 Method

Taking inspiration from other interpretability works on linear probes that by and large perform *selective classification* (distinguishing correct from incorrect predictions) (Kossen et al., 2024; Azaria and Mitchell, 2023; Zou et al., 2023; Burns et al., 2023; Li et al., 2024b), we train probes to improve *calibration* of models. We train a linear regression model on the judge's residual stream activations from the last token, with verdict accuracy as the label. The probe output is converted to a [0,1] probability via a sigmoid, with no additional post-hoc calibration.

Layer selection is based on validation performance, and we optimize the Brier score (Glenn, 1950) loss:

$$\mathcal{L}_{Brier} = \frac{1}{N}\sum_{i=1}^{N}(\hat{y}_i - y_i)^2$$

where N is the number of samples, $\hat{y}_i$ is the predicted probability of verdict accuracy, and $y_i \in \{0, 1\}$ is the ground truth label. See A.5 for more details on the training of our probes.

In all our experiments, we prompt the judge to verbally express its uncertainty before its verdict, allowing the probe to leverage both internal representations and linguistic hedging signals, which correlate with uncertainty (Lin et al., 2022; Lyu et al., 2025).

Datasets. We train our probes using the Preference Proxy Evaluation (PPE) dataset (Frick et al., 2024), which features real-world prompts sourced from LM Arena and addresses two key LLM judge tasks: preference alignment and correctness. The dataset includes PPE Preference (10.2K samples) with human preference pairs from 20 LLMs across 121 languages, and PPE Correctness (12.7K samples) with response pairs from four models on five benchmarks (MMLU-Pro, MATH, GPQA, MBPP-Plus, IFEval) spanning knowledge, mathematics, STEM, coding, and instruction following.

For out-of-distribution evaluation, we use JudgeBench (Tan et al., 2024) (620 samples) and RewardBench (Lambert et al., 2024) (3K samples). JudgeBench contains correct-incorrect response pairs across knowledge, reasoning, math, and coding tasks, while RewardBench focuses heavily on chat and human preference, and also includes safety, coding and reasoning. These datasets test our probes' ability to generalize to new prompt types, languages, tasks, and model responses beyond the training distribution.

Models. We evaluate our probes on six dense models with varying architectures, model families, and training strategies: fine-tuned judges from the state-of-the-art J1 family (Whitehouse et al., 2025), as well as their prompted (non-finetuned) variants across 8B, 32B, and 70B parameters—based on (Qwen Team, 2025) and Llama (AI@Meta, 2024). In addition, we assess two Mixture-of-Experts (MoE) models: Llama-Scout (109B) (AI, 2025) and GPT-OSS-20B (OpenAI, 2025). MoE models (Fedus et al., 2021) utilize sparse activation patterns and routing mechanisms, which may encode uncertainty differently than dense models, thereby providing a rigorous test of the robustness of our probing approach.

Judge Formulations. We consider three judge formulations: **Pairwise Judge with Verdict (PaV):** Given a question x and response pair (a, b), the judge generates reasoning trace followed by the preferred response y. **Pairwise Judge with Scores (PaS):** The judge generates reasoning trace, followed by real-valued scores s_a, s_b for each response, selecting the higher-scoring response as

Method	LLAMA 8B	LLAMA 70B	QWEN 32B	J1 QWEN 32B	J1 LLAMA 8B	J1 LLAMA 70B
			PPE Correctness			
Verbalized	0.182 / 0.174	0.135 / 0.131	0.107 / 0.108	0.103 / 0.105	0.219 / 0.213	0.076 / 0.074
Consistency	0.159 / 0.170	0.211 / 0.208	0.132 / 0.145	0.098 / 0.130	0.278 / 0.260	0.134 / 0.144
Majority	0.159 / 0.154	0.215 / 0.202	0.125 / 0.137	0.085 / 0.109	0.279 / 0.261	0.138 / 0.128
Probe	**0.047 / 0.075**	**0.017 / 0.025**	**0.020 / 0.028**	**0.029 / 0.039**	**0.047 / 0.069**	**0.017 / 0.018**
			PPE Preference			
Verbalized	0.173 / 0.166	0.123 / 0.121	0.196 / 0.194	0.195 / 0.195	0.192 / 0.188	0.122 / 0.120
Consistency	0.153 / 0.155	0.231 / 0.226	0.048 / 0.083	0.051 / 0.071	0.324 / 0.311	0.207 / 0.200
Majority	0.150 / 0.143	0.234 / 0.223	**0.038 / 0.065**	**0.050 / 0.069**	0.323 / 0.310	0.211 / 0.197
Probe	**0.062 / 0.093**	**0.040 / 0.047**	0.066 / 0.077	0.056 / 0.072	**0.051 / 0.066**	**0.034 / 0.039**

Table 1: Calibration on Preference Policy Evaluation (PPE) datasets measured by Kuiper/ECE (lower is better). We evaluate dense-architecture models as prompted and fine-tuned judges across multiple sizes. On objective PPE-Correctness tasks, probes outperform all baselines across all models. On subjective PPE-Preference tasks, probes achieve best performance on Llama variants and on Qwen closely match majority voting, which requires 10× higher computational cost.

Method	LLAMA 8B	LLAMA 70B	QWEN 32B	J1 QWEN 32B	J1 LLAMA 8B	J1 LLAMA 70B
			JudgeBench			
Verbalized	0.205 / 0.188	0.152 / 0.145	0.103 / 0.107	0.114 / 0.113	0.274 / 0.263	0.156 / 0.155
Consistency	0.077 / 0.193	0.231 / 0.243	0.205 / 0.276	0.087 / 0.124	0.282 / 0.259	0.188 / 0.204
Majority	0.076 / 0.171	0.238 / 0.224	0.159 / 0.198	0.076 / 0.105	0.271 / 0.259	0.200 / 0.182
Probe	**0.072 / 0.119**	**0.062 / 0.077**	**0.055 / 0.059**	**0.037 / 0.048**	**0.122 / 0.135**	**0.075 / 0.074**
			Reward Bench			
Verbalized	**0.032 / 0.038**	**0.035** / 0.045	**0.023 / 0.027**	**0.041 / 0.043**	**0.009 / 0.007**	0.067 / 0.078
Consistency	0.039 / 0.071	0.039 / 0.052	0.033 / 0.035	0.234 / 0.251	0.078 / 0.078	**0.031** / 0.033
Majority	0.035 / 0.059	0.040 / **0.041**	0.034 / 0.035	0.232 / 0.249	0.076 / 0.075	**0.031 / 0.031**
Probe	0.120 / 0.145	0.065 / 0.067	0.092 / 0.106	0.111 / 0.128	0.166 / 0.190	0.128 / 0.140

Table 2: Out-of-distribution calibration (Kuiper/ECE, lower is better). Probes outperform baselines on JudgeBench but lag on RewardBench, where higher accuracy makes verbalized method's overconfidence appear well-calibrated.

the verdict. **Pairwise Judge with Likert Scale (PaL):** After generating the reasoning trace, the judge selects from $\{A \gg B, A > B, \text{Tie}, B > A, B \gg A\}$, where correctness is verified by checking if the winning response is ranked higher.

We intentionally use the prompting/scoring format that each judge family is designed and commonly evaluated with. In particular, the J1 Llama judge is trained to output pairwise verdicts, whereas J1 Qwen judges are trained to output pairwise scores (Whitehouse et al., 2025). To keep our comparisons aligned with the corresponding fine-tuned variants, we therefore use PaV for Llama-based judges and PaS for Qwen-based judges. For GPT-based judges, we use PaL, following the LM Arena hard prompt protocol (Tan et al., 2024; Li et al., 2024c) (prompts in Appendix A.12).

Evaluation metrics. We evaluate probe performance using the Kuiper statistic and Expected Calibration Error (ECE). We use Kuiper statistic as the main indicator of calibration as it is the most stable of the two metrics. We train and evaluate probes on three different train-test splits of the PPE datasets, reporting average metrics across splits. Standard deviations were near zero for all results.

ECE (Expected Calibration Error). ECE (Guo et al., 2017; Naeini et al., 2015) quantifies the difference between predicted confidence and actual accuracy by partitioning predictions into bins and measuring the gap within each bin. It is computed as:

$$\text{ECE} = \sum_{m=1}^{M} \frac{|B_m|}{n} |\text{acc}(B_m) - \text{conf}(B_m)|$$

where M is the number of bins, B_m is the set of samples in bin m, n is the total number of samples, $\text{acc}(B_m)$ is the accuracy in bin m, and $\text{conf}(B_m)$ is the average confidence in bin m. In our evaluation, we use a bin size of 0.1, as is common in the literature (Guo et al., 2017; Tao et al., 2025), to partition the 0–1 interval. However, it is a known weakness that ECE is sensitive to bin size (Roelofs et al., 2020; Nixon et al., 2019).

Kuiper statistic. The Kuiper metric (Tygert, 2025; Arrieta-Ibarra et al., 2022) measures maximum cumulative miscalibration by computing the spread between over- and under-confidence across all thresholds. Given sorted scores $S_1 < S_2 <$

$\cdots < S_n$ and binary labels $R_1, R_2, \ldots, R_n$, we calculate weighted cumulative differences $C_k = \frac{1}{n}\sum_{j=1}^{k}(R_j - S_j)W_j$ for $k = 1, 2, \ldots, n$:

$$\text{Kuiper} = \max_{0\leq k\leq n} C_k - \min_{0\leq k\leq n} C_k$$

where $C_0 = 0$ and W_j are weights. Lower values indicate better calibration. We use $W_j = S_j$ to prioritize calibration in high-confidence regions, reflecting production scenarios where high-confidence judgments are retained while low-confidence cases are delegated to more capable (and costlier) systems or human reviewers.

Baselines. We compare to three baselines: verbalized confidence, self-consistency, and majority. **Verbalized confidence** uses chain-of-thought explanations followed by a verdict and confidence score (Xiong et al., 2024; Tian et al., 2023). **Self-consistency** estimates confidence as the fraction of N independent samples voting for a response (Wang et al., 2022). **Majority** selects the most frequent verdict among N samples, with confidence as the proportion of times the majority answer appears (Wang et al., 2022). For verbalized confidence, we ablate prompts and score ranges (see A.13). For consistency and majority, we vary N (5–30) and temperature (0–1.5), with optimal results at $N = 10$, temperature $= 0.7$ (see ablations in A.9– A.10).

4 Results

Performance on Dense Models In-Distribution. Table 1 summarizes calibration results across dense model families and datasets. Our probes consistently achieve the best calibration, with the lowest Kuiper/ECE scores on both PPE Correctness and PPE Preference tasks. Llama-family models show the largest gains, with 70–92% improvements over multi-generation methods and 64–87% improvement over verbalized methods, across both prompted and finetuned variants. For Qwen models, probes deliver the best calibration on PPE Correctness and competitive results on PPE Preference, while requiring 10x less compute than multi-generation approaches. Calibration across task categories in the PPE dataset (see A.3) demonstrates that probes consistently outperform verbalized uncertainty and perform as well as or better than multi-generation methods, even when model accuracy varies across these subsets (i.e., between more difficult and easier tasks). We also see that

Method	LLAMA Scout	GPT-OSS 20B
PPE PREFERENCE		
Verbalized	0.1179 / 0.0892	0.2265 / 0.2233
Consistency	0.3533 / 0.3287	0.2045 / 0.1922
Majority	0.3506 / 0.3246	0.2056 / 0.1934
Probe	**0.0434 / 0.0582**	**0.1461 / 0.1713**
PPE CORRECTNESS		
Verbalized	0.1931 / 0.1597	0.1462 / 0.1520
Consistency	0.3212 / 0.2978	0.0932 / **0.0980**
Majority	0.3197 / 0.2940	0.0990 / 0.1034
Probe	**0.0643 / 0.0684**	**0.0689** / 0.1193
JudgeBench		
Verbalized	0.1388 / 0.1312	0.1276 / 0.1267
Consistency	0.3011 / 0.2918	0.0756 / 0.1097
Majority	0.2955 / 0.2842	0.0738 / **0.0997**
Probe	**0.0563 / 0.0766**	**0.0601** / 0.1263
RewardBench		
Verbalized	0.0742 / **0.0671**	0.0198 / **0.0191**
Consistency	0.2233 / 0.2037	**0.0165** / 0.0248
Majority	0.2178 / 0.1996	0.0166 / 0.0282
Probe	**0.0683** / 0.0879	0.1404 / 0.1917

Table 3: MoE model evaluation. Probes consistently outperform baselines for Llama Scout (109B) both in- and out-of-distribution. For GPT-OSS (20B), probes excel in-distribution and on JudgeBench but not on RewardBench, exhibiting the same conservative behavior on RewardBench as observed with dense models.

finetuned and non-finetuned variants achieve similar calibration despite different accuracy levels (see A.4), demonstrating that high accuracy does not guarantee good calibration.

Performance on Out of Distribution Datasets. Probes generalize effectively to unseen benchmarks (Table 2), consistently outperforming baselines on JudgeBench across all model families, demonstrating strong robustness under domain shift. However, on RewardBench, probe performance lags behind verbalized confidence for several models. This suggests that while probes excel on challenging tasks, their conservative calibration can underperform on easier datasets where models already achieve high accuracy.

Relationship Between Accuracy and Calibration. We attribute the relative underperformance of probes on RewardBench to their conservative confidence estimation. RewardBench exhibits higher overall accuracy across models, leading verbalized confidence (which tends to overestimate certainty) to perform deceptively well. In contrast, probes are designed to temper overconfidence, producing smoother and more cautious probability distributions.

By analyzing model accuracy at different confidence thresholds in Appendix A.2, we observe that

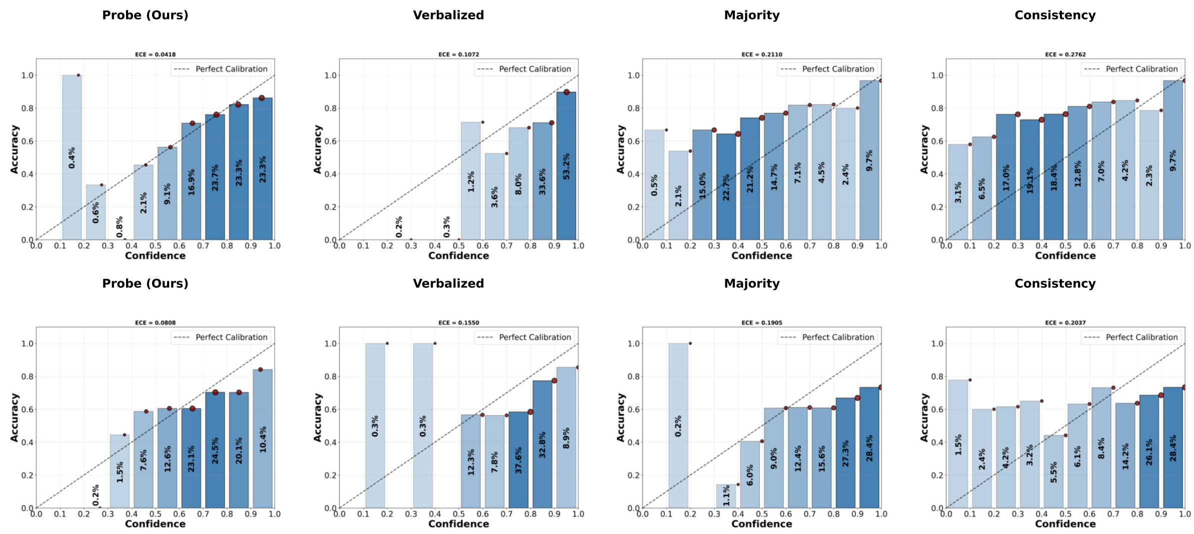


Figure 2: Reliability diagrams using 10 bins for Qwen 32B (Prompted Judge) and J1 LLAMA 70B (Finetuned Judge) on JudgeBench. We notice probes generally improve calibration. The color and percentage in each bar present the proportion of data samples in each bin. The verbalized method is generally overconfident, while multi-generation methods (consistency, majority) may be underconfident (Qwen) or overconfident (LLAMA) depending on model families. Similar trends are observed in finetuned and prompted variants of the judges, as shown in Appendix A.8.

probes yield higher accuracy among the most confident predictions, but fewer samples are assigned high confidence. Verbalized confidence, by contrast, spreads high confidence too liberally, leading to apparent calibration gains on easy datasets but poor reliability on harder ones.

In safety-critical applications such as medical advice, legal reasoning, or financial decision-making, where false positives are costly, this conservative behavior is highly desirable. Probes could present a favorable trade-off between accuracy and risk, offering more trustworthy confidence estimates even if slightly underconfident on easier datasets.

Verbalized confidence is often the second best option. Verbalized Confidence is Often the Second Best Option. Among our three baselines, verbalized confidence consistently performs second best after probes. In contrast, multigeneration methods can skew model confidence distributions—making models overconfident (LLaMA) or underconfident (Qwen, GPT-OSS) depending on the model family. We also note that prompting models to reason through uncertainty with hedging phrases yields more calibrated confidence (Appendix A.13), supporting the hypothesis that verbally expressing uncertainty improves calibration (Lin et al., 2022; Lyu et al., 2025).

Performance on MOE models. Table 3 extends our evaluation to Mixture-of-Experts (MoE) architectures, LLAMA Scout and GPT-OSS 20B. Probes consistently outperform all baselines across both in-distribution and out-of-distribution datasets, except on RewardBench, where GPT-OSS occasionally shows marginally better Kuiper values for the consistency method. However, probes remain competitive while needing 10× fewer inference calls than consistency, offering an efficiency gain.

Ablations with different losses and layers to train the probe. We conduct ablation studies in Appendix A.7 comparing different loss functions for probe training, evaluating binary cross-entropy, focal loss (Lin et al., 2017), and Brier score loss. We also experiment with probes at various layers (see A.6). Our results demonstrate that Brier score loss at the middle layer of models yields the most well-calibrated uncertainty estimates.

Single-Pass Selective Classification Baselines. In Appendix A.1 we compare against single-pass selective classification baselines—including Perplexity, Maximum Softmax Probability, Energy (Liu et al., 2020), Chain of Embedding (Wang et al., 2024), CoT-kinetics (Bi et al., 2025)—which use AUROC to measure how well uncertainty scores distinguish correct from incorrect answers. As shown, our probes achieve superior AUROC performance. Additionally, while some baselines require storing hidden states for every token ($\mathcal{O}$(tokens × layers × hidden_dim)), our probes only require

$\mathcal{O}$(layers $\times$ hidden_dim) memory.

5 Conclusion

We present a practical approach to calibrate LLM judges using linear probes on model representations, achieving strong calibration without additional training, multi-sample generation, or per-token storage, enabling plug-and-play, cost-effective deployment. Probe-based calibration significantly outperforms verbalized and multi-generation methods across models, architectures, and tasks, delivering order-of-magnitude computational savings and strong out-of-distribution generalization for industry-scale deployment.

6 Limitations

Future work could involve more investigation into interpreting which features are learned by the probes and further experimentation to optimize probe generalizability.

One limitation of this approach is that it requires access to the judge's hidden states in the middle layers and a labeled dataset with ground truth verdicts. The method doesn't work for domains where ground truth verdicts aren't available. Also, in this work, we train one probe for each judge model, but if the judge is retrained or finetuned, the probe may also need to be retrained as the information stored in the hidden states could change.

References

Meta AI. 2025. Llama-4-scout-17b-16e-instruct. https://huggingface.co/meta-llama/Llama-4-Scout-17B-16E-Instruct. Llama 4 Scout is a 17B parameter, 16 expert mixture-of-experts multimodal model. Licensed under the Llama 4 Community License Agreement.

AI@Meta. 2024. Llama 3 model card.

Lora Aroyo and Chris Welty. 2015. Truth is a lie: Crowd truth and the seven myths of human annotation. *AI Magazine*, 36(1):15–24.

Imanol Arrieta-Ibarra, Paman Gujral, Jonathan Tannen, Mark Tygert, and Cherie Xu. 2022. Metrics of calibration for probabilistic predictions. *Journal of Machine Learning Research*, 23(351):1–54.

Amos Azaria and Tom M Mitchell. 2023. The internal state of an llm knows when it's lying. *Findings of the Association for Computational Linguistics*.

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, and 1 others. 2022. Constitutional ai: Harmlessness from ai feedback. *arXiv preprint arXiv:2212.08073*.

Jinhe Bi, Danqi Yan, Yifan Wang, Wenke Huang, Haokun Chen, Guancheng Wan, Mang Ye, Xun Xiao, Hinrich Schuetze, Volker Tresp, and Yunpu Ma. 2025. Cot-kinetics: A theoretical modeling assessing lrm reasoning process. *arXiv preprint arXiv: 2505.13408*.

Collin Burns, Haotian Ye, Dan Klein, and Jacob Steinhardt. 2023. Discovering latent knowledge in language models without supervision. *International Conference on Learning Representations*.

Jiuhai Chen and Jonas Mueller. 2024. Quantifying uncertainty in answers from any language model and enhancing their trustworthiness. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5186–5200, Bangkok, Thailand. Association for Computational Linguistics.

Lingjiao Chen, Matei Zaharia, and James Zou. 2023. Frugalgpt: How to use large language models while reducing cost and improving performance. *arXiv preprint arXiv:2305.05176*.

Nuo Chen, Zhiyuan Hu, Qingyun Zou, Jiaying Wu, Qian Wang, Bryan Hooi, and Bingsheng He. 2025. Judgelrm: Large reasoning models as a judge. *Preprint*, arXiv:2504.00050.

Wei-Lin Chiang, Lianmin Zheng, Ying Sheng, Anastasios Nikolas Angelopoulos, Tianle Li, Dacheng Li, Banghua Zhu, Hao Zhang, Michael Jordan, Joseph E Gonzalez, and 1 others. 2024. Chatbot arena: An open platform for evaluating llms by human preference. In *Forty-first International Conference on Machine Learning*.

Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. 2017. Deep reinforcement learning from human preferences. *Advances in Neural Information Processing Systems*, 30.

Mehul Damani, Isha Puri, Stewart Slocum, Idan Shenfeld, Leshem Choshen, Yoon Kim, and Jacob Andreas. 2025. Beyond binary rewards: Training lms to reason about their uncertainty. *arXiv preprint arXiv:2507.16806*.

DeepSeek-AI. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Yann Dubois, Chen Xuechen Li, Rohan Taori, Tianyi Zhang, Ishaan Gulrajani, Jimmy Ba, Carlos Guestrin, Percy S Liang, and Tatsunori B Hashimoto. 2023. Alpacafarm: A simulation framework for methods that learn from human feedback. *Advances in Neural Information Processing Systems*, 36:30039–30069.

William Fedus, Barret Zoph, and Noam Shazeer. 2021. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *arXiv preprint arXiv: 2101.03961*.

Evan Frick, Tianle Li, Connor Chen, Wei-Lin Chiang, Anastasios Nikolas Angelopoulos, Jiantao Jiao, Banghua Zhu, Joseph Gonzalez, and I. Stoica. 2024. How to evaluate reward models for rlhf. *International Conference on Learning Representations*.

Leo Gao, John Schulman, and Jacob Hilton. 2022. Scaling laws for reward model overoptimization. *arXiv preprint arXiv: 2210.10760*.

W. Brier Glenn. 1950. Verification of forecasts expressed in terms of probability. *Monthly weather review*, 78(1):1–3.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. 2017. On calibration of modern neural networks. *arXiv preprint arXiv: 1706.04599*.

Yuheng Huang, Jiayang Song, Zhijie Wang, Huaming Chen, Likun Shen, Xin Lei, and Felix Juefei-Xu Wang. 2023. Look before you leap: An exploratory study of uncertainty measurement for large language models. *arXiv preprint arXiv:2307.10236*.

Ziwei Ji, Lei Yu, Yeskendir Koishekenov, Yejin Bang, Anthony Hartshorn, Alan Schelten, Cheng Zhang, Pascale Fung, and Nicola Cancedda. 2025. Calibrating verbal uncertainty as a linear feature to reduce hallucinations. *arXiv preprint arXiv: 2503.14477*.

Jaehun Jung, Faeze Brahman, and Yejin Choi. 2024. Trust or escalate: Llm judges with provable guarantees for human agreement. *arXiv preprint arXiv:2407.18370*.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, Scott Johnston, Sheer El-Showk, Andy Jones, Nelson Elhage, Tristan Hume, Anna Chen, Yuntao Bai, Sam Bowman, Stanislav Fort, and 17 others. 2022. Language models (mostly) know what they know. *arXiv preprint arXiv: 2207.05221*.

Sanyam Kapoor, Nate Gruver, Manley Roberts, Arka Pal, Samuel Dooley, Micah Goldblum, and Andrew Wilson. 2024. Calibration-tuning: Teaching large language models to know what they don't know. In *Proceedings of the 1st Workshop on Uncertainty-Aware NLP (UncertaiNLP 2024)*, pages 1–14, St Julians, Malta. Association for Computational Linguistics.

Seungone Kim, Juyoung Shin, Yejin Cho, Joel Jang, Shayne Longpre, Hwaran Lee, Sangdoo Yun, Sungdong Shin, Seongjin Kim, James Thorne, and 1 others. 2024a. Prometheus: Inducing fine-grained evaluation capability in language models. *arXiv preprint arXiv:2310.08491*.

Seungone Kim, Juyoung Shin, Yejin Kwak, Yujin Zhao, Jungwoo Jang, Ahra Hwang, Geon Kim, Hyeonbin Kim, Junho Yoon, Hahyeon Park, and 1 others. 2024b. Prometheus 2: An open source language model specialized in evaluating other language models. *arXiv preprint arXiv:2405.01535*.

Hannah Rose Kirk, Alexander Whitefield, Paul Rottger, Andrew M Bean, Katerina Margatina, Rafael Mosquera-Gomez, Juan Ciro, Max Bartolo, Adina Williams, He He, and 1 others. 2024. The prism alignment dataset: What participatory, representative and individualised human feedback reveals about the subjective and multicultural alignment of large language models. *Advances in Neural Information Processing Systems*, 37:105236–105344.

Jannik Kossen, Jiatong Han, Muhammed Razzak, Lisa Schut, Shreshth Malik, and Yarin Gal. 2024. Semantic entropy probes: Robust and cheap hallucination detection in llms. *arXiv preprint arXiv: 2406.15927*.

Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. 2023. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. *International Conference on Learning Representations*.

Nathan Lambert, Valentina Pyatkin, Jacob Morrison, LJ Miranda, Bill Yuchen Lin, Khyathi Chandu, Nouha Dziri, Sachin Kumar, Tom Zick, Yejin Choi, Noah A. Smith, and Hannaneh Hajishirzi. 2024. Rewardbench: Evaluating reward models for language modeling. *arXiv preprint arXiv: 2403.13787*.

Harrison Lee, Samrat Phatale, Hassan Mansoor, Thomas Mesnard, Johan Ferret, Kellie Lu, Colton Bishop, Ethan Hall, Victor Carbune, Abhinav Rastogi, and Sushant Prakash. 2023. Rlaif vs. rlhf: Scaling reinforcement learning from human feedback with ai feedback. *arXiv preprint arXiv: 2309.00267*.

Junlong Li, Shichao Sun, Weizhe Yuan, Run-Ze Fan, Hai Zhao, and Pengfei Liu. 2024a. Generative judge for evaluating alignment. *arXiv preprint arXiv:2310.05470*.

Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. 2024b. Inference-time intervention: Eliciting truthful answers from a language model. *Advances in Neural Information Processing Systems*, 36.

Tianle Li, Wei-Lin Chiang, Evan Frick, Lisa Dunlap, Tianhao Wu, Banghua Zhu, Joseph E. Gonzalez, and Ion Stoica. 2024c. From crowdsourced data to high-quality benchmarks: Arena-hard and benchbuilder pipeline. *arXiv preprint arXiv: 2406.11939*.

Yibo Li, Miao Xiong, Jiaying Wu, and Bryan Hooi. 2025a. Conftuner: Training large language models to express their confidence verbally. *arXiv preprint arXiv: 2508.18847*.

Zhuohang Li, Xiaowei Li, Chengyu Huang, Guowang Li, Katayoon Goshvadi, Bo Dai, Dale Schuurmans, Paul Zhou, Hamid Palangi, Yiwen Song, Palash Goyal, Murat Kantarcioglu, Bradley A. Malin, and Yuan Xue. 2025b. Judging with confidence: Calibrating autoraters to preference distributions. *arXiv preprint arXiv: 2510.00263*.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. Teaching models to express their uncertainty in words. *arXiv preprint arXiv: 2205.14334*.

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

Weitang Liu, Xiaoyun Wang, John D. Owens, and Yixuan Li. 2020. Energy-based out-of-distribution detection. *arXiv preprint arXiv: 2010.03759*.

Qing Lyu, Kumar Shridhar, Chaitanya Malaviya, Li Zhang, Yanai Elazar, Niket Tandon, Marianna Apidianaki, Mrinmaya Sachan, and Chris Callison-Burch. 2025. Calibrating large language models with sample consistency. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 19260–19268.

Potsawee Manakul, Adian Liusie, and Mark JF Gales. 2023. Selfcheckgpt: Zero-resource black-box hallucination detection for generative large language models. *arXiv preprint arXiv:2303.08896*.

Mohammad Pezeshk Naeini, Gregory Cooper, and Milos Hauskrecht. 2015. Obtaining well calibrated probabilities using bayesian binning. In *Proceedings of the AAAI conference on artificial intelligence (AAAI)*, volume 29, pages 2901–2907.

Yixin Nie, Xiang Zhou, and Mohit Bansal. 2020. What can we learn from collective human opinions on natural language inference data? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9131–9143, Online. Association for Computational Linguistics.

Jeremy Nixon, Mike Dusenberry, Ghassen Jerfel, Timothy Nguyen, Jeremiah Liu, Linchuan Zhang, and Dustin Tran. 2019. Measuring calibration in deep learning. *arXiv preprint arXiv: 1904.01685*.

OpenAI. 2024. Learning to reason with llms. *OpenAI Blog*.

OpenAI. 2025. gpt-oss-120b gpt-oss-20b model card. *Preprint*, arXiv:2508.10925.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, and 1 others. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744.

Ellie Pavlick and Tom Kwiatkowski. 2019. Inherent disagreements in human textual inferences. *Transactions of the Association for Computational Linguistics*, 7:677–694.

Benjamin Plaut, Nguyen X. Khanh, and Tu Trinh. 2024. Probabilities of chat llms are miscalibrated but still predict correctness on multiple-choice qa. *arXiv preprint arXiv: 2402.13213*.

Qwen Team. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

Bhaktipriya Radharapu, Manon Revel, Megan Ung, Sebastian Ruder, and Adina Williams. 2025. Arbiters of ambivalence: Challenges of using llms in no-consensus tasks. *arXiv preprint arXiv: 2505.23820*.

Rebecca Roelofs, Nicholas Cain, Jonathon Shlens, and Michael C. Mozer. 2020. Mitigating bias in calibration error estimation. *arXiv preprint arXiv: 2012.08668*.

Candace Ross, Melissa Hall, Adriana Romero-Soriano, and Adina Williams. 2024. What makes a good metric? evaluating automatic metrics for text-to-image consistency. In *First Conference on Language Modeling*.

Gaurang Sriramanan, Siddhant Bharti, Vinu Sankar Sadasivan, Shoumik Saha, Priyatham Kattakinda, and Soheil Feizi. 2024. LLM-check: Investigating detection of hallucinations in large language models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Zeerak Talat, Hagen Blix, Josef Valvoda, Maya Indira Ganesh, Ryan Cotterell, and Adina Williams. 2022. On the machine learning of ethical judgments from natural language. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 769–779, Seattle, United States. Association for Computational Linguistics.

Sijun Tan, Siyuan Zhuang, Kyle Montgomery, William Y. Tang, Alejandro Cuadron, Chenguang Wang, R. Popa, and Ion Stoica. 2024. Judgebench: A benchmark for evaluating llm-based judges. *International Conference on Learning Representations*.

Linwei Tao, Yi-Fan Yeh, Minjing Dong, Tao Huang, Philip Torr, and Chang Xu. 2025. Revisiting uncertainty estimation and calibration of large language models. *arXiv preprint arXiv: 2505.23854*.

Katherine Tian, Eric Mitchell, Allan Zhou, Archit Sharma, Rafael Rafailov, Huaxiu Yao, Chelsea Finn, and Christopher Manning. 2023. Just ask for calibration: Strategies for eliciting calibrated confidence scores from language models fine-tuned with human feedback. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5433–5442, Singapore. Association for Computational Linguistics.

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

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, and 49 others. 2023. Llama 2: Open foundation and fine-tuned chat models. *Preprint*, arXiv:2307.09288.

Mark Tygert. 2025. Calibration and bias in algorithms, data, and models: a tutorial on metrics and plots for measuring calibration, bias, fairness, reliability, and robustness.

Peiyi Wang, Lei Li, Liang Chen, Dawei Zhu, Binghuai Lin, Yunbo Cao, Qi Liu, Tianyu Liu, and Zhifang Sui. 2023. Large language models are not fair evaluators. *arXiv preprint arXiv:2305.17926*.

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

Yiming Wang, Pei Zhang, Baosong Yang, Derek F Wong, and Rui Wang. 2024. Latent space chain-of-embedding enables output-free llm self-evaluation. *arXiv preprint arXiv:2410.13640*.

Chenxi Whitehouse, Tianlu Wang, Ping Yu, Xian Li, Jason Weston, Ilia Kulikov, and Swarnadeep Saha. 2025. J1: Incentivizing thinking in llm-as-a-judge via reinforcement learning. *arXiv preprint arXiv: 2505.10320*.

Johnathan Xie, Annie S. Chen, Yoonho Lee, Eric Mitchell, and Chelsea Finn. 2024a. Calibrating language models with adaptive temperature scaling. *arXiv preprint arXiv: 2409.19817*.

Zhihui Xie, Jizhou Guo, Tong Yu, and Shuai Li. 2024b. Calibrating reasoning in language models with internal consistency. In *Advances in Neural Information Processing Systems*, volume 37, pages 114872–114901. Curran Associates, Inc.

Miao Xiong, Zhiyuan Hu, Xinyang Lu, YIFEI LI, Jie Fu, Junxian He, and Bryan Hooi. 2024. Can LLMs express their uncertainty? an empirical evaluation of confidence elicitation in LLMs. In *The Twelfth International Conference on Learning Representations*.

Wenjun Zeng, Yuchi Liu, Ryan Mullins, Ludovic Peran, Joe Fernandez, Hamza Harkous, Karthik Narasimhan, Drew Proud, Piyush Kumar, Bhaktipriya Radharapu, and 1 others. 2024. Shieldgemma: Generative ai content moderation based on gemma. *arXiv preprint arXiv:2407.21772*.

Anqi Zhang, Yulin Chen, Jane Pan, Chen Zhao, Aurojit Panda, Jinyang Li, and He He. 2025a. Reasoning models know when they're right: Probing hidden states for self-verification. *arXiv preprint arXiv: 2504.05419*.

Lily Hong Zhang, Smitha Milli, Karen Jusko, Jonathan Smith, Brandon Amos, Wassim Bouaziz, Manon Revel, Jack Kussman, Yasha Sheynin, Lisa Titus, and 1 others. 2025b. Cultivating pluralism in algorithmic monoculture: The community alignment dataset. *arXiv preprint arXiv:2507.09650*.

Michael JQ Zhang, Zhilin Wang, Jena D. Hwang, Yi Dong, Olivier Delalleau, Yejin Choi, Eunsol Choi, Xiang Ren, and Valentina Pyatkin. 2024. Diverging preferences: When do annotators disagree and do models know? *arXiv preprint arXiv: 2410.14632*.

Tony Z Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021. Calibrate before use: Improving few-shot performance of language models. *International Conference on Machine Learning*, pages 12697–12706.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, and 1 others. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in neural information processing systems*, 36:46595–46623.

Han Zhou, Xingchen Wan, Lev Proleev, Diana Mincu, Jilin Chen, Katherine A Heller, and Subhrajit Roy. 2024a. Batch calibration: Rethinking calibration for in-context learning and prompt engineering. In *The Twelfth International Conference on Learning Representations*.

Hongli Zhou, Hui Huang, Yunfei Long, Bing Xu, Conghui Zhu, Hailong Cao, Muyun Yang, and Tiejun Zhao. 2024b. Mitigating the bias of large language model evaluation. In *Proceedings of the 23rd Chinese National Conference on Computational Linguistics (Volume 1: Main Conference)*, pages 1310–1319, Taiyuan, China. Chinese Information Processing Society of China.

Lianghui Zhu, Xinggang Wang, and Xinlong Wang. 2025. Judgelm: Fine-tuned large language models are scalable judges. *Preprint*, arXiv:2310.17631.

Andy Zou, Long Phan, Sarah Chen, James Campbell, Phillip Guo, Richard Ren, Alexander Pan, Xuwang Yin, Mantas Mazeika, Ann-Kathrin Dombrowski, and 1 others. 2023. Representation engineering: A top-down approach to ai transparency. *arXiv preprint arXiv:2310.01405*.

A Appendix

A.1 Comparison with Other Single Pass Methods

Selective classification evaluates how effectively uncertainty scores distinguish between correct and incorrect predictions. The most widely used metric for this is AUROC, which quantifies the likelihood that a correct prediction will have a lower uncertainty score than an incorrect one. An AUROC of 0.5 indicates no discriminative ability (equivalent to random guessing), whereas values approaching 1.0 reflect strong discriminative performance.

Note that some of these methods like (Wang et al., 2024; Bi et al., 2025) outputs are not constrained to the [0, 1] range; instead, they yield arbitrary scores that serve as relative signals—higher values may indicate greater correctness, but these scores are not interpretable as probabilities.

We evaluate various single-pass uncertainty estimation methods on the JudgeBench dataset. These methods provide signals for detecting answer correctness, which we measure using AUROC scores.

Table 4 shows the performance comparison across different model architectures.

Method	LLaMA 8B	LLaMA 70B	J1 LLaMA 8B	J1 LLaMA 70B	J1 QWEN 32B	GPT-OSS OSS 20B	QWEN 32B
MaxProb	46.77	51.48	50.67	59.61	64.27	43.54	51.47
Perplexity	46.44	51.31	50.81	59.62	63.88	43.15	51.20
Entropy	46.81	51.08	50.85	59.60	64.33	46.82	50.79
TempScl	46.54	51.55	50.72	59.60	63.74	42.11	51.53
Energy	47.99	50.07	53.34	50.00	49.34	**71.53**	67.79
CoER	52.06	47.81	49.98	49.82	60.36	69.12	46.25
CoEC	49.99	52.24	50.56	59.13	60.26	68.46	45.99
CoTK	46.69	51.11	50.87	59.98	62.90	63.72	48.80
Verbalised	46.83	57.95	53.50	62.04	68.86	64.47	69.20
Sem. Entropy	**54.71**	58.13	54.19	60.63	66.87	64.42	65.63
Probe	52.38	**58.14**	**55.36**	**63.17**	**69.05**	62.03	**71.17**

Table 4: AUROC performance on JudgeBench, comparison of single-pass uncertainty estimation methods across model architectures. Higher values indicate better performance. Bold values indicate best performance per model.

A.2 Accuracy at Different Thresholds

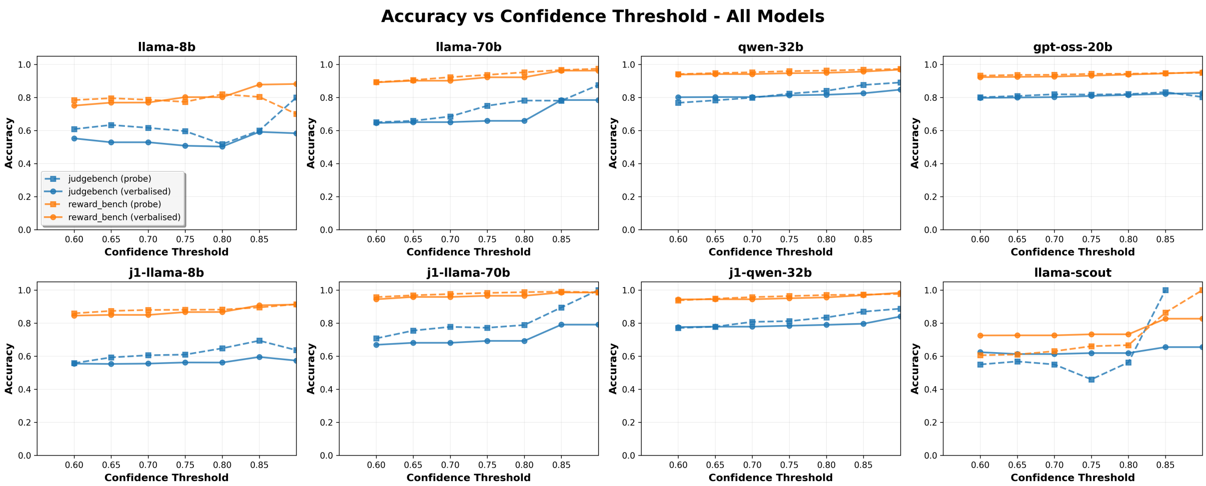


Figure 3: Accuracy of models at different confidence thresholds for Probe and Verbalized uncertainity estimation methods.

In Figure 3, we analyze the accuracy of models at different confidence thresholds to better understand the conservative behavior of our probe-based uncertainty estimation compared to verbalized confidence. As the confidence threshold increases, the accuracy of the probe method consistently rises and achieves higher peak accuracy among the most confident predictions across all evaluated models, including both dense

(LLaMA, Qwen) and MoE (GPT-OSS, LLaMA-Scout) architectures. However, the probe assigns high confidence to fewer samples overall, resulting in a more cautious and smoother probability distribution. In contrast, verbalized confidence tends to spread high confidence too liberally, which can lead to deceptive calibration gains on easier datasets (such as RewardBench) where overall model accuracy is already high, but results in poor reliability on harder tasks. This conservative nature of the probe is particularly advantageous in safety-critical applications where minimizing false positives is essential, as it provides a more trustworthy and reliable measure of uncertainty even if it appears slightly underconfident on simpler tasks.

A.3 Additional Results –Performance on various PPE Correctness Benchmarks

In Table 5 and Table 6, we present the results on the different subsets of PPE correctness dataset to see if the performance varies by domain. We see that the probe performs well across the board.

Method	LLAMA 8B	LLAMA 70B	QWEN 32B	J1 QWEN 32B	J1 LLAMA 8B	J1 LLAMA 70B
			PPE Correctness Gpqa			
Verbalized	0.1846 / 0.1694	0.1431 / 0.1365	0.1411 / 0.1369	0.1444 / 0.1409	0.2604 / 0.2508	0.1109 / 0.1049
Consistency	0.1824 / 0.1857	0.2774 / 0.2653	0.1136 / 0.1328	0.0635 / 0.1075	0.3078 / 0.2872	0.2270 / 0.2323
Majority	0.1792 / 0.1734	0.2790 / 0.2623	0.1079 / 0.1245	**0.0490** / 0.0809	0.3086 / 0.2851	0.2325 / 0.2175
Probe	**0.0457 / 0.0469**	**0.0746 / 0.0717**	**0.0682 / 0.0673**	0.0581 / **0.0599**	**0.0782 / 0.0767**	**0.0404 / 0.0444**
			PPE Correctness Ifeval			
Verbalized	0.1738 / 0.1690	0.1570 / 0.1440	0.1738 / 0.1729	0.1615 / 0.1603	0.2280 / 0.2191	0.0788 / 0.0732
Consistency	0.1781 / 0.1865	0.2244 / 0.2235	0.0735 / 0.0985	0.0749 / 0.1201	0.3729 / 0.3603	0.1920 / 0.1836
Majority	0.1810 / 0.1704	0.2301 / 0.2180	0.0708 / 0.0904	0.0626 / 0.0986	0.3771 / 0.3583	0.1946 / 0.1796
Probe	**0.0368 / 0.0437**	**0.0161 / 0.0237**	**0.0596 / 0.0652**	**0.0541 / 0.0586**	**0.0239 / 0.0343**	**0.0393 / 0.0393**
			PPE Correctness Math			
Verbalized	0.1456 / 0.1447	0.1349 / 0.1374	0.0336 / 0.0362	0.0355 / 0.0382	0.1259 / 0.1241	0.0345 / **0.0341**
Consistency	0.0501 / 0.0939	0.1025 / 0.1223	0.2789 / 0.2950	0.2195 / 0.2424	0.0843 / 0.0820	0.0523 / 0.0905
Majority	0.0509 / 0.0725	0.1098 / 0.1041	0.2747 / 0.2903	0.2153 / 0.2365	0.0841 / 0.0796	0.0384 / 0.0662
Probe	**0.0443 / 0.0459**	**0.0340 / 0.0383**	**0.0336 / 0.0316**	**0.0353 / 0.0349**	**0.0155 / 0.0165**	**0.0310 / 0.0342**
			PPE Correctness Mbpp			
Verbalized	0.2506 / 0.2415	0.1905 / 0.1829	0.2280 / 0.2333	0.2196 / 0.2265	0.3073 / 0.2995	0.1331 / 0.1291
Consistency	0.2573 / 0.2364	0.2995 / 0.2762	**0.0255** / 0.0412	0.0690 / 0.0976	0.3943 / 0.3782	0.1538 / 0.1606
Majority	0.2511 / 0.2276	0.2929 / 0.2743	0.0293 / **0.0365**	0.0747 / 0.0751	0.3947 / 0.3829	0.1578 / 0.1452
Probe	**0.0486 / 0.0471**	**0.0445 / 0.0644**	0.0366 / 0.0379	**0.0215 / 0.0401**	**0.0335 / 0.0406**	**0.0322 / 0.0551**
			PPE Correctness Mmlu			
Verbalized	0.1546 / 0.1469	0.0607 / 0.0608	0.0470 / 0.0475	0.0357 / 0.0405	0.1779 / 0.1766	0.0430 / 0.0422
Consistency	0.1209 / 0.1533	0.1586 / 0.1697	0.1869 / 0.2000	0.1439 / 0.1662	0.2157 / 0.2032	0.1046 / 0.1156
Majority	0.1235 / 0.1315	0.1674 / 0.1603	0.1843 / 0.1994	0.1329 / 0.1506	0.2164 / 0.2063	0.1077 / 0.1015
Probe	**0.0386 / 0.0406**	**0.0287 / 0.0304**	**0.0202 / 0.0277**	**0.0291 / 0.0387**	**0.0271 / 0.0312**	**0.0355 / 0.0413**

Table 5: Performance on PPE Correctness subsets (dense models)

Method	LLAMA Scout	GPT OSS 20B
PPE Correctness Gpqa		
Verbalized	0.2631 / 0.2452	0.1505 / 0.1534
Consistency	0.2684 / 0.2448	**0.1084** / 0.1241
Majority	0.2714 / 0.2443	0.1110 / **0.1176**
Probe	**0.0652 / 0.0730**	0.1301 / 0.1582
PPE Correctness Ifeval		
Verbalized	0.1168 / 0.0904	0.1584 / 0.1627
Consistency	0.3413 / 0.3186	**0.0514** / 0.0625
Majority	0.3410 / 0.3112	0.0517 / **0.0558**
Probe	**0.0673 / 0.0803**	0.0657 / 0.1049
PPE Correctness Math		
Verbalized	0.2546 / 0.2505	**0.0091 / 0.0122**
Consistency	0.2009 / 0.2066	0.0113 / 0.0205
Majority	0.2061 / 0.2016	0.0115 / 0.0206
Probe	**0.0587 / 0.0653**	0.0648 / 0.0790
PPE Correctness Mbpp		
Verbalized	**0.0289 / 0.0161**	0.3545 / 0.3645
Consistency	0.4737 / 0.4660	0.2765 / 0.2759
Majority	0.4748 / 0.4669	0.2908 / 0.3009
Probe	0.0453 / 0.0514	**0.0472 / 0.0493**
PPE Correctness Mmlu		
Verbalized	0.2350 / 0.2186	0.0882 / 0.0894
Consistency	0.2760 / 0.2559	**0.0659** / 0.0754
Majority	0.2750 / 0.2508	0.0691 / **0.0711**
Probe	**0.0643 / 0.0684**	0.0689 / 0.1193

Table 6: Performance on PPE Correctness subsets (MOE models)

A.4 Accuracy of Models on various Datasets

We study calibration on models with varying difficulty for the models. RewardBench and PPE Math are the best performing across models.

PPE MBPP(Coding) and and PPE IfEval and PPE Preference are among the harder datasets across models.

Model	JudgeBench	PPE Preference	PPE GPQA	PPE IFEval	PPE MBPP	PPE MMLU	PPE Math	Reward Bench
Average	0.670	0.627	0.633	0.624	0.583	0.733	0.793	0.856
	±0.108	±0.051	±0.078	±0.074	±0.073	±0.119	±0.157	±0.117
LLAMA-8B	0.549	0.577	0.548	0.553	0.508	0.589	0.606	0.743
J1-LLAMA-8B	0.549	0.605	0.560	0.545	0.537	0.654	0.716	0.844
LLAMA-70B	0.644	0.660	0.624	0.605	0.599	0.750	0.694	0.890
J1-LLAMA-70B	0.669	0.665	0.656	0.675	0.651	0.794	0.869	0.944
QWEN-32B	0.797	0.672	0.723	0.684	0.673	0.849	0.956	0.938
J1-QWEN-32B	0.776	0.664	0.707	0.695	0.666	0.854	0.952	0.943
GPT-OSS-20B	0.797	0.639	0.711	0.705	0.533	0.821	0.953	0.923
LLAMA-SCOUT	0.578	0.531	0.532	0.530	0.501	0.555	0.598	0.621

Table 7: Verbalized method accuracy across different models and datasets. Values show mean accuracy with standard deviation across models. RewardBench is the easiest for the models overall followed by PPE Math, PPE MBPP and PPE Preference are the hardest.

A.5 Training the probe

We use the HuggingFace transformers library to train the probes. The probe architecture has one linear layer with input dimension equal to the judge's hidden state dimension and output dimension of 1 (this is essentially linear regression). We train on 4000 examples randomly selected from the PPE datasets, with 2000 from all of the PPE correctness datasets and 2000 from the PPE preference dataset. The rest of the data ($\approx$10K examples) is used for in-distribution evaluation. We use a learning rate of 10^{-4}, weight decay of 0.01, batch size of 4, and train for 10 epochs. We experiment with two loss functions, mean squared error (MSE) or Brier Score Loss and Focal loss(FL), which is a reweighted binary cross entropy. The formulations of the two loss functions where p is the prediction and y is the label are as follows:

$$\mathrm{MSE}(p, y) = (y - p)^2$$

$$\mathrm{FL}(p, y) = -\alpha\, y\, (1 - p)^{\gamma} \log(p)$$
$$-(1 - \alpha)(1 - y)p^{\gamma} \log(1 - p)$$

We take the probe's logits as the confidence scores, after thresholding them to be between 0 and 1.

A.6 Layer Ablation Study

We experiment with training the probes on different hidden state layers to determine the optimal layer for uncertainty estimation. We find that middle layers tend to perform better, and use this to inform our ultimate choice of which layers to report results on. Specifically, we choose layer 8 for GPT OSS 20B, layer 16 for the 8B and 32B models and LLaMA Scout, and layer 32 for the 70B models. Figure 4 shows the probe performance across different layers.

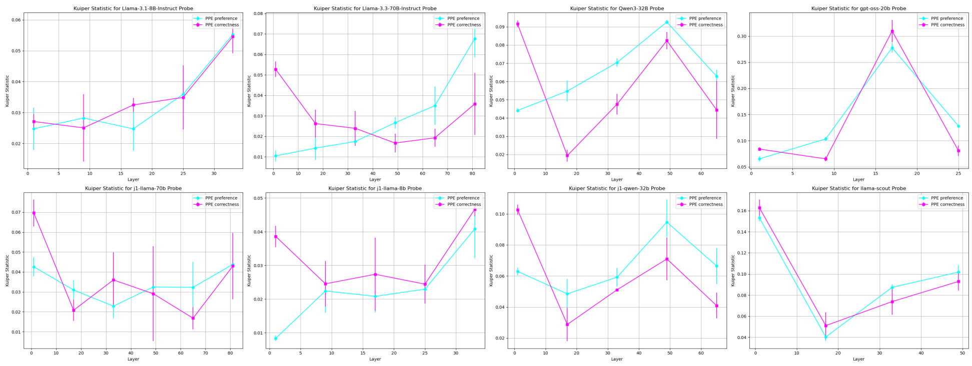


Figure 4: **Probes trained on the middle layers perform best.** Probe performance by transformer layer. Probes perform better when trained on middle layers, with performance typically peaking around layers 16-64 depending on model size.

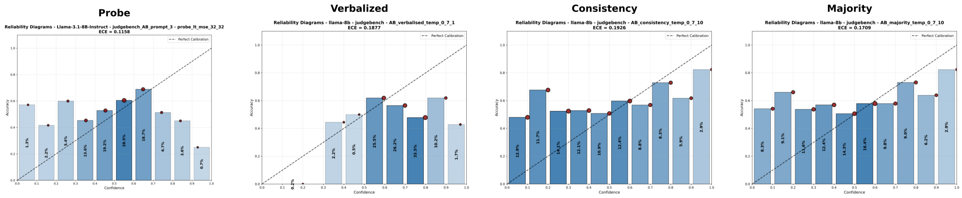
A.7 Loss Function Ablation

We compare different loss functions for probe training, including focal loss with various hyperparameters α and γ, against MSE loss. Table 8 presents the results across different datasets and models.

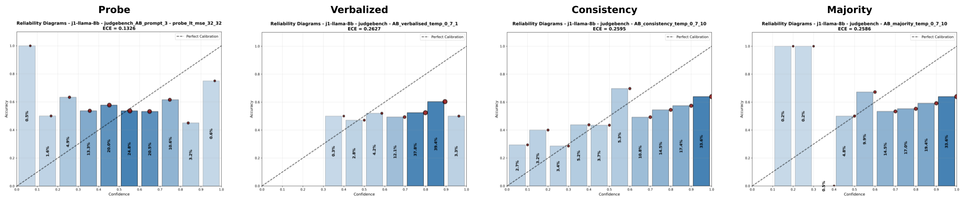
Note focal loss with $\gamma = 0$ is the binary-cross entropy loss.

A.8 Reliability Diagrams for all models on JudgeBench

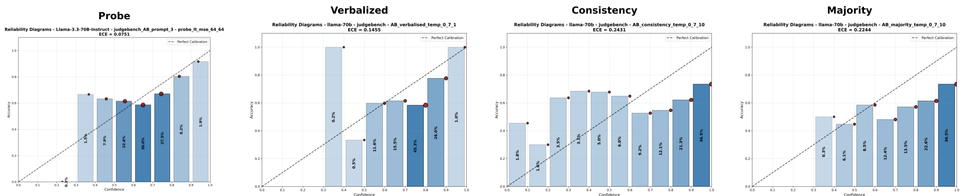
We notice that verbalized confidence is generally overconfident across models. Multi-generation methods like majority or consistency are overconfident for the LLaMA family and the J1 LLaMa variants both across dense and MoE models. Similarly, for GPT-OSS-20B, models tend to get overconfident with multi-generation methods. However, for Qwen 32B and J1-Qwen 32B, consistency and majority methods make the model underconfident.



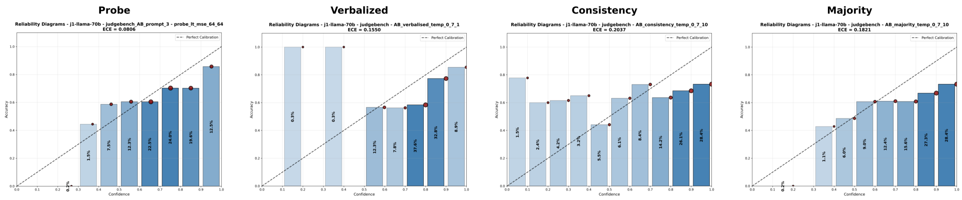
(a) LLAMA 8B



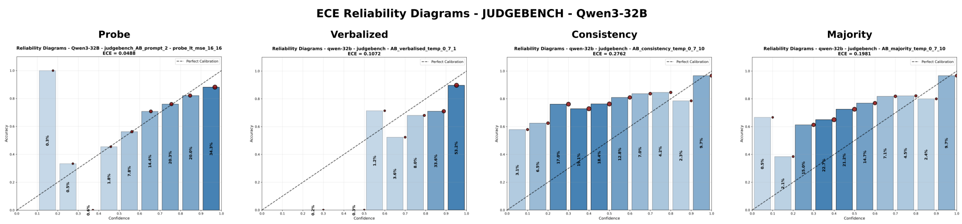
(b) J1 LLAMA 8B



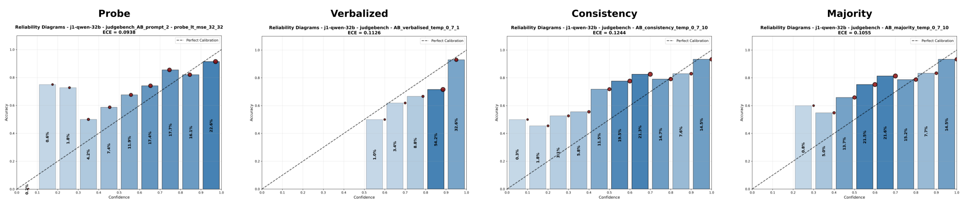
(c) LLAMA 70B



(d) J1 LLAMA 70B



(e) QWEN 32B

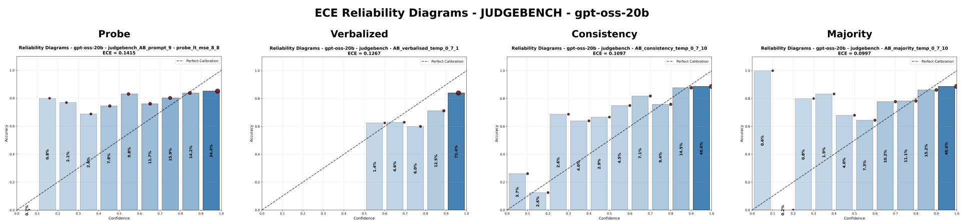


(f) J1 QWEN 32B

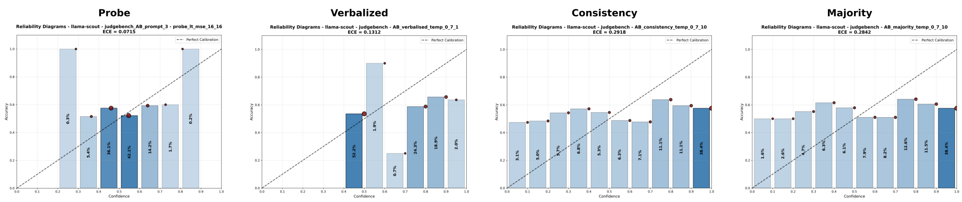
Figure 5: Reliability plots for Dense Models

Loss Function	LLaMA 70B	QWEN 32B	GPT OSS 20B
	PPE PREFERENCE (Kuiper / ECE)		
Focal: $\alpha = 0.25, \gamma = 10$	0.5528 / 0.0986	0.1875 / 0.1266	**0.1441 / 0.1138**
Focal: $\alpha = 0.5, \gamma = 0$	0.1177 / 0.1644	0.1900 / 0.2111	0.2384 / 0.1869
Focal: $\alpha = 0.75, \gamma = 10$	0.2600 / 0.2522	0.2431 / 0.2402	0.2669 / 0.2452
Focal: $\alpha = 0.75, \gamma = 20$	0.0421 / 0.0629	0.1339 / 0.1629	0.2439 / 0.2326
MSE	**0.0303 / 0.0367**	**0.0765 / 0.1031**	0.1676 / 0.1567
	PPE CORRECTNESS (Kuiper / ECE)		
Focal: $\alpha = 0.25, \gamma = 10$	0.5865 / 0.2651	0.1433 / 0.1988	**0.0738 / 0.1252**
Focal: $\alpha = 0.5, \gamma = 0$	0.1364 / 0.1504	0.1221 / 0.1275	0.1270 / 0.1291
Focal: $\alpha = 0.75, \gamma = 10$	0.2234 / 0.2236	0.1380 / 0.1358	0.1440 / 0.1398
Focal: $\alpha = 0.75, \gamma = 20$	0.0500 / 0.0513	0.0821 / 0.0940	0.1365 / 0.1378
MSE	**0.0323 / 0.0367**	**0.0559 / 0.0814**	0.0999 / 0.1325
	JUDGEBENCH (Kuiper / ECE)		
Focal: $\alpha = 0.25, \gamma = 10$	0.5991 / 0.1400	0.1649 / 0.2177	0.1154 / 0.1671
Focal: $\alpha = 0.5, \gamma = 0$	0.1904 / 0.2176	0.1551 / 0.1526	0.1602 / 0.1619
Focal: $\alpha = 0.75, \gamma = 10$	0.2907 / 0.2823	0.1891 / 0.1936	0.1677 / 0.1688
Focal: $\alpha = 0.75, \gamma = 20$	0.0737 / 0.0967	0.1165 / 0.1220	0.1603 / 0.1639
MSE	**0.0546 / 0.0618**	**0.0682 / 0.0837**	**0.0986 / 0.1457**
	REWARD BENCH (Kuiper / ECE)		
Focal: $\alpha = 0.25, \gamma = 10$	0.6005 / 0.4541	0.1746 / 0.2379	0.1452 / 0.1992
Focal: $\alpha = 0.5, \gamma = 0$	0.0365 / 0.0652	**0.0316** / 0.0800	**0.0361** / 0.0962
Focal: $\alpha = 0.75, \gamma = 10$	0.0828 / 0.0852	0.0409 / **0.0468**	0.0446 / **0.0742**
Focal: $\alpha = 0.75, \gamma = 20$	**0.0348 / 0.0443**	0.0367 / 0.0700	0.0456 / 0.0944
MSE	0.1165 / 0.1198	0.1232 / 0.1475	0.1427 / 0.1914

Table 8: Ablations with different hyperparameters for focal loss. Note focal loss is equal to binary cross entropy loss at $\gamma = 0$



(a) GPT OSS 20B



(b) LLAMA SCOUT

Figure 6: Reliability plots for MoE Models

A.9 Temperature and Calibration Performance

We investigate the relationship between sampling temperature and calibration performance. Figure 7 shows how ECE and Kuiper statistics vary with temperature settings across different models.

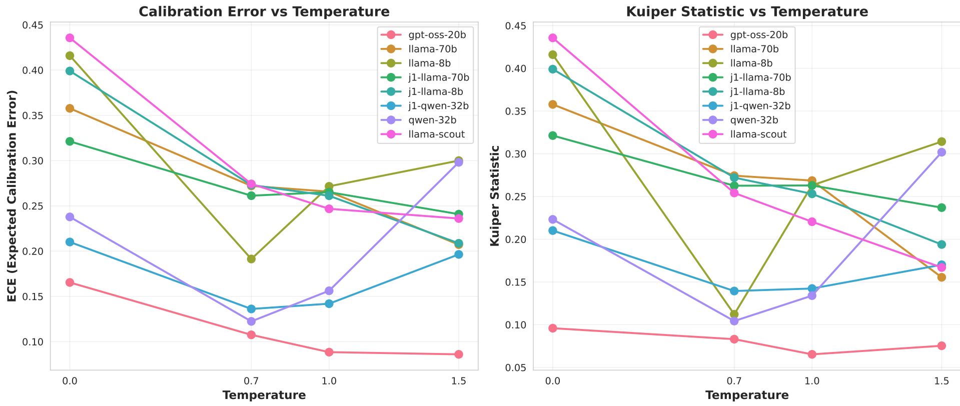


Figure 7: ECE and Kuiper statistics as a function of temperature. We observe the lowest ECE and Kuiper values for most models at temperature 0.7, with minimal gains beyond this point. Performance is worst at temperature 0, improves at 0.7, and begins to decrease as temperature increases further.

A.10 Number of Runs and Consistency-Based Calibration

We examine how the number of consistency sampling runs affects calibration performance. Figure 8 demonstrates the relationship between the number of runs and calibration metrics.

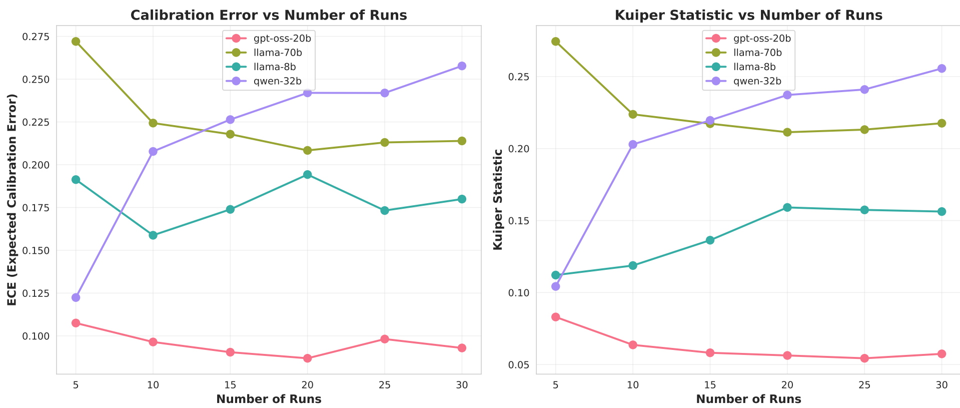


Figure 8: ECE and Kuiper statistics as a function of the number of consistency sampling runs. We observe the lowest ECE and Kuiper values for most models at 10 runs, with minimal gains beyond this point. Models tend to become overconfident as the number of runs increases.

A.11 Prompts used for Multigeneration (Consistency, Majority) baselines

Pairwise Judge with Verdicts

Prompt:
You are given a user question and two responses from two AI assistants. Your task is to act as an impartial judge and evaluate which response better follows the user's instructions and provides a higher-quality answer.
First, provide your reasoning within <think> and </think> tags. This should include your evaluation criteria for a high-quality response, a detailed comparison of the two responses, and when helpful, a reference answer as part of your evaluation. Be explicit in your thought process, referencing your criteria and explaining how each response aligns with or deviates from them.
If you are not entirely sure of the correctness of your verdict, you should verbally express your uncertainty in your thought process.
Avoid any position biases and ensure that the order in which the responses were presented does not influence your decision. Do not allow the length of the responses to influence your evaluation. Do not favor certain names of the assistants. Be as objective as possible.
Finally, provide your verdict within <answer> and </answer> tags, strictly following this format: - <answer> [[A]] </answer> if Assistant A is better - <answer> [[B]] </answer> if Assistant B is better
Below are the user's question and the two responses:

User Question] {input}
[The Start of Assistant A's Answer]
{response_1}
[The End of Assistant A's Answer]
[The Start of Assistant B's Answer]
{response_2}
[The End of Assistant B's Answer]

Pairwise Judge with Scores

Prompt: You are given a user question and two responses from two AI assistants. Your task is to act as an impartial judge and evaluate which response better follows the user's instructions and provides a higher-quality answer.
First, provide your reasoning within <think> and </think> tags. This should include your evaluation criteria for a high-quality response, a detailed comparison of the two responses, and when helpful, a reference answer as part of your evaluation. Be explicit in your thought process, referencing your criteria and explaining how each response aligns with or deviates from them.
If you are not entirely sure of the correctness of your verdict, you should verbally express your uncertainty in your thought process.
Avoid any position biases and ensure that the order in which the responses were presented does not influence your decision. Do not allow the length of the responses to influence your evaluation. Do not favor certain names of the assistants. Be as objective as possible.
Finally, assign the assistant's response a score from 0 to 10, using either an integer or a decimal with up to 0.1 precision, with a higher score indicating a higher-quality response that better satisfies the criteria. Enclose the scores within the tags <score_A> </score_A>, and <score_B> </score_B>.
Format your output like this:
<think> your_thinking_process </think>
<score_A> your_score_a </score_A> <score_B> your_score_b </score_B>
Below are the user's question and the two responses:

[User Question]
{input}
[The Start of Assistant A's Answer]
{response_1}
[The End of Assistant A's Answer]
[The Start of Assistant B's Answer]
{response_2}
[The End of Assistant B's Answer]

Pairwise Judge with Likert Score – LM Arena Hard Prompt

Please act as an impartial judge and evaluate the quality of the responses provided by two AI assistants to the user prompt displayed below. You will be given Assistant A's answer and Assistant B's answer. Your job is to evaluate which assistant's answer is better.
Begin your evaluation by generating your own answer to the prompt. You must provide your answers before judging any answers.
When evaluating the assistants' answers, compare both assistants' answers with your answer. You must identify and correct any mistakes or inaccurate information.
Then consider if the assistants' answers are helpful, relevant, and concise. Helpful means the answer correctly responds to the prompt or follows the instructions. Note that when a user prompt has any ambiguity or more than one interpretation, it is more helpful and appropriate to ask for clarifications or more information from the user than providing an answer based on assumptions. Relevant means all parts of the response closely connect or are appropriate to what is being asked. Concise means the response is clear and not verbose or excessive.
Then consider the creativity and novelty of the assistants' answers when needed. Finally, identify any missing important information in the assistants' answers that would be beneficial to include when responding to the user prompt.
Verbalize any uncertainty in your thought process when you're unsure of your conclusion.
If your verdict is not based on absolute certainty, be sure to articulate that hesitation in your analysis.
After providing your explanation, you must output only one of the following choices as your final verdict with a label:
1. Assistant A is significantly better: [[A»B]]
2. Assistant A is slightly better: [[A>B]]
3. Tie, relatively the same: [[A=B]]
4. Assistant B is slightly better: [[B>A]]
5. Assistant B is significantly better: [[B»A]]
Example output: "My final verdict is tie: [[A=B]]."

[User Question]
{input}
[The Start of Assistant A's Answer]
{response_1}
[The End of Assistant A's Answer]
[The Start of Assistant B's Answer]
{response_2}
[The End of Assistant B's Answer]

A.12 Prompts for Verbalized confidence

Pairwise Judge with Verdicts

Prompt:
You are given a user question and two responses from two AI assistants. Your task is to act as an impartial judge and evaluate which response better follows the user's instructions and provides a higher-quality answer.
First, provide your reasoning within <think> and </think> tags. This should include your evaluation criteria for a high-quality response, a detailed comparison of the two responses, and when helpful, a reference answer as part of your evaluation. Be explicit in your thought process, referencing your criteria and explaining how each response aligns with or deviates from them.
If you are not entirely sure of the correctness of your verdict, you should verbally express your uncertainty in your thought process.
Avoid any position biases and ensure that the order in which the responses were presented does not influence your decision. Do not allow the length of the responses to influence your evaluation. Do not favor certain names of the assistants. Be as objective as possible.
Finally, provide your verdict within <answer> and </answer> tags, strictly following this format: - <answer> [[A]] </answer> if Assistant A is better - <answer> [[B]] </answer> if Assistant B is better

Also, provide a confidence score (0-100) for your judgment, representing how likely your answer is to be correct, enclosed within <confidence> and </confidence> tags.
- <confidence> 50 </confidence>
Below are the user's question and the two responses:

User Question] {input}
[The Start of Assistant A's Answer]
{response_1}
[The End of Assistant A's Answer]
[The Start of Assistant B's Answer]
{response_2}
[The End of Assistant B's Answer]

Pairwise Judge with Scores

Prompt: You are given a user question and two responses from two AI assistants. Your task is to act as an impartial judge and evaluate which response better follows the user's instructions and provides a higher-quality answer.
First, provide your reasoning within <think> and </think> tags. This should include your evaluation criteria for a high-quality response, a detailed comparison of the two responses, and when helpful, a reference answer as part of your evaluation. Be explicit in your thought process, referencing your criteria and explaining how each response aligns with or deviates from them.
If you are not entirely sure of the correctness of your verdict, you should verbally express your uncertainty in your thought process.
Avoid any position biases and ensure that the order in which the responses were presented does not influence your decision. Do not allow the length of the responses to influence your evaluation. Do not favor certain names of the assistants. Be as objective as possible.
Finally, assign the assistant's response a score from 0 to 10, using either an integer or a decimal with up to 0.1 precision, with a higher score indicating a higher-quality response that better satisfies the criteria. Enclose the scores within the tags <score_A> </score_A>, and <score_B> </score_B>.
Format your output like this:
<think> your_thinking_process </think>
<score_A> your_score_a </score_A> <score_B> your_score_b </score_B>

Also, provide a confidence score (0-100) for your judgment, representing how likely your answer is to be correct, enclosed within <confidence> and </confidence> tags.
- <confidence> 50 </confidence>
Below are the user's question and the two responses:

[User Question]
{input}
[The Start of Assistant A's Answer]
{response_1}
[The End of Assistant A's Answer]
[The Start of Assistant B's Answer]
{response_2}
[The End of Assistant B's Answer]

Pairwise Judge with Likert Score – Chatbot Arena Hard Prompt

Please act as an impartial judge and evaluate the quality of the responses provided by two AI assistants to the user prompt displayed below. You will be given Assistant A's answer and Assistant B's answer. Your job is to evaluate which assistant's answer is better.
Begin your evaluation by generating your own answer to the prompt. You must provide your answers before judging any answers.
When evaluating the assistants' answers, compare both assistants' answers with your answer. You must identify and correct any mistakes or inaccurate information.
Then consider if the assistants' answers are helpful, relevant, and concise. Helpful means the answer correctly responds to the prompt or follows the instructions. Note that when a user prompt has any ambiguity or more than one interpretation, it is more helpful and appropriate to ask for clarifications or more information from the user than providing an answer based on assumptions. Relevant means all parts of the response closely connect or are appropriate to what is being asked. Concise means the response is clear and not verbose or excessive.
Then consider the creativity and novelty of the assistants' answers when needed. Finally, identify any missing important information in the assistants' answers that would be beneficial to include when responding to the user prompt.

Verbalize any uncertainty in your thought process when you're unsure of your conclusion.
If your verdict is not based on absolute certainty, be sure to articulate that hesitation in your analysis.
Your judgment must also include a confidence score (0–100) reflecting its likelihood of being accurate.

After providing your explanation, you must output only one of the following choices as your final verdict with a label:
1. Assistant A is significantly better: [[A»B]]
2. Assistant A is slightly better: [[A>B]]
3. Tie, relatively the same: [[A=B]]
4. Assistant B is slightly better: [[B>A]]
5. Assistant B is significantly better: [[B»A]]

Example output: "My final verdict is tie: [[A=B]]. Confidence: 0.85".

[User Question]
{input}
[The Start of Assistant A's Answer]
{response_1}
[The End of Assistant A's Answer]
[The Start of Assistant B's Answer]
{response_2}
[The End of Assistant B's Answer]

A.13 Prompt ablations for verbalized confidence

We also conduct ablation studies to determine which approach yields the most effective verbalized confidence from the models. Specifically, we experiment with permitting expressions of uncertainty within the chain-of-thought reasoning, as opposed to the default chain-of-thought approach. Additionally, we test different confidence scales, including both the 0–1 and 0–100 ranges.

A.13.1 Varying confidence ranges

Prompt 1: Range 0-1

"Also, provide a confidence score (0-1) for your judgment, representing how confident you are about your answer, enclosed within <confidence> and </confidence> tags."
Example output: `<confidence> 0.3 </confidence>`

Prompt 2: Range 0-100

"Also, provide a confidence score (0-100) for your judgment, representing how confident you are about your answer, enclosed within <confidence> and </confidence> tags."
Example output: `<confidence> 30 </confidence>`

A.13.2 Varying elicitation mode

Prompt 3: With uncertainty in reasoning

"If you are not entirely sure of the correctness of your verdict, you should verbally express your uncertainty in your thought process. Also, provide a confidence score (0-100) for your judgment, representing how likely your answer is to be correct, enclosed within <confidence> and </confidence> tags."
Example output: `<confidence> 50 </confidence>`

Prompt 4: Without uncertainty in reasoning

"Also, provide a confidence score (0-100) for your judgment, representing how confident you are about your answer, enclosed within <confidence> and </confidence> tags."
Example output: `<confidence> 30 </confidence>`

Model	With uncertainty in reasoning		Without uncertainty in reasoning	
	0-1	0-100	0-1	0-100
LLAMA 70b	0.1483	**0.14496**	0.19916	0.17803
J1 LLAMA 70b	**0.15651**	0.16230	0.18181	0.19956
LLAMA 8b	0.18799	**0.14251**	0.26258	0.20280
J1 LLAMA 8b	0.26245	0.18474	0.21141	**0.17139**

Table 9: Calibration (Kuiper, lower is better) for various prompts eliciting verbalized confidence

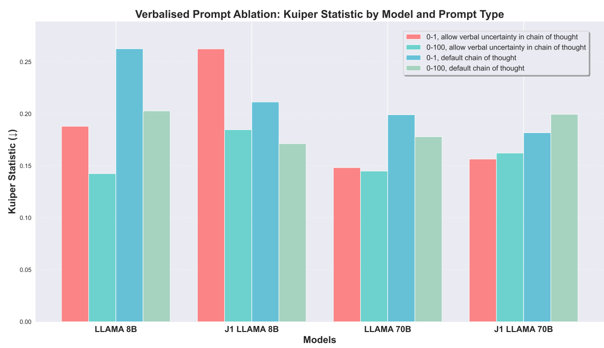


Figure 9: Best prompts for Verbalized confidence as measured by the Kuiper metric. Lower is better. 0-100 ranges and allowing uncertainty in reasoning work best.