Self-Aware Knowledge Probing: Evaluating Language Models' Relational Knowledge through Confidence Calibration

Anonymous ACL submission

Abstract

Knowledge probing quantifies how much relational knowledge a language model (LM) has acquired during pre-training. Existing knowledge probes evaluate model capabilities through metrics like prediction accuracy and precision. Such evaluations fail to account for the model's reliability, reflected in the calibration of its confidence scores. In this paper, we propose a novel calibration probing framework for relational knowledge, covering three modalities of model confidence: (1) intrinsic confidence, (2) structural consistency and (3) semantic grounding. Our extensive analysis of ten causal and six masked language models reveals that most models, especially those pre-trained with the masking objective, are overconfident. The best-calibrated scores come from confidence estimates that account for inconsistencies due to statement rephrasing. Moreover, even the largest pre-trained models fail to encode the semantics of linguistic confidence expressions accurately.

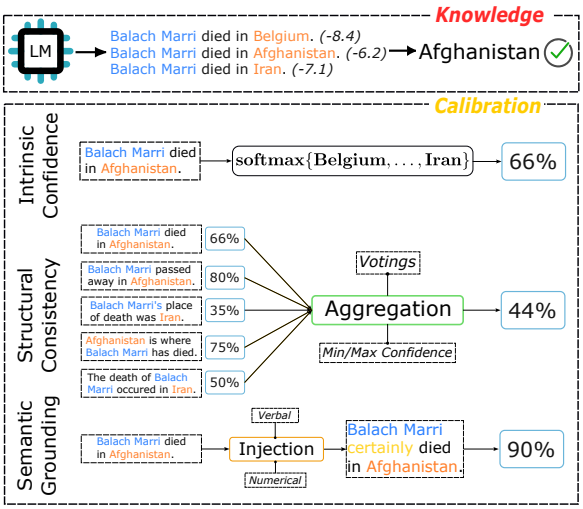

Figure 1: Using existing factual knowledge (like the place of death of a particular person), we measure an LM's awareness of its relational knowledge in three modalities: (1) intrinsic confidence of predictions, (2) structural consistency of predictions across multiple rephrasings of the same statement, and (3) semantic grounding using epistemic markers.

1 Introduction

Knowledge probes serve as the primary diagnostic tool for quantifying the relational knowledge internalized by language models during pre-training. By querying models with factual triplets—typically formatted as cloze-style statements such as "Balach Marri died in [MASK]"—researchers can assess a model's grasp of diverse semantic relations (Petroni et al., 2019; Youssef et al., 2023). Such evaluations are critical for understanding how internalized knowledge correlates with performance on complex downstream reasoning tasks.

However, existing knowledge probing frameworks (Youssef et al., 2023; Kalo and Fichtel, 2022; Wiland et al., 2024) focus almost exclusively on attainment: the accuracy and precision with which a model retrieves a fact. While accurate recall is imperative, it provides an incomplete picture of model reliability. A truly reliable model must be "self-aware," exhibiting confidence scores that are well-calibrated with its probability of being correct (Guo et al., 2017). In a perfectly calibrated system, a confidence score of 0.8 implies an 80% likelihood of factual correctness. Without such alignment, even high-accuracy models are prone to silent failures and overconfident hallucinations, as they lack the mechanisms to signal the limits of their own knowledge.

Knowledge calibration probing. In this paper, we propose a novel calibration probing framework for relational knowledge. Unlike prior work that treats knowledge as a binary state of presence or absence, our approach evaluates the epistemic self-awareness of language models. We decompose model confidence into three distinct modalities: (1) *intrinsic confidence*, derived from raw log-probabilities; (2) *structural consistency*, measured

through agreement across semantically equivalent rephrasings; and (3) *semantic grounding*, evaluated through the model's response to explicit linguistic markers of uncertainty. See Figure 1 for an overview.

This multi-faceted approach allows for a granular analysis of how model architecture and scale influence the reliability of factual retrieval. Specifically, we leverage a closed-set probing setup (Wiland et al., 2024) to facilitate a direct, head-to-head comparison between causal language models (CLMs) and masked language models (MLMs). While these architectures differ fundamentally in their training objectives—autoregressive prediction versus denoising—our framework provides a unified metric space to evaluate their respective epistemic self-awareness. To ensure our findings are robust across diverse knowledge domains and model families, we evaluate 16 distinct models ranging from 109M to 7B parameters.

Contributions. We therefore offer the following contributions:

- We present a novel setup for knowledge probing that supports six distinct confidence estimates across intrinsic, structural, and semantic dimensions.
- We provide the first large-scale comparison of calibration between CLMs and MLMs, analyzing 16 models to isolate the impact of pretraining objectives on epistemic awareness.
- We evaluate the efficacy of various confidence estimation strategies—ranging from raw log-probabilities to multi-prompt aggregations—for improving calibration and selective prediction.
- We investigate the impact of explicit linguistic confidence expressions on model reliability, testing whether LMs can integrate verbalized uncertainty into their factual retrieval process.

Our analysis reveals a critical "reliability gap" in current architectures; specifically, that MLMs—despite competitive factual accuracy—remain fundamentally less aware of their own knowledge than CLMs. Furthermore, our results suggest a clear scale-dependency for semantic grounding: while smaller models are "semantically blind" to expressions of doubt, larger models begin to align their internal confidence with linguistic cues. To facilitate further research into model reliability, we provide our proposed framework as an extensible benchmark for evaluating the epistemic awareness of future language models.

2 Related Work

The evaluation of model reliability in factual retrieval lies at the intersection of two research areas: uncertainty quantification for deep learning and relational knowledge probing.

2.1 Confidence Estimation for LMs

Confidence estimation methods can be broadly categorized into three modalities, mirroring the framework we propose in this paper.

Intrinsic confidence. The most direct approach utilizes the model's internal probability distribution. Early work on calibration in neural networks (Guo et al., 2017) focused on softmax-normalized scores, a technique recently extended to LMs for tasks like hallucination detection (Huang et al., 2025) and selective prediction (Duan et al., 2024). Our work specifically examines how these intrinsic signals behave when the answer space is constrained to a fixed set of factual candidates.

Structural consistency. This modality relies on the intuition that a model "knows" a fact if it provides the same answer across various semantically equivalent formulations. Techniques such as *Self-Consistency* (Wang et al., 2023) and *Semantic Entropy* (Kuhn et al., 2023) leverage multiple model outputs to derive uncertainty. While consistency is often used as a proxy for reliability in knowledge bases (Zheng et al., 2024), our framework evaluates whether these consistency signals are well-calibrated with factual truth.

Semantic grounding. A growing body of research explores "self-reported" or "verbalized" confidences, where the model expresses its certainty through natural language (Tian et al., 2023). Previous studies have found that LMs are often reluctant to express uncertainty (Zhou et al., 2024) or fail to align their verbalized scores with their internal log-probabilities (Xiong et al., 2024). We extend this line of inquiry to relational knowledge statements by injecting explicit epistemic markers (e.g., *certainly*, *possibly*) into the probing templates.

2.2 Relational Knowledge Probing

The field of knowledge probing was pioneered by the LAMA benchmark (Petroni et al., 2019), which

used cloze-style prompts to quantify the factual knowledge of MLMs. Subsequent benchmarks like KAMEL (Kalo and Fichtel, 2022) and Myriad-LAMA (Zhao et al., 2024) expanded the diversity of relations and entities. However, these probes focus primarily on attainment—measuring whether a model can predict the correct token—rather than the model's awareness of its knowledge boundaries.

2.3 The BEAR Framework

To evaluate calibration across CLMs and MLMs, we require a probing mechanism that provides a unified scoring rule. The BEAR probe (Wiland et al., 2024) achieves this by shifting the task from open-ended generation to a closed-set ranking.

For a given relation instance such as (BALACH MARRI, `died-in`, AFGHANISTAN), BEAR provides a set of K candidate objects, including the true answer plus plausible distractors (e.g. IRAN). The probe tests whether an LM distinguishes the true fact from the distractors by estimating the full log-likelihood of each statement created with a template: *Balach Marri died in Afghanistan* and *Balach Marri died in Iran*.

Scorings. For CLMs, a full sentence-level log-likelihood is simply the sum of the token log-likelihoods in the sequence:

$$\log P(x) = \sum_{t=1}^{|x|} \log P(x_t \mid x_{<t})$$

For MLMs, BEAR utilizes pseudo log-likelihoods (PLLs) (Salazar et al., 2020). We adopt the refinement by (Kauf and Ivanova, 2023), which masks tokens one-by-one and accounts for multi-token words by masking all future within-word tokens:

$$\text{PLL}(x) = \sum_{t=1}^{|x|} \log P(x_t \mid x_{\setminus t})$$

Answer selection. Once a log-likelihood score ℓ_i is assigned to every candidate statement o_i in the answer set, the model's prediction $\hat{y}$ is the object that maximizes this score: $\hat{y} := \arg\max_{i \in \{1,\ldots,K\}} \ell_i$.

3 Confidence Estimation

We propose to evaluate model reliability through confidence calibration on closed-answer set benchmarks like BEAR by normalizing the log-likelihoods using the softmax function (σ). This yields probability values which can be interpreted as the model's confidence in the correctness of a given answer (Figure 1).

3.1 Intrinsic Confidence

Estimates in this category are characterized by the fact that log-likelihood estimation for each instance is done using a single relation-specific template.

Base-Confidence. We define a model agnostic baseline estimate as the maximum value after softmax normalization:

$$C_{\text{Base}}(\hat{y}) := \max_{i \in \{1,\ldots,K\}} \sigma(\boldsymbol{\ell})_i \tag{1}$$

where $\boldsymbol{\ell}$ represents the vector of log-likelihoods of the answer options. We also used this baseline estimate to find the token log-likelihood reduction strategy which maximizes accuracy and minimizes calibration error (Appendix B).

Margin-Confidence. Another single-template estimate is the confidence margin, also commonly used in the context of image classification (Pleiss et al., 2020; Joshi et al., 2009). With *Margin-Confidence*, we compute the difference between the largest and second-largest value returned by the softmax, reflecting whether the model's answer is clearly preferred over the second-best answer:

$$C_{\text{Margin}}(\hat{y}) := \sigma(\boldsymbol{\ell})_{[1]} - \sigma(\boldsymbol{\ell})_{[2]} \tag{2}$$

Here, the subscripts [1] and [2] refer to the largest and second-largest values returned by the softmax.

3.2 Structural Consistency

Repeating the probe across multiple equivalent templates yields multiple answers $\tilde{y}_1, \ldots, \tilde{y}_5$. Hence, structural consistency estimates require an intermediate step which we call aggregation to obtain a single prediction $\hat{y}$ (Figure 1).

Aggregations. The selected answers can be aggregated through voting schemes or confidence-based aggregation. Voting schemes reject predictions with insufficient agreement, yielding no answer and zero calibration error by design. Confidence-based aggregation always produces a final answer, potentially based on a single candidate.

In our analysis, we use plurality voting and minimum confidence aggregation. Plurality voting selects the most frequent answer. Minimum confidence aggregation selects the predicted answer with minimum C_{Base} (Equation 1). We denote voting aggregation by superscript $Vote$ and minimum-confidence aggregation by superscript Min.

Average-Confidence. Given an aggregation strategy, we define *Average-Confidence* as the per-

template average of the selected answer:

$$C_{\text{Average}}(\hat{y}) := \frac{\sum_{i=1}^{5} \mathbb{1}(\tilde{y}_i = \hat{y}) \cdot C_{\text{Base}}(\tilde{y}_i)}{5} \quad (3)$$

Consistency-Confidence. Instead of averaging the softmax scores, confidence can be reflected through the relative frequency of the answers:

$$C_{\text{Consistency}}(\hat{y}) := \frac{\sum_{i=1}^{5} \mathbb{1}(\tilde{y}_i = \hat{y})}{5} \quad (4)$$

Hence, $C_{\text{Consistency}}$ bypasses the step of softmax normalization and is discretely distributed. It therefore more closely aligns how humans naturally express confidence, typically in multiples of five (Xiong et al., 2024).

3.3 Semantic Grounding

We study the impact of *verbalized* and *numerical* confidence expressions on both accuracy and calibration in a zero-shot setting. To achieve this, we modify the templates by injecting verbalized and numerical confidence expressions (Zhou et al., 2023). As a confidence score, we use the single-template C_{Base}, as defined in Equation 1.

Verbalized. For the verbalized setup, we inject the epistemic marker *possibly* to lower the confidence of the statement and the marker *certainly* to strengthen it. For instance, *Balach Marri certainly died in Afghanistan*.

Numerical. To evaluate the impact of numerical confidence expressions in relational knowledge statements, we modified the relation templates to include formulations such as *"I'm x% confident that Balach Marri died in Afghanistan."*, with $x \in \{0, 25, \ldots, 100\}$.

4 Experimental Setup

In this section, we give an overview of the datasets, models and evaluation metrics we used.

4.1 Datasets

We base our calibration probe around the BEAR framework (Wiland et al., 2024). The dataset contains 7,731 instances across 60 relations. Each relation has three semantically equivalent template statements. As part of this work, we extended the number of templates to five per relation using ChatGPT (see Appendix G).

Out of all relations, 46 have cardinality N:1 (different subjects can share the same answer) and 14 have cardinality 1:1 (each object is the answer to exactly one subject). For the N:1 relations, the average number of answer options per instance is 6.5. Each 1:1 relation consists of 60 instances, with 60 answer options per instance.

Additionally, we run experiments with the Wiki subset of the FACTOR (Muhlgay et al., 2024) dataset, which has one correct and three incorrect completions for each statement. These results are presented in Appendix F.

4.2 Language Models

We evaluated 16 different models, 10 CLMs and 6 MLMs (listed in Appendix A). The MLMs range from a size of 109 million parameters to 561 million parameters. For the CLMs, the smallest model is the `opt-125m` with 125 million parameters, and the largest is `gemma-7b` with 7 billion parameters.

4.3 Evaluation Metrics

Adaptive Calibration Error. We avoid the commonly used Expected Calibration Error (ECE) (Naeini et al., 2015) with fixed bin widths to measure the average deviation between confidence and accuracy. Instead, we use the *Adaptive Calibration Error* (*ACE*) (Nixon et al., 2020) to account for the underlying confidence distributions. Low *ACEs* indicate high knowledge awareness.

Calibration Curves. To assess over- and under-confidence, we rely on calibration curves for which we plot the confidence scores against accuracy, i.e. the fractions of $\mathbb{1}(\hat{y}_i = y_i)$ in a corresponding confidence bin. For both, *ACE* and calibration curves, we use quantile binning with 20 bins.

Brier Score. Alongside *ACE* as a scalar metric, we report the *Brier Score* (Brier, 1950). Unlike *ACE*, the *Brier Score* is not a pure calibration metric and also captures the discriminative power of an estimate. We use correctness labels $\mathbb{1}(y_i = \hat{y}_i)$ as a target (Tian et al., 2023; Lin et al., 2022).

5 Experiment 1: Calibration Evaluation

This experiment evaluates the calibration of the confidence estimates for each model.

5.1 Comparison of Confidence Estimates

Table 1 presents the *ACEs* and the *Brier Scores* for all models and each confidence estimate. Figure 2 shows the calibration curves for selected models.

C^{Min}_{Average} is the best-performing estimate. The structural consistency estimates $C^{Vote}_{\text{Average}}$ and

Model	ACE (↓)				Brier Score (↓)					
	C_{Base}	C_{Margin}	$C^{Vote}_{\text{Average}}$	C^{Min}_{Average}	C_{Base}	C_{Margin}	$C^{Vote}_{\text{Average}}$	C^{Min}_{Average}	$C^{Vote}_{\text{Consistency}}$	$C^{Min}_{\text{Consistency}}$
`bert-base-cased`	0.454	0.295	0.298	**0.218**	0.367	0.276	0.244	**0.205**	0.408	0.338
`bert-large-cased`	0.452	0.302	0.278	**0.198**	0.373	0.285	0.230	**0.196**	0.369	0.310
`gemma-2b`	0.113	0.053	0.068	**0.032**	0.162	0.155	0.133	**0.098**	0.190	0.157
`gemma-7b`	0.077	0.046	0.081	**0.042**	0.133	0.134	0.114	**0.083**	0.144	0.124
`gpt2`	0.295	0.157	0.158	**0.100**	0.237	0.181	0.140	**0.117**	0.315	0.259
`gpt2-medium`	0.249	0.121	0.116	**0.071**	0.218	0.179	0.140	**0.110**	0.286	0.230
`gpt2-large`	0.236	0.104	0.101	**0.070**	0.206	0.163	0.139	**0.116**	0.290	0.238
`gpt2-xl`	0.240	0.128	0.086	**0.068**	0.228	0.193	0.152	**0.121**	0.284	0.235
`opt-125m`	0.299	0.153	0.164	**0.116**	0.241	0.186	0.147	**0.130**	0.320	0.273
`opt-350m`	0.265	0.139	0.137	**0.098**	0.224	0.177	0.150	**0.127**	0.320	0.263
`opt-1.3b`	0.207	0.100	0.068	**0.056**	0.214	0.187	0.163	**0.126**	0.290	0.232
`opt-6.7b`	0.154	0.069	0.029	**0.024**	0.187	0.168	0.144	**0.110**	0.232	0.191
`roberta-base`	0.482	0.321	0.268	**0.165**	0.387	0.291	0.196	**0.162**	0.322	0.265
`roberta-large`	0.441	0.291	0.228	**0.144**	0.367	0.278	0.188	**0.158**	0.304	0.254
`xlm-roberta-base`	0.585	0.427	0.395	**0.274**	0.481	0.368	0.283	**0.225**	0.448	0.362
`xlm-roberta-large`	0.559	0.412	0.360	**0.249**	0.461	0.359	0.272	**0.224**	0.430	0.354

Table 1: *ACE* and *Brier Score* of our estimates. **Bold values** indicate the best-performing estimate per model. ***ACE:*** • [0,0.05), • [0.05,0.10), • [0.10,0.20), • ≥ 0.20; ***Brier Score:*** • [0–0.10), • [0.10–0.20), • [0.20–0.25), • ≥ 0.25. A *Brier Score* of 0.25 corresponds to a *random baseline*, always predicting a confidence of 0.5. For *ACE*, we consider values < 0.1 *well-calibrated*, where the confidence deviates from accuracy by 10 percentage points at most. We do not report *ACE* for $C_{\text{Consistency}}$ due to its discrete distribution.

C^{Min}_{Average} clearly outperform the other approaches across all models in terms of *Brier Score* and across most models in terms of *ACE*. For example, for `opt-6.7b`, C^{Min}_{Average} achieves an *ACE* of 0.024, which is close to perfect calibration. This is reflected in the calibration curve shown in Figure 2, which deviates only slightly from the identity line. We further note that for larger models, the difference between C_{Average} and C_{Margin} is often small. Finally, $C_{\text{Consistency}}$ performs very poorly in terms of *Brier Score*, showing that confidence is much better captured by aggregating log-likelihoods than through answer agreement.

Better calibration with minimum confidence aggregation. Minimum confidence aggregation reduces *ACE* and *Brier Score* more effectively than plurality voting, even tough we hard code vote failures with zero calibration error. This holds for both, C_{Average} and $C_{\text{Consistency}}$. With C^{Min}_{Average}, only `bert-base-cased` and the `xlm-roberta` models exhibit an $ACE \geq 0.2$, while eight out of 14 models achieve an $ACE \leq 0.1$. As shown in Figure 2, both strategies behave similarly in low-confidence regions, but minimum confidence aggregation shows less underconfidence in high-confidence regions. Appendix C provides a broader evaluation of aggregation strategies, particularly stricter voting schemes.

Structural consistency has high costs. The advantage of structural consistency comes at a cost. For example, in the case of `gpt2`, the number of estimated token log-likelihoods for the full dataset using only the first template is 2.6M. When using all templates, this increases by more than a factor of five to 13.2M log-likelihoods. The cost is even higher for MLMs, where sequences must be duplicated and tokens masked.

C_{Margin} strongly improves over C_{Base}. For intrinsic confidence estimates derived using only a single template, we find that C_{Margin} consistently performs well in terms of *ACE* and *Brier Score* across all LMs. With C_{Margin}, `gemma-7b` even achieved an *ACE* of 0.05, indicating good calibration and reliability. By plotting the calibration curves in Figure 2, we see that C_{Base} is markedly overconfident. The C_{Margin} curves are closer to the identity line, which confirms that C_{Margin} is an effective strategy for mitigating overconfidence.

We identify another key advantage of C_{Margin}. For all models included in Figure 2, C_{Base} does not produce enough scores in the lower confidence ranges to derive a calibration curve that covers the full $[0, 1]$ interval. This prevents a complete evaluation of calibration across the entire confidence range. C_{Margin}, however, extends the curve into these lower confidence regions without limiting the

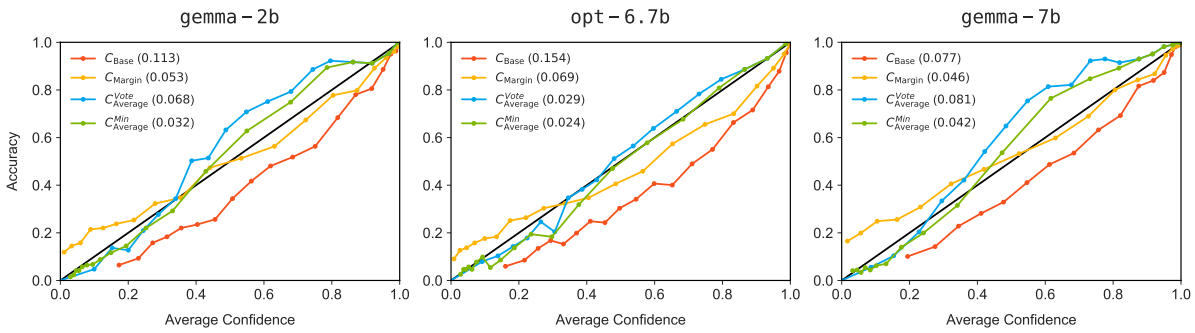

Figure 2: Calibration curves of our confidence estimates. *ACE* in parentheses.

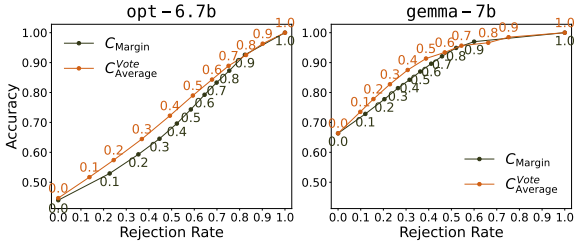

Figure 3: Accuracy Rejection Curves (Nadeem et al., 2008) (fraction of rejected answers versus accuracy among the non-rejected answers) for confidence thresholds of $0.1, \ldots, 0.9$. For a given threshold, curves closer to the upper left indicate better performance.

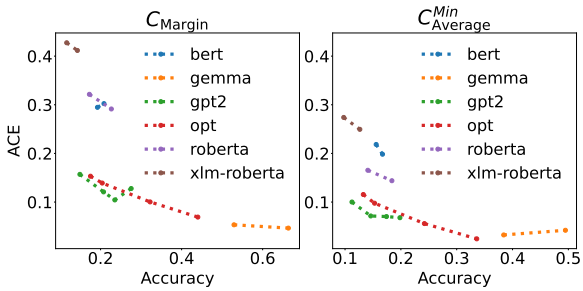

Figure 4: *ACE*-accuracy scatter plot, demonstrating negative correlation between the two metrics.

model's ability to output enough high-confidences.

We also observe that C_{Margin} can introduce underconfidence in the lower range of the confidence scores, pushing the calibration curve above the identity line. From a practical perspective, we would always prefer a model that is underconfident rather than one that is overconfident.

The largest models are also most calibrated. The six largest models: `gemma-7b`, `gemma-2b`, `opt-6.7b`, `opt-1.3b`, `gpt2-large` and `gpt2-xl` have the lowest *ACEs* and *Brier Scores*. Also, these six models all achieve, with their best estimate, an $ACE \leq 0.1$, which we consider as well-calibrated. The only model achieving this threshold for all estimates is `gemma-7b`. Moreover, most CLMs and MLMs show a better *Brier Score* than the random baseline (always predicting a confidence of 0.5).

5.2 Selective Prediction Evaluation

We evaluate the practical usability of the confidence estimates for selective prediction — where we discard low-confidence data points in order to increase the accuracy on the remaining ones. Figure 3 shows the accuracy-rejection curves — plotting the share of discarded samples against the accuracy of the remaining ones — of C_{Margin} and $C^{Vote}_{\text{Average}}$ for the two largest models. These curves show that the confidence estimates are effective for selective prediction. For `gemma-7b`, applying a confidence threshold of 0.1 to $C^{Vote}_{\text{Average}}$ increases accuracy by 8 percentage points while rejecting only 10% of the dataset.

5.3 Correlation between ACE and Accuracy

We find an overall negative correlation between accuracy and *ACE*, exemplary shown in Figure 4 for C_{Margin} and C^{Min}_{Average}. Since larger models generally achieve higher accuracy, this result also implies a negative correlation between calibration error and model size. With some exceptions, this trend holds across pre-training objectives and model families. From the perspective of overall trustworthiness, this is concerning: it suggests that gains in calibration primarily arise from higher accuracy rather than from well-aligned confidences. This trend is the weakest for C_{Margin}, when compared to the other confidences. This indicates that calibration under C_{Margin} depends less on model size or accuracy, and smaller models can be as aware of their knowledge limitations as larger ones.

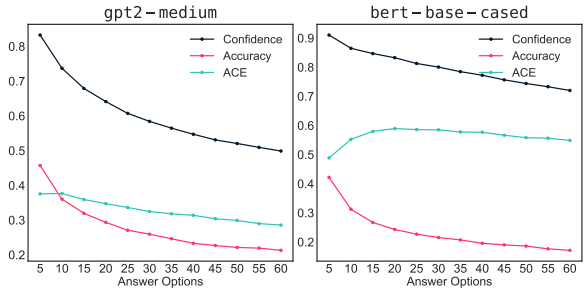

Figure 5: Top-label (average) confidence, accuracy and *ACE* as a function of the number of answer options (including the correct answer). Each point represents the mean over three repeated samplings.

5.4 Impact of the Number of Answer Options

We investigate how average confidence of C_{Base}, accuracy and *ACE* change as a function of the number of answer options. We create varying number of options by sampling subsets of the 60 options in the 1:1 relations in BEAR. In Figure 5, we plot this relationship for `gpt2-medium` and `bert-base-cased`, and we observed similar findings for the other CLMs and MLMs. We find that increasing the number of answer options can significantly reduce calibration error. This contradicts the finding in Li et al. (2024), however they only experiment with low numbers (up to 4 options).

Average confidence and accuracy decrease as the number of answer options grows. In CLMs, confidence drops faster than accuracy, which results in *ACE* decreasing because the models are overconfident. In contrast, in MLMs, accuracy drops faster than confidence. Thus, *ACE* increases before stabilizing after 15–20 options. Therefore, to obtain robust estimates with both MLMs and CLMs, a fairly high number of answer options is required, which can lead to substantial computational costs.

To confirm our findings, we evaluate C_{Base} and C_{Margin} on the Wiki-FACTOR benchmark in Appendix F. As this dataset only includes four answer options, we found significantly higher calibration errors. Through this benchmark, we also confirm the benefits of C_{Margin} over C_{Base}, highlighting it as a more reliable confidence estimate.

6 Experiment 2: Comparison of CLMs and MLMs

To examine the differences in calibration between the two pre-training objectives, we matched CLMs and MLMs with similar accuracies. The goal was to minimize the influence of differing accuracies on calibration. The matched model pairs are listed in Table 2, together with the *Brier Scores* of the best-performing estimates C_{Margin} and C^{Min}_{Average}. To further reduce the impact of differing training data for each of the five pairs, we also identified the domain where the models are most similar in terms of accuracy. An overview of the most similar domain per model pair is given in Appendix D.

Better Calibration of CLMs. Although MLMs with the same or even lower number of parameters can achieve a similar, and in some cases even higher accuracy, we find that CLMs are much better calibrated using most of our estimates (see Table 2). Across all five model pairs, we observe substantial calibration differences, often reaching differences of approximately 0.1 to 0.2 for C_{Margin} and C^{Min}_{Average}. For C_{Margin} and C^{Min}_{Average}, all CLMs achieve a *Brier Score* of 0.2 or lower, while none of the MLMs do. When restricting the analysis to the selected domains, we still find the CLMs much better calibrated for most of the estimates and model pairs, similar to the results on the full dataset.

Softmax MLM scores do not reflect model confidence. MLMs do not explicitly model a sentence-level log-likelihood during their pre-training phase and we rely on the pseudo log-likelihoods, as explained in subsection 2.3. While this scoring is sufficient to distinguish between correct and incorrect answers for the BEAR probe, we believe that the absence of a well-defined sentence-level log-likelihood is the main reason behind the large calibration discrepancies between CLMs and MLMs.

7 Experiment 3: Impact of Linguistic Confidence Expressions

We examine the impact of adding epistemic markers to the relational knowledge statements. For this setup of semantic grounding, we inject the verbal epistemic markers "certainly" (strengthener) and "possibly" (weakener) into the statements. Additionally, we modified the templates with the numeric confidence expressions "I'm x% confident..." for $x \in \{0, 25, \ldots, 100\}$.

Injected markers mostly hurt calibration. In the previous experiments we showed that LMs are consistently overconfident. We therefore expected that weakening the confidence of the statements would reduce this overconfidence, but Figure 6 indicates that for most models *ACE* actually increases under these injections. Verbalized strengthening injections consistently increase *ACE* across all models.

Model	All Domains			Selected Domain		
	Acc.	C_{Margin}	C^{Min}_{Average}	Acc.	C_{Margin}	C^{Min}_{Average}
`opt-125m` (125M)	15.6	**0.186**	**0.130**	10.6	**0.121**	**0.095**
`roberta-base` (125M)	16.0	0.291	0.162	10.4	0.291	0.135
`gpt2-medium` (355M)	18.4	**0.179**	**0.110**	16.0	**0.179**	**0.114**
`bert-base-cased` (109M)	18.1	0.276	0.205	14.8	0.239	0.152
`gpt2-large` (812M)	21.5	**0.163**	**0.116**	19.6	**0.185**	**0.126**
`roberta-large` (355M)	21.1	0.278	0.158	19.0	0.301	0.172
`opt-350m` (350M)	19.0	**0.177**	**0.127**	10.5	**0.149**	0.113
`bert-large-cased` (335M)	19.3	0.285	0.196	10.1	0.198	**0.127**
`gpt2` (137M)	12.8	**0.181**	**0.117**	10.0	**0.137**	**0.091**
`xlm-roberta-large` (561M)	13.5	0.359	0.224	10.0	0.387	0.217

Table 2: *Brier Scores* (↓) for matched CLM/MLM pairs. Accuracy (in %) is the average over all templates.

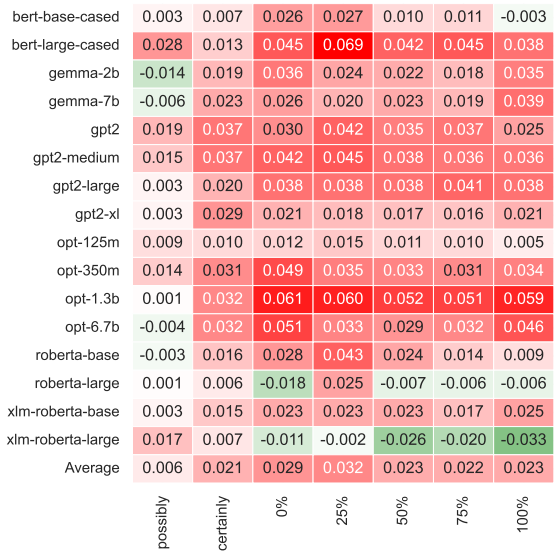

Figure 6: Difference in *ACE* between the non-injected and injected templates. The x-axis shows the applied injection. Negative values (green) show an improvement.

The situation is similar with numerical injections — they largely increase *ACE* and worsen calibration. The only exception to this are some MLMs, such as `roberta-large` and `xlm-roberta-large` where we see improved *ACE* with numerical confidence expressions. Additionally, we found that the markers hurt model accuracy on the BEAR probe in all settings (see Appendix E).

Verbal markers correctly shift calibration curves for large models. While the weakener marker counter-intuitively degrades calibration for most models, it does successfully improve it for the largest models in our analysis: `gemma-2b`, `gemma-7b` and `opt-6.7b`. For those models, "possibly" effectively reduced the overconfidence, while "certainly" further increases the overconfidence relative to the original variant. This is also demonstrated in the calibration curves in Figure 7, where the curves got shifted in the expected directions.

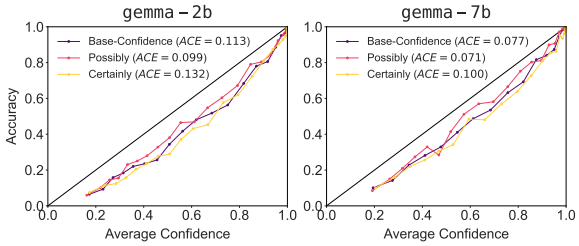

Figure 7: Calibration curves of C_{Base} (Equation 1) derived from the original and verbal injected templates.

8 Conclusion

This paper proposes self-aware knowledge probing, which assesses the reliability of language models' relational knowledge through the lens of calibration. Our probing setup covers three confidence estimation modalities: intrinsic confidence, structural consistency and semantic grounding.

We show that obtaining well-calibrated confidences for relational knowledge is highly challenging, strongly indicating that LMs lack a deeper understanding of their own knowledge boundaries. Regardless of the pre-training objective and model, confidences are consistently overconfident, and the MLM scoring generally worsens this behavior.

Structural consistency estimates effectively address this issue, but they also reveal a high sensitivity to formulation changes. If this sensitivity is not accounted for, LMs remain unreliable in terms of calibration, especially for low-accuracy models. In practice, addressing this sensitivity requires prompt perturbations, which limits the practical usability of LMs.

Limitations

Following are some limitations in our results that we want to address.

For our purposes, quantifying calibration with respect to the top answer is sufficient, since BEAR contains no N:M relations. However, a complete assessment of reliability would also account for such relations. Calibration must then be evaluated over all answer options, for example using the *Brier Score* with one-hot-encoded ground-truth labels $\boldsymbol{y}_i$ or the Static Calibration Error (Nixon et al., 2020).

Our calibration probing serves mainly as a diagnostic tool and is quite specific to closed-answer set scenarios. The calibration error values are therefore unlikely to transfer directly to more practical settings. Such scenarios typically lack predefined answer options, particularly across different formulations, necessary to derive structural consistencies. Moreover, the predefined answer options limit our evaluation to output uncertainty, but unreliable responses can also stem from input uncertainty, such as vaguely formulated prompts.

References

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

Jinhao Duan, Hao Cheng, Shiqi Wang, Alex Zavalny, Chenan Wang, Renjing Xu, Bhavya Kailkhura, and Kaidi Xu. 2024. Shifting attention to relevance: Towards the predictive uncertainty quantification of free-form large language models. *Preprint*, arXiv:2307.01379.

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

Benjamin Heinzerling and Kentaro Inui. 2021. Language models as knowledge bases: On entity representations, storage capacity, and paraphrased queries. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1772–1791, Online. Association for Computational Linguistics.

Yuheng Huang, Jiayang Song, Zhijie Wang, Shengming Zhao, Huaming Chen, Felix Juefei-Xu, and Lei Ma. 2025. Look before you leap: An exploratory study of uncertainty analysis for large language models. *IEEE Transactions on Software Engineering*, 51(2):413–429.

Ajay J. Joshi, Fatih Porikli, and Nikolaos Papanikolopoulos. 2009. Multi-class active learning for image classification. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2372–2379.

Jan-Christoph Kalo and Leandra Fichtel. 2022. Kamel: Knowledge analysis with multitoken entities in language models. In *Automated Knowledge Base Construction*.

Carina Kauf and Anna Ivanova. 2023. A better way to do masked language model scoring. *Preprint*, arXiv:2305.10588.

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

Wangyue Li, Liangzhi Li, Tong Xiang, Xiao Liu, Wei Deng, and Noa Garcia. 2024. Can multiple-choice questions really be useful in detecting the abilities of llms? *Preprint*, arXiv:2403.17752.

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

Dor Muhlgay, Ori Ram, Inbal Magar, Yoav Levine, Nir Ratner, Yonatan Belinkov, Omri Abend, Kevin Leyton-Brown, Amnon Shashua, and Yoav Shoham. 2024. Generating benchmarks for factuality evaluation of language models. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 49–66, St. Julian's, Malta. Association for Computational Linguistics.

Malik Sajjad Ahmed Nadeem, Jean-Daniel Zucker, and Blaise Hanczar. 2008. Accuracy-rejection curves (arcs) for comparing classification methods with a reject option. In *Proceedings of the JMLR: Workshop and Conference Proceedings, Machine Learning in Systems Biology*, volume 8, pages 65–81, Paris, France. JMLR.org.

Mahdi Pakdaman Naeini, Gregory Cooper, and Milos Hauskrecht. 2015. Obtaining well calibrated probabilities using bayesian binning.

Jeremy Nixon, Mike Dusenberry, Ghassen Jerfel, Timothy Nguyen, Jeremiah Liu, Linchuan Zhang, and Dustin Tran. 2020. Measuring calibration in deep learning. *Preprint*, arXiv:1904.01685.

Fabio Petroni, Tim Rocktäschel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, Alexander H. Miller, and Sebastian Riedel. 2019. Language models as knowledge bases? *Preprint*, arXiv:1909.01066.

Geoff Pleiss, Tianyi Zhang, Ethan R. Elenberg, and Kilian Q. Weinberger. 2020. Identifying mislabeled data using the area under the margin ranking. *Preprint*, arXiv:2001.10528.

Max Ploner, Jacek Wiland, Sebastian Pohl, and Alan Akbik. 2024. Lm-pub-quiz: A comprehensive framework for zero-shot evaluation of relational knowledge in language models. *Preprint*, arXiv:2408.15729.

Julian Salazar, Davis Liang, Toan Q. Nguyen, and Katrin Kirchhoff. 2020. Masked language model scoring. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics.

Katherine Tian, Eric Mitchell, Allan Zhou, Archit Sharma, Rafael Rafailov, Huaxiu Yao, Chelsea Finn, and Christopher D. Manning. 2023. Just ask for calibration: Strategies for eliciting calibrated confidence scores from language models fine-tuned with human feedback. *Preprint*, arXiv:2305.14975.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Jacek Wiland, Max Ploner, and Alan Akbik. 2024. Bear: A unified framework for evaluating relational knowledge in causal and masked language models. *Preprint*, arXiv:2404.04113.

Miao Xiong, Zhiyuan Hu, Xinyang Lu, Yifei Li, Jie Fu, Junxian He, and Bryan Hooi. 2024. Can llms express their uncertainty? an empirical evaluation of confidence elicitation in llms. *Preprint*, arXiv:2306.13063.

Paul Youssef, Osman Koraş, Meijie Li, Jörg Schlötterer, and Christin Seifert. 2023. Give me the facts! a survey on factual knowledge probing in pre-trained language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 15588–15605, Singapore. Association for Computational Linguistics.

Xin Zhao, Naoki Yoshinaga, and Daisuke Oba. 2024. What matters in memorizing and recalling facts? multifaceted benchmarks for knowledge probing in language models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 13186–13214, Miami, Florida, USA. Association for Computational Linguistics.

Danna Zheng, Mirella Lapata, and Jeff Z. Pan. 2024. How reliable are llms as knowledge bases? rethinking facutality and consistency. *Preprint*, arXiv:2407.13578.

Kaitlyn Zhou, Jena Hwang, Xiang Ren, and Maarten Sap. 2024. Relying on the unreliable: The impact of language models' reluctance to express uncertainty. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3623–3643, Bangkok, Thailand. Association for Computational Linguistics.

Kaitlyn Zhou, Dan Jurafsky, and Tatsunori Hashimoto. 2023. Navigating the grey area: How expressions of uncertainty and overconfidence affect language models. *Preprint*, arXiv:2302.13439.

A Overview of Models

Table 3 lists all language models included in our experiments.

Model	# Param.	Objective	Accuracy
`bert-base-cased`	109M	MLM	18.1
`bert-large-cased`	335M	MLM	19.3
`gemma-2b`	2.0b	CLM	50.0
`gemma-7b`	7.0b	CLM	62.4
`gpt2`	137M	CLM	12.8
`gpt2-medium`	355M	CLM	18.4
`gpt2-large`	812M	CLM	21.5
`gpt2-xl`	1.6b	CLM	25.3
`opt-125m`	125M	CLM	15.6
`opt-350m`	350M	CLM	19.0
`opt-1.3b`	1.3b	CLM	31.0
`opt-6.7b`	6.7b	CLM	42.8
`roberta-base`	125M	MLM	16.0
`roberta-large`	355M	MLM	21.1
`xlm-roberta-base`	279M	MLM	11.2
`xlm-roberta-large`	561M	MLM	13.5

Table 3: Models considered in this work.

B Reduction Strategy Evaluation

We compare different strategies to reduce a set of token log-likelihoods to a single sentence-level log-likelihood. Following are the four different strategies, i.e. variants of C_{Base} (Equation 1):

1. *Sum*: summation of all tokens in the sequence: tokens corresponding to the object, subject, and template.

2. *Mean*: averaging over all tokens in the sequence.

3. *Sum (A)*: summation over (answer) object tokens only.

4. *Mean (A)*: averaging over answer (object) tokens only.

We compare these strategies in terms of accuracy and *ACE*. We additionally calculate their harmonic mean weighting accuracy and *ACE* equally:

$$H := \frac{2 \cdot \text{ACC} \cdot (1 - ACE)}{\text{ACC} + (1 - ACE)} \in [0, 1]$$

where we subtracted the *ACE* from 1 so that higher values reflect better overall performance.

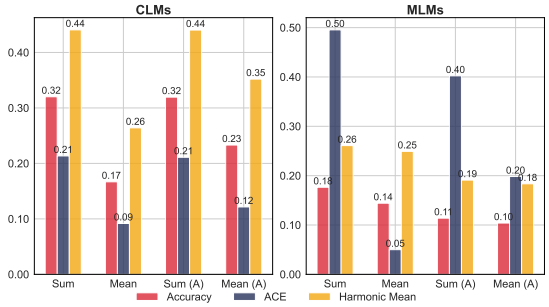

Figure 8: Performance of different token log-likelihood reduction strategies, measured by Accuracy, *ACE* and harmonic mean. Higher harmonic mean is better. Scores are averaged across CLMs and MLMs.

In Figure 8, we show the three metrics for each reduction strategy, averaged separately across CLMs and MLMs. We selected the full-sequence summation — Sum — as our preferred approach, as it resulted in highest H scores for both CLMs and MLMs. $Mean$ and *Mean (A)* do result in better calibration scores than the summation variants, however this comes with a significant decrease in accuracy, making it overall less beneficial.

Furthermore, as shown in Figure 9, mean reduction yields strongly right-skewed confidence distributions, which can result in artificially low calibration errors, especially for low-accuracy models. For high-accuracy models, however, the effect is reversed: calibration error increases because confidences remain too low despite correct predictions. Moreover, the choice of reduction strategy itself limits the model's ability to express high confidences. Under full-sequence mean reduction, the confidences of `opt-1.3b` do not exceed 0.5.

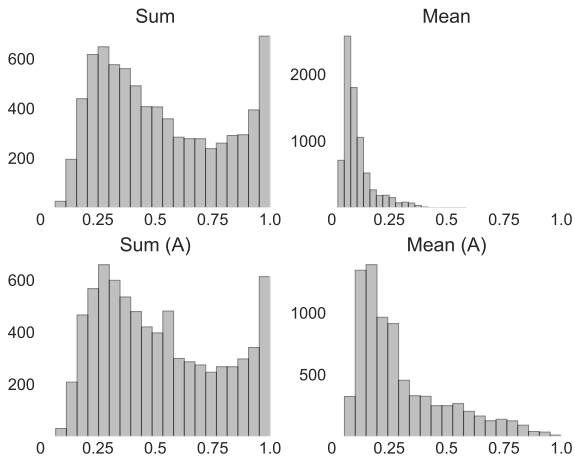

Figure 9: Distributions of C_{Base} for `opt-1.3b` and the different reduction strategies.

C Aggregation Strategies

We evaluate different aggregation strategies for structural consistency. Minimum- and Maximum-Confidence select the candidate answer with the minimum or maximum confidence, respectively. Integers denote how many of the five answers must match for a successful vote, e.g., *2* for plurality, *3* for absolute majority and *5* for unanimity.

Maximum-Confidence aggregation performs worst. In Table 4, we report the *Brier Scores* for the different aggregation strategies. As expected, maximum-confidence aggregation performs worst overall. This is unsurprising, as LMs were shown to be overconfident and maximum-confidence aggregation will only introduce more overconfidence. Compared to minimum-confidence aggregation, the *Brier Score* often increases by around five percentage points. For some models, such as `gemma-2b`, `gemma-7b`, and `opt-6.7b`, the *Brier Score* is almost twice as large.

Votings have high discriminative power. Since the *Brier Score* also captures an estimate's discriminative power, it is unsurprising that voting-based aggregation minimizes it. As expected, stricter voting further increases the discriminative power. Despite the relatively small number of candidate answers, vote failures are highly effective in doing so. For most LMs, requiring all five candidate answers to match yields *Brier Scores* around 0.05. This shows that the models are highly sensitive to the available template variants and confirms a high sensitivity to prompt perturbations (Heinzerling and Inui, 2021).

Model	*Min*	*Max*	*2*	*3*	*4*	*5*
`bert-base-cased`	0.205	0.247	0.244	0.235	0.180	0.114
`bert-large-cased`	0.196	0.237	0.230	0.218	0.161	0.096
`gemma-2b`	0.098	0.173	0.133	0.119	0.071	0.030
`gemma-7b`	0.083	0.145	0.114	0.102	0.061	0.026
`gpt2`	0.117	0.147	0.140	0.124	0.072	0.032
`gpt2-medium`	0.110	0.156	0.140	0.123	0.073	0.025
`gpt2-large`	0.116	0.161	0.139	0.123	0.075	0.028
`gpt2-xl`	0.121	0.178	0.152	0.133	0.083	0.037
`opt-125m`	0.130	0.152	0.147	0.137	0.085	0.043
`opt-350m`	0.127	0.174	0.150	0.138	0.090	0.046
`opt-1.3b`	0.126	0.203	0.163	0.147	0.103	0.050
`opt-6.7b`	0.110	0.189	0.144	0.128	0.081	0.037
`roberta-base`	0.162	0.206	0.196	0.179	0.117	0.056
`roberta-large`	0.158	0.198	0.188	0.171	0.111	0.053
`xlm-roberta-base`	0.225	0.285	0.283	0.275	0.215	0.128
`xlm-roberta-large`	0.224	0.276	0.272	0.260	0.208	0.124

Table 4: *Brier Scores* of C_{Average} for different aggregation strategies. Integers denote the number of matching answers required for successful voting aggregation. • [0–0.10), • [0.10–0.20), • [0.20–0.25), • ≥ 0.25.

This sensitivity and the high discriminative power of voting failures are also evident in the accuracy-rejection curves in Figure 10. We observe that moving from a plurality vote (relative majority) to an unanimous vote (all answers match) increases accuracy among the non-rejected answers by roughly 20 percentage points, with the only exception being `xlm-roberta-large`. Apart from `xlm-roberta-large`, the jumps in accuracy and rejection rates are relatively consistent across the models shown in Figure 10. Sensitivity to prompt perturbations is reflected in the fraction of retained answers, (one minus the rejection rate). Even for a large model like `gemma-7b`, only about 40% of instances have matching answers across all templates. For `opt-6.7b`, this drops to 30%, and for `gpt2-xl`, only 16%.

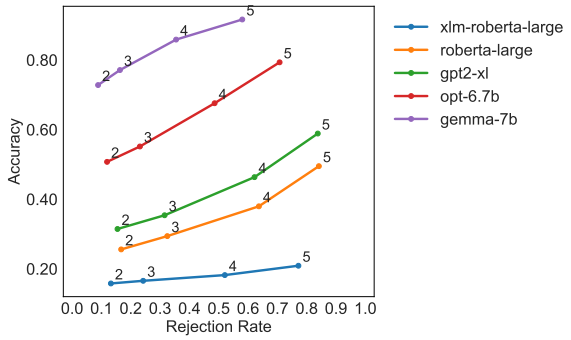

Figure 10: Accuracy-rejection curves for different voting schemes, where (early) answer rejections occur only through vote failures.

D Overview of Selected Domains per Model Pair

Ploner et al. (2024) annotated each of the 60 relations with up to three domains. BEAR covers the domains of *Biographical, Arts, Geographic, Movies, Economic, Historical, Sports, Scientific*, and *Political*. To further reduce the impact of differing training data on calibration in Experiment 2 (section 6), we identified the domain in which each pair is most similar in terms of accuracy. Table 5 reports the selected domain, the corresponding model accuracies, as well as the number of instances.

E Impact of Forced Epistemicity on Accuracy

While forced epistemicity can improve reliability for some models, we find that the injections significantly reduce accuracy, as shown in Figure 11.

Model	Domain	# Instances	Accuracy
`opt-125m` `roberta-base`	Arts	1,623	10.6 10.4
`gpt2-medium` `bert-base-cased`	Economic	1,110	16.0 14.8
`gpt2-large` `roberta-large`	Biographical	2,847	19.6 19.0
`opt-350m` `bert-large-cased`	Movies	1,350	10.5 10.1
`gpt2` `xlm-roberta-large`	Arts	1,623	10.0 10.1

Table 5: Selected domains per matched CLM-MLM pair. The selected domain has the smallest difference in accuracy (%) between the CLM and MLM model.

For instance, `gemma-7b` shows a drop of 15 percentage points in accuracy for the injection of "I'm 25% confident....", the largest decrease observed overall. The only LM where we observe a slight improvement compared to the unaltered template is `roberta-base`, where accuracy increased by 0.2 percentage points for the injection of "certainly". We also observe that larger models are more sensitive to these injections, particularly to numerical confidence expressions.

	possibly	certainly	0%	25%	50%	75%	100%
bert-base-cased	-0.011	-0.009	-0.024	-0.021	-0.017	-0.016	-0.011
bert-large-cased	-0.030	-0.005	-0.039	-0.040	-0.034	-0.035	-0.027
gemma-2b	-0.030	-0.024	-0.107	-0.099	-0.098	-0.079	-0.092
gemma-7b	-0.054	-0.032	-0.130	-0.150	-0.142	-0.110	-0.129
gpt2	-0.013	-0.012	-0.015	-0.012	-0.010	-0.011	-0.008
gpt2-medium	-0.020	-0.020	-0.027	-0.024	-0.023	-0.021	-0.021
gpt2-large	-0.017	-0.020	-0.031	-0.031	-0.030	-0.031	-0.032
gpt2-xl	-0.025	-0.025	-0.042	-0.043	-0.040	-0.037	-0.034
opt-125m	-0.013	-0.013	-0.016	-0.013	-0.013	-0.011	-0.013
opt-350m	-0.019	-0.016	-0.037	-0.032	-0.032	-0.031	-0.030
opt-1.3b	-0.029	-0.028	-0.058	-0.063	-0.059	-0.054	-0.054
opt-6.7b	-0.039	-0.032	-0.098	-0.109	-0.095	-0.085	-0.087
roberta-base	-0.014	0.002	-0.018	-0.023	-0.015	-0.011	-0.008
roberta-large	-0.031	-0.001	-0.019	-0.042	-0.027	-0.024	-0.025
xlm-roberta-base	-0.007	-0.014	-0.024	-0.024	-0.024	-0.024	-0.019
xlm-roberta-large	-0.023	-0.009	-0.006	-0.015	-0.008	-0.011	-0.004
Average	-0.023	-0.016	-0.043	-0.046	-0.042	-0.037	-0.037

Figure 11: Difference in Accuracy between the non-injected and injected templates. Positive values show an improvement over the non-injected version.

F Wiki-FACTOR Benchmark

While structural consistency and semantic grounding are specific to template-based probes such as BEAR, intrinsic confidence estimates apply to any

multiple-choice setting. For a further benchmark, we implemented the estimates C_{Base} and C_{Margin} on the Wiki-FACTOR dataset (Muhlgay et al., 2024). FACTOR defines a multiple-choice task in which each question consists of a multi-sentence context, a factual continuation and three non-factual distractors. In the original setup, the model's answer is chosen as the completion with the highest mean log-probability. As shown on BEAR, mean reduction often results in very low confidence scores, which are not indicative of a well-calibrated model. We observe the same behavior on FACTOR: log-likelihoods do not span the full confidence interval and often differ only marginally, resulting in random guessing confidence splits. Combined with C_{Margin}, the vast majority of confidences would be close to zero. We therefore aggregate token log-likelihoods by summation as we did it for BEAR.

Higher calibration errors on FACTOR due to the low number of answer options. In Table 6 we present the *ACEs* on FACTOR for C_{Base} and C_{Margin} for the CLMs. The first thing that stands out is that the calibration errors are significantly higher than the ones we obtained on BEAR. On BEAR we found an average *ACE* of 0.213 for C_{Base} and 0.107 for C_{Margin} for the CLMs. In contrast, on FACTOR the *ACEs* are 0.475 and 0.446, respectively. We primarily attribute this to the much lower number of answer options. To show this, Figure 12 plots *ACE* against accuracy for FACTOR and for the 1:1 relations of BEAR with five sampled answer options, i.e. one more than on FACTOR. We find that if the number of answer options is similar, we recover an almost identical relationship between accuracy and *ACE* on both datasets. Hence, for identical accuracies, we find highly similar *ACE* values on BEAR and FACTOR. For example, `gpt2-large` achieves an accuracy of 48.97% on BEAR with an *ACE* of 0.33, while `opt-6.7b` has a comparable accuracy on FACTOR of 49.33% and a nearly identical *ACE* of 0.323. Another example is `gpt2-xl` with an accuracy of 37.81% on BEAR and `gpt2` with an accuracy of 37.5% on FACTOR. For these two models we find an *ACE* of 0.427 and 0.448, respectively.

C_{Margin} also improves on FACTOR. We further note that C_{Margin} also consistently improves compared to C_{Base} on FACTOR, but with much smaller magnitude. On BEAR we found an average improvement in *ACE* among the CLMs of 0.106, while on FACTOR, the average improvement of C_{Margin} is only 0.03. However, we find again a more similar performance if we restrict BEAR to only five answer options per instance, as we did it before. Then average improvement of C_{Margin} is 0.07.

		ACE (↓)		Brier Score (↓)	
Model	*Accuracy*	C_{Base}	C_{Margin}	C_{Base}	C_{Margin}
`gemma-2b`	40.21	0.433	**0.387**	0.457	**0.432**
`gemma-7b`	16.50	0.740	**0.690**	0.753	**0.720**
`gpt2`	26.95	0.528	**0.482**	0.545	**0.513**
`gpt2-medium`	31.93	0.481	**0.438**	0.504	**0.477**
`gpt2-large`	35.74	0.448	**0.433**	0.484	**0.468**
`gpt2-xl`	37.81	0.427	**0.388**	0.459	**0.439**
`opt-125m`	29.53	0.505	**0.481**	0.535	**0.515**
`opt-350m`	33.03	0.475	**0.452**	0.506	**0.485**
`opt-1.3b`	41.88	0.395	**0.385**	0.436	**0.427**
`opt-6.7b`	49.33	0.323	**0.321**	0.377	**0.376**

Table 6: *ACE* and *Brier Score* for C_{Base} and C_{Margin}, as well as accuracy on the Wiki-FACTOR (Muhlgay et al., 2024) dataset. **Bold values** indicate the best-performing estimate per model. **ACE:** • [0,0.05), • [0.05,0.10), • [0.10,0.20), • $\geq$ 0.20; **Brier Score:** • [0–0.10), • [0.10–0.20), • [0.20–0.25), • $\geq$ 0.25.

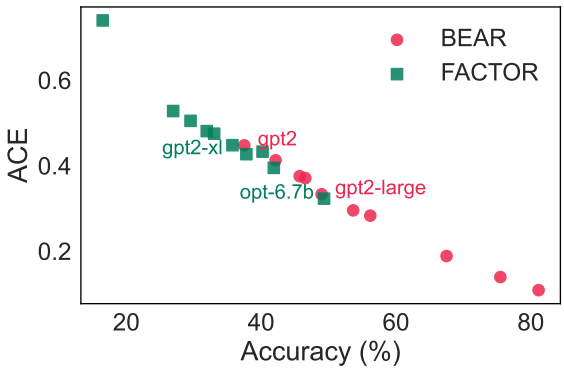

Figure 12: *Accuracy* against *ACE* of C_{Base} for the 1:1 relations of BEAR and the Wiki-FACTOR dataset. For BEAR, we sampled the number of answer options to five per instance. On FACTOR, each question consists of four answer options.

G Extending the BEAR Templates

To better justify the structural consistency estimates, we extended the number of templates from three to five per relation. To generate the additional templates, we utilized the free version of ChatGPT. The following prompt has been used:

As a research assistant, your task is to create an evaluation dataset to assess the relational knowledge of language models. You are provided with a specific relation label, and for each relation label, three semantically equivalent templates are already given, where '[X]' is a placeholder for the subject and '[Y]' for the object (answer). Your objective is to craft two additional semantically similar cloze sentence templates that capture the semantic meaning of the existing templates. Ensure that these new sentence templates are straightforward and devoid of superfluous elements.

For instance, given the following existing templates: ["[X] was authored by [Y].", "[X] is a written work by [Y].", "The author of [X] is [Y]."], an additional template might be: "[Y] wrote [X]."

Present your response in the JSON file format as you received it. Do not provide only the added templates, but include all five templates for each relation. You will be provided with the relation labels and the corresponding templates one by one in the following prompts.