

Pinocchio: Fast Uncertainty Estimates for Black-Box Language Models

Kevin David Hayes¹, Arka Pal², Haosong Zhang³, Tom Goldstein¹, Micah Goldblum⁴

¹University of Maryland ²Ritual AI ³Fudan University ⁴Columbia University

Abstract

In high-stakes decision-making applications of large language models (LLMs), practitioners require not only accurate LLMs but also uncertainty estimates for their predictions. Existing approaches to uncertainty estimation for LLMs require access to log-probabilities output by the model or require fine-tuning access. However, the vast majority of industrial LLM products use closed-source API models, and many such API models like GPT-5 do not return log-probabilities and may not allow fine-tuning. We introduce PINOCCHIO, a suite of 8B checkpoints for assigning uncertainty estimates to predictions made by popular API models. To this end, we curate a large collection of hard datasets, ranging from long-context to vision-language, that expose conditions under which frontier models output incorrect responses. We then train PINOCCHIO to recognize the patterns in prompts and model responses that indicate confidence or uncertainty. PINOCCHIO checkpoints significantly outperform existing baselines for black-box uncertainty estimation in calibration metrics and transfer to current frontier models whose responses they were not trained on, including Claude Fable 5, Claude Opus 5, GPT-5.6, and Kimi 3. We release code for adding uncertainty estimation to existing repos in only two additional lines of code.

1 Introduction

A physician asks GPT-5 whether two medications interact. The model replies confidently, and incorrectly. Nothing in the API response flags this failure. This failure mode is not hypothetical; it is the default. As large language models are deployed in medicine, law, scientific research, and software engineering, wrong answers delivered with high confidence remain the central obstacle to trustworthy deployment. The closed-source models that dominate production (GPT-5, Claude, Gemini) expose little of the internal signal that might warn when they are wrong, and recent models increasingly hide even token log-probabilities, yet these are precisely the models that most need reliable uncertainty estimates.

Existing approaches face a fundamental tension between access and performance. Logit- and representation-based methods [23] achieve reasonable discrimination but require white-box access that closed-source providers do not expose. Black-box alternatives (verbalized confidence [43, 50], self-evaluation [23], and semantic entropy [29]) pay a steep cost: verbalized confidence is poorly calibrated, self-evaluation inherits the model’s blind spots, and semantic entropy requires 5–10× the inference budget. A frontier model’s own signals may be calibrated on average yet weak at judging whether *this* response is right, and sampling a surrogate or the target itself does not recover that per-response signal on hard reasoning. None of these methods inherently transfers across models, and the problem compounds for vision-language models [56, 57]. **Fast and accurate uncertainty estimation for black-box API models remains an open problem.**

¹Department of Computer Science, University of Maryland. ²Ritual AI. ³Fudan University.

⁴Department of Computer Science and Department of Electrical Engineering, Columbia University. Correspondence: khayes1@umd.edu, micah.g@columbia.edu.

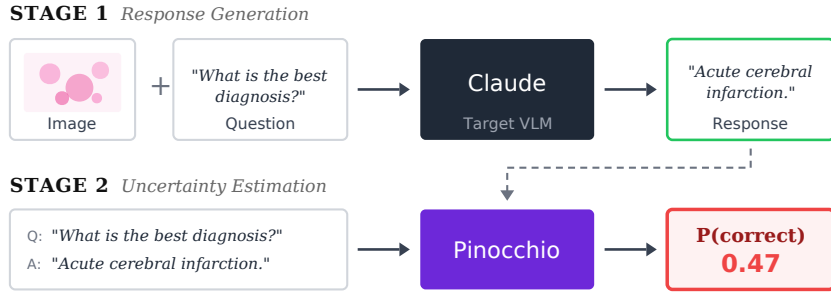


Figure 1: **An external calibrator scores a target model’s response without touching the model.** It reads the question, response, and image and outputs a probability of correctness. This example illustrates the intended workflow; the displayed score $P(\text{correct}) = 0.47$ is illustrative. Adapted from MMMU [53].

We introduce PINOCCHIO, a family of learned uncertainty estimators that predict whether a model’s response is correct from a single (question, response) pair, without access to log-probabilities, multiple samples, or model internals. Rather than extracting uncertainty from the target model itself, we train a small *calibrator* (Figure 1) that takes the question, the response, and the identity of the model that produced it, plus any associated images for vision-language tasks, and outputs a probability of correctness.

Concretely, we train PINOCCHIO by fine-tuning Qwen3-VL-8B [3] with LoRA [22] on 10,892 graded (question, response, correctness) triples from three source models across 20 benchmarks. A single forward pass produces a calibrated correctness probability, with no sampling, no logit access, and no modification of the target model. On held-out data with a strict question-level split (zero question overlap), PINOCCHIO achieves 0.878 AUROC, outperforming the strongest single-pass black-box baseline (0.649) by over 22 points, and a single calibrator transfers to model families it never saw in training, including Claude, Gemini, and LLaMA. When absolute calibration matters on a new target, a practitioner recovers it by recalibrating on roughly 100 labeled examples, without retraining (Section 4.4).

We summarize our contributions as follows:

1. **A tiny, fast uncertainty estimator for black-box APIs.** We train a small uncertainty estimator for closed API models that needs only a single forward pass, with no access to log-probabilities, samples, or model weights, and it assigns well-calibrated correctness probabilities to responses from models that expose no internal signal.
2. **Vision-language data supplies training signal that transfers to text.** Frontier models saturate many text-only datasets, so hard errors are scarce. Vision-language benchmarks still elicit frequent errors, and what the estimator learns from them transfers to text-only benchmarks, improving text-only calibration when hard text examples are limited (Section 4.3).
3. **Stronger discrimination at lower inference cost than black-box baselines.** PINOCCHIO outperforms every single-pass black-box baseline we evaluate, and it avoids the repeated sampling and multi-pass overhead those methods require, so it estimates uncertainty at a fraction of their inference cost.
4. **One auxiliary model serves many target models and transfers to new ones.** A single auxiliary model produces calibrated uncertainty for multiple target models, and it transfers to the newest frontier models it never trained on, including **Claude Fable 5**, **Claude Opus 5**, **GPT-5.6**, and **Kimi 3**, which rules out memorization of their responses.

Table 1: **PINOCCHIO is the only method that meets all three desiderata.** Comparison with existing methods.

Method	Black-box	Single-pass	Scales w/ data
Logit-based [23]	✗	✓	✗
MC Dropout [11]	✗	✗	✗
Semantic Entropy [29]	✓	✗	✗
Verbalized Conf. [43]	✓	✓	✗
ORM / Verifiers [8]	✗	✓	✓
LLM-as-Judge [58]	✓	✓	✗
White-box Probe [24]	✗	✓	✓
PINOCCHIO (ours)	✓	✓	✓

2 Related work

Uncertainty quantification in LLMs. Existing methods fall into three categories. *Token-level* approaches use output probabilities [23, 39] but conflate fluency with correctness. *Sampling-based* methods measure consistency across multiple generations [5, 10, 19, 29], but require $O(K)$ forward passes ($K=5-20$), multiplying API costs. *Verbalized* approaches prompt models to state confidence [31, 43], but LLMs are systematically overconfident [15, 21, 50]. When model internals are accessible, probing hidden states [2, 4, 27] or training supervised UQ modules [26, 41] yields stronger signals, but these white-box methods are inapplicable to closed-source models.

Fine-tuned uncertainty judges. Post-hoc calibration [17] can help when validation data is available, though instruction-tuning degrades base-model calibration [35]. In the reward modeling setting, Yang et al. [51] show that uncertainty-aware reward models reduce annotation costs and mitigate reward hacking, which demonstrates the value of calibrated confidence in LLM post-training. Most relevant to our work, Kapoor et al. [24] train probes and LoRA adapters on the target model’s hidden states, outperforming black-box baselines but requiring open-weight access to the target model. We extend their white-box approach to the black-box API model setting: we train PINOCCHIO, a lightweight auxiliary uncertainty estimator, on LLM outputs across 20 benchmarks and three source models, handling both text and vision inputs in a unified model.

Data scaling and vision-language uncertainty. Table 1 summarizes how existing UQ methods meet the three desiderata we target. Non-learned methods (verbalized confidence, semantic entropy, LLM-as-judge) have no mechanism to improve with additional data. PINOCCHIO’s performance scales log-linearly with training examples: 5,000 samples achieve 96% of the full 10,892-example training set’s performance (Section 4.5). Frontier models err frequently on vision-language benchmarks [16, 30], which makes them a rich source of the hard examples a calibrator needs; uncertainty estimation for VLMs itself remains underexplored. Existing VLM uncertainty methods each drop at least one property we require: they need multiple forward passes [56], are evaluated on a single model family [28, 57], depend on grounding annotations [36], or train a critic bound to one target model [54]. To our knowledge, no prior work delivers a black-box, single-pass uncertainty estimator that transfers across model families. PINOCCHIO treats these frequent vision-language errors as a rich training signal, and that signal transfers to text, so a single jointly trained checkpoint scores both text-only and vision-language responses.

3 Method

3.1 Problem formulation

Given question Q (optionally with image V) and candidate answer A from a target model M_{target} , we seek $f(Q, A, V, M) \in [0, 1]$ estimating the probability that A is correct, without access to M_{target} ’s logits, weights, or internal states, where M is a lightweight identifier (a source-model token) for the model that produced A . We frame this as binary classification

on $\mathcal{D} = \{(Q_i, A_i, V_i, y_i)\}_{i=1}^N$ where $y_i \in \{0, 1\}$, training a model to predict $P(y = 1 \mid Q, A, V, M)$, an explicitly conditional probability that includes the source-model token M . We allow the calibrator to condition on the target model name M so that it can use model-specific patterns to identify uncertainty, noting a modest gain of 0.014 AUROC over target-model-agnostic uncertainty estimation (0.878 vs. 0.863; Section 4.4). When predicting uncertainty for new target models, we simply omit the name of the target model. When a new target model’s labeled responses are available, a practitioner can also train PINOCCHIO directly on them, which modestly exceeds zero-shot transfer (Appendix S.2). A well-calibrated estimator satisfies:

$$\mathbb{P}(y = 1 \mid f(Q, A, V, M) = p) = p \quad \forall p \in [0, 1] \quad (1)$$

3.2 Architecture and training

We fine-tune Qwen3-VL-8B-Instruct [3] with LoRA [22] (rank 32, $\alpha = 64$) on all linear layers. This VLM processes both text and image inputs in a unified architecture, and its forward pass requires an image on every example, so for text-only benchmarks we pass a fixed uniform-gray image, which fills the required slot without adding visual signal. We format each input as a chat-style prompt containing the question, response, and optional benchmark/source model metadata (Appendix O); including metadata improves AUROC by 5.0 points (Section 4.5).

The model predicts a single token, `i` (incorrect) or `ii` (correct), and we extract the correctness probability via softmax over the two logits:

$$P(\text{correct}) = \frac{\exp(z_{ii})}{\exp(z_i) + \exp(z_{ii})} \quad (2)$$

We train the model with binary cross-entropy loss on this final token only. Full training configuration is in Appendix C; hyperparameters in Appendix B; training data breakdown in Appendix A.

3.3 Training data

Since frontier models achieve very high scores on benchmarks, the central challenge in training a calibrator is assembling enough examples where frontier models are wrong. We address this deficit in three ways: (1) selecting hard benchmarks in the 20–80% accuracy sweet spot, where models fail often enough and simultaneously succeed often enough to provide learning signal for the calibrator; (2) including vision-language benchmarks, which are naturally harder for frontier models and provide a rich source of errors; and (3) collecting responses from multiple source models to the same questions, which pushes the calibrator toward patterns of correctness that hold across models rather than the phrasing habits of any one model, a claim we test directly in Section 4.4.

We aim for a single calibrator that, though trained on a fixed set of source models, transfers to the newest frontier systems it never saw, including **Claude Fable 5**, **Claude Opus 5**, **Kimi 3**, and **GPT-5.6** (Section 4.4). Concretely, we run three source models (**GPT-5-mini**, **GPT-5.2**, and **Qwen3.5-397B**) on 20 benchmarks spanning mathematics, science, coding, general knowledge, and multimodal reasoning (Table 2). We grade each response against ground-truth answers to produce binary correctness labels using exact/fuzzy matching for 15 benchmarks and an LLM judge (GPT-5-mini) for the remaining five rubric benchmarks (grading robustness in Appendix M); a post-hoc audit confirmed only 18 of 12,972 labels (0.14%) are affected by grading edge cases. The resulting dataset contains $\sim 11,000$ training examples (54% correct). We hold out 15% using a *question-level split*: all instances of a given question across source models go to the same split, yielding 1,953 test samples with zero question overlap.

Benchmark selection. We design the benchmark mixture around three criteria. First, *difficulty calibration*: we prioritize benchmarks where frontier models achieve 20–80% accuracy, ensuring a naturally balanced class distribution without artificial resampling. Benchmarks

Table 2: **The mixture spans 7 domains and 2 modalities in the 20–80% accuracy band.** Benchmark mixture spanning 7 domains and 2 modalities. Of 20 benchmarks, 9 are text-only and 11 are vision-language.

Domain	Benchmarks	Modality
Mathematics	OmniMath, MathVista, MathVerse, MathVision	T+V
Science	GPQA Diamond, ChemBench	T
Coding	PRBench, ARC-AGI	T
General	SimpleQA, LiveBench, BBEH	T
Multimodal	MMMU, MMStar, MM-Vet, CharXiv, VizWiz	V
Spatial	RealWorldQA, HallusionBench	V
Frontier	HLE, HLE-Multimodal	T+V

that are too easy (e.g., MMLU, >85% accuracy) or too hard (e.g., FrontierMath, <5%) provide little discriminative signal. Second, *domain diversity*: we span seven domains (Table 2) to prevent the calibrator from learning domain-specific shortcuts: consistent with this, leave-K-out cross-validation shows that dropping any 5 benchmarks costs only +0.003 mean AUROC (Appendix Table 14), i.e. no single domain is load-bearing. Third, *format and modality heterogeneity*: we include multiple-choice, open-ended, and binary formats across both text-only and vision-language modalities, teaching the calibrator general correctness signals rather than format-specific heuristics.

Multi-model training. Training on responses from multiple source models drives cross-model transfer. We evaluate each single-source calibrator on held-out target models it never trained on, where it reaches 0.672–0.759 AUROC; the multi-source calibrator reaches 0.878, a transfer gain of 12–21 points (Section 4.4). Multi-source training forces the calibrator to learn model-agnostic features of correctness (reasoning quality, factual consistency) rather than model-specific response patterns. That these patterns are largely model-agnostic is borne out downstream: removing the source-model token costs only 0.014 AUROC (0.878 to 0.863), and leave-one-model-out transfer holds at 0.843 (Section 4.4).

4 Experiments

4.1 Experimental setup

Target models. We train our flagship calibrator on three source models, **GPT-5-mini** and **GPT-5.2** (OpenAI) and **Qwen3.5-397B** (open-weight). The headline test is generalization to the current frontier: with no further training, this single calibrator scores the newest systems it never saw, **Claude Fable 5**, **Claude Opus 5**, **GPT-5.6**, and **Kimi 3**, and transfers more broadly to fifteen unseen models across eight organizations, including **LLaMA-3.1-8B-Instruct** [14] (Section 4.4, Table 3).

Benchmarks. We evaluate on 20 benchmarks across 7 domains, including 9 text-only and 11 vision-language benchmarks. Full benchmark details and citations are in Appendix N.

Baselines. We compare against verbalized confidence, Platt and isotonic recalibration, response length, an LLM judge, and proxy semantic entropy and self-consistency computed from $N=5$ Qwen3-VL-8B samples. The first six use the full held-out test set of 1,953 samples (question-level split); the two proxy sampling baselines use its 1,376-example text-only subset. Because the proxy variants sample a stand-in model rather than the target, we additionally evaluate a full suite of *faithful* sampling estimators run directly on open targets: semantic entropy [29], self-consistency [45], the graph-theoretic measures of Lin et al. [32], SPUQ [13], and the logit-free conformal method of Su et al. [42] (Section 4.2; Appendix Q). Implementation details are found in Appendix P.

Metrics. We report **AUROC** (area under the receiver operating characteristic curve), which measures the ability to rank correct responses above incorrect ones, regardless of threshold.

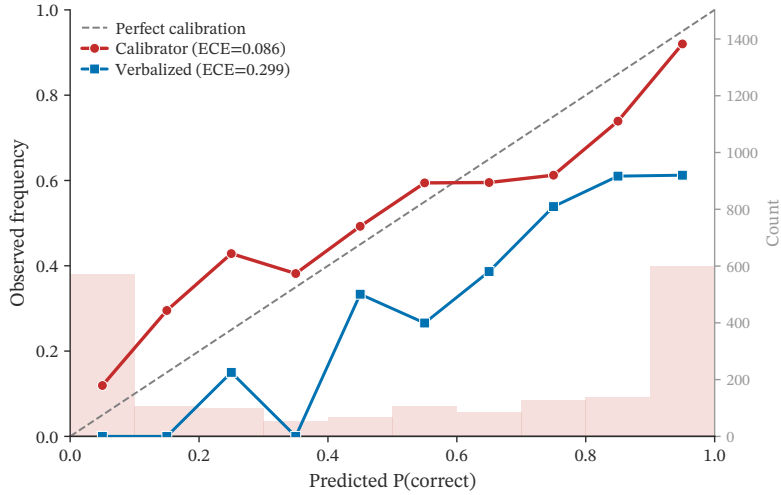


Figure 2: **PINOCCHIO’s probabilities are well calibrated; raw verbalized confidence is not.** Reliability diagram on the held-out test set. PINOCCHIO’s probabilities track observed accuracy far more closely than raw verbalized confidence, which remains overconfident on incorrect responses.

We include 95% BCa bootstrap confidence intervals from 2,000 resamples. We additionally report Brier score and expected calibration error (ECE; 10 equal-width bins) for calibration, and use DeLong’s test [9] for paired full-set AUROC comparisons.

4.2 Main results

On a closed API, a deployer can observe only the question, the model’s answer, and whatever confidence the model reports about itself. Table 7 (Appendix F) compares PINOCCHIO against these single-pass signals: verbalized confidence, response length, and an LLM judge. All are weak, sitting near chance, while a calibrator that reads the same question and answer but is trained to discriminate reaches 0.878 AUROC (Figures 2 and 3). We headline the uncalibrated score because it needs no labels from the target and preserves discrimination that post-hoc recalibration cannot recover; a deployer who can spare roughly 100 target labels should prefer per-model Platt scaling, which matches this AUROC at lower ECE (Appendix S.2).

Appendices L and S.1 report effect sizes and additional input ablations.

Takeaway

A small external calibrator predicts black-box correctness far better than the confidence a model expresses about itself (0.88 vs. 0.65 AUROC).

We test whether the calibrator reads correctness from the response itself rather than merely learning which *questions* are hard. We keep only the 228 test questions that at least one source model answered correctly and at least one answered incorrectly, so a question-only predictor scores 0.5 AUROC on this set by construction. A within-question permutation test, which shuffles labels only among responses to the same question and thus controls for question identity exactly, gives 0.841 AUROC ($p < 0.0001$; Appendix F.7). This ablation shows that the calibrator reads correctness from the content of the response, not from the difficulty of the question.

One might think PINOCCHIO keys on surface words like “maybe” or “certainly” rather than the model’s actual knowledge. A second ablation isolates what the calibrator reads (Table 9).

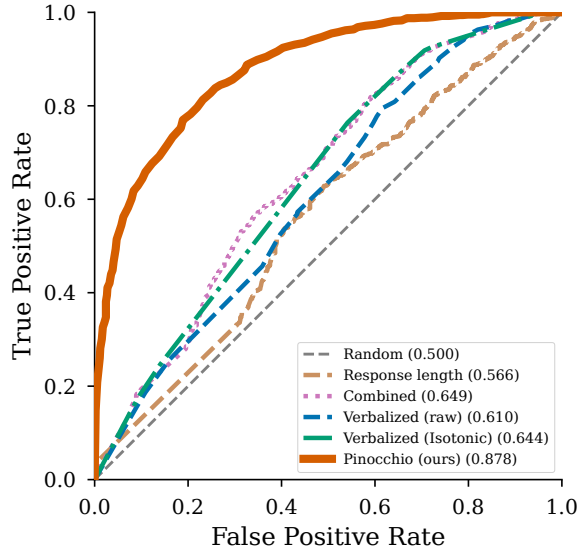


Figure 3: **PINOCCHIO separates correct from incorrect responses far more cleanly than single-pass baselines.** ROC curves on the held-out test set of 1,953 examples.

Scoring from the response alone, with the question blanked, retains most of the signal, so it does not require knowing what was asked; adding the question back raises AUROC further, so the question contributes too. Shuffling words within each response, which preserves lexical content but destroys meaning, drops AUROC sharply, whereas shuffling whole sentences barely moves it. The calibrator therefore draws on *both* the model’s knowledge and the content of the response, and the operative cue is meaning rather than surface wording.

Takeaway

The calibrator relies on both identifying questions that are hard for the target model and interpreting linguistic cues in the target model’s response.

If the calibrator merely read hedges, adding or removing words like “maybe” should swing its score. Instead, hedges such as “maybe” or “I think” that make a human reader judge an answer as less confident change the calibrator’s score only slightly: stripping hedge words from every response shifts AUROC by +0.0015 and injecting them shifts it by −0.0185 (Table 10). The calibrator responds to a property of the response that the surface hedges do not control.

One might object that the sampling baselines look weak only because they sample a stand-in rather than the target. We therefore run semantic entropy, self-consistency, the estimators of Lin et al. [32], SPUQ [13], and the conformal method of Su et al. [42] directly on generations from LLaMA-3.1-8B. Even granted the target’s own samples, they stay near chance on the hard suite, far below PINOCCHIO scored on the same responses. These estimators work well where they were designed to, reaching 0.740 AUROC on an easier TriviaQA control, but they collapse on the hard, low-accuracy suite, where the target rarely produces a correct sample for consistency to detect. Full results and controls are in Appendix Q.

4.3 Performance across benchmarks and modalities

One might worry that forcing a single checkpoint to serve both text and vision costs accuracy on each. It does not: the calibrator scores text-only and vision-language responses at comparable AUROC, so a deployer needs no separate per-modality calibrator. The more

important role of vision-language data is as training signal. Because frontier models saturate text benchmarks, hard text errors are scarce, and vision-language benchmarks supply the errors the calibrator learns from. The calibrator is strongest on structured mathematical reasoning and weakest on specialized or frontier-difficulty tasks (Appendix Table 11).

Beyond coverage, vision-language data acts as *training-signal augmentation* in the regime our premise implies: frontier models saturate text benchmarks, so hard text examples are scarce. Training on a fixed budget of N text examples with and without the full vision-language set (matched configuration, same held-out text benchmarks) shows that adding vision-language data improves text AUROC at every budget, with the largest gains where text is scarcest and tapering as text data grows (Appendix Table 31). Adding it to a small text budget matches a far larger text-only budget, substituting for hard text examples that are expensive to collect at frontier difficulty. With plentiful text data it neither helps nor hurts; when hard text is scarce, the common case at frontier difficulty, it recovers much of the gap.

Takeaway

A single calibrator covers both text and vision at comparable accuracy, and vision-language data improves text calibration when hard text examples are scarce.

4.4 Cross-model transfer

A more difficult question is whether one predictor can stay well-calibrated across models, including models it never trained on. A calibrator that memorized a single model’s response style would fail as soon as a deployer switched providers, so the real test is whether a calibrator trained on a few models generalizes to the rest. Multi-model training is the mechanism: by exposing the calibrator to cases where different models disagree on the same question, we reduce its reliance on any single model’s idiosyncrasies and push it toward what correct responses share.

Training on all three source models yields a large gain over any single-source calibrator (Table 8), and leave-one-model-out evaluation still transfers to the held-out model (Appendix Table 30). The stronger test evaluates families absent from training entirely: a single calibrator transfers to 15 models across 8 organizations (Table 3), six released *after* the calibrator was trained, ruling out memorization. Among them are the current-generation Claude Fable 5, Claude Opus 5, and Kimi 3. The newest OpenAI model, GPT-5.6, transfers just as well despite sharing an organization with the source models, as does the separate 700-question LLaMA-3.1-8B hard suite.

The most demanding deployment target is the current frontier. We apply the same calibrator, trained only on older models, to the four newest frontier systems and compare it against the signals a deployer actually has for them: the model’s own verbalized confidence and its response length. On all four, the calibrator ranks correct answers far better than either deployer-available signal, with pooled AUROC 0.837 against 0.691 for the models’ own verbalized confidence and 0.343 for response length. These numbers use no labels from the new models; a calibrator trained directly on their own responses reaches 0.85–0.86 AUROC (Appendix S.2), so the zero-shot transfer is a no-label lower bound.

The source-model token is useful but not required: removing it reduces AUROC from 0.878 to 0.863. Leave- K -out benchmark evaluation is in Appendix Table 14.

Transfer preserves how responses are ranked, but it can shift their absolute calibration. Because that shift is monotonic, it is cheap to undo: Platt or isotonic regression fit on roughly 100 labeled examples from a new target roughly halves ECE without changing the calibrator’s weights, while leaving AUROC unchanged (Table 4). Recalibration cannot add discrimination that the underlying score lacks, which is why rescaling leaves the verbalized baseline near 0.61 AUROC. What transfers across models is the ranking signal, not the absolute confidence value.

Table 3: **A single calibrator transfers to the newest frontier models, and to 15 unseen families in all.** Cross-family transfer across 8 organizations. The calibrator is trained on GPT-5-mini, GPT-5.2, and Qwen3.5 only; every model below is absent from training. The current-generation frontier systems (top, bold) and all models marked [†] were released *after* the calibrator was trained, ruling out memorization of their responses.

Model	Organization	AUROC $\uparrow$
Claude Fable 5[†]	Anthropic	0.822
Claude Opus 5[†]	Anthropic	0.845
Kimi 3[†]	Moonshot	0.854
LLaMA-3.1-8B	Meta	0.811
Mistral-7B	Mistral	0.840
Devstral-2-24B [†]	Mistral	0.836
gemma-2-9b	Google	0.826
OLMo-2-7B	AllenAI	0.824
Olmo-3.1-32B [†]	AllenAI	0.801
DeepSeek-R1-Distill-32B	DeepSeek	0.816
granite-4.1-30b [†]	IBM	0.806
Claude Opus 4.6	Anthropic	0.791
Claude Sonnet 4.6	Anthropic	0.820
Gemini 3.1 Pro	Google	0.861
Gemini 3 Flash	Google	0.840

Table 4: **Unseen frontier models drift out of calibration, but ~ 100 labels restore it.** Text-only ECE on four models absent from training. Platt or isotonic recalibration fit on 100 labeled examples and evaluated on the remainder (mean over 25 random splits) roughly halves the pooled ECE, while AUROC is unchanged because recalibration is monotone.

Model	AUROC	ECE $\downarrow$		
		Off-the-shelf	Platt	Isotonic
Claude Fable 5	0.822	0.179	0.101	0.086
GPT-5.6	0.830	0.133	0.116	0.086
Kimi 3	0.854	0.086	0.094	0.074
Claude Opus 5	0.845	0.200	0.099	0.080
Pooled	0.837	0.134	0.077	0.067

Takeaway

One calibrator transfers to more than a dozen unseen model families across eight organizations, several released after training.

4.5 Scaling and input ablations

The three factors that could drive the calibrator’s accuracy do not matter equally, so we measure each in turn: the amount of training data, the size of the calibrator, and the structured input it is given. Increasing the training set produces substantially larger gains than increasing calibrator size (Figure 8); the smallest 0.8B calibrator retains 99% of the 9B model’s AUROC (Appendix Figure 7), while performance keeps improving as data grows. The structured input also contributes: removing the benchmark and source-model metadata drops AUROC from 0.878 to 0.827, so the calibrator uses more than the bare response. Full input and elicitation ablations are in Appendices S.1 and F.9.

Beyond aggregate AUROC, PINOCCHIO drives three deployment workflows (adaptive clarification, confidence-gated actions, and human-escalation review), outperforming every

baseline on each; Appendices H and S.2 detail these production use cases and per-model recalibration.

Takeaway

Training data matters more than calibrator size, so accurate calibration runs on a small, cheap model.

5 Using PINOCCHIO

Disclaimer! PINOCCHIO may be miscalibrated on new data distributions

How to adapt PINOCCHIO to new data distributions and API models. We train and calibrate PINOCCHIO models on a fixed set of datasets we aggregate. Out of the box, its correctness ranking already transfers across target model families and nearby domains, so calibrated probabilities are usually just a ~ 100 -label Platt or isotonic fit away, with no retraining (Table 4). For a task far from the training mix, fine-tune the calibrator on a sample of your own labeled data to recover accuracy and calibration (Appendix S.2).

Incorporating PINOCCHIO in your code. The released package (`pip install pinocchio-uf`) adds two lines on top of the API call you already make, with no access to the target’s logits or weights (full examples in Appendix O):

```
from pinocchio import Pinocchio
judge = Pinocchio()                                # load once (downloads the model)

# your existing OpenAI call:
response = client.chat.completions.create(model="gpt-5", messages=messages)
# ...plus one line for a calibrated P(correct):
p_correct = judge.score(response, messages=messages)
```

Limitations. The calibrator requires labeled training data with known ground-truth correctness, limiting applicability to tasks where automated grading is possible. Cross-family transfer degrades compared to within-family (Section 4.4). Including benchmark meta-data improves performance by ~ 5 points; without it, users should expect lower but still baseline-beating AUROC. Our training data is English-only. The calibrator is weakest on frontier-difficulty benchmarks where even correct responses hedge and qualify, the same markers associated with incorrectness (Appendix F.4).

Acknowledgements

This project was supported by a research award from the Center for AI and Responsible Financial Innovation at Columbia University and by NVIDIA’s Academic Grant Program.

References

- [1] Afra Feyza Akyürek, Advait Gosai, Chen Bo Calvin Zhang, Vipul Gupta, Jaehwan Jeong, Anisha Gunjal, Tahseen Rabbani, Maria Mazzone, David Randolph, Mohammad Mahmoudi Meymand, Gurshaan Chattha, Paula Rodriguez, Diego Mares, Pavit Singh, Michael Liu, Subodh Chawla, Pete Cline, Lucy Ogaz, Ernesto Hernandez, Zihao Wang, Pavi Bhattar, Marcos Ayestaran, Bing Liu, and Yunzhong He. PRBench: Large-scale expert rubrics for evaluating high-stakes professional reasoning, 2025.
- [2] Amos Azaria and Tom M. Mitchell. The internal state of an LLM knows when it’s lying. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 967–976, 2023. doi: 10.18653/v1/2023.findings-emnlp.68. URL <https://aclanthology.org/2023.findings-emnlp.68/>.
- [3] Shuai Bai, Yuxuan Cai, Ruizhe Chen, et al. Qwen3-VL technical report. *arXiv preprint arXiv:2511.21631*, 2025.
- [4] Collin Burns, Haotian Ye, Dan Klein, and Jacob Steinhardt. Discovering latent knowledge in language models without supervision. In *ICLR*, 2023.
- [5] Jiuhai Chen and Jonas Mueller. 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. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.acl-long.283. URL <https://aclanthology.org/2024.acl-long.283/>.
- [6] Lin Chen, Jinsong Li, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Zehui Chen, Haodong Duan, Jiaqi Wang, Yu Qiao, Dahua Lin, and Feng Zhao. Are we on the right way for evaluating large vision-language models? In *Advances in Neural Information Processing Systems*, 2024. NeurIPS 2024; also arXiv:2403.20330.
- [7] François Chollet. On the measure of intelligence, 2019.
- [8] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.
- [9] Elizabeth R DeLong, David M DeLong, and Daniel L Clarke-Pearson. Comparing the areas under two or more correlated receiver operating characteristic curves: A nonparametric approach. *Biometrics*, 44(3):837–845, 1988.
- [10] Sebastian Farquhar, Jannik Kossen, Lorenz Kuhn, and Yarin Gal. Detecting hallucinations in large language models using semantic entropy. *Nature*, 630:625–630, 2024. doi: 10.1038/s41586-024-07421-0. URL <https://doi.org/10.1038/s41586-024-07421-0>.
- [11] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *Proceedings of the 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 1050–1059, 2016. URL <https://proceedings.mlr.press/v48/gal16.html>.
- [12] Bofei Gao et al. Omni-MATH: A universal olympiad level mathematic benchmark for large language models. *arXiv preprint arXiv:2410.07985*, 2024.
- [13] Xiang Gao, Jiaxin Zhang, Lalla Mouatadid, and Kamalika Das. SPUQ: Perturbation-based uncertainty quantification for large language models. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2336–2346, 2024. doi: 10.18653/v1/2024.eacl-long.143. URL <https://aclanthology.org/2024.eacl-long.143/>.
- [14] Aaron Grattafiori, Abhimanyu Dubey, et al. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

- [15] Tobias Groot and Matias Valdenegro-Toro. Overconfidence is key: Verbalized uncertainty evaluation in large language and vision-language models. In *Proceedings of the 4th Workshop on Trustworthy Natural Language Processing*, pages 145–171, 2024. doi: 10.18653/v1/2024.trustnlp-1.13. URL <https://aclanthology.org/2024.trustnlp-1.13/>.
- [16] Tianrui Guan, Fuxiao Liu, Xiyang Wu, Ruiqi Xian, Zongxia Li, Xiaoyu Liu, Xijun Wang, Lichang Chen, Furong Huang, Yaser Yacoob, Dinesh Manocha, and Tianyi Zhou. HallusionBench: An advanced diagnostic suite for entangled language hallucination and visual illusion in large vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024. URL https://openaccess.thecvf.com/content/CVPR2024/html/Guan_HallusionBench_An_Advanced_Diagnostic_Suite_for_Entangled_Language_Hallucination_and_CVPR_2024_paper.html.
- [17] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1321–1330, 2017. URL <https://proceedings.mlr.press/v70/guo17a.html>. ICML 2017.
- [18] Danna Gurari et al. VizWiz grand challenge: Answering visual questions from blind people. In *CVPR*, 2018.
- [19] Kimia Hamidieh, Veronika Thost, Walter Gerych, Mikhail Yurochkin, and Marzyeh Ghassemi. Complementing self-consistency with cross-model disagreement for uncertainty quantification. In *International Conference on Learning Representations*, 2026. ICLR 2026.
- [20] Kevin David Hayes, Micah Goldblum, Vikash Sehwal, Gowthami Somepalli, Ashwinee Panda, and Tom Goldstein. FineGRAIN: Evaluating failure modes of text-to-image models with vision language model judges. In *Advances in Neural Information Processing Systems (Datasets and Benchmarks Track)*, 2025. NeurIPS 2025.
- [21] Juyeon Heo, Miao Xiong, Christina Heinze-Deml, and Jaya Narain. Do LLMs estimate uncertainty well in instruction-following? In *International Conference on Learning Representations*, 2025.
- [22] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.
- [23] Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, et al. Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*, 2022.
- [24] Sanyam Kapoor, Nate Gruver, Manley Roberts, Katherine Collins, Arka Pal, Umang Bhatt, Adrian Weller, Samuel Dooley, Micah Goldblum, and Andrew Gordon Wilson. Large language models must be taught to know what they don’t know. In *NeurIPS*, 2024.
- [25] Mehran Kazemi et al. BIG-Bench Extra Hard. *arXiv preprint arXiv:2502.19187*, 2025.
- [26] Reza Khanmohammadi, Erfan Miah, Mehrsa Mardikoraem, Simerjot Kaur, Ivan Brugere, Charese Smiley, Kundan S. Thind, and Mohammad M. Ghassemi. Calibrating LLM confidence by probing perturbed representation stability. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 10448–10514, 2025. doi: 10.18653/v1/2025.emnlp-main.530. URL <https://aclanthology.org/2025.emnlp-main.530/>.
- [27] Jannik Kossen, Jiatong Han, Muhammed Razzak, Lisa Schut, Shreshth Malik, and Yarin Gal. Semantic entropy probes: Robust and cheap hallucination detection in LLMs, 2024.

- [28] Vasily Kostumov, Bulat Nutfullin, Oleg Pilipenko, and Eugene Ilyushin. Uncertainty-aware evaluation for vision-language models, 2024.
- [29] Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. In *International Conference on Learning Representations*, 2023.
- [30] Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Xin Zhao, and Ji-Rong Wen. Evaluating object hallucination in large vision-language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 292–305. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.emnlp-main.20. URL <https://aclanthology.org/2023.emnlp-main.20/>.
- [31] Stephanie Lin, Jacob Hilton, and Owain Evans. Teaching models to express their uncertainty in words. *TMLR*, 2022.
- [32] Zhen Lin, Shubhendu Trivedi, and Jimeng Sun. Generating with confidence: Uncertainty quantification for black-box large language models. *Transactions on Machine Learning Research*, 2024.
- [33] Pan Lu et al. MathVista: Evaluating mathematical reasoning of foundation models in visual contexts. In *ICLR*, 2024.
- [34] Adrian Mirza et al. Are large language models superhuman chemists? *arXiv preprint arXiv:2404.01475*, 2024.
- [35] Preetum Nakkiran, Arwen Bradley, Adam Goliński, Eugene Ndiaye, Michael Kirchhof, and Sinead Williamson. Trained on tokens, calibrated on concepts: The emergence of semantic calibration in llms. In *International Conference on Learning Representations*, 2026.
- [36] Trilok Padhi, Ramneet Kaur, Adam D. Cobb, Manoj Acharya, Anirban Roy, Colin Samplawski, Brian Matejek, Alexander M. Berenbeim, Nathaniel D. Bastian, and Susmit Jha. Calibrating uncertainty quantification of multi-modal LLMs using grounding, 2025.
- [37] Long Phan et al. Humanity’s Last Exam. *arXiv preprint arXiv:2501.14249*, 2025.
- [38] John C. Platt. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In Alexander J. Smola, Peter Bartlett, Bernhard Schölkopf, and Dale Schuurmans, editors, *Advances in Large-Margin Classifiers*, pages 61–74. MIT Press, 1999. URL <https://mitpress.mit.edu/9780262194488/advances-in-large-margin-classifiers/>.
- [39] Benjamin Plaut, Nguyen X. Khanh, and Tu Trinh. Probabilities of chat LLMs are miscalibrated but still predict correctness on multiple-choice Q&A. *Transactions on Machine Learning Research*, 2025.
- [40] David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R Bowman. GPQA: A graduate-level google-proof Q&A benchmark. In *First Conference on Language Modeling*, 2024.
- [41] Artem Shelmanov, Ekaterina Fadeeva, Akim Tsvigun, Ivan Tsvigun, Zhuohan Xie, Igor Kiselev, Nico Daheim, Caiqi Zhang, Artem Vazhentsev, Mrinmaya Sachan, Preslav Nakov, and Timothy Baldwin. A head to predict and a head to question: Pre-trained uncertainty quantification heads for hallucination detection in LLM outputs. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 35712–35731, 2025. doi: 10.18653/v1/2025.emnlp-main.1809. URL <https://aclanthology.org/2025.emnlp-main.1809/>.
- [42] Jiayuan Su, Jing Luo, Hongwei Wang, and Lu Cheng. API is enough: Conformal prediction for large language models without logit-access. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 979–995. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.findings-emnlp.54. URL <https://aclanthology.org/2024.findings-emnlp.54/>.

- [43] Katherine Tian, Eric Mitchell, Allan Zhou, Archit Sharma, Rafael Rafailov, Huaxiu Yao, Chelsea Finn, and Christopher D Manning. 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, 2023. doi: 10.18653/v1/2023.emnlp-main.330. URL <https://aclanthology.org/2023.emnlp-main.330/>.
- [44] Ke Wang, Junting Pan, Weikang Shi, Zimu Lu, Houxing Ren, Aojun Zhou, Mingjie Zhan, and Hongsheng Li. Measuring multimodal mathematical reasoning with MATH-Vision dataset. *Advances in Neural Information Processing Systems*, 2024.
- [45] Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *ICLR*, 2023.
- [46] Zirui Wang et al. CharXiv: Charting gaps in realistic chart understanding in multimodal LLMs. In *Advances in Neural Information Processing Systems*, 2024.
- [47] Jason Wei et al. SimpleQA: Measuring short-form factuality in large language models. *OpenAI Technical Report*, 2024. URL <https://openai.com/index/introducing-simpleqa/>.
- [48] Colin White, Samuel Dooley, Manley Roberts, Arka Pal, et al. LiveBench: A challenging, contamination-limited LLM benchmark. In *International Conference on Learning Representations*, 2025.
- [49] xAI. RealWorldQA. <https://huggingface.co/datasets/xai-org/RealworldQA>, 2024.
- [50] Miao Xiong, Zhiyuan Hu, Xinyang Lu, Yifei Li, Jie Fu, Junxian He, and Bryan Hooi. Can llms express their uncertainty? an empirical evaluation of confidence elicitation in llms. In *International Conference on Learning Representations*, 2024.
- [51] Daniel Yang, Samuel Stante, Florian Redhardt, Lena Libon, Parnian Kassraie, Ido Hakimi, Barna Pásztor, and Andreas Krause. RewardUQ: A unified framework for uncertainty-aware reward models. *arXiv preprint arXiv:2602.24040*, 2026.
- [52] Weihao Yu, Zhengyuan Yang, Linjie Li, Jianfeng Wang, Kevin Lin, Zicheng Liu, Xinchao Wang, and Lijuan Wang. MM-Vet: Evaluating large multimodal models for integrated capabilities. In *Proceedings of the 41st International Conference on Machine Learning*, 2024. ICML 2024; also arXiv:2308.02490.
- [53] Xiang Yue et al. MMMU: A massive multi-discipline multimodal understanding and reasoning benchmark for expert AGI. In *CVPR*, 2024.
- [54] Di Zhang et al. Critic-V: VLM critics help catch VLM errors in multimodal reasoning. In *CVPR*, 2025.
- [55] Renrui Zhang et al. MathVerse: Does your multi-modal LLM truly see the diagrams in visual math problems? In *European Conference on Computer Vision*, 2024.
- [56] Ruiyang Zhang, Hu Zhang, and Zhedong Zheng. VL-uncertainty: Detecting hallucination in large vision-language model via uncertainty estimation, 2024.
- [57] Ruiyang Zhang, Hu Zhang, Hao Fei, and Zhedong Zheng. Uncertainty-o: One model-agnostic framework for unveiling uncertainty in large multimodal models, 2025.
- [58] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, et al. Judging LLM-as-a-judge with MT-Bench and chatbot arena. In *NeurIPS*, 2023.

Table 5: **Training data spans seven domains and both modalities.** Training data summary by benchmark domain.

Domain	Benchmarks	Examples	Modality
Mathematics	OmniMath, MathVista, MathVerse, MathVision	~2,500	T+V
Science	GPQA Diamond, ChemBench	~800	T
Professional and coding	PRBench, ARC-AGI	~600	T
General	SimpleQA, LiveBench, BBEH	~2,800	T
Multimodal	MMMU, MMStar, MM-Vet, CharXiv, VizWiz	~1,800	V
Spatial	RealWorldQA, HallusionBench	~800	V
Frontier	HLE, HLE-Multimodal	~700	T+V
Total	20 benchmarks, 3 source models	~11,000	T+V

Table 6: **The calibrator is trained with lightweight LoRA fine-tuning.** Unified calibrator training hyperparameters.

Hyperparameter	Value
Base model	Qwen3-VL-8B-Instruct
LoRA rank (r)	32
LoRA alpha (α)	64
LoRA target modules	All linear layers
Learning rate	1×10^{-4}
Weight decay	0.01
Batch size	1
Gradient accumulation steps	16
Effective batch size	16
Epochs	3
Max question length	1,500 characters
Max response length	800 characters
Train/test split	85/15 (question-level split)
Precision	bf16
Optimizer	AdamW
Hardware	4× GPU (DDP)
Training time	2–3 hours

A Training data details

Table 5 shows the training data breakdown for the unified calibrator. Three target models (GPT-5-mini, GPT-5.2, Qwen3.5-397B) generated the responses across 20 benchmarks. The dataset contains 10,892 training examples with 54% correct responses, and 1,953 held-out test examples (15% split, question-level stratified by benchmark and source model).

B Hyperparameters

Table 6 lists training hyperparameters for the unified calibrator.

C Training configuration and input processing

Training configuration. Training uses AdamW with learning rate 10^{-4} , batch size 1 with gradient accumulation over 16 steps (effective batch size 16), BF16 mixed precision, and runs for 3 epochs on 4 GPUs with DDP. The full training takes approximately 2–3 hours on A100 GPUs. See Table 6 for complete hyperparameters.

Truncation and image processing. We truncate questions to 1,500 characters and responses to 800 characters, and report truncation results in Appendix S.1. For VLM benchmarks, we resize real images to fit within 200,704 to 401,408 pixels (corresponding to 256–512 tiles of 28×28 in the model’s dynamic-resolution tiling scheme). For text-only benchmarks, we use a 28×28 gray placeholder image.

D Training data details: multi-model training, benchmark selection, and data sufficiency

Why multi-model training? Training on responses from multiple target models is critical for cross-model transfer. Models trained on a single source model’s responses achieve 0.672–0.759 AUROC, while the combined model achieves 0.878, a gain of 12–21 points (Section 4.4). We hypothesize that multi-model training forces the calibrator to learn model-agnostic features of correctness (reasoning quality, factual consistency) rather than model-specific response patterns. A leave-one-model-out (LOMO) evaluation confirms this: training on two source models and testing on the held-out third yields a mean AUROC of 0.843, only 3.5 points below the full model, demonstrating genuine cross-model transfer rather than memorization of model-specific artifacts.

Training data sufficiency and split integrity. The training size ablation in Appendix S.1 shows that 5,000 samples achieve 96% of full-data performance and 2,000 samples achieve 91%. We enforce a *question-level split*: since the same question may appear across all three source models, we assign all instances of a given question to the same split. This is essential: a naïve sample-level split allows 82–96% of test questions to leak into training, inflating AUROC from 0.878 to 0.953. All results in this paper use the question-level split with zero question overlap between train and test.

E Near-duplicate contamination analysis

To verify that the question-level train/test split does not inadvertently leak information through near-duplicate questions, we conduct a cross-split contamination analysis using character 5-gram Jaccard similarity. For each test sample, we compute the Jaccard similarity of its character 5-gram set against every training sample; a test sample is flagged as a near-duplicate if any training sample exceeds a similarity threshold of 0.8.

Results. Of 1,953 test samples, 405 (20.7%) have at least one near-duplicate in the training set. The worst-affected benchmarks are ARC-AGI (65.8%), MM-Vet (41.2%), ChemBench (39.8%), and HallusionBench (39.1%). These benchmarks contain templated or formulaic question structures that naturally produce high textual overlap even across semantically distinct questions (e.g., ARC-AGI grid descriptions share boilerplate formatting).

Impact on performance. The calibrator achieves AUROC 0.878 on the full test set and 0.872 after excluding flagged overlaps, a difference of -0.006 . The small change indicates that the flagged overlaps do not explain the reported discrimination.

F Additional analyses

F.1 Full results tables

Table 7 reports the in-distribution main results on the held-out test set of the three source models, where PINOCCHIO reaches 0.878 AUROC, far above every single-pass baseline. The full cross-family transfer to 15 unseen models is reported in the main body (Table 3).

Table 7: **A trained calibrator predicts black-box correctness far better than any single-pass baseline.** Main results on the held-out test set (1,953 examples, question-level split). Brackets give 95% BCa bootstrap CIs. Differences between PINOCCHIO and each full-set baseline are significant by DeLong’s test (Appendix L).

Method	AUROC $\uparrow$	Brier $\downarrow$	ECE $\downarrow$
Verbalized (raw)	0.610 [.590, .624]	0.326	0.299
Verbalized (Platt)	0.610 [.590, .624]	0.232	0.034
Verbalized (Isotonic)	0.644 [.620, .667]	0.231	0.009
Response length	0.566 [.542, .592]	0.237	0.028
Combined (verb.+len.)	0.649 [.624, .673]	0.229	0.021
LLM-as-judge	0.532 [.524, .540]	0.257	0.110
Proxy sem. entropy [†]	0.467	0.340	0.266
Proxy self-consist. [†]	0.466	0.340	0.266
PINOCCHIO (ours)	0.878 [.863, .892]	0.151	0.086
PINOCCHIO (Platt) [‡]	0.874	–	0.064
PINOCCHIO (Isotonic) [‡]	0.866	–	0.063

[†]Text-only subset (1,376 samples); CIs omitted. [‡]Recalibrated per source model on ~ 100 held-out labels, evaluated on the remaining 1,648 samples (Appendix S.2).

Table 8: **Training on multiple source models is what makes the calibrator transfer.** Cross-model transfer. AUROC when training on single vs. multiple source models. Multi-model training yields a 12–21 point gain over single-source training. Single-source rows evaluate on held-out target models the calibrator never trained on (the diagonal is N/A).

Training data	GPT-5-mini	GPT-5.2	Qwen3.5	Mean
GPT-5-mini only	N/A	0.715	0.693	0.704
GPT-5.2 only	0.741	N/A	0.776	0.759
Qwen3.5 only	0.658	0.687	N/A	0.672
All three (ours)	0.882	0.877	0.873	0.878

F.2 Cross-model training

Table 8 shows that training on multiple source models, rather than a single one, is what makes the calibrator transfer: each single-source calibrator is evaluated on held-out target models it never trained on.

F.3 Input and robustness ablations

Table 9 decomposes what the calibrator reads (the question versus the response, and word-versus sentence-level shuffling of the response), and Table 10 shows that stripping or injecting explicit hedge words barely moves its score.

F.4 Per-benchmark breakdown

Table 11 reports per-benchmark AUROC and effect size on the held-out test set.

F.5 Response length analysis

The residualized analysis in Table 12 uses negative log-length as a fixed score. The separately evaluated response-length baseline in Table 7 fits logistic regression on log-length and reports 0.566 AUROC; the two entries therefore use different scoring rules.

F.6 Difficulty stratification

Table 13 breaks down calibrator performance by benchmark difficulty tier.

Table 9: **The calibrator reads response meaning, not surface wording, and draws on both the question and the answer.** Model knowledge vs. linguistic cues. Blanking the question leaves most of the signal intact, while word-level shuffling, which keeps the response’s words but destroys their meaning, degrades AUROC far more than sentence-level shuffling. Brackets give 95% bootstrap CIs. “Empty question” leaves the question field blank; “question redacted” replaces it with the literal string “[QUESTION REDACTED]”; both use the trained calibrator without retraining, changing only the inference-time question field.

Input	AUROC $\uparrow$	Δ
Question + response (canonical)	0.878	–
Sentence-shuffle response	0.866	–0.012
Word-shuffle response	0.781 [.760, .801]	–0.097
Response only, empty question	0.804 [.783, .823]	–0.074
Response only, question redacted	0.817 [.796, .835]	–0.061

Table 10: **Explicit hedge words barely move the calibrator’s score.** Hedging probe. Stripping or injecting explicit hedge words (“maybe”, “I think”, “possibly”) barely changes AUROC; the calibrator leverages signals beyond superficial hedging. The last column is the mean shift in predicted $P(\text{correct})$.

Condition	AUROC	Δ vs original	Mean shift in $P(\text{correct})$
Original responses	0.8778	–	–
Hedging stripped	0.8793	+0.0015	+0.0027
Hedging injected	0.8593	–0.0185	–0.049

F.7 Difficulty-controlled disagreement test

To isolate response-level correctness from question-level difficulty, we select every test question that multiple source models attempted and on which at least one model was correct and at least one was incorrect. This selection rule is fixed in advance and applies no further filtering. It yields 228 questions ($n=810$ graded responses) on which difficulty is held constant by construction: a predictor that uses only the question scores 0.5 AUROC. On this set the calibrator reaches 0.841 AUROC. We assess significance with a within-question permutation test that shuffles correctness labels among the responses to each question (preserving the per-question difficulty structure) and recomputes AUROC; the observed value exceeds all permuted values ($p < 0.0001$). The calibrator therefore reads correctness from the (question, response) pair, not difficulty alone.

F.8 Leave- K -out cross-validation

Table 14 reports per-fold results for leave- K -out cross-validation ($K = 5, 4$ folds). The mean gap between in-distribution and held-out AUROC is +0.003.

F.9 Elicitation strategy ablation

Table 15 compares different ways to extract the uncertainty signal from calibrator checkpoints on an internal validation split (absolute values therefore differ from the final model in Table 7). The standard logit-based approach (softmax over the i/ii tokens) achieves 0.859 AUROC. MC Dropout [11] ($N=5$ forward passes) provides no improvement (0.859), consistent with LoRA’s low-rank perturbations providing insufficient stochasticity. Verbalized probability (prompting the calibrator to output a number) degrades to 0.749. A hidden-state MLP probe achieves 0.878 (+2 points), but requires open-weight access to the calibrator’s internals. Temperature scaling ($T = 1.34$) leaves AUROC unchanged while marginally improving ECE (0.023 to 0.019), indicating the model is already well-calibrated.

Table 11: **The calibrator discriminates correctness across every benchmark.** Per-benchmark calibrator performance on the held-out test set (question-level split, zero question overlap with training). d = Cohen’s d between calibrator scores for correct vs. incorrect responses. Sorted by AUROC descending.

Benchmark	Modality	N	Acc	AUROC	d
LiveBench	Text	102	.676	.995	4.77
MathVista	VLM	117	.333	.954	2.82
OmniMath	Text	78	.244	.929	2.26
RealWorldQA	VLM	113	.451	.911	1.77
ARC-AGI	Text	114	.412	.901	2.00
MathVerse	VLM	93	.495	.883	1.78
HallusionBench	VLM	115	.800	.875	1.65
MMMU	VLM	69	.826	.867	1.59
BBEH	Text	115	.496	.865	1.56
MathVision	VLM	81	.346	.862	1.54
VizWiz	VLM	85	.706	.846	1.44
MM-Vet	VLM	80	.338	.843	1.55
MMStar	VLM	118	.864	.809	1.26
SimpleQA	Text	124	.339	.798	1.28
CharXiv	VLM	118	.678	.728	0.85
GPQA	Text	90	.711	.726	0.74
ChemBench	Text	108	.732	.698	0.79
HLE-Multimodal	VLM	46	.174	.669	0.54
PRBench	Text	91	.648	.639	0.45
HLE	Text	96	.167	.616	0.30
Mean	–	98	.520	.821	1.46

Table 12: **Response length explains almost none of the calibrator’s signal.** Response length analysis ($N=1,953$). Length accounts for only 3.2% of the calibrator’s discriminative signal.

Metric	Value
Calibrator AUROC	0.878
Length-only baseline AUROC	0.631
Residual AUROC (length regressed out)	0.866
AUROC drop from length control	0.012
Length signal fraction	3.2%
$\rho(\text{cal, correct} \mid \text{length})$	0.638
$\rho(\text{length, correct} \mid \text{cal})$	−0.060
<i>AUROC by response length quartile</i>	
Q1 (≤ 319 chars)	0.873
Q2 (319–809)	0.891
Q3 (809–2240)	0.849
Q4 (> 2240)	0.861

G Per-model results

Table 16 shows per-model AUROC for the main comparison.

H Production use case details

AUROC measures discrimination in aggregate, but practitioners need to know: *what can I actually do with a calibrator score?* We evaluate three deployment scenarios on the same held-out test set (1,953 examples), comparing the calibrator against every baseline from the main results.

Table 13: **Calibrator performance holds across difficulty tiers.** Calibrator performance by benchmark difficulty tier.

Tier	Benchmarks	N	Mean acc	AUROC
Easy (≥ 0.65)	8	805	0.749	0.832
Medium (0.40–0.65)	5	526	0.494	0.862
Hard (< 0.40)	7	622	0.288	0.848
All	20	1,953	0.534	0.878

Table 14: **Held-out benchmarks match in-distribution performance.** Leave- K -out cross-validation ($K = 5, 4$ folds, question-level split). In-dist. and held-out AUROC for each fold. Gap = in-dist. minus held-out (positive = degradation on unseen benchmarks).

Fold	Held-out benchmarks	In-dist.	Held-out	Gap
0	GPQA, Hallusion., MathVerse, MM-Vet, VizWiz	0.879	0.866	+0.013
1	HLE, MMMU, MMStar, OmniMath, SimpleQA	0.873	0.885	−0.011
2	BBEH, CharXiv, MathVision, MathVista, RealWorldQA	0.878	0.876	+0.002
3	ARC-AGI, ChemBench, HLE-Multimodal, LiveBench, PRBench	0.879	0.872	+0.007
Mean $\pm$ std		0.877	0.875	+0.003 $\pm$ 0.009

Adaptive clarification (error detection). Consider a customer-facing chatbot where incorrect responses damage user trust. The system monitors each response and flags likely errors for human review before delivery. We measure this as a binary detection task: given a response, predict whether it is incorrect. The calibrator achieves AUPRC of 0.867 and best F1 of 0.782 across three target models, compared to 0.576 AUPRC for verbalized confidence (Table 17). We find numerous examples where the target model reports 100% verbalized confidence while the calibrator correctly assigns $P(\text{correct}) < 0.01$; all such cases are indeed incorrect (Figure 4a).

Confidence-gated actions (agentic safety). Consider an agentic pipeline where an LLM takes irreversible actions (e.g., sending emails, executing trades, modifying databases). Only responses above a confidence threshold $P(\text{correct}) > t$ are auto-executed; the rest are held for human verification. The calibrator achieves 27–35% auto-execution coverage at 90% accuracy across target models; no baseline achieves comparable coverage at the same accuracy (Figure 4b).

Human-in-the-loop escalation (enterprise triage). Consider an enterprise helpdesk where thousands of queries arrive daily but only a limited number of human reviewers are available. By reviewing low-confidence responses first (ranked by calibrator score), reviewers catch errors faster than random ordering. To reach 95% system accuracy, UQ-guided review requires examining only 79% of responses vs. 100% with random ordering, a 21% workload reduction (Figure 4c). A three-tier routing (high/medium/low confidence) assigns 35–43% of queries to auto-delivery at 88–90% accuracy, containing only 9% of total errors (Table 19).

H.1 Per-model use case breakdown

Figure 5 and Tables 18–19 break down all three use cases by target model.

H.2 Additional downstream applications

Beyond the three primary use cases above, we briefly evaluate PINOCCHIO on additional downstream tasks:

DPO pair selection. In reinforcement learning from human feedback (RLHF), training requires pairs of preferred and dispreferred responses. We use the calibrator’s confidence gap between two responses to the same question to select *informative* pairs where the

Table 15: **Simple logit-based elicitation captures nearly all available signal.** Elicitation strategy ablation evaluated on an internal validation split. Absolute AUROC values differ from Table 7 (which reports the final model on the held-out test set); relative comparisons between strategies are unchanged. Logit-based extraction captures 95%+ of the available signal.

Strategy	AUROC
Logit (softmax over i/ii)	0.859
MC Dropout (N=5)	0.859
Hidden-state MLP probe	0.878
Hidden + logit combined	0.884
Verbalized probability	0.749
Token entropy	0.520

Table 16: **PINOCCHIO outperforms baselines consistently across target models.** Per-model AUROC on held-out test set. PINOCCHIO is consistent across all target models.

Method	GPT-5-mini	GPT-5.2	Qwen3.5
Verbalized (raw)	0.645	0.637	0.569
Verbalized (Isotonic)	0.647	0.646	0.659
Response length	0.633	0.609	0.500
Combined	0.693	0.658	0.656
PINOCCHIO	0.882	0.877	0.873

calibrator is confident one is correct and the other incorrect. This achieves 88.8% informative pair accuracy, compared to random pairing which yields many uninformative pairs where both responses are correct or both incorrect.

Model selection. Given N candidate models, we use the calibrator to select the most confident model’s response for each query. With $N=3$ models, this achieves 67.4% accuracy, a +5.4% improvement over always using the single best model. The calibrator effectively routes each query to the model most likely to answer it correctly.

Data filtering. We use calibrator confidence to filter training data, retaining only high-confidence samples for downstream fine-tuning. At 50% retention, filtered accuracy reaches 90.9%, substantially higher than the unfiltered baseline. This suggests PINOCCHIO can serve as a quality filter for synthetic data pipelines.

Step-level uncertainty. We attempted to apply the calibrator at the reasoning-step level (predicting whether individual chain-of-thought steps are correct). This achieved only 0.53 AUROC, near random, indicating that the calibrator’s signal is response-level rather than step-level. Developing step-level uncertainty estimation remains an open problem.

I Domain-specific deployment details

We evaluate PINOCCHIO in three domain-specific scenarios to illustrate how calibrated uncertainty enables practical deployment decisions. In each case, we simulate a tiered routing system where the calibrator’s confidence score determines whether an LLM response is auto-delivered, reviewed by a junior expert, or escalated to a senior expert.

I.1 Healthcare deployment

Consider a telehealth platform where patients submit medical questions to an LLM. Incorrect medical advice can cause direct harm, so confidence-based routing is critical: high-confidence responses can be sent directly, moderate-confidence responses reviewed by a

Table 17: **PINOCCHIO detects incorrect responses far better than baselines.** Error detection for adaptive clarification. AUPRC and best F1 for detecting incorrect responses, averaged across three target models.

Method	AUPRC	Best F1
PINOCCHIO (ours)	0.867	0.782
Verbalized confidence	0.576	0.405
Combined baseline	0.647	0.650
Isotonic (verbalized)	0.628	0.637
Length baseline	0.534	0.642

Table 18: **The calibrator enables meaningful auto-execution coverage at strict error targets.** Confidence-gated actions: coverage at target accuracy levels, per model (v3 test set, $n = 1,953$).

Risk Profile	GPT-5-mini	GPT-5.2	Qwen3.5
Aggressive ($\leq 10\%$ error)	29.3%	35.2%	26.9%
Moderate ($\leq 5\%$ error)	14.4%	16.8%	21.3%

nurse, and low-confidence responses escalated to a physician. We evaluate on 1,006 samples from medical/scientific benchmarks (ChemBench, MMMU, HLE, GPQA, SimpleQA), achieving 0.898 AUROC with a base accuracy of 54.9%.

Clinical triage. Table 20 shows the tradeoff between AI coverage and patient safety. At the most conservative threshold (0.9), only 4.1% of auto-approved responses are wrong, but coverage drops to 27%. At 0.7, coverage increases to 43% while matching a resident physician’s error catch rate (90%).

Harm severity. Of 454 total errors, only 26 (5.7%) fall in the high-confidence tier ($p > 0.8$), while 340 (74.9%) are correctly assigned low confidence ($p < 0.5$) and would be flagged automatically. Reviewing the least-confident responses yields a number-needed-to-review (NNR) of ≈ 1.0 for the first 25 reviews, meaning nearly every reviewed response is an actual error.

Telehealth routing. We model a three-tier system with illustrative per-query costs: physician review (\$25), nurse review (\$10), AI auto-response (\$0.50). Queries with $p > 0.8$ are auto-responded, 0.5–0.8 routed to a nurse, and < 0.5 escalated to a physician. Under this model, total cost per 1,006 queries drops from \$25,150 (all-physician baseline) to \$11,654, a 54% reduction, with 26 AI-facing errors (7.7% of auto-approved).

I.2 Finance deployment

Consider an automated financial analysis tool where an LLM answers quantitative reasoning queries for investment analysts. Incorrect outputs could lead to costly trading errors, so a tiered system routes confident outputs directly, uncertain ones to a junior analyst, and low-confidence ones to a senior analyst for verification. We evaluate on 1,930 samples from quantitative reasoning benchmarks (GPQA, SimpleQA, LiveBench, BBEH, HLE, OmniMath, MathVista, MathVision, MathVerse), achieving 0.952 AUROC with a base accuracy of 43.7%.

Analyst triage. Table 21 shows error detection efficiency. Reviewing the bottom 50% by confidence catches 83% of errors, exceeding a junior analyst’s baseline catch rate (75%) at that review level.

Risk tiering. The calibrator assigns 30% of outputs to the GREEN tier (reliable) at 95.3% accuracy, 11% to YELLOW (verify) at 72.9%, and 59% to RED (unreliable) at 12.1%. The false GREEN rate is 4.7% (27 wrong outputs labeled reliable out of 571 in the GREEN tier).

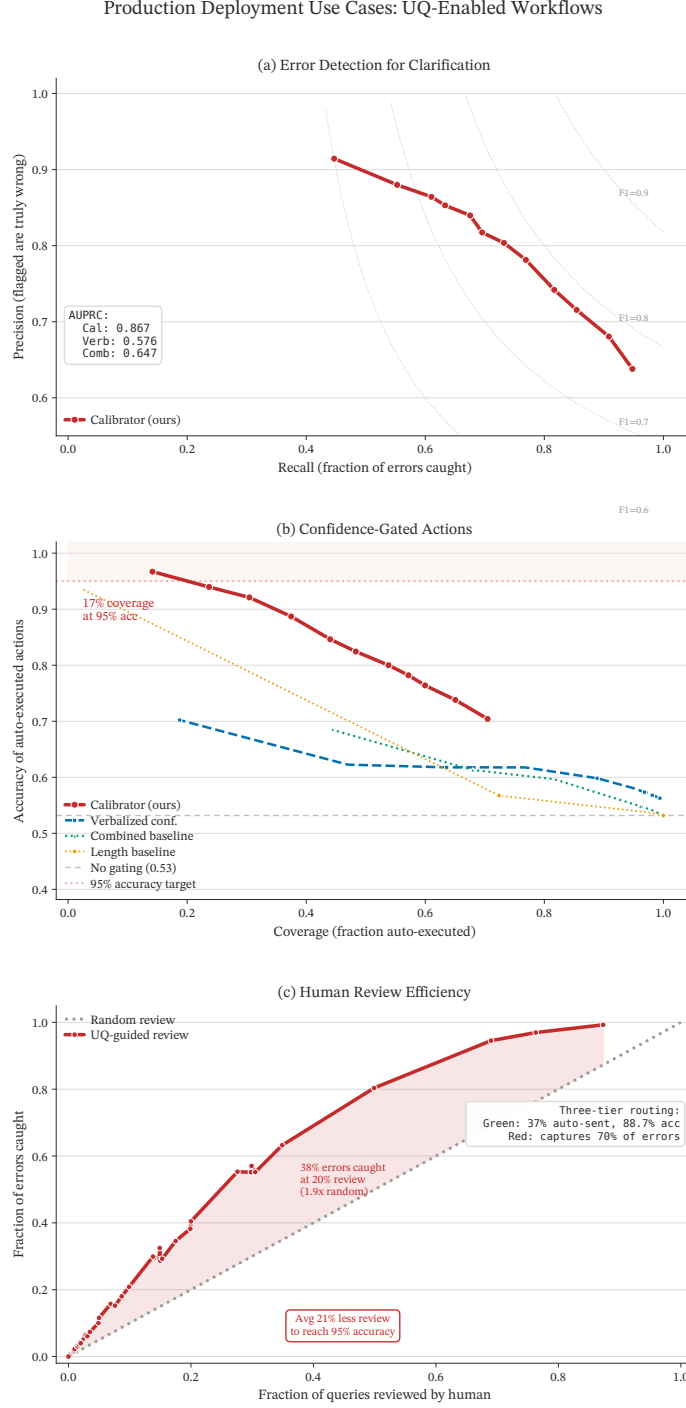


Figure 4: **Across three production workflows, the calibrator beats every baseline.** (a) Error detection: calibrator AUPRC = 0.867 vs. verbalized 0.576. (b) Confidence-gated actions: only the calibrator achieves meaningful coverage at $\geq 90\%$ accuracy. (c) Human escalation: UQ-guided review reduces workload by 21% to reach 95% accuracy.

Robo-advisor routing. We model a three-tier system with illustrative per-query costs: senior analyst (\$50), junior analyst (\$20), AI auto-response (\$0.50). Queries with $p > 0.8$ are auto-delivered, 0.5–0.8 reviewed by a junior analyst, and < 0.5 escalated to a senior analyst.

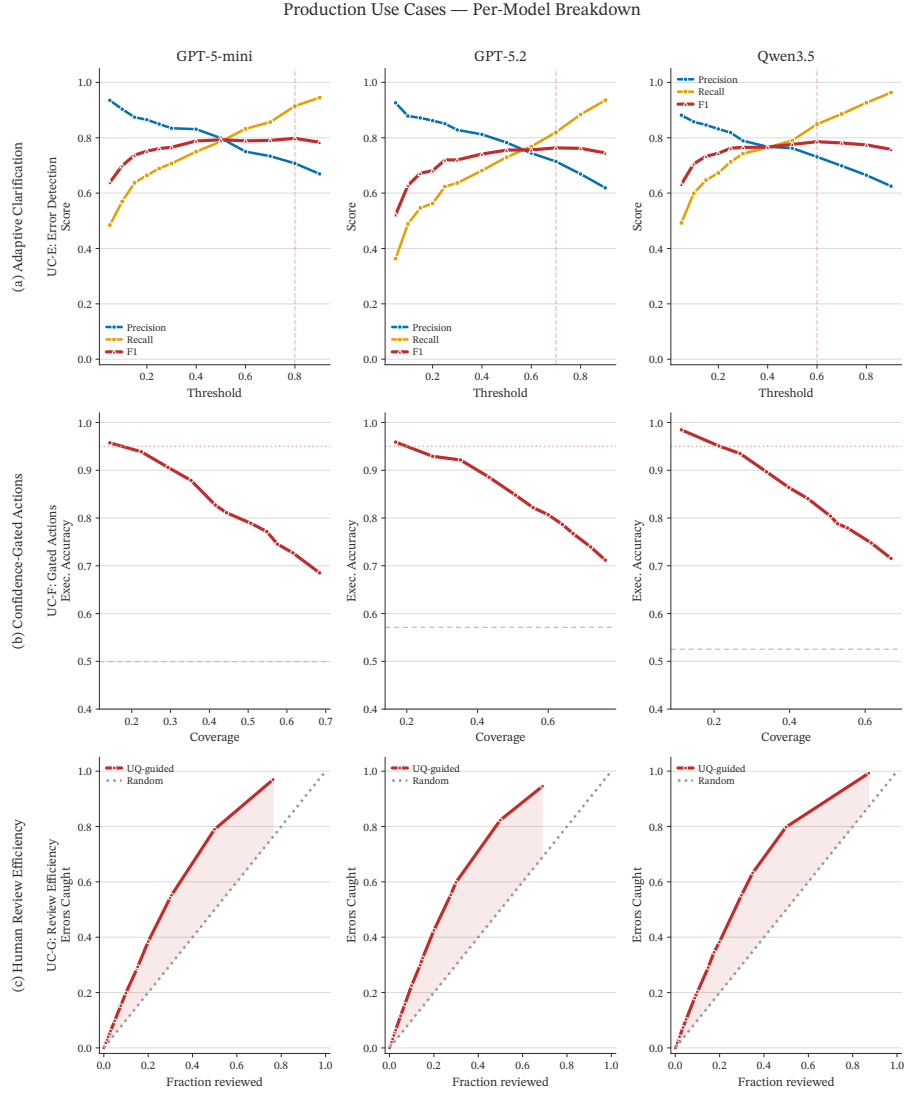


Figure 5: The three production use cases hold consistently across every target model. Top row: error detection (precision, recall, F1 vs. threshold). Middle row: coverage vs. accuracy for confidence-gated actions. Bottom row: human review efficiency (UQ-guided vs. random). Columns: GPT-5-mini, GPT-5.2, Qwen3.5. Results are consistent across all target models.

Under this model, total cost per 1,930 queries drops from \$96,500 (all-senior baseline) to \$60,244, a 38% reduction, with 27 AI-facing errors (4.7% of auto-approved).

J Cross-domain transfer to text-to-image evaluation

We test whether PINOCCHIO can transfer to a fundamentally different domain: detecting failure modes in text-to-image (T2I) generation. Using the FineGRAIN benchmark [20], which evaluates T2I outputs across 27 failure modes with human-annotated labels, we frame T2I evaluation as visual QA (“Does this image accurately depict: *[prompt]*?”) and evaluate on 3,750 samples across 5 T2I models.

Without any T2I-specific training, the calibrator achieves AUROC 0.736 [0.719, 0.751]. With domain adaptation (leave-one-model-out CV on 5 T2I models), AUROC improves to 0.953

Table 19: **Confidence-based routing cuts review workload while concentrating errors in the red tier.** Human escalation: three-tier routing and workload reduction, per model (question-level split).

	GPT-5-mini	GPT-5.2	Qwen3.5
Green tier volume	35%	43%	34%
Green tier accuracy	0.879	0.885	0.897
Green tier error share	9%	12%	7%
Red tier error share	71%	64%	74%
Review for 95% acc (UQ)	78%	79%	79%
Review for 95% acc (random)	100%	100%	100%
Workload reduction	22%	21%	21%

Table 20: **Confidence thresholds trade coverage for safety in clinical triage.** Healthcare clinical triage. Resident physician baseline catches 90% of errors when reviewing all responses.

Threshold	Coverage	Auto Error Rate	Catch Rate	$\geq$ Resident?
0.5	59%	19.3%	75%	No
0.6	50%	13.2%	85%	No
0.7	43%	10.4%	90%	✓
0.8	34%	7.7%	94%	✓
0.9	27%	4.1%	98%	✓

± 0.028 . This suggests PINOCCHIO’s learned correctness signals generalize beyond LLM text evaluation to visual quality assessment.

K Error analysis and failure cases

Of 1,953 test examples, the calibrator makes 116 “hard errors”: 48 confident-but-wrong ($p > 0.9$, incorrect; 2.5% of test set) and 68 unconfident-but-right ($p < 0.1$, correct; 3.5%). Table 22 shows the distribution across benchmarks.

Systematic patterns. The confident-but-wrong failures cluster in *visual reasoning* benchmarks (RealWorldQA, VizWiz, MMStar, CharXiv) where the target model produces a plausible-sounding answer that is visually wrong. These are cases where the response text appears well-formed and confident, giving the calibrator insufficient signal that the answer is wrong.

The unconfident-but-right failures concentrate in benchmarks with terse or unconventional correct answers (SimpleQA, RealWorldQA, MM-Vet, ARC-AGI, MathVista, HLE). This suggests the calibrator has learned to associate longer, more detailed reasoning with correctness, penalizing correct but brief responses.

L Statistical significance

We assess AUROC differences using paired comparisons only when methods are evaluated on the same examples. The full-set baselines use the held-out test set of 1,953 samples; the two proxy sampling baselines use the 1,376-example text-only subset and are excluded from the full-set significance statements below.

DeLong’s test [9] compares correlated AUROC curves directly. Comparisons between PINOCCHIO and each full-set baseline yield $p < 0.001$.

Bootstrap confidence intervals. We compute 2,000 BCa bootstrap resamples of AUROC for the full-set methods. PINOCCHIO’s 95% CI [0.863, 0.892] does not overlap with the full-set baseline intervals. The best full-set baseline is Combined at 0.649 [0.624, 0.673].

Table 21: **Confidence-ranked review catches errors faster than a junior analyst.** Finance analyst triage. Junior analyst baseline catches 75% of errors when reviewing all outputs.

Review %	Errors Caught	Errors Slipped	Catch Rate	$\geq$ Junior?
10%	193	894	18%	No
20%	383	704	35%	No
30%	567	520	52%	No
40%	744	343	68%	No
50%	899	188	83%	✓

Table 22: **Hard errors are rare and spread thinly across benchmarks.** Failure case distribution by benchmark. CW = confident-but-wrong ($p > 0.9$, incorrect); UR = unconfident-but-right ($p < 0.1$, correct). Benchmarks sorted by total failure count.

Benchmark	CW	UR	Total
RealWorldQA	7	8	15
SimpleQA	1	9	10
MM-Vet	1	8	9
VizWiz	6	3	9
ARC-AGI	1	7	8
MMStar	5	3	8
MathVista	1	7	8
CharXiv	5	2	7
BBEH	1	4	5
HLE	0	5	5
All others (10 benchmarks)	20	12	32
Total	48	68	116

Effect sizes. Figure 6 visualizes the AUROC difference between the calibrator and each full-set baseline with 95% bootstrap CIs. The smallest gap is +0.229 against Combined, and all displayed differences are significant at $p < 0.001$.

M Grading robustness and self-preference

Fifteen of the twenty benchmarks use exact-match or programmatic grading with no LLM judge, so grader self-preference cannot affect the majority of our results. The remaining five use rubric-based grading with an LLM judge (GPT-5-mini). To check whether a GPT-family judge favors GPT-family responses, we re-graded these five with five lab-independent judges from five organizations: Claude Sonnet 4.6 (Anthropic), Llama-3.1-70B (Meta), granite-4.1-30b (IBM), Olmo-3.1-32B (AllenAI), and Gemma-4-31B (Google); two of the five (Claude, Gemma) are neither GPT- nor Qwen-family. Pairwise inter-judge agreement (Cohen’s κ) tracks rubric subjectivity rather than grader identity (Table 23): high on the factual rubric (HLE $\kappa = 0.54$) and near zero on the subjective rubrics (TutorBench 0.15, HealthBench 0.21, PRBench 0.28), uniformly across every judge pair. Were GPT-5-mini self-prefering, the non-GPT judges would agree with one another more than with it; they do not, across all ten judge pairs. The low rubric agreement matches the ~ 63 – 66% inter-human agreement reported by Zheng et al. [58] on subjective grading, i.e. expected human-level noise rather than a bias signature.

N Benchmark descriptions

N.1 Text benchmarks

- **BBEH** [25] (Big-Bench Extra Hard): 23 challenging reasoning tasks from BIG-Bench requiring multi-step inference, logical deduction, and causal reasoning. Frontier models achieve $\sim 50\%$ accuracy.

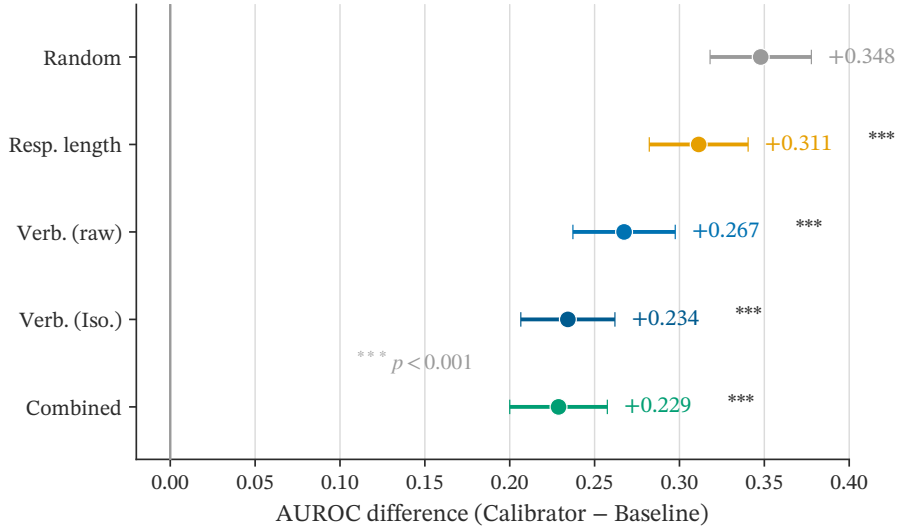


Figure 6: **PINOCCHIO beats every full-set baseline by a wide, significant margin.** AUROC difference (calibrator minus baseline) with 95% bootstrap CIs. The two proxy sampling baselines, evaluated on a different text-only subset, are excluded. All displayed differences are significant at $p < 0.001$.

Table 23: **Inter-judge agreement follows rubric subjectivity, not grader identity.** Mean pairwise Cohen’s κ across five lab-independent judges on the five LLM-graded benchmarks; agreement is high on the factual rubric and near zero on the subjective ones, uniformly across judge pairs.

Benchmark	Mean pairwise κ
HLE (factual)	0.54
PRBench	0.28
HealthBench	0.21
TutorBench	0.15

- **GPQA Diamond** [40]: 198 PhD-level multiple-choice questions in physics, chemistry, and biology, written by domain experts. Even expert humans achieve only $\sim 65\%$ accuracy.
- **OmniMath** [12]: Olympiad-level competition mathematics problems requiring symbolic reasoning and multi-step proofs.
- **SimpleQA** [47]: Short-form factual questions with unambiguous, verifiable answers. Tests factual recall rather than reasoning.
- **HLE** [37] (Humanity’s Last Exam): Expert-level questions across diverse academic domains, designed to be at the frontier of model capabilities.
- **LiveBench** [48]: Continuously updated benchmark with fresh questions, minimizing contamination risk. Covers math, coding, reasoning, and data analysis.
- **ChemBench** [34]: Chemistry knowledge and reasoning spanning organic, inorganic, and physical chemistry.
- **PRBench** [1]: Professional reasoning benchmark with expert-written rubrics spanning legal and finance domains. Uses LLM-as-judge grading.
- **ARC-AGI** [7]: Abstraction and Reasoning Corpus requiring novel pattern completion on grid-based visual tasks. Tests generalization to unseen transformation rules.

Table 24: **The benchmark mixture is broad and multimodal.** Benchmark mixture spanning seven domains and two modalities.

Domain	Benchmarks	Modality
Mathematics	OmniMath, MathVista, MathVerse, MathVision	T+V
Science	GPQA Diamond, ChemBench	T
Professional and coding	PRBench, ARC-AGI	T
General	SimpleQA, LiveBench, BBEH	T
Multimodal	MMMU, MMStar, MM-Vet, CharXiv, VizWiz	V
Spatial	RealWorldQA, HallusionBench	V
Frontier	HLE, HLE-Multimodal	T+V

N.2 Vision-language benchmarks

- **MMMU** [53]: Multimodal questions requiring college-level knowledge across 30 subjects including art, science, engineering, and medicine. Images include diagrams, charts, and photographs.
- **MMStar** [6]: Vision-indispensable questions specifically designed so that text-only models cannot solve them, ensuring genuine visual reasoning is required.
- **CharXiv** [46]: Questions about scientific figures and charts extracted from arXiv papers, requiring chart comprehension and numerical reasoning.
- **HallusionBench** [16]: Diagnostic suite for visual hallucination and illusion, testing whether models fabricate visual details or fall for optical illusions.
- **MathVista** [33]: Visual math reasoning combining geometric diagrams, statistical charts, and function plots with mathematical problems.
- **MathVerse** [55]: Math problems where the visual diagram is necessary for correct interpretation; removing the image makes problems unsolvable.
- **MathVision** [44]: Competition-level math problems with diagram dependencies, sourced from mathematical olympiads.
- **RealWorldQA** [49]: Spatial reasoning about real-world photographs, testing understanding of 3D scenes, object relationships, and physical properties.
- **VizWiz** [18]: Visual questions captured by blind users using smartphone cameras, including unanswerable queries due to image quality issues.
- **MM-Vet** [52]: Open-ended visual questions evaluating integrated capabilities including recognition, OCR, knowledge, spatial awareness, and language generation. Uses LLM-as-judge grading.

O Prompt templates

O.1 Calibrator prompt (combined format)

Benchmark: {benchmark_name}
Source model: {source_model}

Question: {question}

Answer: {response}

Is the answer correct? (i) No (ii) Yes

For VLM benchmarks, we prepend the image to the question in the model’s native multi-modal format. For text-only benchmarks, we use a 28×28 gray placeholder image.

O.2 Using the calibrator

Using the calibrator. PINOCCHIO is released as a pip package (pip install pinocchio-ug; source at github.com/khayes95/pinocchio, weights at huggingface.co/KevinDavidHayes/pinocchio-0.8b). The released checkpoint is the 0.8B variant, which runs on CPU and retains nearly all of the 8B model’s accuracy (Section 4.5). It wraps an existing API call: the judge downloads its weights on first use and returns $P(\text{correct})$ from a single forward pass. The two additional lines referenced in the abstract are instantiating the judge and calling score.

```
from openai import OpenAI
from pinocchio import Pinocchio

client = OpenAI()
judge = Pinocchio()                                # downloads the 0.8B weights

messages = [{"role": "user", "content": question}]
response = client.chat.completions.create(model="gpt-5", messages=messages)

# score the API response directly...
p_correct = judge.score(response, messages=messages)
# ...or from raw question/answer strings:
p_correct = judge.score(question=question, answer=answer)
```

O.3 Verbalized confidence prompt

You answered the following question:

Question: {question}
Your answer: {response}

How confident are you that your answer is correct?
Respond with only a number between 0 and 100,
where 0 means certainly wrong and 100 means
certainly correct.

Confidence:

P Baseline implementation details

All post-hoc baselines use scikit-learn with a fixed random seed of 42 and a 50/50 train/test split (stratified by correctness label) for fitting. We did not tune hyperparameters; we use library defaults except where noted.

Verbalized confidence (raw). The target model’s self-reported confidence, elicited via the prompt in Appendix O. Parsed as a float in $[0, 1]$; used directly as $\hat{p}(\text{correct})$ with no post-processing.

Platt scaling (logistic regression). A single-feature logistic regression fit on the verbalized confidence score: `LogisticRegression(C=1.0, solver='lbfgs', max_iter=1000)`. The sigmoid output serves as the calibrated $\hat{p}(\text{correct})$. This is a standard two-parameter recalibration [38].

Isotonic regression. A non-parametric monotone recalibration of verbalized confidence: `IsotonicRegression(y_min=0.01, y_max=0.99, out_of_bounds='clip')`. Unlike Platt scaling, isotonic regression makes no parametric assumptions about the calibration curve and can correct non-sigmoid miscalibration. The number of segments is determined automatically by the pool-adjacent-violators algorithm (typically 10–30 pieces for our sample sizes).

Response length. The feature is $\log(1 + t)$ where t is the response token count, fit with the same logistic regression configuration as Platt scaling. The log transform handles the heavy-tailed length distribution (response lengths span ~ 10 to $>5,000$ tokens). This baseline tests whether response verbosity alone predicts correctness.

Combined (verbalized + length). A two-feature logistic regression on $[\text{verbalized_confidence}, \log(1 + t)]$, using the same LogisticRegression configuration. Fit on a 50/50 train/test split to avoid overfitting the two-dimensional feature space.

LLM-as-judge (GPT-5-mini). A zero-shot prompt asks GPT-5-mini to estimate $P(\text{correct}) \in [0.0, 1.0]$ given only the question and response (no reference answer). We send requests asynchronously with up to 20 concurrent API calls. Response parsing: (1) attempt direct float conversion; (2) regex search for a decimal in $[0, 1]$; (3) if the response contains “yes”/“correct”, assign 0.8; “no”/“incorrect”, assign 0.2; (4) otherwise assign 0.5. The fallback values are not optimized but are sufficient for AUROC evaluation: since AUROC depends only on rank ordering, any values satisfying $\text{no} < \text{fallback} < \text{yes}$ yield identical discrimination. The 0.5 fallback approximates the dataset base rate (53.4% correct). We truncate questions and responses to 1,500 and 800 characters, respectively, matching the calibrator’s input format.

Proxy semantic entropy. We generate $N=5$ responses from Qwen3-VL-8B (temperature 0.7) for each question and compute the negative entropy of the semantic cluster distribution as a confidence score, following Kuhn et al. [29]. We call this a *proxy* because the generations come from a different model than the target: since closed-source API models make repeated sampling prohibitively expensive, we use a small open-weight model as a stand-in. If the proxy model’s uncertainty were informative about the target model’s correctness, this would provide a cheap alternative to true semantic entropy. Evaluated on the 1,376 text-only test samples (VLM benchmarks excluded due to image input requirements).

Proxy self-consistency. Using the same $N=5$ proxy generations from Qwen3-VL-8B, we compute the fraction that agree with the target model’s response as a confidence score. We measure agreement by exact string match after normalization. As with proxy semantic entropy, this tests whether consistency among a proxy model’s responses predicts correctness of the target model. Evaluated on the same 1,376 text-only samples.

Q Faithful same-target sampling baselines

The proxy semantic-entropy and self-consistency baselines in Table 7 sample a stand-in open model rather than the target. This is a low-cost workaround for closed APIs, but it does not satisfy the methods’ same-target assumption. We therefore also sample an open target directly, LLaMA-3.1-8B-Instruct, on the hard test suite (700 questions) and compute each method from $N=10$ generations. We use the official implementations of semantic entropy [29], the black-box estimators of Lin et al. [32], SPUQ [13], and the logit-free conformal method of Su et al. [42].

Every faithful baseline stays near random on the hard suite (Table 25), and the gap to PINOCCHIO is significant for all of them. Increasing the sample budget does not change this: same-target semantic entropy is 0.476/0.486/0.477 at $N=10/20/40$. The pipeline is not broken; on the short-form QA benchmarks where these methods were designed it behaves as expected (TriviaQA semantic entropy 0.740, BoolQ 0.679, NaturalQA 0.629). The near-random result on the hard suite is a domain limitation of sampling-based uncertainty, not a setup artifact. We observe the same pattern on two further open targets (DeepSeek-R1-Distill-Qwen-32B, semantic entropy 0.479) and on a frontier closed target with stronger accuracy (Claude Sonnet 4.6, semantic entropy 0.489 at 25.3% accuracy), which rules out the explanation that the open targets are simply too weak.

Table 25: **Faithful same-target sampling baselines stay near random while PINOCCHIO leads.** Faithful same-target sampling baselines on LLaMA-3.1-8B ($N=10$, 700 hard-suite questions), versus PINOCCHIO scored on the same responses. Paired Δ is the AUROC difference on matched questions, with uncertainty estimated from 1,000 paired bootstrap resamples.

Method	AUROC $\uparrow$	Paired Δ vs. PINOCCHIO
Semantic entropy (NLI)	0.476	+0.335
Self-consistency (NLI)	0.477	+0.334
Lin et al. [32] (best of 3)	0.553	+0.258
SPUQ [13]	0.482	+0.329
Su et al. [42] (LofreeCP)	0.574	+0.237
PINOCCHIO (ours)	0.811	–

Table 26: **PINOCCHIO’s advantage replicates across three lab-independent targets.** Faithful same-target sampling baselines on three lab-independent targets. Each baseline is run directly on $N=10$ generations from the target model itself on the hard suite, satisfying the methods’ same-target assumption. Lin (best) is the strongest of NumSemSets / Deg / EigV. The bottom row is the paired AUROC difference between PINOCCHIO and the strongest baseline for that target (1,000 paired bootstrap resamples).

Method	LLaMA-3.1-8B ($n=700$)	Granite-4.1-30B ($n=1576$)	Devstral-2-24B ($n=1574$)
Semantic entropy (NLI)	0.476	0.558	0.514
Self-consistency (NLI)	0.477	0.559	0.514
Lin et al. [32] (best)	0.553	0.534	0.520
SPUQ [13]	0.482	0.588	0.553
Su et al. [42] (LofreeCP / conformal)	0.574	0.521	0.497
PINOCCHIO (ours)	0.811	0.806	0.836
Paired Δ vs. best baseline	+0.237	+0.218	+0.253

Granite-4.1-30B and Devstral-2-24B are text-only; VLM benchmarks are excluded for those two targets.

The result also replicates across lab-independent targets at larger scale (Table 26): on Granite-4.1-30B ($n=1576$) and Devstral-2-24B ($n=1574$), every faithful baseline stays near random while PINOCCHIO reaches 0.806 and 0.836, paired $\Delta = +0.218$ and $+0.253$.

Table 27 breaks the hard suite down by benchmark for LLaMA-3.1-8B. PINOCCHIO wins in aggregate by a wide margin and on the hardest long-form benchmarks (hle 0.869, livebench 0.951), where the sampling methods fall to or below random. On a few lower-accuracy benchmarks whose answers are short enough that response diversity is informative, a sampling baseline is competitive or better (SPUQ on omnimath and simpleqa, the graph measure of Lin et al. [32] on bbeh), which is consistent with sampling-based uncertainty working precisely when consistency tracks correctness.

The per-benchmark SE/SC values also expose a second result: where sampling methods fail, they fail by *inverting* below random, and the inversion is domain-specific rather than uniform. Strong inversion concentrates on benchmarks where the model commits to a single long-form wrong trajectory across all N samples (bbeh 0.352, livebench 0.404); it weakens where some answer-space diversity survives (chembench 0.493, gpqa 0.516); and it disappears on omnimath (0.582), whose short numerical answers paraphrase cleanly enough for NLI clustering to work. A wiring error would invert every benchmark uniformly. The pattern instead follows the two mechanisms behind the collapse: confidently wrong trajectories that repeated sampling cannot escape, and NLI-based semantic clustering (DeBERTa-large-MNLI, trained on short sentence pairs) failing on multi-paragraph reasoning, where it neither merges paraphrases of the same correct answer nor separates distinct wrong answers of similar surface form.

Table 27: **PINOCCHIO wins overall and dominates on long-form benchmarks.** Per-benchmark AUROC on the hard suite for LLaMA-3.1-8B ($N=10$, $n=100$ per benchmark, text-only). SE and SC are NLI-based semantic entropy and self-consistency; Lin is the best of the three graph measures of Lin et al. [32]. Bold marks the best method on each benchmark. PINOCCHIO is scored black-box on the same responses.

Benchmark	n	Acc%	SE	SC	Lin	SPUQ	LofreeCP	PINOCCHIO
bbeh	100	3	0.352	0.347	0.644	0.478	0.629	0.641
chembench	100	43	0.493	0.517	0.468	0.568	0.579	0.583
gpqa	100	29	0.516	0.517	0.534	0.556	0.506	0.598
hle	100	14	0.532	0.532	0.503	0.549	0.630	0.869
livebench	100	53	0.404	0.405	0.682	0.422	0.541	0.951
omnimath	100	6	0.582	0.582	0.709	0.876	0.766	0.783
simpleqa	100	2	0.441	0.462	0.495	0.827	0.776	0.791
Overall	700	21	0.476	0.477	0.537	0.482	0.574	0.811

Table 28: **Even white-box log-probabilities carry no usable correctness signal on the hard suite.** White-box sequence-likelihood baselines on LLaMA-3.1-8B with full logit access, on the hard suite. Even with direct access to the target’s token log-probabilities, sequence likelihood carries no usable correctness signal on hard reasoning, whereas PINOCCHIO (black-box, scored on the same responses) reaches 0.811.

Method	AUROC $\uparrow$
Mean log-prob (= $-\text{perplexity}$)	0.413
Min-token log-prob	0.447
Sum log-prob	0.521
Token entropy	0.520
Verbalized confidence	0.610
PINOCCHIO (ours)	0.811

Even *white-box* access to the target’s token log-probabilities gives no usable correctness signal on the hard suite (Table 28): on LLaMA-3.1-8B, mean log-probability scores 0.413 AUROC (below chance), and no sequence-likelihood statistic exceeds verbalized confidence (0.610), while PINOCCHIO, scored black-box on the same responses, reaches 0.811.

R Computational resources

Training efficiency. The use of LoRA adapters (rank 32) reduces trainable parameters to $<1\%$ of the base model, enabling training on 4 GPUs with standard VRAM ($\geq 40\text{GB}$ per GPU for VLM training with images).

Inference cost. At inference, PINOCCHIO requires a single forward pass through the 8B calibrator model per query (~ 0.1 seconds on a single GPU). This contrasts with sampling-based methods like semantic entropy [29] that require 5–10 forward passes through the *target* model, representing a 5–10 $\times$ reduction in cost per uncertainty estimate.

Smaller models. Our model size ablation (Figure 7) shows that a 0.8B parameter model retains 99% of the largest (9B) model’s AUROC (0.863 vs. 0.871), making PINOCCHIO viable for edge deployment at $\sim 4\text{GB}$. This means uncertainty estimation can run on-device without a GPU, enabling offline or latency-sensitive applications.

Table 29: **Training and evaluation are inexpensive.** Approximate training and evaluation times.

Task	Time
Unified calibrator training (10,892 examples)	2–3 hours
Test set evaluation (1,953 examples)	<30 minutes
Use case scoring (1,953 examples)	<1 hour

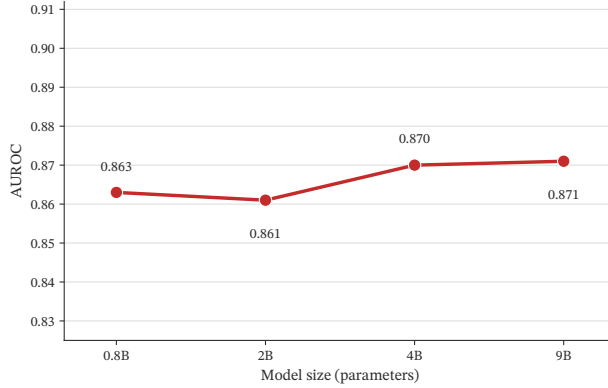


Figure 7: **AUROC is nearly flat across calibrator sizes.** Calibrator model size. AUROC across the Qwen3.5 calibrator family (0.8B to 9B) is nearly flat: the smallest 0.8B model retains 99% of the 9B model’s AUROC (0.863 vs. 0.871). Training-set size (Figure 8) has a much larger effect.

S Additional main-body figures and tables

S.1 Training scale and input ablations

Performance improves most rapidly up to $N=1,000$ and then shows diminishing returns. Removing benchmark and source-model metadata reduces AUROC from 0.878 to 0.827. Removing only the source-model token gives 0.863 AUROC, with larger reductions on Qwen3.5 and VLM benchmarks. Across five random seeds, mean AUROC is 0.861 ± 0.018 . Word- and sentence-level response shuffling are reported in Table 9.

Vision-language data as text augmentation. Table 31 reports the text-budget sweep from Section 4.3: calibrators trained on N text examples, with and without the full vision-language training set added, all at matched configuration (LoRA rank 32, seed 42, two epochs) and evaluated on the same held-out text benchmarks.

Response length. The response content is load-bearing: truncating it degrades AUROC monotonically (Table 32). At a 200-character budget AUROC falls to 0.805, and beyond the 800-character canonical setting it continues to rise slightly, so more of the response helps.

S.2 Per-model recalibration

For each target model, we fit Platt scaling and isotonic regression on 100 labeled examples and evaluate on the remainder. The procedure changes the mapping from scores to probabilities without retraining the calibrator.

Platt scaling preserves AUROC because it is monotone. On the three source models, isotonic regression introduces ties while reducing aggregate ECE from 0.096 to 0.063.

The same procedure applies to models absent from training entirely: on the four current-generation frontier models (Claude Fable 5, GPT-5.6, Kimi 3, Claude Opus 5, text-only),

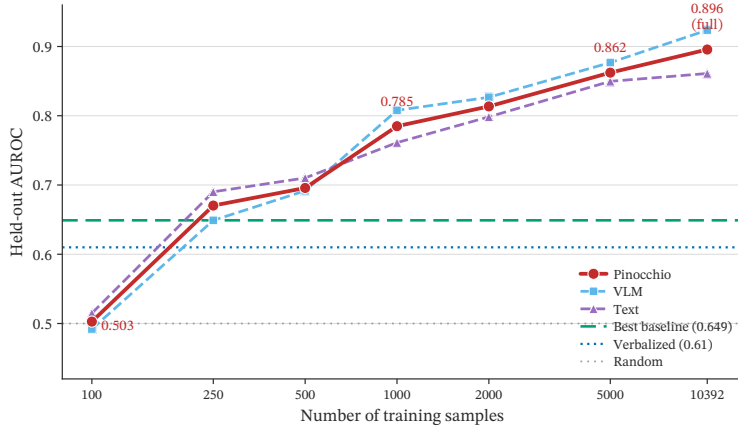


Figure 8: **More training data helps more than a larger calibrator.** Training-set size. AUROC is measured on the ablation’s held-out split, on which the full-data calibrator reaches 0.896; on the main test set it reaches the 0.878 reported throughout (Table 7). Calibrator parameter count has a much smaller effect (Figure 7).

Table 30: **The calibrator transfers to a held-out source model with only a small drop.** Leave-one-model-out (LOMO) cross-model transfer. Each row trains on two source models and evaluates on the held-out third. Δ = difference from the full three-model calibrator (0.878).

Held-out model	Training models	AUROC	Δ
GPT-5-mini	GPT-5.2 + Qwen3.5	0.877	−0.001
GPT-5.2	GPT-5-mini + Qwen3.5	0.861	−0.017
Qwen3.5	GPT-5-mini + GPT-5.2	0.790	−0.088
Mean (LOMO)		0.843	−0.035
Best baseline (combined)		0.649	−

a 100-example fit roughly halves pooled ECE while leaving AUROC unchanged (Table 4, main body). This is the empirical basis for the recalibration disclaimer in Section 5.

Training directly on the new models. The transfer numbers above use no labels from the new frontier models. When labels are available, a calibrator trained directly on the four frontier models’ own responses (same recipe and combined-metadata prompt) reaches 0.851 in-distribution text AUROC (ECE 0.081); a three-model variant reaches 0.860 with 0.880 leave-one-model-out transfer to the held-out frontier model. Direct training thus modestly exceeds the 0.837 pooled zero-shot transfer (Section 4.4), so the zero-shot results are a no-label lower bound on what a labeled deployment achieves.

Recalibration also holds when we hold out whole *domains* rather than models. We run four leave-one-domain-out folds, in each of which the calibrator excludes a domain group from training and then scores it, and we fit recalibration on about 100 labels from the held-out domain (Table 35). Recalibrating on those ~ 100 labels cuts pooled ECE several-fold, from 0.271 off-the-shelf to 0.050 with Platt scaling, and adding the target’s verbalized confidence as a feature does not consistently help beyond that. We note that AUROC on held-out domains is lower than on held-out models: domain holdout is the harder setting, and recalibration restores calibration but not the discrimination lost when an entire domain is unseen.

Error analysis. Per-benchmark AUROC varies from 0.616 (HLE) to 0.995 (LiveBench), and the variation is driven by domain alignment rather than difficulty: the Spearman rank correlation between benchmark accuracy and calibrator AUROC is just $\rho = 0.045$ ($p = 0.85$).

Table 31: **Vision-language data improves text calibration when hard text data is scarce.** Text-benchmark AUROC for calibrators trained on N text examples, with and without the vision-language training set added (matched configuration, same held-out text set). Gains are largest at small N and taper as text data grows.

N (text examples)	Text only	+ vision-language	Δ
500	0.690	0.731	+0.041
1,000	0.730	0.789	+0.059
2,500	0.791	0.815	+0.025

Table 32: **Longer response context monotonically improves AUROC.** Response-truncation ablation. AUROC as a function of the response character budget, relative to the 800-character canonical setting.

Response budget	AUROC	Δ vs canonical
$r=200$	0.8053	−0.072
$r=400$	0.8462	−0.031
$r=800$ (canonical)	0.8778	—
$r=1600$	0.8882	+0.010
$r=3000$	0.8900	+0.012

We identify two complementary failure modes. First, on frontier-difficulty benchmarks where target model accuracy is very low (HLE: 17%, HLE-Multimodal: 17%), the calibrator is *underconfident on rare correct answers*: 50–63% of correct responses receive $P(\text{correct}) < 0.2$, producing weak score separation (Cohen’s $d = 0.30$ – 0.54). Second, on domain-specialized benchmarks (PRBench, ChemBench, GPQA), the calibrator is *confidently wrong*: 17–22% of incorrect answers receive $P(\text{correct}) > 0.8$. In contrast, benchmarks involving structured mathematical reasoning (LiveBench, MathVista, OmniMath) achieve AUROC > 0.929 , likely because tasks with clear solution steps produce more distinctive textual signals for correct vs. incorrect responses. This pattern does not reduce to answer format alone: GPQA Diamond is multiple-choice yet remains among the harder benchmarks, suggesting that expert-level domain knowledge requirements matter more than surface structure.

Table 33: **The response adds discriminative signal beyond question and metadata.** Input decomposition. Starting from the question text alone, we add benchmark/source metadata and then the model response. The response adds discriminative signal beyond question and metadata, showing PINOCCHIO is an integrated calibrator rather than a pure question-difficulty regressor.

Input	AUROC $\uparrow$	Δ
Pure question (no metadata, no response)	0.783	–
+ benchmark / source metadata	0.826	+0.043
+ response (full)	0.854	+0.028

Table 34: **Platt scaling preserves AUROC while cutting ECE.** Per-model recalibration for the three source models with a 100-sample fit set. Aggregate is the unweighted mean across models.

Model	Raw		Platt		Isotonic		N_{eval}
	AUROC	ECE	AUROC	ECE	AUROC	ECE	
GPT-5-mini	0.878	0.106	0.878	0.067	0.865	0.076	551
GPT-5.2	0.875	0.078	0.875	0.062	0.870	0.050	624
Qwen3.5	0.869	0.103	0.869	0.064	0.864	0.064	473
Aggregate	0.874	0.096	0.874	0.064	0.866	0.063	N/A

Table 35: **Recalibration restores calibration even on held-out domains the calibrator never trained on.** Pooled ECE across four leave-one-domain-out folds. Off-the-shelf is the excluded-domain calibrator with no recalibration; the others fit on about 100 labels from the held-out domain. Verbal elicitation adds the target’s self-reported confidence as a feature.

Condition	Pooled ECE $\downarrow$
Off-the-shelf (no recalibration)	0.271
Domain scaling (Platt)	0.050
Isotonic	0.080
Isotonic + verbal elicitation	0.065