The ACUTE Protocol: Operationalizing Language Model Activations for Better Calibration, Utility, and Trust

Anonymous Authors[1]

Abstract

As language models improve and become increasingly deployed to solve a variety of tasks, trustworthiness becomes essential. Calibration is a good proxy for trust: well-calibrated confidence estimates help inform the risk versus reward tradeoff when trusting a specific model output. Unfortunately, even as models improve, they remain poorly calibrated, often biasing towards overconfidence. Additionally, calibration can be gamed: a policy that always predicts the base rate is perfectly calibrated, but completely uninformative. To resolve this, we develop a new metric, expected utility renormalized by the oracle (EURO), that balances calibration and informativeness. We also propose a general-purpose **a**ctivation-based **c**onfidence, **u**tility, and **t**rust **e**stimation protocol (ACUTE) to appropriately adjudicate uncertainty. The ACUTE protocol provides flexible, sample-efficient, and compute-efficient confidence estimators for 3 tasks including multiple choice question answering, tool-calling, and scientific document summarization across 6 models from 4 model families. ACUTE outperforms strong baselines on EURO, while maintaining low calibration error. Taken together, our work shows that equipping LLMs with the ACUTE protocol can improve calibration, utility, and trustworthiness in numerous settings.

1. Introduction

Users increasingly rely on large language models (LLMs) for many tasks such as information seeking, writing, and agentic tool-calling. Many technological solutions integrate LLMs into their workflow at many stages, trusting and using model output for further computation. As a result, automatically knowing whether to trust a model output is essential.

A confidence estimator is often an auxiliary model that estimates the probability that the output of another model (*e.g.*, an LLM) is correct. *Well-calibrated* confidence estimates can provide a signal of whether to trust a generation and what the associated risks could be. Practitioners often rely on a simple confidence estimator to evaluate whether to trust an LLM output: the probability the LLM assigns to the generation (*i.e.*, the product of token probabilities), even though this is known to be *poorly calibrated* in both single- and multi-token generation settings (Desai & Durrett, 2020; Jiang et al., 2021; Zhong et al., 2023; Stengel-Eskin & Van Durme, 2023a; Subramani et al., 2025).

We want a metric to adjudicate how much to trust a model generation, while accounting for how risky the task is (*i.e.*, for a high risk task we require more certainty to trust an LLM output than a low risk task). Calibration, often measured using expected calibration error (ECE; Murphy & Epstein (1967); Naeini et al. (2015)), solely measures how well predicted probabilities match observed outcomes. This has *two major drawbacks* for trust:

ECE cannot distinguish between an oracle and a perfectly uninformative estimator that just predicts the base rate for all outputs independent of correctness. Suppose we have a task that an LLM gets exactly 50% accuracy on. Both an oracle estimator (assigns a probability of 1 to correct and 0 to incorrect outputs) and a baserate estimator (assigns the base rate of 0.5 to all outputs) achieve an ECE of 0 (Figure 1).

ECE is invariant to the associated risk of a task. Suppose there exists two tasks with different risks: high risk requiring 0.9 probability to trust and medium risk requiring only 0.5 to trust and a probabilistic oracle (*e.g.*, one that assigns 0.75 to correct and 0.25 to incorrect outputs). This estimator gets an ECE of 0.25 for both tasks, even though it perfectly solves the medium-risk task, and fails on the high risk task (never trusting a single output).

Our contributions are:

1. A new general and easily interpretable metric called **expected utility renormalized by the oracle** (EURO), which balances calibration and decision-making util-

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

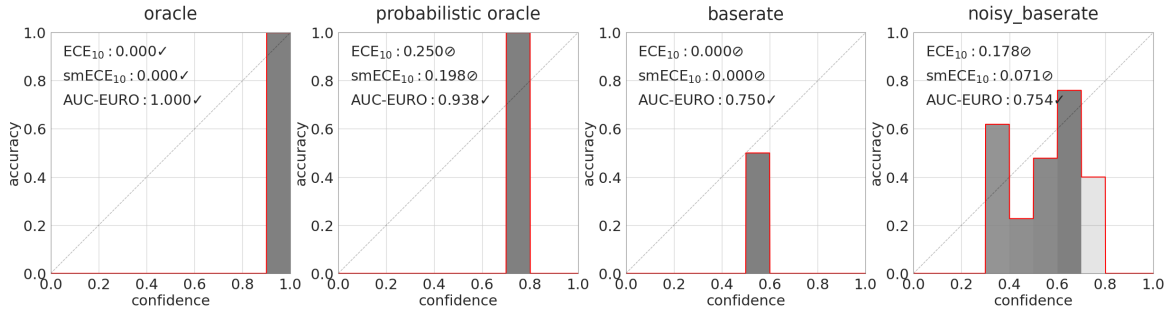

Figure 1. Here we show four different confidence estimators in order of decreasing quality and how the metrics ECE, smECE, and our metric AUC-EURO adjudicate them. Both ECE and smECE are errors so they should start at 0 and increase, while AUC-EURO is a utility so it should decrease. ECE and smECE break and decide that the two worst systems (baserate and noisy_baserate) are *much* better than the probabilistic oracle. AUC-EURO appropriately ranks them. Details of what each estimator are in §2.2.

ity. EURO resolves both drawbacks of ECE, granting practitioners a more reliable trust estimator (§2).

2. An **activation-based confidence, utility, and trust estimation protocol** (ACUTE) to assess the confidence of language model outputs (§3). Experiments on multiple-choice question answering (MMLU; Hendrycks et al. (2021)), tool-calling (APIGen; Liu et al. (2024)), and scientific document summarization (SCITLDR; Cachola et al. (2020)) show that applying the ACUTE protocol on 6 LLMs improves confidence estimation via EURO, while maintaining low calibration error.

2. Expected Utility Renormalized by the Oracle (EURO)

2.1. Calibration, ECE, and Smooth ECE

Calibration measures how well predicted probabilities match the observed outcomes for a task. To measure how good a confidence estimator is for a task, expected calibration error (ECE) is often used (Murphy & Epstein, 1967; Dawid, 1982; Naeini et al., 2015; Naeini & Cooper, 2016). To calculate ECE, we build a histogram where bins correspond to predicted confidence bands (*e.g.*, 3 bins equate to 0.0 - $\frac{1}{3}$, $\frac{1}{3}$ - $\frac{2}{3}$, $\frac{2}{3}$ - 1.0). For every example i within a bin, the accuracy of that example is compared to the estimator's predicted confidence: $|acc_i - \hat{p}_i|$. All of these absolute differences are averaged to produce a final ECE score. The number of bins can be challenging to set and different choices lead to different calibration scores.

Błasiok & Nakkiran (2024) introduce smooth ECE (smECE) a hyperparameter-free method to remove the reliance on the number of bins. [1] SmECE smooths out the histogram using Nadaraya-Watson kernel regression and measures calibration more precisely than ECE, but suffers from the same major drawbacks that ECE does. We introduce a metric that balances calibration with informativeness called expected utility renormalized by the oracle (EURO) to assess the decision-making utility of a confidence estimator. EURO has *just one* parameter: the normalized net utility of correctly abstaining from trusting an incorrect generation (u_{ca}).

2.2. Calibration Alone is Insufficient

Suppose for some task, we have language model M that outputs predictions $y_i, \ldots, y_n$ and has 50% accuracy on these (*i.e.*, $\sum_i^n y_i = \frac{n}{2}$). Consider the following estimators:

1. oracle (1 if y_i is correct and 0 otherwise)
2. probabilistic oracle (0.75 correct, 0.25 incorrect)
3. baserate (0.5 regardless of what y_i is)
4. noisy baserate ($p \sim U(0.25, 0.75)$ regardless of y_i)

The oracle is perfect and the probabilistic oracle is indistinguishable from the oracle as long as the probability threshold $\tau \in (0.25, 0.75]$ (*i.e.*, for $\hat{p}$ emit by the estimator, trust if $\hat{p} > \tau$). Both base rate and noisy base rate estimators are uninformative for decision-making: they generate middling probabilities that are uncorrelated with correctness.

In Figure 1, we plot the reliability diagrams for these four estimators and their associated ECE, smECE, and our metric AUC-EURO, and find two concerning trends with both ECE and smECE. [2] First, they cannot distinguish between the oracle estimator and the perfectly uninformative base rate estimator (ECE=smECE=0). Our metric, AUC-EURO, however, does distinguish between them (oracle gets a score of 1, while the baserate estimator gets a score of 0.75). Secondly, the probabilistic oracle gets the worst performance score on ECE and smECE despite being a great confidence estimator. AUC-EURO appropriately scores this as worse than the true oracle, but much better than either the baserate or noisy baserate estimator.

Both ECE metrics have a third drawback: they are invariant to the associated risk of a task. Consider two tasks with dif-

[1] Kernel width is determined automatically from data.

[2] See §2.3 for details on how to calculate AUC-EURO.

ferent risk levels: a high risk task (*e.g.*, evaluating a trading agent that can buy and sell equities in a stock market) and a medium risk task (*e.g.*, scientific document summarization to use in a literature review for a course project). For the high risk task, we must be very sure the output is correct before trusting the agent to execute a trade; lets suppose that the threshold τ to trust a generation is 0.9 (*i.e.*, $\hat{p} > 0.9$ to trust). For the medium risk task, lets assume that $\tau = 0.5$. The probabilistic oracle estimator is good at abstaining for the high risk task, but it never trusts the output because all probabilities are less than 0.9. An ideal metric should give the estimator a lot of credit for abstaining often, but with some penalty for never trusting. For the medium risk task however, the probabilistic oracle estimator is perfect. Since both calibration error metrics are invariant to risk level, the estimator gets the same scores regardless of task. EURO, on the other hand, incorporates risk level; the probabilistic oracle achieves EURO = 0.9 and EURO = 1.0 for the high and medium risk tasks respectively.

2.3. EURO

For a given query q, a language model M generates a candidate solution $\hat{y}$; a confidence estimator C assigns a probability $\hat{p}$ to the example. There are four outcomes based on the Minimum Bayes Risk (MBR) decision:

1. True Positives (tp) := $\hat{p} > \tau$ and $\hat{y}$ is correct
2. False Positives (fp) := $\hat{p} > \tau$ and $\hat{y}$ is incorrect
3. False Negatives (fn) := $\hat{p} \leq \tau$ and $\hat{y}$ is correct
4. True Negatives (tn) := $\hat{p} \leq \tau$ and $\hat{y}$ is incorrect

Each of these four outcomes has an associated reward or utility: R_{tp}, R_{fp}, R_{fn}, R_{tn} that can be set by the user based on the task at hand. Given the number of true positives, false positives, false negatives, and true negatives ($N_{tp}, N_{fp}, N_{fn}, N_{tn}$), the total utility $\mathbb{U} = R_{tp} * N_{tp} + R_{fp} * N_{fp} + R_{fn} * N_{fn} + R_{tn} * N_{tn}$. As a result, the MBR decision threshold for $\hat{p}$ is defined as: [3]

$$\hat{p} > \tau \stackrel{\text{def}}{=} \frac{R_{tn} - R_{fp}}{(R_{tp} - R_{fn}) + (R_{tn} - R_{fp})} \tag{1}$$

In our setting, the confidence estimator trusts the output generation if $\hat{p} > \tau$, and abstains if $\hat{p} \leq \tau$ because it is Bayes-optimal to do so. Since a confidence estimator should get a greater reward for trusting when the generation is correct and abstaining when incorrect than the opposite, $R_{tp} > R_{fn}$ and $R_{tn} > R_{fp}$. [4] Setting four values that have large ranges can be challenging, unintuitive, and difficult to analyze. To remedy this, we introduce two intuitive and tightly bounded quantities that help reparametrize this setup without loss of generality to have *just one* degree of freedom.

Reparametrization: We are interested in comparing confidence estimators C_1 and C_2 on the outputs of the same language model M on the same dataset D. Crucially, M generates $\hat{y}_1 \ldots \hat{y}_n$ and C_1 and C_2 *both* estimate the confidences of these candidate generations. R_{tp}, R_{tn}, R_{fp}, and R_{fn} are based on the dataset, not the LLM or confidence estimator. When comparing C_1 and C_2 and the probabilities they emit, we only care about τ from equation 1.

Net Utility of Correctly Trusting (U_{ct}): When the model generates a correct answer ($\hat{y}$ is correct), we want our confidence estimator to emit $\hat{p} > \tau$. We assign a reward R_{tp} or R_{fn} based on whether $\hat{p} > \tau$ or not. Recall that $R_{tp} > R_{fn}$. Our reformulation looks at their difference, the net utility of trusting a correct generation.

Net Utility of Correctly Abstaining (U_{ca}): Similarly when M generates an incorrect answer, we want the confidence estimator C to output $\hat{p} \leq \tau$ and assign R_{tn} or R_{fp} accordingly. We look again to their difference, the net utility of abstaining from trusting an incorrect generation.

$$U_{ct} \stackrel{\text{def}}{=} R_{tp} - R_{fn}; U_{ca} \stackrel{\text{def}}{=} R_{tn} - R_{fp}$$

Normalizing and Recomputing the MBR Threshold: To simplify further, we calculate a normalized version of both utilities u_{ct}, u_{ca} that are bounded between 0 and 1:

$$\begin{bmatrix} u_{ct} \\ u_{ca} \end{bmatrix} = \begin{bmatrix} U_{ct} \\ U_{ca} \end{bmatrix} \cdot \frac{1}{U_{ct} + U_{ca}} \tag{2}$$

Using u_{ct} and u_{ca}, equation 1 can then be rewritten as: [5]

$$\hat{p} > \tau \stackrel{\text{def}}{=} \frac{U_{ca}}{U_{ct} + U_{ca}} = u_{ca} \tag{3}$$

This formulation allows us to use τ as precisely the risk level associated with a task (*i.e.*, high risk means that an estimator gets a larger penalty for a false positive than reward for a true positive.).

Calculating the Relative Utility: To calculate the relative utility between two confidence estimators C_1 and C_2, remember that the rewards (R_{tp}, R_{tn}, R_{fp}, and R_{fn}) and language model generations ($\hat{y}_1, \ldots \hat{y}_n$) are fixed. The only differences are the confusion matrices. The relative utility between two systems is the difference in total utility, $\mathbb{RU}_{C_1,C_2} = \mathbb{U}_{C_1} - \mathbb{U}_{C_2}$. The fixed generations ensure that the number of correct and incorrect samples are

[3] See the Appendix §A for a detailed derivation.

[4] Often $R_{tp}, R_{tn} \in [0, \infty)$ and $R_{fn}, R_{fp} \in (-\infty, 0]$ because getting a classification decision correct generally yields a positive reward and getting it incorrect yields at most 0 reward. In a standard binary classification task $R_{tp} = R_{tn} = 1$ and $R_{fp} = R_{fn} = 0$ and thus $\tau = 0.5$.

[5] Given the normalization scheme we chose, u_{ct} = 1 - u_{ca}.

preserved: $N_{tp,C_1} + N_{fn,C_1} = N_{tp,C_2} + N_{fn,C_2}$ and $N_{tn,C_1} + N_{fp,C_1} = N_{tn,C_2} + N_{fp,C_2}$. Thus, simplifying and normalizing with respect to u_{ca} and u_{ct} yields: [6]

$$\mathbb{RU}_{C_1,C_2} = \frac{1}{U_{ct} + U_{ca}} \cdot \begin{bmatrix} u_{ct} \\ u_{ca} \end{bmatrix} \cdot \begin{bmatrix} (N_{tp,C_1} - N_{tp,C_2}) \\ (N_{tn,C_1} - N_{tn,C_2}) \end{bmatrix} \tag{4}$$

Renormalizing with respect to the Oracle and Anti-Oracle: We want to put relative utilities on a scale with respect to the Oracle policy (O), a system that only gets true positives and true negatives, and the Anti-Oracle policy (AO), a system that only gets false positives and false negatives. Since the Oracle policy should get the highest possible utility and the Anti-Oracle policy get the lowest possible utility, we normalize the utility of an estimator accordingly: $\text{EURO}_C(u_{ca}) \in [0, 1]$, approaching 0 as C gets closer to the AO policy and approaching 1 as C gets closer to the O policy. For a confidence estimator C, EURO is:

$$\text{EURO}_C(u_{ca}) = \frac{N_{tp,C} + u_{ca} \cdot (N_{tn,C} - N_{tp,C})}{N_{tp,O} + u_{ca} \cdot (N_{tn,O} - N_{tp,O})} \tag{5}$$

Area under the EURO curve Since EURO is parametrized by u_{ca}, each value is a setting for the net utility of correctly abstaining from trusting an incorrect candidate. We can compute the EURO for each of these, which traces a curve. To compare confidence estimators across all possible settings, we summarize EURO by taking the area under this curve and term this AUC-EURO. [7]

Analyzing EURO across risk levels: u_{ca} naturally stratifies into low, medium, and high risk settings. In equation 3, we observe that $\tau \stackrel{\text{def}}{=} u_{ca}$, so we split u_{ca} into three risk bins, where low corresponds to $u_{ca} \in (0, \frac{1}{3})$, medium to $u_{ca} \in [\frac{1}{3}, \frac{2}{3})$, and high to $u_{ca} \in [\frac{2}{3}, 1)$. For the highest risk tasks, we should be most weary of trusting a generation and get the most credit for correctly abstaining from trusting an incorrect generation. For the lowest risk tasks, we should be the least weary of trusting a generation, but get the least credit for correctly abstaining from trusting an incorrect generation. The threshold τ and thus u_{ca} exactly determines how much confidence we need to trust a generation. We measure a total of 4 AUC-EURO values: one corresponding to the entire spectra of u_{ca} values and one for each of the 3 risk-split settings mentioned above. [8]

3. ACUTE

Activation-based confidence estimation leverages the activations of the language model while generating a candidate answer to learn a confidence estimator. In particular, the activations are used as input features to a classifier that predicts whether the resulting generation is correct or not. The probability that the classifier assigns to the "correct" class *is* the new calibrated confidence. Given that the latent spaces of language models are steerable via steering vectors (Subramani et al., 2022), there exist circuits and individual directions in latent space that are controllable (Li et al., 2023; Conmy et al., 2023), and that activations at different layers of models correspond to different linguistic phenomenon (Tenney et al., 2019; Ethayarajh, 2019; Li & Subramani, 2025), we hypothesize that the activations contain decipherable signals for confidence estimation.

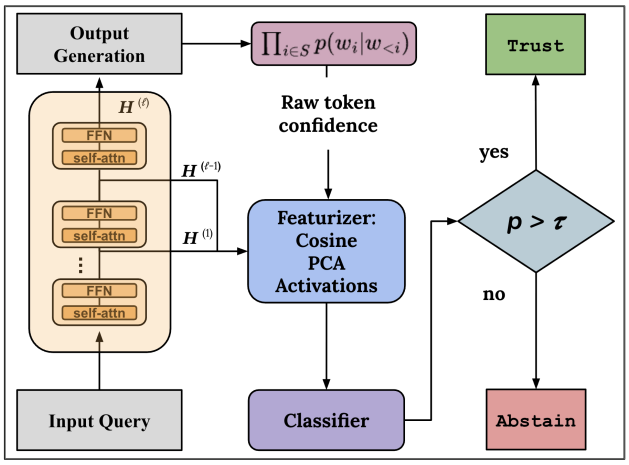

Figure 2. The ACUTE Protocol. We use input features from model internals to train confidence estimators to adjudicate whether to trust or abstain from trusting a model output. Details are in §3.

3.1. Input features

Mean-pooled Activations Since we hypothesize that the activation spaces contain decipherable information for confidence estimation, the raw activations themselves are the most natural place to start. Consider a language model M that processes an input query q and produces a candidate generation $\hat{y} = w_1, \ldots, w_T$. At each layer j, we extract the activations for the output sequence: $H^{(j)} = \begin{bmatrix} \mathbf{h}_1^{(j)} & \ldots & \mathbf{h}_T^{(j)} \end{bmatrix}$. We collapse this activation into a single vector by mean pooling across the sequence such that for every layer j, we have a single activation vector per query: $\bar{\mathbf{h}}^{(j)}$. These become the input features to our confidence estimators. We train separate confidence estimators at each layer, but present the performance of the best performing estimator at different layer depths (early, middle, and late) rather than at all layers for simplicity.

Layer-wise Cosine Similarities with the Final Layer Activations at a single layer are high-dimensional and can be limited in the information it could contain. Layers

[6] See Appendix §A.3 for all the EURO derivations.

[7] Since EURO $(u_{ca}) \in [0, 1]$, AUC-EURO $\in [0, 1]$.

[8] See §B for a discussion on how EURO relates to other calibration and decision-making utility tools.

	MMLU					APIGen					SCITLDR				
		AUC-EURO (↑)					AUC-EURO (↑)					AUC-EURO (↑)			
estimator	**smECE** (↓)	low	med	high	all	**smECE** (↓)	low	med	high	all	**smECE** (↓)	low	med	high	all
Raw Conf	0.17	0.90	0.71	0.54	0.72	0.22	0.28	0.53	0.80	0.53	0.15	0.36	0.71	**0.92**	0.66
HRE	0.11	0.87	0.72	0.71	0.77	**0.02**	0.82	0.70	0.82	0.78	**0.08**	0.67	0.71	**0.92**	0.77
NWKR	**0.07**	0.90	0.73	0.74	0.79	**0.02**	0.82	0.69	0.82	0.78	**0.08**	0.67	0.71	**0.92**	0.77
ACUTE early act	**0.07**	**0.91**	0.73	0.74	0.79	0.05	0.90	0.82	0.87	0.86	**0.08**	0.69	**0.72**	**0.92**	**0.78**
ACUTE mid act	**0.07**	**0.91**	0.76	0.78	0.82	0.06	0.90	**0.84**	**0.88**	0.87	**0.08**	**0.70**	**0.72**	**0.92**	**0.78**
ACUTE late act	**0.07**	**0.91**	**0.77**	**0.80**	**0.83**	0.07	0.90	**0.84**	**0.88**	0.87	**0.08**	0.69	**0.72**	**0.92**	0.77
ACUTE cosine	0.09	0.90	0.75	0.78	0.81	0.03	0.87	0.77	0.84	0.83	**0.08**	0.68	0.71	**0.92**	0.77
ACUTE pca10	0.08	**0.91**	0.76	0.78	0.82	0.04	0.90	0.82	0.87	0.87	0.09	**0.70**	**0.72**	**0.92**	**0.78**
ACUTE pca20	0.08	**0.91**	0.76	0.77	0.81	0.06	**0.91**	**0.84**	**0.88**	**0.88**	0.09	**0.70**	**0.72**	**0.92**	**0.78**

Table 1. Results on the MMLU test set averaged across all 57 subtasks (left), on the APIGen test subset (middle), and on the SCITLDR dev set (right). All results for all tasks are averaged across the 6 LLMs we test. Lower smECE is better, while higher AUC-EURO is better. **Bold** indicates the best result and underline indicates the second best result in each column.

tend to specialize; activations at earlier layers are better at probing word or subword-level phenomenon and those at layer layers are better at higher-level, sequence-level phenomenon (Rogers et al., 2020). Since concatenating the representations from all layers is infeasible as the number of input features would be too large (*e.g.*, 200,000 input features for a model with a hidden size of 4000 and 50 layers), we choose an aggressive reduction. For a model with ℓ layers, we collect the mean pooled activations at each layer $\bar{\mathbf{h}}^{(1)}, \ldots, \bar{\mathbf{h}}^{(\ell)}$. and compute the cosine similarity between the pooled activation at each layer and that of the final layer:

$$\mathbf{x}_{\text{COSINE}} = \left[\text{SIM}(\bar{\mathbf{h}}^{(1)}, \bar{\mathbf{h}}^{(\ell)}) \ \ldots \ \text{SIM}(\bar{\mathbf{h}}^{(\ell-1)}, \bar{\mathbf{h}}^{(\ell)})\right]$$

This aggregation method yields $\ell - 1$ total features, *i.e.*, $\mathbf{x}_{\text{COSINE}} \in \mathbb{R}^{\ell-1}$. We chose COSINE as our similarity metric following previous work on aggregating embeddings to measure similarity, *e.g.*, BERTScore and CodeBERTScore (Zhang et al., 2020; Zhou et al., 2023). In practice, one could replace COSINE with any other measure. We use these $\ell - 1$ features to learn a confidence estimator. [9]

PCA Transformed Activations As mentioned earlier, the number of input features is still large, even when considering a single layer. Aggregating across layers is infeasible without dimensionality reduction, but collapsing each layer down to just one feature like our cosine approach removes a lot of information. We use Principal Components Analysis and compute the top m components at each layer, leading to $m\ell$ features that we use to learn the estimator. We use 10 and 20 components in our experiments.

[9]This estimator is a much more efficient version of the MICE estimators used to recalibrate tool-calling agents (Subramani et al., 2025), see §C for more details.

3.2. ACUTE models

We use these input features to train a simple classifier to predict whether a model generation is correct or not. Random forests are strong correctness predictors for tool-calling and question-answering tasks (Subramani et al., 2025; Liu et al., 2025). [10] Traditionally, random forests require Platt scaling when used as confidence estimators as decision-trees favor extreme probabilities (Platt et al., 1999; Zadrozny & Elkan, 2001), but Shaker & Hüllermeier (2025) show that well-trained random forests do not need recalibration. We found no improvement with scaling, so we omit scaling in our experiments. Hyperparameter details are in §4.

4. Experiments

We apply the ACUTE protocol on 6 models across 3 tasks and measure the degree to which calibration (via smECE) and utility (via EURO) improve over strong, efficient baselines. [11]

Models In our experiments, we evaluate 6 different models that come from 4 model families: gemma-3-4b-it, gemma-3-12b-it (Team et al., 2025), Qwen3-4B-Instruct-2507, Qwen3-14B (Yang et al., 2025), phi-4 (Abdin et al., 2024), and SmolLM3-3B (Bakouch et al., 2025). We apply the ACUTE protocol to each LLM separately and evaluate performance relative to strong baselines independently.

Datasets & Tasks To measure how ACUTE generalizes, we choose three distinct tasks: language understanding (MMLU; 5-shot in-context learning), tool-calling (APIGen; zeroshot), and scientific document summarization (SCITLDR; zeroshot). MMLU is a multiple choice English-

[10]In early experiments, we tested other simple classifiers like logistic regression and support vector machines, and found that random forests were slightly more performant.

[11]See §C for a more detailed discussion.

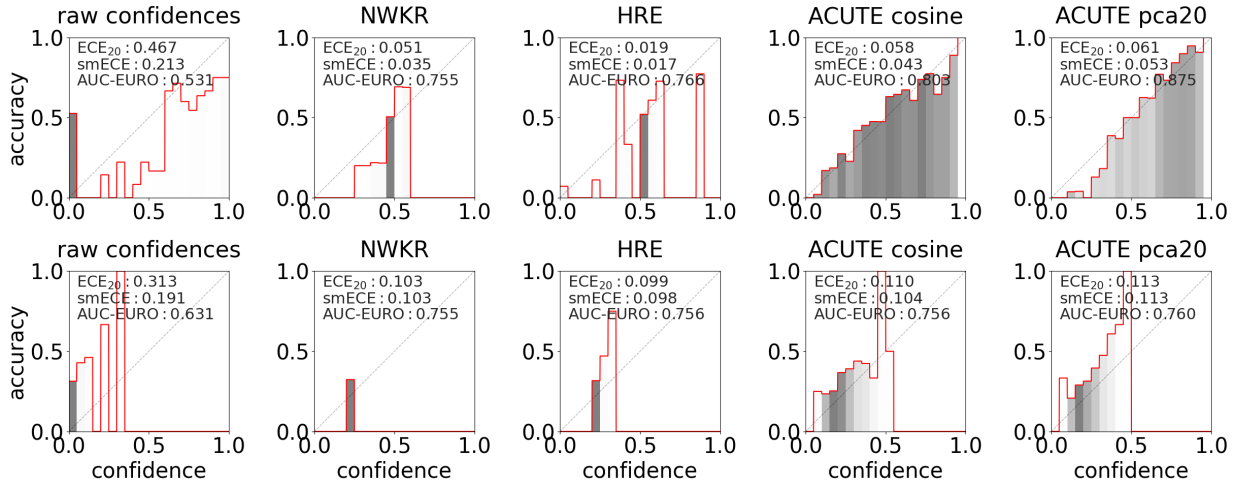

Figure 3. Here we show reliability diagrams for the gemma-3-12b-it model for APIGen (top row) and SCITLDR (bottom row) across 3 baseline estimators and two ACUTE estimators. Darker shading corresponds to higher density of examples in that confidence bin.

language benchmark with 57 distinct subtasks ranging from elementary school mathematics and sciences to college and post-graduate level medicine and law (Hendrycks et al., 2021). Crucially, these subtasks are all multiple choice, so LLMs generate a single token corresponding to the answer choice. APIGen is a tool-calling dataset where the model is given a description and set of APIs to potentially call (Liu et al., 2024). We instruct the LLM to generate a tool call or set of tool calls with API names and parameters in a parsable format. We experiment with the abstract only subtask of SCITLDR, where the model has to generate a short summary of approximately 25 words looking at just the abstract and title of the paper (Cachola et al., 2020). Both APIGen and SCITLDR require generating multi-token output and are evaluated using approximate exact match (APIGen) and rouge-L score (SCITLDR). [12] See §D for system prompts.

4.1. Baseline Confidence Estimators

Raw Confidences Raw confidences are the default baseline that our language models emit. For an output sequence $y = w_1, \ldots, w_T$ of length T, the raw confidence is the joint probability that the language model M assigns to y: $\hat{p} = \prod_{i=1}^{T} p_M(w_i|w_{<i})$. Computing this does not require any supervised training data or training an auxiliary confidence estimator as these are emitted directly from the language model. Raw confidence is used as an auxiliary feature for all of our ACUTE estimators because in early experiments, we saw a small performance gain when added.

Histogram Regression Estimator (HRE) Our second baseline, HRE, is a standard recalibration method. We construct a histogram of confidences based on a training set with equal sized bins. To recalibrate a raw confidence estimate, we look up the bin associated with that estimate and return the average accuracy of all the samples in that bin. [13]

Nadaraya-Watson Kernel Regression (NWKR) For our strongest baseline, we use the Nadaraya-Watson kernel regression estimator (Nadaraya, 1964; Watson, 1964), which is the method used to calculate smECE. NWKR uses the raw confidence emitted by the language model and learns a reflected Gaussian kernel to map how over or under-confident a confidence estimate is. Formally, to map from a confidence estimate emitted by the language model p_M to a recalibrated confidence $\hat{p}$, we look up the NWKR estimate for p_M and return that estimate: $\hat{p} = NWKR(p_M)$, like with HRE.

4.2. ACUTE Estimators

We experiment with the different versions of the ACUTE estimators discussed in §3: early activations, middle activations, late activations, cosine, pca10, and pca20. As mentioned earlier, ACUTE early, medium, and late are just the highest performing classifier using mean-pooled activations as input features over the first, second, and final third of the layers. ACUTE cosine uses the layer-wise cosine similarities with the final layer as input features to learn a confidence estimator. ACUTE pca10 and pca20 first PCA transform the activations at each layer to have 10 or 20 components and aggregate those across layers to produce input features. For all ACUTE models, we train a random forest classifier on these different features to predict whether the input query is correct or not. Our random forest uses 500 estimators, a max depth of 10, and does not restrict the number of features that can be used in each tree.

[12] Since we need binary labels for correctness, we use 0.3 as a threshold to stratify correct vs. incorrect generations for SCITLDR.

[13] See Subramani et al. (2025) for more details.

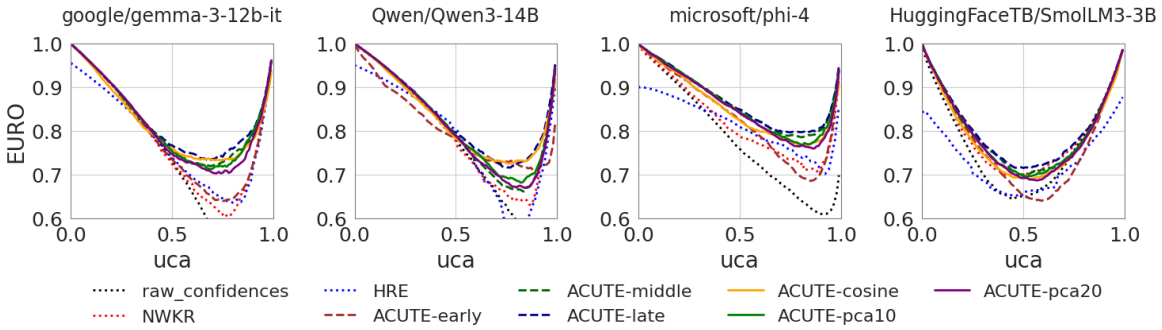

Figure 4. Results plotting EURO versus risk level (u_{ca}) for 4 of 6 LLMs (the other two are very similar to SmolLM3-3B). Results are averaged across the 57 subtasks of MMLU. We include 3 baselines (raw confidences, HRE, and NWKR) along with our 6 configurations of ACUTE. All policies perform similarly at the lowest risk levels, but at high risk settings, the ACUTE estimators perform best.

5. Results & Analysis

5.1. Main Results

We present the results of evaluating the confidence estimators on MMLU, APIGen, and SCITLDR and evaluate both smECE and AUC-EURO in Table 1. We report 4 AUC-EURO values: low risk ($u_{ca} \in [0, \frac{1}{3})$), medium risk (med; $u_{ca} \in [\frac{1}{3}], \frac{2}{3})$), high risk ($u_{ca} \in [\frac{2}{3}, 1)$), and all ($u_{ca} \in [0, 1)$).

Raw confidences have high calibration error and low AUC-EURO values across the board; it is the worst performing estimator across all tasks. Inspecting the probabilities themselves (Figure 3), we observe rampant over-confidence in MMLU and under-confidence in APIGen and SCITLDR. Both HRE and NWKR improve upon the raw confidences by moving the probabilities into a much narrower range, whereby dramatically lowering smECE and increasing AUC-EURO. Due to this, both baselines lead to low utility confidence estimates that make decision-making challenging, especially in high risk settings (see Figure 4 when $u_{ca} \in [0.4, 0.9]$). In contrast, all of our ACUTE estimators have probability mass across the full spectrum, still maintain the low calibration error of HRE and NWKR, but have higher AUC-EURO scores at all risk levels on both tasks.

In Figure 3, we show reliability diagrams for the gemma3-12b-it model on APIGen and SCITLDR. Across tasks, HRE and NWKR bias for being a base rate estimator, whereas ACUTE systems have probability mass across the entire probability spectrum. ACUTE offers recalibration with decision-making utility, whereas HRE and NWKR bias for calibration alone. The best confidence estimators are ACUTE late and ACUTE pca20, having the highest AUC-EURO values across risk settings. Given SCITLDR is a challenging task, NWKR, HRE, and all ACUTE methods perform comparably metric-wise, but baselines maintain degenerate behavior, nearly always predicting the base rate. ACUTE always outperforms baselines on AUC-EURO, while maintaining smECE.

5.2. Does ACUTE have strong sample complexity?

Here, we look at how sample efficient our methods are in comparison to the highly sample-efficient NWKR baseline. We look at effective training sizes n_i = 25, 50, 100, 200, 500, 1000 on the APIGen dataset. Recall the full training set has 1000 examples. For each effective training set size n_i, we randomly sample n_i examples without replacement from the overall training set 20 times and train different estimators for each of these 20 samples.

In Figure 5 we show the results of this for the gemma3-12b-it model with error bands corresponding to one standard deviation above and below the mean. We find that all ACUTE methods outperform both baselines at all effective training set sizes, with the activation-based and pca-based estimators (ACUTE early, middle, late, ACUTE pca10, and ACUTE pca20) performing better with just 25 training examples than either baseline (NWKR, HRE) on all 1000 training examples.

5.3. Which layers are best for ACUTE?

One of our configurations for ACUTE is using mean-pooled activations at a specific layer as input features to our random forest classifier. We dive deeper into this by looking at the performance of every layer across each of the 6 LLMs we tested on the MMLU benchmark, averaging results across all 57 tasks. In Figure 6, we find that layerwise performance is very low at the first layer, indicating that the subword embeddings alone lack a strong confidence estimation signal. Additionally, we find that there exist three phases that roughly correspond to the splits of early, middle, and late layers. Early corresponds to $\sim$40% of the layers, middle between 40 and 65%, and late the remaining. The early and late phases are flat, while the middle phase has a linearly increasing trend. These trends are consistent across all 6 LLMs, suggesting that models tend to encode information useful for confidence estimation similarly across layers.

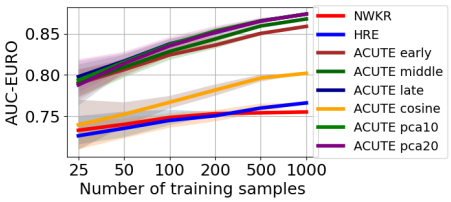

Figure 5. Sample complexity as the size of the training set varies of confidence estimators measured on the APIGen dataset on the gemma3-12b-it model. The error bands correspond to $\pm\sigma$.

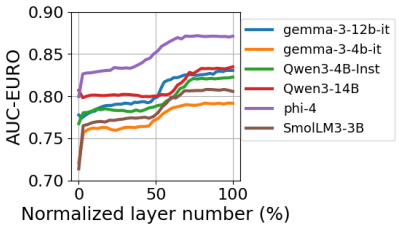

Figure 6. AUC-EURO performance of our mean-pooled activation ACUTE estimators on all 6 LLMs on MMLU across layers.

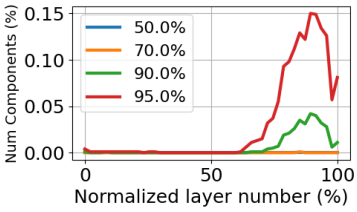

Figure 7. Intrinsic dimension of the activations for gemma-3-12b-it on the APIGen test set.

5.4. Intrinsic Dimension Analysis

To better understand how complex the representations extracted from our language models are, we perform an intrinsic dimension analysis using PCA. [14] On the APIGen dataset, we look at how many components (as a percentage of the total) are needed to explain 50%, 70%, 90%, and 95% of the variance in our test set. Note that our test set has 1000 instances and our representation size for each of these models is larger than 1000. Thus, trivially, 1000 components are sufficient to fully explain the variance in our data sample, so our maximum number of components is 1000. Figure 7 shows the results of inspecting the activations of gemma3-12b-it model on APIGen and analyzing its intrinsic dimension. We find that just a couple of components explain over 95% of the variance for the first 60% of the layers, but this increases at later layers perhaps providing better confidence estimation signal for our random forests. These trends are consistent with all LLMs we tested.

6. Related Work

Calibration & Confidence Estimation: Calibration has been studied in a variety of contexts often applied to binary classifiers. Often these works consider only a single predicted confidence and include Platt scaling (Platt et al., 1999), isotonic regression (Barlow, 1972), beta calibration (Kull et al., 2017), and histogram-based adaptive binning methods (Nobel, 1996). Additionally different language technology systems have had their calibration assessed including large language models (Jiang et al., 2021; Kadavath et al., 2022), and machine translation systems (Niculescu-Mizil & Caruana, 2005), while others leverage confidence estimates to ask for confirmation (Stengel-Eskin & Van Durme, 2023b), and provide an estimate of trust to users (Stengel-Eskin et al., 2024). Finally, as language model capabilities have improved, linguistic calibration, *i.e.*, asking the language model of its confidence, has become popular and has mixed utility (Mielke et al., 2022; Band et al., 2024). Prior work focus on single-token, classification-like settings, ACUTE, on the other hand, works with both single- and multi-token generation settings and offers a generalized activation-based calibration solution.

Model Internals: Many works have studied the internals of language models to identify how linguistic information is encoded throughout the layers of a model (Hewitt & Manning, 2019; Jawahar et al., 2019; Tenney et al., 2019; Niu et al., 2022; He et al., 2024; Li & Subramani, 2025; Acevedo et al., 2026). Language models can also be steered for exact generation (Subramani et al., 2019) and attribute-based steering (Subramani et al., 2022; Li et al., 2023). Others have found that there exist vectors that loosely correspond to functions or tasks (Todd et al., 2024; Hendel et al., 2023) Work combining mechanistic interpretability with calibration is rare: Beigi et al. (2024) use contrastive learning on model internals to improve trustworthiness, Subramani et al. (2025) show that decoding from intermediate layers can improve calibration for a tool-using dataset, and Liu et al. (2025) learn probes on activations to predict correctness in question-answering tasks. ACUTE operationalizes the activation spaces to build compute- and sample-efficient confidence estimators to improve trustworthiness.

7. Conclusion

We identify major problems with calibration metrics when used for decision-making and propose a new general, interpretable metric called **expected utility renormalized by the oracle** (EURO), which balances calibration and decision-making utility. EURO can reliably score confidence estimators, resolving some drawbacks of ECE. Also, we introduce a light-weight, compute- and sample-efficient activation-based confidence, utility, and trust estimation protocol (ACUTE) to recalibrate LLM outputs. Experiments on MCQA (MMLU), tool-calling (APIGen), and scientific document summarization (SCITLDR) show that incorporating the ACUTE protocol on 6 LLMs improves confidence estimation (EURO), while maintaining low calibration error.

[14] This is common practice to understand the effective dimension or intrinsic dimension of representations (Li et al., 2018; Subramani et al., 2019; Li & Subramani, 2025).

Impact Statement

In this paper, we present EURO and ACUTE. EURO is a metric to measure how well calibrated and how useful a confidence estimator is for decision-making. This gives users an alternative metric to measure how well their systems work across a variety of risk-levels. ACUTE provides a light-weight, compute- and sample-efficient method to directly recalibrate LLM output probabilities. Since better calibration improves decision-making for users, both of our contributions can help increase trust in language model outputs, either by helping to provide more reliable confidence estimates (through ACUTE) or by evaluating an already deployed confidence estimator (through EURO).

We believe that as LLMs become increasingly embedded in everyday technology, reliability and trust is paramount. We believe our work takes a small step to reiterate the need for better metrics and methods to appropriately adjudicate trust. All that being said, our decision theoretic framework is not a replacement for careful data analysis and system design.

References

Abdin, M., Aneja, J., Behl, H., Bubeck, S., Eldan, R., Gunasekar, S., Harrison, M., Hewett, R. J., Javaheripi, M., Kauffmann, P., et al. Phi-4 technical report. *arXiv*, 2024.

Acevedo, S., Laio, A., and Baroni, M. Differential syntactic and semantic encoding in llms. *arXiv*, 2026.

Bakouch, E., Ben Allal, L., Lozhkov, A., Tazi, N., Tunstall, L., Patiño, C. M., Beeching, E., Roucher, A., Reedi, A. J., Gallouédec, Q., Rasul, K., Habib, N., Fourrier, C., Kydlicek, H., Penedo, G., Larcher, H., Morlon, M., Srivastav, V., Lochner, J., Nguyen, X.-S., Raffel, C., von Werra, L., and Wolf, T. SmolLM3: smol, multilingual, long-context reasoner. https://huggingface.co/blog/smollm3, 2025.

Band, N., Li, X., Ma, T., and Hashimoto, T. Linguistic calibration of long-form generations. In *ICML*, 2024.

Barlow, R. E. Statistical inference under order restrictions: The theory and application of isotonic regression. *(No Title)*, 1972.

Beigi, M., Shen, Y., Yang, R., Lin, Z., Wang, Q., Mohan, A., He, J., Jin, M., Lu, C.-T., and Huang, L. InternalInspector i^2: Robust confidence estimation in LLMs through internal states. In *EMNLP Findings*, 2024. URL https://aclanthology.org/2024.findings-emnlp.751/.

Błasiok, J. and Nakkiran, P. Smooth ECE: Principled reliability diagrams via kernel smoothing. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=XwiA1nDahv.

Cachola, I., Lo, K., Cohan, A., and Weld, D. TLDR: Extreme summarization of scientific documents. In *EMNLP Findings*, 2020. URL https://aclanthology.org/2020.findings-emnlp.428/.

Conmy, A., Mavor-Parker, A., Lynch, A., Heimersheim, S., and Garriga-Alonso, A. Towards automated circuit discovery for mechanistic interpretability. *NeurIPS*, 2023.

Dawid, A. P. The well-calibrated bayesian. *Journal of the American statistical Association*, (379), 1982.

Desai, S. and Durrett, G. Calibration of pre-trained transformers. In *EMNLP*, 2020. URL https://aclanthology.org/2020.emnlp-main.21/.

Ethayarajh, K. How contextual are contextualized word representations? Comparing the geometry of BERT, ELMo, and GPT-2 embeddings. In *EMNLP*, 2019. URL https://aclanthology.org/D19-1006/.

He, L., Chen, P., Nie, E., Li, Y., and Brennan, J. R. Decoding probing: Revealing internal linguistic structures in neural language models using minimal pairs. In *COLING*. ELRA and ICCL, 2024. URL https://aclanthology.org/2024.lrec-main.402/.

Hendel, R., Geva, M., and Globerson, A. In-context learning creates task vectors. In *EMNLP Findings*, 2023. URL https://aclanthology.org/2023.findings-emnlp.624/.

Hendrycks, D., Burns, C., Basart, S., Zou, A., Mazeika, M., Song, D., and Steinhardt, J. Measuring massive multitask language understanding. In *ICLR*, 2021. URL https://openreview.net/forum?id=d7KBjmI3GmQ.

Hewitt, J. and Manning, C. D. A structural probe for finding syntax in word representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 2019. URL https://aclanthology.org/N19-1419/.

Jawahar, G., Sagot, B., and Seddah, D. What does BERT learn about the structure of language? In *ACL*, 2019. URL https://aclanthology.org/P19-1356/.

Jiang, Z., Araki, J., Ding, H., and Neubig, G. How can we know when language models know? on the calibration of language models for question answering. *TACL*, 2021. URL https://aclanthology.org/2021.tacl-1.57/.

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

Kull, M., Filho, T. S., and Flach, P. Beta calibration: a well-founded and easily implemented improvement on logistic calibration for binary classifiers. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, Proceedings of Machine Learning Research. PMLR, 2017. URL https://proceedings.mlr.press/v54/kull17a.html.

Li, C., Farkhoor, H., Liu, R., and Yosinski, J. Measuring the intrinsic dimension of objective landscapes. In *ICLR*, 2018.

Li, K., Patel, O., Viégas, F., Pfister, H., and Wattenberg, M. Inference-time intervention: Eliciting truthful answers from a language model. *NeurIPS*, 2023.

Li, M. and Subramani, N. Model internal sleuthing: Finding lexical identity and inflectional morphology in modern language models. *arXiv*, 2025.

Liu, J., Jain, J., Diab, M., and Subramani, N. Llm microscope: What model internals reveal about answer correctness and context utilization. *arXiv*, 2025.

Liu, Z., Hoang, T., Zhang, J., Zhu, M., Lan, T., Tan, J., Yao, W., Liu, Z., Feng, Y., RN, R., et al. Apigen: Automated pipeline for generating verifiable and diverse function-calling datasets. *NeurIPS*, 2024.

Mielke, S. J., Szlam, A., Dinan, E., and Boureau, Y.-L. Reducing conversational agents' overconfidence through linguistic calibration. *TACL*, 2022. URL https://aclanthology.org/2022.tacl-1.50/.

Murphy, A. H. and Epstein, E. S. Verification of probabilistic predictions: A brief review. *Journal of Applied Meteorology and Climatology*, (5), 1967.

Nadaraya, E. A. On estimating regression. *Theory of Probability & Its Applications*, (1), 1964.

Naeini, M. P. and Cooper, G. F. Binary classifier calibration using an ensemble of near isotonic regression models. In *2016 IEEE 16th International Conference on Data Mining (ICDM)*. IEEE, 2016.

Naeini, M. P., Cooper, G., and Hauskrecht, M. Obtaining well calibrated probabilities using bayesian binning. In *AAAI*, number 1, 2015.

Niculescu-Mizil, A. and Caruana, R. Predicting good probabilities with supervised learning. In *ICML*, 2005.

Niu, J., Lu, W., and Penn, G. Does BERT rediscover a classical NLP pipeline? In *COLING*. International Committee on Computational Linguistics, 2022. URL https://aclanthology.org/2022.coling-1.278/.

Nobel, A. Histogram regression estimation using data-dependent partitions. *The Annals of Statistics*, (3), 1996.

Platt, J. et al. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in large margin classifiers*, (3), 1999.

Rogers, A., Kovaleva, O., and Rumshisky, A. A primer in BERTology: What we know about how BERT works. *TACL*, 2020. URL https://aclanthology.org/2020.tacl-1.54/.

Shaker, M. H. and Hüllermeier, E. Random forest calibration. *Knowledge-Based Systems*, 2025. ISSN 0950-7051. URL https://www.sciencedirect.com/science/article/pii/S0950705125011840.

Stengel-Eskin, E. and Van Durme, B. Calibrated interpretation: Confidence estimation in semantic parsing. *TACL*, 2023a. URL https://aclanthology.org/2023.tacl-1.69/.

Stengel-Eskin, E. and Van Durme, B. Did you mean...? confidence-based trade-offs in semantic parsing. In *EMNLP*, 2023b. URL https://aclanthology.org/2023.emnlp-main.159/.

Stengel-Eskin, E., Hase, P., and Bansal, M. Lacie: Listener-aware finetuning for calibration in large language models. *NeurIPS*, 2024.

Subramani, N., Bowman, S., and Cho, K. Can unconditional language models recover arbitrary sentences? *NeurIPS*, 2019.

Subramani, N., Suresh, N., and Peters, M. Extracting latent steering vectors from pretrained language models. In *ACL Findings*, 2022. URL https://aclanthology.org/2022.findings-acl.48/.

Subramani, N., Eisner, J., Svegliato, J., Van Durme, B., Su, Y., and Thomson, S. MICE for CATs: Model-internal confidence estimation for calibrating agents with tools. In *NAACL*, 2025. URL https://aclanthology.org/2025.naacl-long.615/.

Team, G., Kamath, A., Ferret, J., Pathak, S., Vieillard, N., Merhej, R., Perrin, S., Matejovicova, T., Ramé, A., Rivière, M., et al. Gemma 3 technical report. *arXiv*, 2025.

Tenney, I., Das, D., and Pavlick, E. BERT rediscovers the classical NLP pipeline. In *ACL*, 2019. URL https://aclanthology.org/P19-1452/.

Todd, E., Li, M. L., Sharma, A. S., Mueller, A., Wallace, B. C., and Bau, D. Function vectors in large language models. In *ICLR*, 2024.

Watson, G. S. Smooth regression analysis. *Sankhyā: The Indian Journal of Statistics, Series A*, 1964.

Yang, A., Li, A., Yang, B., Zhang, B., Hui, B., Zheng, B., Yu, B., Gao, C., Huang, C., Lv, C., et al. Qwen3 technical report. *arXiv*, 2025.

Zadrozny, B. and Elkan, C. Obtaining calibrated probability estimates from decision trees and naive bayesian classifiers. In *ICML*, 2001.

Zhang, T., Kishore, V., Wu, F., Weinberger, K. Q., and Artzi, Y. Bertscore: Evaluating text generation with bert. In *ICLR*, 2020.

Zhong, R., Snell, C., Klein, D., and Eisner, J. Non-programmers can label programs indirectly via active examples: A case study with text-to-SQL. In *EMNLP*, 2023. URL https://aclanthology.org/2023.emnlp-main.312/.

Zhou, S., Alon, U., Agarwal, S., and Neubig, G. CodeBERTScore: Evaluating code generation with pretrained models of code. In *EMNLP*, 2023. URL https://aclanthology.org/2023.emnlp-main.859/.

A. EURO derivations

Here we present some of the derivations from §2.

A.1. Bayes Optimality of the MBR Threshold

Suppose a confidence estimator emits a probability $\hat{p}$ for an example. From our vantage point, we have no other information than just this probability $\hat{p}$. We expect that the probability that the example is correct is $\hat{p}$ and thus the probability that the example is incorrect is $1 - \hat{p}$. For this task, lets assume we get the following rewards for true positives, true negatives, false positives, and false negatives: R_{tp}, R_{tn}, R_{fp}, and R_{fn}. The reward associated with predicting the positive class (trusting the estimator) is:

$$\hat{p} * R_{tp} + (1 - \hat{p}) * R_{fp} \tag{6}$$

The reward associated with predicting the negative class (abstaining from trusting the estimator) is:

$$(1 - \hat{p}) * R_{tn} + (\hat{p}) * R_{fn} \tag{7}$$

Given this, we should trust the estimator (predict the positive class) if and only if the reward for this is greater than the reward for abstaining (predicting the negative class):

$$\begin{aligned}
\hat{p}(R_{tp}) + (1 - \hat{p})R_{fp} &> (1 - \hat{p})R_{tn} + \hat{p}(R_{fn}) \\
\hat{p}(R_{tp} - R_{fn}) &> (1 - \hat{p})(R_{tn} - R_{fp}) \\
\hat{p}(R_{tp} - R_{fn}) &> \hat{p}(R_{fp} - R_{tn}) + R_{tn} - R_{fp} \\
\hat{p}(R_{tp} - R_{fn} - R_{fp} + R_{tn}) &> R_{tn} - R_{fp} \\
\hat{p} &> \frac{R_{tn} - R_{fp}}{(R_{tp} - R_{fn} - R_{fp} + R_{tn})}
\end{aligned}$$

We rearrange this to get the Minimum Bayes Risk threshold τ from equation 1:

$$\hat{p} > \tau \stackrel{\text{def}}{=} \frac{R_{tn} - R_{fp}}{(R_{tp} - R_{fn}) + (R_{tn} - R_{fp})} \tag{8}$$

.

A.2. Recomputing the MBR Threshold

Recall that $U_{ct} = R_{tp} - R_{fn}$ and $U_{ca} = R_{tn} - R_{fp}$. Substituting the definitions of U_{ct}, U_{ca}, u_{ct}, and u_{ca} into equation 1 yields:

$$\hat{p} > \tau \stackrel{\text{def}}{=} \frac{R_{tn} - R_{fp}}{(R_{tp} - R_{fn}) + (R_{tn} - R_{fp})} = \frac{U_{ca}}{U_{ct} + U_{ca}} = u_{ca}$$

A.3. Calculating the Relative Utility

Here we show the proof for equation 9. Recall that we are aiming to calculate the relative utility between two confidence estimators C_1 and C_2. The rewards (R_{tp}, R_{tn}, R_{fp}, and R_{fn}) and language model generations ($\hat{y}_1, \ldots \hat{y}_n$) are fixed. The confusion matrices are the only differences between the systems, so the only thing that varies are the counts of true positives, true negatives, false positives, and false negatives between the two systems based on equation 3. We use the total reward calculation $\mathbb{U} = R_{tp} * N_{tp} + R_{tn} * N_{tn} + R_{fp} * N_{fp} + R_{fn} * N_{fn}$ to compute the relative utility between two systems. This is precisely the difference in total utility: $\mathbb{RU}_{C_1,C_2} = \mathbb{U}_{C_1} - \mathbb{U}_{C_2}$. Remember that our generations are fixed. This ensures that the number of correct and incorrect samples are preserved: $N_{tp,C_1} + N_{fn,C_1} = N_{tp,C_2} + N_{fn,C_2}$

and $N_{tn,C_1} + N_{fp,C_1} = N_{tn,C_2} + N_{fp,C_2}$. Rearranging this shows that $N_{tp,C_1} - N_{tp,C_2} = -N_{fn,C_1} + N_{fn,C_2}$ and $N_{tn,C_1} - N_{tn,C_2} = -N_{fp,C_1} + N_{fp,C_2}$.

$$\mathbb{RU}_{C_1,C_2} = \mathbb{U}_{C_1} - \mathbb{U}_{C_2} = \begin{bmatrix} N_{tp,C_1} - N_{tp,C_2} \\ N_{tn,C_1} - N_{tn,C_2} \\ N_{fp,C_1} - N_{fp,C_2} \\ N_{fn,C_1} - N_{fn,C_2} \end{bmatrix} \cdot \begin{bmatrix} R_{tp} \\ R_{tn} \\ R_{fp} \\ R_{fn} \end{bmatrix}$$

$$\mathbb{RU}_{C_1,C_2} = \begin{bmatrix} N_{tp,C_1} - N_{tp,C_2} \\ N_{tn,C_1} - N_{tn,C_2} \\ -N_{tn,C_1} + N_{tn,C_2} \\ -N_{tp,C_1} + N_{tp,C_2} \end{bmatrix} \cdot \begin{bmatrix} R_{tp} \\ R_{tn} \\ R_{fp} \\ R_{fn} \end{bmatrix}$$

$$\mathbb{RU}_{C_1,C_2} = \begin{bmatrix} (R_{tp} - R_{fn}) \\ (R_{tn} - R_{fp}) \end{bmatrix} \cdot \begin{bmatrix} (N_{tp,C_1} - N_{tp,C_2}) \\ (N_{tn,C_1} - N_{tn,C_2}) \end{bmatrix}$$

$$\mathbb{RU}_{C_1,C_2} = \begin{bmatrix} U_{ct} \\ U_{ca} \end{bmatrix} \cdot \begin{bmatrix} (N_{tp,C_1} - N_{tp,C_2}) \\ (N_{tn,C_1} - N_{tn,C_2}) \end{bmatrix}$$

We can write this in terms of u_{ct} and u_{ca}, using equation 2:

$$\mathbb{RU}_{C_1,C_2} = \frac{1}{U_{ct} + U_{ca}} \cdot \begin{bmatrix} u_{ct} \\ u_{ca} \end{bmatrix} \cdot \begin{bmatrix} (N_{tp,C_1} - N_{tp,C_2}) \\ (N_{tn,C_1} - N_{tn,C_2}) \end{bmatrix} \tag{9}$$

A.4. Renormalizing with respect to the Oracle and Anti-Oracle Derivation

Recall that we want our relative utilities to be scaled with respect to the Oracle (O) and Anti-Oracle (AO) policies, with $\text{EURO}_O = 1$ and $\text{EURO}_{AO} = 0$. For a confidence estimator C, EURO is calculated as follows:

$$\text{EURO}_C(u_{ca}) = \frac{\mathbb{RU}_{C,AO}}{\mathbb{RU}_{O,AO}} \tag{10}$$

Notice that the $\frac{1}{U_{ct}+U_{ca}}$ is a constant in both relative utilities, so we can cancel those out.

$$\text{EURO}_C(u_{ca}) = \begin{bmatrix} u_{ct} \\ u_{ca} \end{bmatrix} \cdot \frac{\begin{bmatrix} (N_{tp,C} - N_{tp,AO}) \\ (N_{tn,C} - N_{tn,AO}) \end{bmatrix}}{\begin{bmatrix} (N_{tp,O} - N_{tp,AO}) \\ (N_{tn,O} - N_{tn,AO}) \end{bmatrix}}$$

The Anti-Oracle policy has no true positives or true negatives ($N_{tp,AO} = N_{tn,AO} = 0$). Also recall that u_{ct} = 1 - u_{ca}.

$$\text{EURO}_C(u_{ca}) = \frac{u_{ct} \cdot N_{tp,C} + u_{ca} \cdot N_{tn,C}}{u_{ct} \cdot N_{tp,O} + u_{ca} \cdot N_{tn,O}} = \frac{N_{tp,C} + u_{ca} \cdot (N_{tn,C} - N_{tp,C})}{N_{tp,O} + u_{ca} \cdot (N_{tn,O} - N_{tp,O})}$$

B. How EURO relates to other measures

EURO is unique in that it combines calibration with decision-making utility together in one simple metric. As mentioned earlier, calibration error metrics have major drawbacks that make decision-making challenging. One of these drawbacks is the lack of incorporating the risk-level or the costs (and rewards) of true positives, false positives, true negatives, and false negatives into account. An estimator that emits the same sets of probabilities for examples should be adjudicated differently for a high risk task versus a low risk one. For a task with 0 risk, the right thing to do is to always trust your output regardless of whether it is correct. As a result, any estimator that generates a nonzero probability would be scored equally. Both calibration error metrics (ECE, smECE) lack this. Brier score lacks this too, but has the guarantee that an objective function that minimizes it, will be Bayes-optimal.

Precision, recall, sensitivity, and specificity deal with the confusion matrix directly. The area under the precision recall curve (AUPRC) and the area under the receiver operating characteristic curve (AUROC) both involve summarizing the

confusion matrix trading off these quantities. In general, AUROC treats both classes as equal, so it is invariant to risk level. AUPRC is not symmetric and is often used when there are imbalanced classes. In a high (or low) risk scenario the rewards for the classes are imbalanced and thus AUPRC could be used. However, both AUROC and AUPRC are reward-blind, so a MBR type of formulation needs to be used. EURO balances calibration and decision-making utility, while being able to appropriately score a confidence estimator across risk levels, taking into account calibration.

C. Confidence Estimators can be Computationally Infeasible

Confidence estimators need to be light-weight and computationally-efficient to be useful in a variety of settings and actually be used in practice for language model generation. Most methods for confidence estimation and model recalibration are expensive, nearly all applications ignore confidence estimation. The ones that attempt recalibration rely on raw confidences or adhoc linguistic calibration on a small set of outputs.

Previous work on calibration recalibrates in an inefficient manner. Subramani et al. (2025) propose MICE, which, to our knowledge, is the only model internals based confidence estimation work that leverages hidden states to recalibrate language model outputs in a single shot. MICE requires logit lens decoding from each intermediate layer, which by itself is expensive, requiring additional matrix multiplies with the unembedding matrix for each layer. Additionally, they compute the BERTSCORE between the final output generation and each intermediate generation and requires using a large auxiliary language model (`DeBERTa-xlarge-mnli`) to extract these BERTSCORE features before training their classifier. This is computationally infeasible. Even if the confidence estimator has been trained, each new inference query would require an additional forward passes of the BERTSCORE model equal to the number of layers of the model along with the logit lens decoding overhead. Assuming the underlying language model is the same size as `DeBERTa-xlarge-mnli`, each query takes 40-50x longer than the simple generation.

Our ACUTE cosine method approximates MICE. We remove the reliance on the auxiliary model to compute BERTSCORE by looking at the model's activations and comparing those with other layers. This also removes the reliance on intermediate decoding, improving efficiency by at least 40x, and adding negligible overhead for confidence estimation.

D. Prompts used for tasks

D.1. MMLU

```
"Answer the following multiple choice question by giving the most appropriate response.
    Answer with a single letter: either 'A', 'B', 'C', or 'D'. Even though the answer
    choices may have a period at the end, answer with just the letter. The following are
    multiple choice questions (with answers) about {CURR_SUBJECT} Question: {ICL EXAMPLE1}
     Answer: {ICL ANSWER1} ... Question: {ICL EXAMPLE 5} Answer: {ICL ANSWER5} Now answer
    the following multiple choice question with a single letter. Question: {QUESTION}
    Answer: "
```

Listing 1. System Prompt for all models for MMLU

D.2. APIGen

```
"You are an agent that specializes in function calling. You are given a list of tools,
    each with a name, description, and arguments. Each of the arguments may have a name,
    description, type, and a default value that the argument takes. You will be given a
    query. Please answer the query using the provided tools. Format your final answer as
    list of function calls in a json-like format without any linebreaks. Make sure the
    arugments for the function-calling are called arguments, not parameters. Here are the
    list of tools available: {CURRENT_EXAMPLES_AVAILABLE_TOOLS}. {INPUT_QUERY}"
```

Listing 2. System Prompt for all models for APIGen

D.3. SCITLDR

```
"You are a helpful assistant who is an expert in summarizing scientific papers. Given the
    title and abstract of the paper in the following format TITLE: <title> ABSTRACT: <
```

```
abstract> SUMMARY: , write a short summary of the abstract, which will go after the
SUMMARY: in the prompt. Summaries must be no more than one sentence, but this sentence
 can be longer than an average sentence. Most summaries are between 15 and 25 words.
TITLE: {CURR_EXAMPLE_TITLE} ABSTRACT: {CURR_EXAMPLE_ABSTRACT} SUMMARY: "
```

Listing 3. System Prompt for all models for SCITLDR