Thermometer: Towards Universal Calibration for Large Language Models

Maohao Shen [1] **Subhro Das** [2] **Kristjan Greenewald** [2] **Prasanna Sattigeri** [2] **Gregory Wornell** [1] **Soumya Ghosh** [2]

Abstract

We consider the issue of calibration in large language models (LLM). Recent studies have found that common interventions such as instruction tuning often result in poorly calibrated LLMs. Although calibration is well-explored in traditional applications, calibrating LLMs is uniquely challenging. These challenges stem as much from the severe computational requirements of LLMs as from their versatility, which allows them to be applied to diverse tasks. Addressing these challenges, we propose THERMOMETER, a calibration approach tailored to LLMs. THERMOMETER learns an auxiliary model, given data from multiple tasks, for calibrating a LLM. It is computationally efficient, preserves the accuracy of the LLM, and produces better-calibrated responses for new tasks. Extensive empirical evaluations across various benchmarks demonstrate the effectiveness of the proposed method[1].

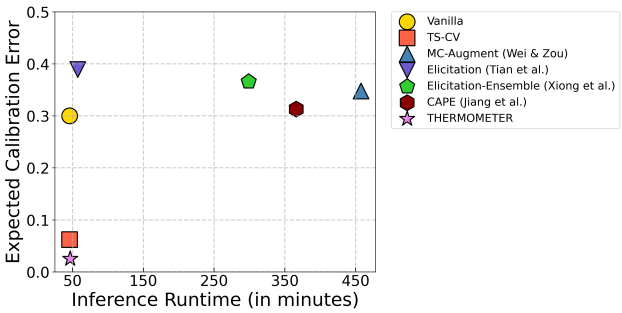

Figure 1. **Calibration performance against inference runtime.** Different methods for calibrating LLaMA-2-Chat 7B compared on the PROFESSIONAL LAW task of MMLU on a A100, 40 GB GPU. The task contains 1533 questions. Our method, THERMOMETER is significantly faster than methods that require multiple forward passes (Wei & Zou, 2019; Xiong et al., 2023; Jiang et al., 2023a) at inference time and achieves lower calibration error compared to methods with comparable runtime (Tian et al., 2023). Vanilla refers to the no-calibration baseline, and TS-CV is a temperature scaling variant (Section 5). Similar trends hold for other benchmarks.

1. Introduction

Well-calibrated forecasts are a desirable property of any probabilistic forecaster. They ensure that probabilities produced by the forecaster can be interpreted as accurate confidence estimates of the forecasts. Informally, this implies that the forecaster is more often wrong on predictions made with low probabilities than those made with high probabilities. Such forecasts are useful for both enabling trust in the forecaster's predictions and incorporating the forecasts as part of a larger autonomous or semi-autonomous system.

Large language models (LLMs) (Brown et al., 2020; Raffel et al., 2020; Touvron et al., 2023) define probability distributions over sequences of tokens and produce probabilistic forecasts over future tokens in a sequence[2]. Their ability to synthesize knowledge from large volumes of data, represented as sequences of tokens, has led to remarkable performance on diverse tasks, including question answering (Hendrycks et al., 2020), commonsense reasoning (Zhong et al., 2019), and machine translation (Zhu et al., 2020) among others. However, much like other probabilistic forecasters, before deploying LLMs in critical applications, it is important that they are well-calibrated in addition to being accurate. Unfortunately, a growing body of evidence suggests that while pre-trained LLMs are often well-calibrated, alignment interventions such as instruction tuning which make the pre-trained LLMs more usable, also harm calibration (OpenAI, 2023; Zhu et al., 2023).

While calibration has long been studied (Brier, 1950; Dawid, 1982; Gneiting et al., 2007) and different approaches for improving calibration properties of probabilistic forecasters exist (Platt et al., 1999; Lakshminarayanan et al., 2017; Gal & Ghahramani, 2016; Guo et al., 2017), LLMs pose unique challenges. Training a LLM is expensive, and even

[1]Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, USA [2]MIT-IBM Watson AI Lab, IBM Research.
Correspondence to: Maohao Shen <maohao@mit.edu>, Soumya Ghosh <ghoshso@us.ibm.com>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

[1]The code is available at https://github.com/maohaos2/Thermometer.

[2]and corresponding semantic entities represented by the tokens.

inference typically incurs non-negligible expenses. This makes any calibration approach that requires multiple training runs prohibitively expensive. Even approaches that only require multiple inferences at test time can be unreasonably expensive for certain applications. Moreover, owing to their versatility, instruction-tuned LLMs are often applied, without further adaptation, to a diverse array of tasks. It is essential that methods for calibrating them do not affect the accuracy of the uncalibrated LLMs and that the calibration methods themselves can adapt to new tasks. Finally, measuring and alleviating calibration is challenging in cases where the LLM is required to generate free-form text. The equivalence class defined by the LLM-generated sequences that map to the same semantic content is large, making it challenging to assign meaningful confidence to the generation or even robustly assess the quality of the generation.

We present THERMOMETER for calibrating LLMs while alleviating the above challenges. THERMOMETER learns, from multiple tasks, a parameterized mapping to map the outputs of the LLM to better-calibrated probabilities. Our approach is (i) computationally efficient: we do not require multiple training runs, and at inference time, we are only $\sim 0.5\%$ slower than the uncalibrated LLM, (ii) accuracy preserving: we build on temperature scaling (Guo et al., 2017) which provably guarantees that greedy-decoded predictions do not change after our calibration procedure, and (iii) takes a step towards being universal: once trained we do not require retraining when exposed to a similar but new task. Moreover, similarly to recent work (Kadavath et al., 2022; Lin et al., 2022), we circumvent the challenges posed by free-form text generation by mapping the free-form text generation task to a next-token prediction task. We empirically evaluate THERMOMETER on diverse benchmarks and models and find it consistently produce better-calibrated uncertainties than competing methods at a fraction of the computational cost. Moreover, we find THERMOMETER transfers across datasets and model scales. THERMOMETER trained for calibrating a smaller model (e.g., LLaMA-2-Chat 7B) also improves calibration of larger models (e.g., LLaMA-2-Chat 70B) from the same family of models.

2. Related Work

A variety of methods exist that aim to produce better-calibrated uncertainties. Post-hoc calibration methods learn to map the outputs of a pre-trained model to well-calibrated uncertainties. These include histogram binning (Zadrozny & Elkan, 2001; Naeini et al., 2015), isotonic regression (Zadrozny & Elkan, 2002), and approaches that assume parametric maps, including matrix, vector, and temperature scaling (Platt et al., 1999; Guo et al., 2017) as well as more ambitious variants (Kull et al., 2019) that aim to calibrate, in multi-class prediction problems, all class probabilities rather than just the probability of the most likely class (*confidence calibration*). The next-token prediction task is also a multi-class prediction problem, albeit one with a large number of classes. Here, the number of classes equals the number of tokens in the LLM's vocabulary. This high dimensionality motivates us to focus on the more modest goal of confidence calibration.

We learn an auxiliary model that, given an unlabeled dataset, predicts a dataset-specific temperature, allowing us to calibrate an LLM's uncertainties on a previously unseen task. Others (Joy et al., 2023; Yu et al., 2022) have also considered parameterized temperature maps but with a focus on more traditional vision models. Joy et al. (2023) predict per-data instance temperatures, do not learn from multiple datasets, and find it necessary to learn a variational auto-encoder for representation learning jointly. In contrast, we leverage data from multiple tasks to generalize to new tasks, find no need for additional representation learning, and empirically find dataset-specific temperatures to better calibrate LLMs than data-instance-specific temperatures. Yu et al. (2022) are motivated by making calibration robust to distribution shifts. Their approach involves a two-step procedure: independently learning dataset-specific temperatures for each dataset and then fitting a linear regression to the learned temperatures. In contrast, we jointly learn a non-linear auxiliary model across multiple datasets by simply maximizing a lower bound to the likelihood. Moreover, empirically, we find that THERMOMETER outperforms linear counterparts.

Other *ab initio* approaches involve training with label-smoothing (Szegedy et al., 2016), mix-up augmentations (Zhang et al., 2017), confidence penalties (Pereyra et al., 2017), focal loss (Mukhoti et al., 2020), or approximate Bayesian procedures (Izmailov et al., 2021). The substantial changes to the training process required by these approaches make them difficult to use with LLMs. Yet other approaches include ensembling over multiple models arrived at by retraining from different random initializations, for example, deep ensembles (Lakshminarayanan et al., 2017), or from perturbations of a single model, for example, Monte-Carlo dropout (Gal & Ghahramani, 2016). Training multiple LLM variants is prohibitively expensive, and while perturbations around a single model are possible, we find them to be not competitive with THERMOMETER.

Jiang et al. (2021); Xiao et al. (2022); Chen et al. (2022) empirically evaluate calibration of LLMs, find evidence of miscalibration, and evaluate existing calibration interventions and combinations thereof with varying degrees of success. Park & Caragea (2022) use mixup while Desai & Durrett (2020) find temperature scaling and label smoothing effective for calibrating smaller encoder-only models. Others (Lin et al., 2022) employ supervised fine-tuning to produce verbalized uncertainties to be better calibrated

on certain tasks. Such fine-tuning, however, is compute-intensive. An alternate body of work (Zhang et al., 2021; Kadavath et al., 2022; Mielke et al., 2022) approach calibration indirectly by learning an auxiliary model for predicting whether a generation is incorrect. There is also a body of work (Abbas et al., 2024; Han et al., 2023; Jiang et al., 2023b; Zhao et al., 2021; Zhou et al., 2024) that use the term calibration to mean de-biasing the predictions of a language model to biases introduced by the choice and ordering of in-context examples. This is distinct from our notion of statistical calibration. Yet others (Tian et al., 2023; Xiong et al., 2023), similar to our approach, consider RLHF-tuned LLMs and find that they can express better-calibrated uncertainties with carefully crafted prompts. Our work is orthogonal, extending temperature scaling to tasks without labeled data, and can be combined with their approach.

3. Background

Setup and Notation We consider question answering tasks, both involving multiple-choice and free form answers. We pose this as a next token prediction problem by concatenating the question, any available context, and the multiple-choice options into a single prompt. In Section 4.3, we describe our treatment of free form answers within this setup. See Appendix C.1 for example prompts. Given the prompt, the next token predicted by the LLM is considered the answer to the question. We denote the nth prompt in a dataset by $\boldsymbol{x}_n$ and the corresponding completion by y_n. We assume that both the prompt and the completion have been tokenized, and the prompt $\boldsymbol{x}_n = x_{n,t_n}, \ldots, x_{n,2}, x_{n,1}$ is a sequence of t_n tokens. The tokens $x_{n,t}$ and y_n take one of V values. We use $\mathcal{D} = \{(\boldsymbol{x}_n, y_n)\}_{n=1}^{N}$ to denote a training set of prompt, completion pairs. We model

$$p(y_n \mid \boldsymbol{x}_n; \mathcal{M}) = \frac{\exp(\boldsymbol{w}_{y_n}^T \phi(\boldsymbol{x}_n; \mathcal{W}))}{\sum_{v'=1}^{V} \exp(\boldsymbol{w}_{v'}^T \phi(\boldsymbol{x}_n; \mathcal{W}))},$$

using a large language model, $\mathcal{M}$, parameterized by weights, $\{\mathcal{W}, \boldsymbol{w}_1, \ldots, \boldsymbol{w}_V\}$. We use $p(y_n \mid \boldsymbol{x}_n; \mathcal{M})$ as a notationally convenient stand in for $p(Y = y_n | X = \boldsymbol{x}_n; \mathcal{M})$, where Y and X are random variables corresponding to the completion and the prompt. We also note that $p(Y \mid X = \boldsymbol{x}_n; \mathcal{M}) \in \Delta^V$ where Δ^V is a V-dimensional probability simplex. We view $\phi(\boldsymbol{x}_n, \mathcal{W})$ as a feature extractor that transforms the input prompt $\boldsymbol{x}_n$ into a D-dimensional representation. For notational brevity, we will suppress the dependence on $\mathcal{W}$ and use $\phi(\boldsymbol{x}_n, \mathcal{W})$ and $\phi(\boldsymbol{x}_n)$ interchangeably.

Confidence calibration We aim to confidence calibrate $\mathcal{M}$, such that among the completions whose top probability is β the accuracy is also β,

$$\Pr(Y = \hat{Y} | \hat{P} = \beta) = \beta, \text{for all } \beta \in [0, 1],$$

where $\hat{Y} = \operatorname{argmax} p(Y|X; \mathcal{M})$ and $\hat{P} = \max p(Y|X; \mathcal{M})$.

One common notion of miscalibration is the expected calibration error (ECE) (Naeini et al., 2015), $\mathbb{E}\left[\left|\Pr(Y = \hat{Y} \mid \hat{P} = \beta) - \beta\right|\right]$. An empirical estimate of the ECE can be computed by partitioning N samples into M bins based on the confidences $\max p(Y \mid \boldsymbol{x}_n; \mathcal{M})$ predicted by the model $\mathcal{M}$. Let B_m denote the set of samples allocated to the M-th bin. Then the empirical ECE estimate is the weighted average over the difference in accuracy, $\text{acc}(B_m) = 1/|B_m| \sum_{n \in B_m} \mathbf{1}\left[y_n = \operatorname{argmax} p(Y|\boldsymbol{x}_n; \mathcal{M})\right]$, and the average confidence within each bin, $\text{conf}(B_m) = 1/|B_m| \sum_{n \in B_m} \max p(Y|\boldsymbol{x}_n; \mathcal{M})$,

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

Temperature Scaling A large ECE score indicates the need for calibration as the model's prediction is poorly calibrated on the given dataset. Temperature scaling (Guo et al., 2017; Platt et al., 1999) is a widely used post-hoc calibration technique that introduces a single positive, scalar parameter $\tau \in \mathbb{R}_+$, the temperature, and defines,

$$p(y_n \mid \boldsymbol{x}_n; \tau, \mathcal{M}) = \frac{\exp(\boldsymbol{w}_{y_n}^T \phi(\boldsymbol{x}_n; \mathcal{W})/\tau)}{\sum_{v'=1}^{V} \exp(\boldsymbol{w}_{v'}^T \phi(\boldsymbol{x}_n; \mathcal{W})/\tau)}.$$

The parameter τ is typically learned by minimizing the negative log likelihood (NLL) on a *labeled* held-out dataset, $\hat{\tau} = \arg\min_{\tau \in \mathbb{R}_+} -\sum_{n=1}^{N^*} \log p(y_n \mid \boldsymbol{x}_n; \tau, \mathcal{M})$, freezing all other model parameters. Since temperature scaling only scales probabilities, it does not change the class with the maximum probability and preserves accuracy. Recent work (Chen et al., 2022; Xiao et al., 2022; Tian et al., 2023) has found temperature scaling effective at calibrating LLMs. Besides, it does not require additional tuning of the LLM or multiple forward passes at inference time. However, the requirement of a held-out labeled dataset can be severely limiting for LLMs. LLMs are often used on new tasks, where no labeled data is available!

To alleviate this challenge, we propose to bypass dataset specific optimization and instead learn to predict the optimal temperature for a previously unseen and *unlabeled* dataset.

4. THERMOMETER

Consider a multi-task setting. Given K tasks with labeled datasets $\{\mathcal{D}_k\}_{k=1}^{K} = \{\{(\boldsymbol{x}_n^k, y_n^k)\}_{n=1}^{N_k}\}_{k=1}^{K}$, our goal is to learn to infer task-specific temperature parameters, in order to accurately infer the temperature for an unseen task given only N_* prompts, $\mathcal{D}_* = \{\boldsymbol{x}_n^*\}_{n=1}^{N_*}$ but not the corresponding completions. To this end, we propose a probabilistic model with task-specific latent variables, τ_k, for modeling

the conditional distribution of completions given a prompt,

$$p(\mathcal{D}_1, \ldots \mathcal{D}_K \mid \nu_0; \mathcal{M}) = \prod_{k=1}^{K} \int p(\tau_k \mid \nu_0) \prod_{n=1}^{N_k} p(y_n^k \mid \boldsymbol{x}_n^k, \tau_k; \mathcal{M}) d\tau_k, \tag{1}$$

where $p(y_n^k \mid \boldsymbol{x}_n^k, \tau_k; \mathcal{M}) = \frac{\exp(\boldsymbol{w}_{y_n^k}^T \phi(\boldsymbol{x}_n^k; \mathcal{W})/\tau_k)}{\sum_{v'=1}^{V} \exp(\boldsymbol{w}_{v'}^T \phi(\boldsymbol{x}_n^k; \mathcal{W})/\tau_k)}$, $p(\tau_k \mid \nu_0)$ is a prior on τ_k for regularizing it away from zero and large positive values, and ν_0 is the prior's hyper-parameters.

Our key idea is to use a recognition network (Dayan et al., 1995), to infer the per-task latent temperatures. We share the recognition network across all tasks, allowing us to amortize computation across tasks and sidestep the need for task specific optimization for inferring τ_k. Crucially, we design our recognition network to condition only on $\phi(\boldsymbol{x}_n^k)$ and not on the completions y_n^k, allowing us to infer the temperature for unlabeled data $\mathcal{D}^*$ at test time. We dub our recognition network, THERMOMETER, which, much like a real THERMOMETER, is designed to estimate temperature, albeit in the context of a dataset.

4.1. A Variational Lower-bound

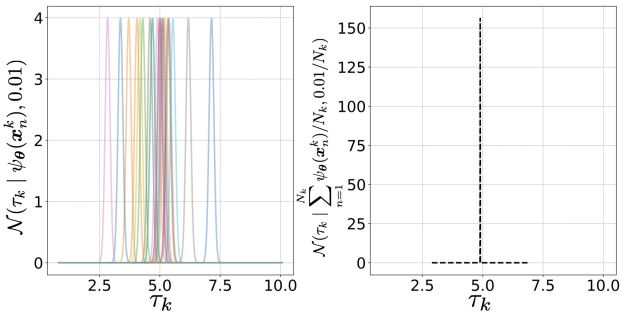

Figure 2. **Evidence accumulation.** We illustrate evidence accumulation for the PROFESSIONAL LAW dataset from MMLU. It contains $N_k = 1533$ instances. The left panel, shows twenty of the possible 1533 Gaussians, $\mathcal{N}(\tau_k \mid \psi_{\boldsymbol{\theta}}(\phi(\boldsymbol{x}_n^k)), \epsilon = 0.01)$. The right panel, plots the Gaussian proportional to $\prod_{n=1}^{1533} \mathcal{N}(\tau_k \mid \psi_{\boldsymbol{\theta}}(\phi(\boldsymbol{x}_n^k)), \epsilon = 0.01)$. The resulting Gaussian is nearly a point mass at $\frac{1}{1533} \sum_{n=1}^{1533} \psi_{\boldsymbol{\theta}}(\phi(\boldsymbol{x}_n^k))$.

We work within the framework of variational inference, which provides us with a coherent objective to optimize for learning the recognition network. We begin by forming a variational lower bound of the logarithm of the marginal likelihood, $p(\mathcal{D}_1, \ldots \mathcal{D}_K \mid \nu_0; \mathcal{M})$,

$$\mathcal{L}(\boldsymbol{\theta}) = \sum_{k=1}^{K} \mathbb{E}_{q(\tau_k; \boldsymbol{\theta})} [\log p(\mathcal{D}_k \mid \tau_k; \mathcal{M})] - \mathrm{KL}\left(q(\tau_k; \boldsymbol{\theta}) \,||\, p(\tau_k \mid \nu_0)\right), \tag{2}$$

where $q(\tau_k; \boldsymbol{\theta})$ is a variational approximation to the posterior distribution $p(\tau_k \mid \mathcal{D}_1, \ldots, \mathcal{D}_k)$. We choose a product of Gaussian PDF for the variational approximation. Each corresponding Gaussian variable has a fixed variance ϵ, and a mean parameterized by a recognition network (for example, a MLP), $\psi_{\boldsymbol{\theta}} : \mathbb{R}^D \to \mathbb{R}_+$, where $\boldsymbol{\theta} \in \mathbb{R}^P$ represents the parameters of the recognition network that are shared across $\mathcal{D}_1, \ldots, \mathcal{D}_k$. The resulting variational approximation is,

$$q(\tau_k; \boldsymbol{\theta}) \propto \prod_{n=1}^{N_k} \mathcal{N}(\tau_k \mid \psi_{\boldsymbol{\theta}}(\phi(\boldsymbol{x}_n^k)), \epsilon), \quad \mathcal{N}\left(\tau_k \mid \frac{1}{N_k} \sum_{n=1}^{N_k} \psi_{\boldsymbol{\theta}}(\phi(\boldsymbol{x}_n^k)), \frac{\epsilon}{N_k}\right), \tag{3}$$

where the last equality follows from standard properties of Gaussian distributions. We provide a proof in Appendix B.2. This particular choice of the variational parameterization affords us several benefits. First, it allows us to *accumulate evidence* (Bouchacourt et al., 2018) across data instances in $\mathcal{D}_k$, with the uncertainty in the inferred τ_k decreasing linearly with N_k. Second, by sharing $\boldsymbol{\theta}$ across datasets we share statistical strength across the training datasets $\mathcal{D}_1, \ldots, \mathcal{D}_K$ which allows us to use $\psi_{\boldsymbol{\theta}}$ to predict τ_* for an unseen dataset $\mathcal{D}_*$. Finally, when ϵ is small, our parameterization leads to a particularly simple algorithm.

Training Objective Plugging in the variational approximation Equation (3) in Equation (2), we have,

$$\mathcal{L}(\boldsymbol{\theta}) = \sum_{k=1}^{K} \mathbb{E}_{q(\tau_k; \boldsymbol{\theta})} [\log p(\mathcal{D}_k | \tau_k; \mathcal{M}) + \log p(\tau_k | \nu_0)] + \mathrm{C}, \tag{4}$$

where C[3] is a constant independent of $\boldsymbol{\theta}$. By further observing that when ϵ is small, for example, 10^{-2}, for even moderate N_k the variational approximation $q(\tau_k \mid \boldsymbol{\theta})$ is reasonably approximated by a point mass distribution $\delta(\tau_k - \frac{1}{N_k} \sum_{n=1}^{N_k} \psi_{\boldsymbol{\theta}}(\phi(\boldsymbol{x}_n^k)))$ (Figure 2). This leads to the objective for training THERMOMETER,

$$\tilde{\mathcal{L}}(\boldsymbol{\theta}) = \sum_{k=1}^{K} \log p\left(\mathcal{D}_k \,\middle|\, \frac{1}{N_k} \sum_{n=1}^{N_k} \psi_{\boldsymbol{\theta}}(\phi(\boldsymbol{x}_n^k)); \mathcal{M}\right) - \lambda_{\text{reg}} \log p\left(\frac{1}{N_k} \sum_{n=1}^{N_k} \psi_{\boldsymbol{\theta}}(\phi(\boldsymbol{x}_n^k)) \,\middle|\, \nu_0\right), \tag{5}$$

where we have dropped the constant, plugged in the point mass approximation in Equation (4), and introduced a hyperparameter λ_{reg} for balancing the prior and the likelihood terms. We place a Gamma prior, with a mode at one, on the temperatures τ_k, $p(\tau_k \mid \nu_0) = \text{Gamma}(\tau_k \mid \alpha_0 = 1.25, \beta_0 = 4)$[4]. This prior enforces non-negative tem-

[3] $\mathrm{C} = K/2 + (K/2)\ln(2\pi\epsilon)$

[4] We use the shape scale parameterization, with shape = 1.25 and scale = 4.

peratures, and although the prior's mode at one encodes our belief that $\mathcal{M}$ is already reasonably calibrated, the high prior variance emphasizes that this is only a weakly held belief. Finally, we train THERMOMETER by minimizing, $\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}\in\mathbb{R}^P} -\tilde{\mathcal{L}}(\boldsymbol{\theta})$, or equivalently maximizing an approximate[5] lower bound to the log-marginal likelihood. We use stochastic optimization with mini-batched gradients to optimize $\tilde{\mathcal{L}}(\boldsymbol{\theta})$. Algorithm 1 summarizes our learning procedure. The temperature estimation step in the algorithm involves averaging over the minibatch which introduces a small bias in the minibatch gradients. However, this bias decreases linearly with increasing batch size (Lemma 4.1), and disappears in the full-batch limit. Empirically, we find performance to be largely invariant to batch size (Table 8).

Algorithm 1 THERMOMETER Training

input Training datasets of K tasks $\{\mathcal{D}_k\}_{k=1}^K$; pre-trained LLM $\mathcal{M}$ with feature extractor ϕ; prior hyper-parameters ν_0; batch size N_b; learning rate γ; number of iterations M; checkpoint parameters $\{m', \text{burnin}\}$; initialization $\boldsymbol{\theta}_{\text{init}}$

▷ Split $\{\mathcal{D}_k\}_{k=1}^K$ into a development and validation set, $\mathrm{D}^{\text{dev}} = \{\mathcal{D}_k^{dev}\}_{k=1}^K, \mathrm{D}^{\text{val}} = \{\mathcal{D}_k^{val}\}_{k=1}^K$
▷ Initialize THERMOMETER, $\boldsymbol{\theta}_0 \leftarrow \boldsymbol{\theta}_{\text{init}}$; Denote the number of mini-batches by $B = \lceil \frac{N}{N_b} \rceil$, where $N = |\mathrm{D}^{\text{dev}}|$.
for $m = 1, 2, \ldots, M$ **do**
  ▷ Sample, uniformly at random, a task k and its dataset $\mathcal{D}_k^{dev}$.
  ▷ Sample a batch of N_b samples from $\mathcal{D}_k^{dev}$, i.e., $\{(\boldsymbol{x}_n^k, y_n^k)\}_{n=1}^{N_b}$.
  ▷ Estimate temperature $\hat{\tau}_k = \frac{1}{N_b}\sum_{n=1}^{N_b} \psi_{\boldsymbol{\theta}}(\phi(\boldsymbol{x}_n^k))$.
  ▷ Evaluate the loss w.r.t batch of samples, i.e.,

$$-\tilde{\mathcal{L}}(\boldsymbol{\theta})_{N_b} = -\sum_{n=1}^{N_b} \log p(y_n^k \mid \boldsymbol{x}_n^k, \hat{\tau}_k; \mathcal{M}) - \frac{\lambda_{\text{reg}}}{B} \cdot \log p(\hat{\tau}_k \mid \nu_0)$$

  ▷ Update the THERMOMETER parameter using AdamW, i.e., $\boldsymbol{\theta}_m \leftarrow \text{AdamW}(\boldsymbol{\theta}_{m-1}, \gamma, -\nabla_{\boldsymbol{\theta}}\tilde{\mathcal{L}}(\boldsymbol{\theta})_b)$.
  if $m > \text{burnin}$ **then**
    ▷ Checkpoint $\boldsymbol{\theta}_m$ every m' iterations.
  end
end
▷ Evaluate $\tilde{\mathcal{L}}(\boldsymbol{\theta})$ on D^{val} for each check-pointed $\boldsymbol{\theta}$ and set $\boldsymbol{\theta}^*$ to the check-pointed $\boldsymbol{\theta}$ with lowest $-\tilde{\mathcal{L}}(\boldsymbol{\theta})$.

output Optimized THERMOMETER, $\boldsymbol{\theta}^*$.

4.2. Test Time Procedure

Given a new test task, we assume we have a collection of N_* inputs $\{\boldsymbol{x}_n^*\}_{i=1}^{N_*}$ with which we compute the temperature to be used for the task,

$$\tau_* = \frac{1}{N_*}\sum_{n=1}^{N_*} \psi_{\boldsymbol{\theta}^*}(\phi(\boldsymbol{x}_n^*)). \tag{6}$$

[5]the approximation is increasingly accurate with larger N_k.

Impact of test data size At test time not all the data may be immediately available when computing the temperature τ_* (e.g. deploying on real-world tasks) and N_* in (6). As the empirical average used to compute τ_* is a proxy for $\mathbb{E}_{\boldsymbol{x}\sim\mathcal{P}_*}[\psi_{\boldsymbol{\theta}}(\phi(\boldsymbol{x}))]$, where $\mathcal{P}_*$ is the unknown data generating process for the test task, it is important that N_* not be too small or this approximation will not be accurate and performance may suffer. We can control this error, however. Under an assumption that for fixed parameters, $\boldsymbol{\theta}$, $0 \le \psi_{\boldsymbol{\theta}}(\phi(\boldsymbol{x})) \le C_{\boldsymbol{\theta}}$ for some $C_{\boldsymbol{\theta}}$ (e.g. because the output of the THERMOMETER model must be bounded if the input is), we can apply a Bernstein inequality to obtain:

Lemma 4.1. *Let $\mathcal{X}$ be the support set of the distribution of $\boldsymbol{x}$, and assume we have N_* i.i.d. samples $\{\boldsymbol{x}_n\}_{n=1}^{N_*}$ from $\mathcal{P}_*$. Assume that for fixed parameters $\boldsymbol{\theta}$, $\sup_{\boldsymbol{x}\in\mathcal{X}} \psi_{\boldsymbol{\theta}}(\phi(\boldsymbol{x})) \le C_{\boldsymbol{\theta}}$ for some $C_{\boldsymbol{\theta}}$, and $\mathrm{Var}[\psi_{\boldsymbol{\theta}}(\phi(\boldsymbol{x}_n))] \le V_\theta$.[6] Further assume that we are interested in measuring calibration error using a metric* CE *(for example,* ECE*) that as a function of temperature,* $\mathrm{CE}(\tau)$ *is L-Lipschitz. Then with probability at least* $1 - \frac{2}{{N_*}^2}$,

$$\mathbb{E}\left|\frac{1}{N_*}\sum_{n=1}^{N_*}\psi_{\boldsymbol{\theta}}(\phi(\boldsymbol{x}_n)) - \mathbb{E}_{\boldsymbol{x}\sim\mathcal{P}_*}\left[\psi_{\boldsymbol{\theta}}(\phi(\boldsymbol{x}))\right]\right| \le \frac{4}{3}C_\theta\frac{\log N_*}{N_*} + (2V_\theta)^{\frac{1}{2}}\sqrt{\frac{\log N_*}{N_*}}, \quad \textit{and,}$$

$$\mathrm{CE}\left(\frac{1}{N_*}\sum_{n=1}^{n}\psi_{\boldsymbol{\theta}}(\phi(\boldsymbol{x}_n))\right) - \mathrm{CE}\left(\mathbb{E}_{\boldsymbol{x}\sim\mathcal{P}_*}\left[\psi_{\boldsymbol{\theta}}(\phi(\boldsymbol{x}))\right]\right) \le L\left(\frac{4}{3}C_\theta\frac{\log N_*}{N_*} + (2V_\theta)^{\frac{1}{2}}\sqrt{\frac{\log N_*}{N_*}}\right).$$

The proof is in Appendix B.1. This result guarantees that these partial sums will converge quickly as N_* increases, and that any resulting loss of CE will be small. We confirm this guarantee experimentally in Appendix A.4. Our bound can be checked by practitioners as a guide to choosing N_*, trading off accuracy and the need for more examples. In practice, both the Lipschitz constant and V_θ can be easily estimated. Specifically, the former is simple since the temperature is a one-dimensional last-layer parameter of an otherwise pretrained model, and the latter can be bounded by using an empirical estimate of the variance.[7]

4.3. Question Answering with Free Form Answers

As we introduced in Section 3, given a sequence of tokens as prompts $\boldsymbol{x}_n$, we aim to calibrate the LLM's prediction on its corresponding completion token y_n. In the context

[6]This can be replaced with $C_\theta^2/4$ if no additional knowledge of the variance is available.

[7]This can be done rigorously by deriving and adding a concentration upper bound for the error of the empirical variance before plugging into the bound in Lemma 4.1 as V_θ.

of a multiple-choice QA problem, y_n typically represents a single token corresponding to one of the multiple-choice options, such as 'A' or 'B'. However, in practice, LLMs are more commonly tasked with generating free-form responses for arbitrary questions without predefined answer options.

We address this issue by converting the free-form QA problem into a multiple-choice format, by posing a 'Yes' or 'No' question to an LLM, inquiring whether its generated response is correct or incorrect. Given a training set $\mathcal{D} = \{(\boldsymbol{x}_n, y_n)\}_{n=1}^{N}$, where $\boldsymbol{x}_n$ represents the question context, and y_n denotes the true answers, we generate a new prompt $\tilde{\boldsymbol{x}}_n$ by concatenating the original question $\boldsymbol{x}_n$ with LLM's generated response $\boldsymbol{z}_n$, i.e., $\tilde{\boldsymbol{x}}_n = [\boldsymbol{x}_n, \boldsymbol{z}_n]$, and construct a new target $\tilde{y}_n$ by comparing generated response $\boldsymbol{z}_n$ and true answer y_n, i.e., $\tilde{y}_n$ is assigned 'Yes' if $\boldsymbol{z}_n$ closely approximates y_n based on a predefined similarity metric (for example, the Rouge-L score). The resulting calibration problem can be tackled in the same manner as multiple-choice QA, i.e., calibrate LLM's prediction of single completion token $\tilde{y}_n$ ('Yes' or 'No') given the prompts $\tilde{\boldsymbol{x}}_n$.

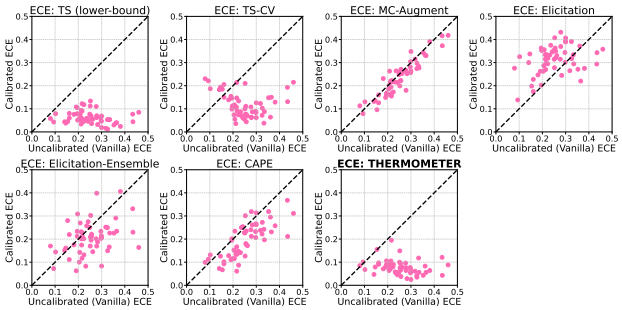

Figure 3. **LLaMA-2-Chat 7B Scatter Plots: ECE Score of 57 MMLU Datasets.** The x-axis and y-axis represent ECE score of uncalibared and calibrated model, respectively. THERMOMETER reduces calibration error on all 57 datasets, often substantially.

5. Experiments

We describe the experimental setup here, and present the main experimental results in Section 5.1, and conduct a comprehensive analysis in Section 5.2. Additional details and results are in Appendix A and Appendix C.

Benchmark Datasets We employ two widely used benchmark datasets for multiple-choice question-and-answer (QA) experiments: MMLU (Hendrycks et al., 2020) and BIG-bench (Srivastava et al., 2022), and adopt MRQA (Fisch et al., 2019) for experiments on QA with free form answers. MMLU contains datasets of exam questions from fifty seven subjects spanning various fields, each question comprises a question context and four possible answers. BIG-bench is a massive benchmark containing more than two hundred datasets covering a wide range of NLP tasks. We extracted multiple-choice QA datasets with over thousand training instances, which resulted in twenty three datasets. The MRQA benchmark comprises of a predefined train and development split, with six training datasets and six development datasets. The style of questions vary greatly both within and across the training and development sets. More details are in Appendix C.2.

Models We aim to enhance the calibration performance of instruction tuned language models. As decoder-only models become more ubiquitous in practice, our main experiments are conducted using decoder-only model LLaMA-2-Chat 7B with seven billion parameters (Touvron et al., 2023). To demonstrate the robust calibration performance of our proposed THERMOMETER with respect to different types of models, we also conduct a set of experiments using encoder-decoder model FLAN-T5-XL with three billion parameters (Chung et al., 2022). THERMOMETER is an auxiliary model with a Multi-Layer Perceptron (MLP)-like structure described in Appendix C.3.

Evaluation Metrics We evaluate the calibration performance using following metrics: (1) Expected Calibration Error (ECE): measures the average error between prediction confidence and accuracy across different confidence intervals, for which we use 10 bins in our evaluation. (2) Maximum Calibration Error (MCE) (Naeini et al., 2015): identifies the largest error between prediction confidence and accuracy among all confidence bins, represents the worst-case calibration scenario. (3) Negative log likelihood (NLL): a proper scoring rule (Gneiting & Raftery, 2007) that measures closeness to the data generating process in the Kullback-Leibler sense (Deshpande et al., 2024). (4) Brier Score (Brier) (Brier, 1950): another proper scoring rule that measures the mean squared error between prediction confidence and one-hot labels. (5) Top Label ECE (TL-ECE) (Gupta & Ramdas, 2021): measures the average error conditioned on both confidence and top-predicted labels. TL-ECE also serves as an upper bound of ECE, and is an alternate measure of calibration error.

Implementation Details For multiple-choice QA tasks, we employ a leave-one-out approach to assess the THERMOMETER's calibration on unseen tasks. We train the model using $K-1$ datasets, and test the trained model on the remaining single testing task, repeating this process for all the K datasets. For the QA tasks with free form answers, we use the established train and dev splits. We train THERMOMETER on MRQA's train split, and evaluate on the six held-out development datasets. In all experiments, we report results averaged over five random trials. See Appendix C.4 for details.

Baselines Our goal of calibrating models on tasks with no labeled data rules out many calibration approaches. Nonetheless, we compare THERMOMETER against several classical

Table 1. **LLaMA-2-Chat 7B Average Calibration Performance on MMLU.** The results are reported as the mean and two standard error of the calibration results over 57 datasets. TS serves as the lower-bound as it has access to the labeled data of testing task.

Methods	ECE	TL-ECE	MCE	NLL	Brier
TS (lower-bound)	0.062±0.008	0.123±0.011	0.298±0.051	1.205±0.043	0.161±0.006
Vanilla	0.260±0.025	0.290±0.023	0.482±0.047	1.760±0.123	0.191±0.012
TS-CV	0.119±0.014	0.167±0.012	0.344±0.037	1.253±0.048	0.166±0.006
MC-Augment	0.242±0.026	0.281±0.021	0.488±0.047	1.716±0.125	0.192±0.012
Elicitation	0.315±0.016	0.340±0.014	0.787 ±0.034	/	/
Elicitation-Ensemble	0.202±0.019	0.255±0.017	0.544±0.055	/	/
CAPE	0.196±0.021	0.247±0.016	0.431±0.054	1.542± 0.097	0.184±0.009
THERMOMETER	**0.078±0.008**	**0.136±0.011**	**0.304±0.037**	**1.220±0.043**	**0.162±0.006**

calibration methods as well as those recently proposed methods: (1) **Vanilla**: calibration performance of LLMs without any calibration techniques applied. (2) Temperature Scaling (**TS**): cheats by using labeled data from the testing task to tune the task-specific temperatures and establishes a lower bound for the error. (3) Temperature Scaling with cross-validation (**TS-CV**): This variant of TS is similar to the setting of our approach. Here, the temperature is tuned using data from $K-1$ tasks, and used to calibrate the held-out task, without any task-specific adaptation. (4) Monte Carlo Dropout (**MC-Dropout**)[8] (Gal & Ghahramani, 2016): performs inference with random dropout multiple times, and averages the output probabilities across these inferences. (5) Monte Carlo Augmentation (**MC-Augment**) (Wei & Zou, 2019): employs random data augmentations of the prompts rather than dropout. (6) **Elicitation** (Tian et al., 2023) directly generates verbalized confidence by providing the LLMs with carefully crafted prompts. (7) **Elicitation-Ensemble** (Xiong et al., 2023) improves the quality of verbalized confidence by aggregation of multiple samples. (8) **CAPE** (Jiang et al., 2023a) performs prompt augmentation using option permutation.

5.1. Main Results

Multiple-choice QA Our main calibration results for multiple-choice QA tasks are presented in two formats. We offer an overview of the overall calibration performance by calculating average results across all datasets. This includes 57 tasks from MMLU and 23 from BIG-bench. The average results and 95% confidence intervals for MMLU are presented in Table 1. We provide a finer granularity view through scatter plots illustrating performance on each task in Figure 3. In this plots, points above the diagonal are instances where the calibration method fails to improve over the uncalibrated model, where THERMOMETER shows zero failure cases. Furthermore, we compare the predicted temperatures of THERMOMETER, which uses no labeled data from the test task, with temperatures learned via temperature scaling that uses all the labeled data from the task in Figure 4. We find THERMOMETER's predictions adapt to different tasks and align with learned temperatures.

[8]We found dropout to not be functional in LLaMA-2-Chat 7B, enabling dropout at inference time did not produced any variability, so we only report MC-Dropout comparisons with FLAN-T5-XL.

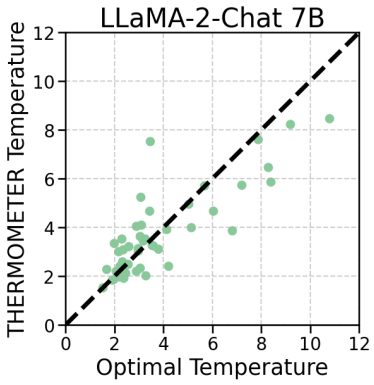

Figure 4. **THERMOMETER against temperature scaling on MMLU.** Comparison of THERMOMETER predictions and temperatures obtained by temperature scaling. Each green dot represents one MMLU task. The x-coordinate is the temperature learned via temperature scaling and the y-coordinate is the THERMOMETER predicted temperature. THERMOMETER accurately predicts the temperature for unseen task.

The diverse nature of the BIG-bench tasks makes learning THERMOMETER more challenging. Nonetheless, the results for BIG-bench, included in Appendix A.1, THERMOMETER still outperforms other baselines with large margin, while stays within noise of TS-CV. Finally, we remark that the similar behavior of THERMOMETER holds for encoder-decoder model FLAN-T5-XL (see Appendix A.2). THERMOMETER shows superior performance over other baseline methods. For the challenging BIG-bench, THERMOMETER with FLAN-T5-XL outperforms all competing approaches, producing lower ECE and TL-ECE scores than the strongest competitor, TS-CV, on 20/23 tasks.

Table 2. **LLaMA-2-Chat 7B Average Calibration Performance on MRQA.** The results are reported as the mean and two standard error of the calibration results over the six MRQA validation datasets.

Methods	ECE	TL-ECE	MCE	NLL	Brier
TS (lower-bound)	0.029±0.005	0.058±0.021	0.115±0.015	0.536±0.042	0.178±0.017
Vanilla	0.127±0.026	0.146±0.023	0.198±0.023	0.656±0.068	0.198±0.022
TS-CV	0.071±0.011	0.099±0.014	0.157±0.015	0.556±0.042	0.183±0.017
MC-Augment	0.335±0.044	0.411±0.034	0.679±0.052	0.997±0.081	0.370±0.026
Elicitation	0.130±0.031	0.207±0.036	0.522±0.169	/	/
Elicitation-Ensemble	0.171±0.047	0.237±0.032	0.552±0.206	/	/
CAPE	0.067±0.016	0.098±0.028	**0.156±0.060**	0.556± 0.087	0.183±0.036
THERMOMETER	**0.065±0.008**	**0.093±0.016**	**0.163±0.027**	**0.551±0.039**	**0.182±0.016**

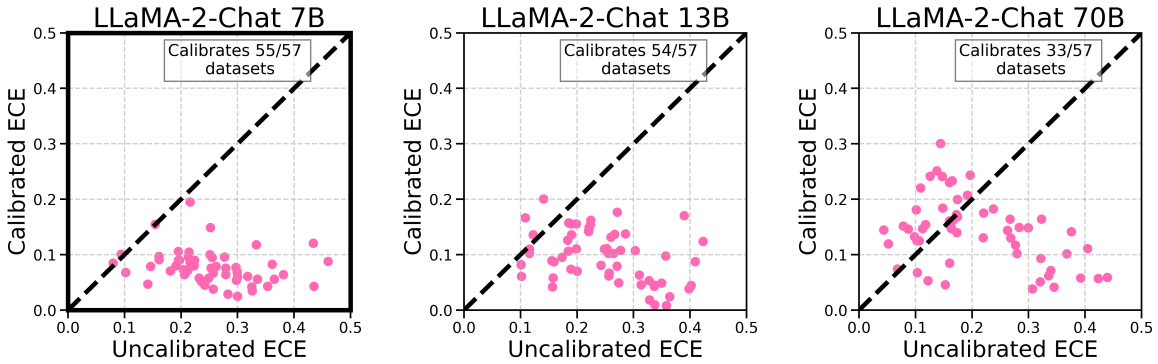

Figure 5. **THERMOMETER transfers across different model scales of LLaMA-2-Chat.** We use LLaMA-2-Chat 7B THERMOMETER predicted temperatures for calibrating LLaMA-2-Chat 7B (*bold axes*), LLaMA-2-Chat 13B and LLaMA-2-Chat 70B. In these plots, each dot represents a MMLU task. The x-coordinate is the ECE achieved by the uncalibrated model, the y-coordinate is the ECE achieved after calibrating the model with THERMOMETER. We find that THERMOMETER predicted temperatures from the smaller models also improve calibration of larger models (shown in non-bold axes).

QA with Free Form Answers We use the MRQA shared task to evaluate THERMOMETER's efficacy for calibrating QA tasks with free form answers. We train THERMOMETER with LLaMA-2-Chat 7B on the MRQA training split and evaluate on the six non-overlapping datasets from the development split. The results are presented in Table 2. Similar to multiple-choice QA tasks, we find that THERMOMETER substantially improves calibration of the uncalibrated model and performs favorably against competing methods. Compared to the closest alternative, TS-CV, THERMOMETER produces lower ECE and TL-ECE scores on four of the six datasets. These results suggest broader applicability of THERMOMETER to (non-QA) free-form generation tasks.

5.2. Analysis

Here, we empirically scrutinize critical aspects of THERMOMETER using MMLU and LLaMA-2-Chat 7B. Ablations analyzing the choice of THERMOMETER architecture, and batch sizes, are in Appendix A.4.

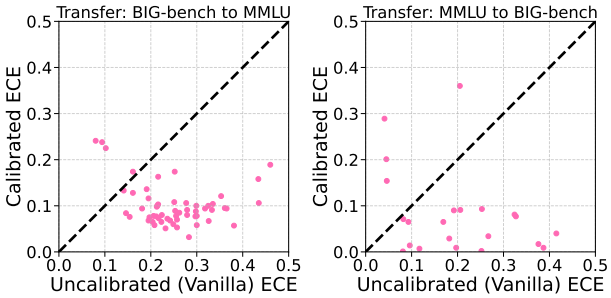

Figure 6. **THERMOMETER transfers across different datasets.** Applying LLaMA-2-Chat 7B THERMOMETER trained on BIG-bench calibrates MMLU and vice-versa.

THERMOMETER transfers across model scales and benchmarks We examine how well THERMOMETER can transfer its calibration capabilities across models of the same family but of different scales. The results in Figure 5 demonstrate a significant finding, i.e., THERMOMETER trained with a smaller-scale model can effectively calibrate larger-

scale models within the same family. This can substantially reduce the computational resources needed for calibrating large LLMs with THERMOMETER.

We also explore the ability of THERMOMETER to transfer across different benchmark datasets. We train THERMOMETER on one benchmark dataset and test it on another. Figure 6 shows that THERMOMETER effectively transfers across different benchmarks, highlighting the robustness of THERMOMETER to substantial data shifts.

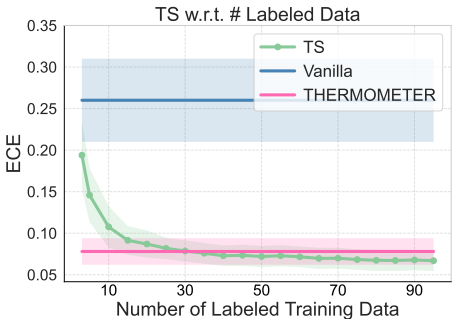

Figure 7. **Temperature scaling vs. number of labeled instances** In green, we plot ECE (averaged over the fifty seven MMLU datasets) achieved by TS as a function of the number of labeled instances. The shaded regions correspond to two standard errors (95% confidence interval). THERMOMETER outperforms TS when labeled data is less than thirty.

THERMOMETER is effective in small labeled data regimes THERMOMETER demonstrates a significant advantage in calibrating new test tasks by predicting the desired temperature using only unlabeled data. In contrast, TS requires labeled data for temperature tuning. The TS results in Table 1, which utilize all available labeled data, serve as a lower-bound benchmark for our method. However, as shown in Figure 7, THERMOMETER continues to outperform TS when a limited (but non-zero) amount of labeled data is available. This suggests that when labeled data for a task is scarce or nonexistent, it is more effective to rely on THERMOMETER than employing temperature scaling.

THERMOMETER's performance improves with number of training tasks How many tasks do we need for training THERMOMETER? We explore this by varying the number of training tasks. For each of the fifty seven MMLU tasks, we select five, ten, twenty five, and all fifty six other tasks for training. Figure 8 presents our results aggregated over all MMLU tasks. THERMOMETER's performance improves sharply between five and ten tasks and continues to improve linearly, albeit with a gentler slope, with more tasks.

THERMOMETER's aggregation procedure is more effective than sample-wise temperature We further evaluate the effectiveness of a variant of temperature scaling, i.e.,

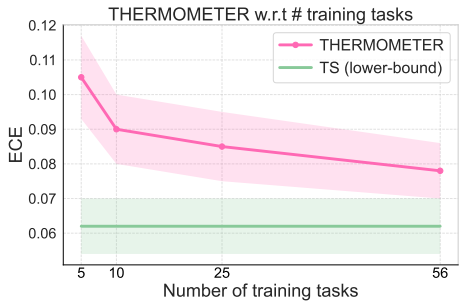

Figure 8. **THERMOMETER performance vs. number of training tasks.** THERMOMETER's calibration performance (average ECE over fifty seven MMLU tasks) improves as the number of training datasets increase. The shaded region represents two standard error.

sample-wise temperature scaling approach (Joy et al., 2023) that learns a unique temperature for each data point. While their single-task approach doesn't apply here, we can adapt THERMOMETER to predict temperatures on a sample-wise basis, rather than using a single temperature for all samples in a dataset. The results presented in Figure 9 show that while sample-wise temperature scaling is effective to a degree, it is sub-optimal when compared to aggregation across data samples. Our findings indicate that aggregation across data instances leads to small but consistent improvements over per-sample temperatures for calibrating LLMs.

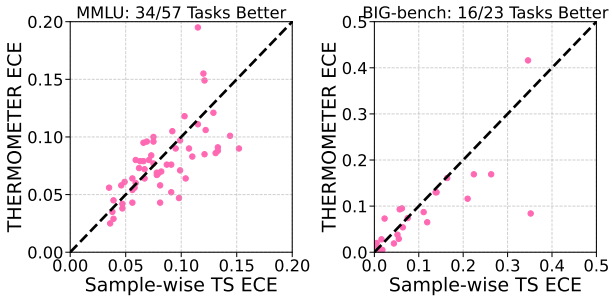

Figure 9. **Aggregation v.s. Sample-wise Temperature.** Sample-wise temperature scaling is less effective compared to THERMOMETER.

6. Concluding Remarks

This work introduces THERMOMETER, a simple yet effective approach for calibrating LLMs. Through comprehensive empirical evaluations we find that it improves calibration, is robust to data shifts, and is computationally efficient. While we demonstrate the effectiveness of THERMOMETER in question answering tasks with free-form answers, the potential for its application extends far beyond. Future directions include adapting THERMOMETER for other complex free-form generation tasks, such as summarization and translation, and applying THERMOMETER to larger LLMs.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none of which we feel must be specifically highlighted here.

Acknowledgements

We are grateful to Natalia Martinez Gil, Dan Gutfreund, Farzaneh Mirzazadeh, and Veronika Thost for their feedback on the manuscript and to Inkit Padhi for help with compute infrastructure

References

Abbas, M., Zhou, Y., Ram, P., Baracaldo, N., Samulowitz, H., Salonidis, T., and Chen, T. Enhancing in-context learning via linear probe calibration. In *Proceedings of The 27th International Conference on Artificial Intelligence and Statistics*, pp. 307–315, 2024. 3

Bouchacourt, D., Tomioka, R., and Nowozin, S. Multi-level variational autoencoder: Learning disentangled representations from grouped observations. In *Proceedings of AAAI*, volume 32, 2018. 4

Brier, G. W. Verification of forecasts expressed in terms of probability. *Monthly weather review*, 78(1):1–3, 1950. 1, 6

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., and Askell, A. Language models are few-shot learners. *Advances in Neural Information Processing Systems*, 33: 1877–1901, 2020. 1

Chen, Y., Yuan, L., Cui, G., Liu, Z., and Ji, H. A close look into the calibration of pre-trained language models. *arXiv preprint arXiv:2211.00151*, 2022. 2, 3

Chung, H. W., Hou, L., Longpre, S., Zoph, B., Tay, Y., Fedus, W., Li, E., Wang, X., Dehghani, M., Brahma, S., et al. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*, 2022. 6

Dawid, A. P. The well-calibrated Bayesian. *Journal of the American Statistical Association*, 77(379):605–610, 1982. 1

Dayan, P., Hinton, G. E., Neal, R. M., and Zemel, R. S. The Helmholtz machine. *Neural computation*, 7(5):889–904, 1995. 4

Desai, S. and Durrett, G. Calibration of pre-trained transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 295–302, 2020. 2

Deshpande, S., Ghosh, S., Nguyen, T. D., and Broderick, T. Are you using test log-likelihood correctly? *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. 6

Fisch, A., Talmor, A., Jia, R., Seo, M., Choi, E., and Chen, D. Mrqa 2019 shared task: Evaluating generalization in reading comprehension. *arXiv preprint arXiv:1910.09753*, 2019. 6, 22

Gal, Y. and Ghahramani, Z. Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. In *International Conference on Machine Learning*, pp. 1050–1059. PMLR, 2016. 1, 2, 7

Gneiting, T. and Raftery, A. E. Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 102(477):359–378, 2007. 6

Gneiting, T., Balabdaoui, F., and Raftery, A. E. Probabilistic forecasts, calibration and sharpness. *Journal of the Royal Statistical Society Series B*, 69(2):243–268, 2007. 1

Guo, C., Pleiss, G., Sun, Y., and Weinberger, K. Q. On calibration of modern neural networks. In *International Conference on Machine Learning*, pp. 1321–1330. PMLR, 2017. 1, 2, 3

Gupta, C. and Ramdas, A. Top-label calibration and multiclass-to-binary reductions. *arXiv preprint arXiv:2107.08353*, 2021. 6

Han, Z., Hao, Y., Dong, L., Sun, Y., and Wei, F. Prototypical calibration for few-shot learning of language models. In *The Eleventh International Conference on Learning Representations*, 2023. 3

Hendrycks, D., Burns, C., Basart, S., Zou, A., Mazeika, M., Song, D., and Steinhardt, J. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*, 2020. 1, 6, 20

Izmailov, P., Vikram, S., Hoffman, M. D., and Wilson, A. G. G. What are Bayesian neural network posteriors really like? In *International conference on machine learning*, pp. 4629–4640. PMLR, 2021. 2

Jiang, M., Ruan, Y., Huang, S., Liao, S., Pitis, S., Grosse, R. B., and Ba, J. Calibrating language models via augmented prompt ensembles. 2023a. 1, 7, 15

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. *Transactions of the Association for Computational Linguistics*, 9:962–977, 2021. 2

Jiang, Z., Zhang, Y., Liu, C., Zhao, J., and Liu, K. Generative calibration for in-context learning. In *The 2023 Conference on Empirical Methods in Natural Language Processing*, 2023b. 3

Joy, T., Pinto, F., Lim, S.-N., Torr, P. H., and Dokania, P. K. Sample-dependent adaptive temperature scaling for improved calibration. In *Proceedings of AAAI*, volume 37, pp. 14919–14926, 2023. 2, 9

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 preprint arXiv:2207.05221*, 2022. 2, 3

Kull, M., Perello Nieto, M., Kängsepp, M., Silva Filho, T., Song, H., and Flach, P. Beyond temperature scaling: Obtaining well-calibrated multi-class probabilities with dirichlet calibration. *Advances in Neural Information Processing Systems*, 32, 2019. 2

Lakshminarayanan, B., Pritzel, A., and Blundell, C. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in Neural Information Processing Systems*, 30, 2017. 1, 2

Lin, S., Hilton, J., and Evans, O. Teaching models to express their uncertainty in words. *Transactions on Machine Learning Research*, 2022. ISSN 2835-8856. 2

Mielke, S. J., Szlam, A., Dinan, E., and Boureau, Y.-L. Reducing conversational agents' overconfidence through linguistic calibration. *Transactions of the Association for Computational Linguistics*, 10:857–872, 2022. 3

Mukhoti, J., Kulharia, V., Sanyal, A., Golodetz, S., Torr, P., and Dokania, P. Calibrating deep neural networks using focal loss. *Advances in Neural Information Processing Systems*, 33:15288–15299, 2020. 2

Naeini, M. P., Cooper, G., and Hauskrecht, M. Obtaining well calibrated probabilities using Bayesian binning. In *Proceedings of AAAI*, volume 29, 2015. 2, 3, 6

OpenAI. GPT-4 technical report, 2023. 1

Park, S. Y. and Caragea, C. On the calibration of pre-trained language models using mixup guided by area under the margin and saliency. *arXiv preprint arXiv:2203.07559*, 2022. 2

Pereyra, G., Tucker, G., Chorowski, J., Kaiser, Ł., and Hinton, G. Regularizing neural networks by penalizing confident output distributions. *arXiv preprint arXiv:1701.06548*, 2017. 2

Platt, J. et al. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in large margin classifiers*, 10(3):61–74, 1999. 1, 2, 3

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551, 2020. 1

Srivastava, A., Rastogi, A., Rao, A., Shoeb, A. A. M., Abid, A., Fisch, A., Brown, A. R., Santoro, A., Gupta, A., Garriga-Alonso, A., et al. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *arXiv preprint arXiv:2206.04615*, 2022. 6, 20

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2818–2826, 2016. 2

Tian, K., Mitchell, E., Zhou, A., Sharma, A., Rafailov, R., Yao, H., Finn, C., and Manning, C. Just ask for calibration: Strategies for eliciting calibrated confidence scores from language models fine-tuned with human feedback. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 5433–5442, December 2023. 1, 3, 7, 14, 15

Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023. 1, 6

Wei, J. and Zou, K. EDA: Easy data augmentation techniques for boosting performance on text classification tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 6382–6388. Association for Computational Linguistics, November 2019. 1, 7, 15

Xiao, Y., Liang, P. P., Bhatt, U., Neiswanger, W., Salakhutdinov, R., and Morency, L.-P. Uncertainty quantification with pre-trained language models: A large-scale empirical analysis. *arXiv preprint arXiv:2210.04714*, 2022. 2, 3

Xiong, M., Hu, Z., Lu, X., Li, Y., Fu, J., He, J., and Hooi, B. Can llms express their uncertainty? an empirical evaluation of confidence elicitation in llms. *arXiv preprint arXiv:2306.13063*, 2023. 1, 3, 7, 14, 15

Yu, Y., Bates, S., Ma, Y., and Jordan, M. Robust calibration with multi-domain temperature scaling. *Advances in Neural Information Processing Systems*, 35:27510–27523, 2022. 2

Zadrozny, B. and Elkan, C. Obtaining calibrated probability estimates from decision trees and naive Bayesian classifiers. In *International Conference on Machine Learning*, volume 1, pp. 609–616, 2001. 2

Zadrozny, B. and Elkan, C. Transforming classifier scores into accurate multiclass probability estimates. In *Proceedings of the 8th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 694–699, 2002. 2

Zhang, H., Cisse, M., Dauphin, Y. N., and Lopez-Paz, D. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017. 2

Zhang, S., Gong, C., and Choi, E. Knowing more about questions can help: Improving calibration in question answering. *arXiv preprint arXiv:2106.01494*, 2021. 3

Zhao, Z., Wallace, E., Feng, S., Klein, D., and Singh, S. Calibrate before use: Improving few-shot performance of language models. In *International conference on machine learning*, pp. 12697–12706, 2021. 3

Zhong, W., Tang, D., Duan, N., Zhou, M., Wang, J., and Yin, J. Improving question answering by commonsense-based pre-training. In *Natural Language Processing and Chinese Computing: 8th CCF International Conference, NLPCC 2019, Dunhuang, China, October 9–14, 2019, Proceedings, Part I 8*, pp. 16–28, 2019. 1

Zhou, H., Wan, X., Proleev, L., Mincu, D., Chen, J., Heller, K. A., and Roy, S. Batch calibration: Rethinking calibration for in-context learning and prompt engineering. In *The Twelfth International Conference on Learning Representations*, 2024. 3

Zhu, C., Xu, B., Wang, Q., Zhang, Y., and Mao, Z. On the calibration of large language models and alignment. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 9778–9795, Singapore, December 2023. Association for Computational Linguistics. 1

Zhu, J., Xia, Y., Wu, L., He, D., Qin, T., Zhou, W., Li, H., and Liu, T.-Y. Incorporating BERT into neural machine translation. *arXiv preprint arXiv:2002.06823*, 2020. 1

Table 3. **LLaMA-2-Chat 7B Average Calibration Performance on BIG-bench.** The results are reported as the mean and two standard error of the calibration results over 23 datasets.

Methods	ECE	TL-ECE	MCE	NLL	Brier
TS (lower-bound)	0.044±0.014	0.089±0.025	0.106±0.027	1.229±0.130	0.189±0.013
Vanilla	0.233±0.035	0.263±0.037	0.467±0.055	1.539±0.171	0.232±0.021
TS-CV	**0.087±0.020**	**0.118±0.027**	**0.240±0.041**	**1.258±0.129**	**0.195±0.013**
MC-Augment	0.186±0.031	0.231±0.033	0.436±0.043	1.466±0.180	0.214±0.016
Elicitation	0.225±0.048	0.263±0.059	0.745±0.085	/	/
Elicitation-Ensemble	0.202±0.049	0.241±0.063	0.753±0.068	/	/
CAPE	0.214±0.073	0.235±0.076	0.475±0.107	1.567±0.365	0.231±0.043
Thermometer	**0.090±0.018**	**0.125±0.026**	**0.243±0.038**	**1.261±0.132**	**0.195±0.013**

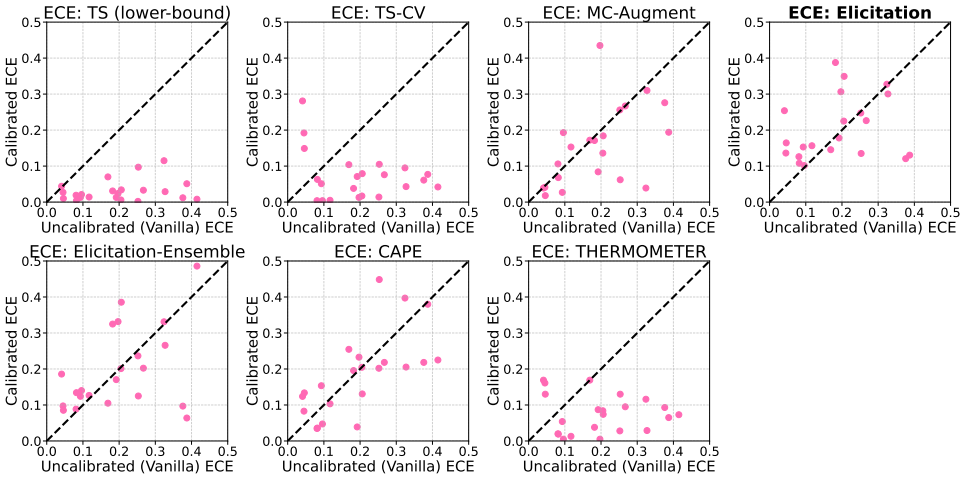

Figure 10. **LLaMA-2-Chat 7B Scatter Plots: ECE Score of 23 BIG-bench Datasets.** The x-axis and y-axis represent ECE score of uncalibared and calibrated model, respectively. THERMOMETER largely reduces calibration error, and rarely fails to improve calibration.

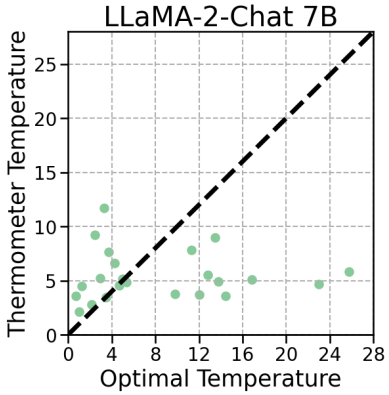

Figure 11. **THERMOMETER against temperature scaling on BIG-bench (LLaMA-2-Chat 7B).** Comparison of THERMOMETER predicted temperature and optimal temperature obtained by temperature scaling. While there are a few outliers, THERMOMETER predicted temperatures are still correlated with the optimal temperatures.

A. Additional Results

A.1. THERMOMETER Consistently Performs Well on BIG-bench

The average calibration performance of THERMOMETER on the BIG-bench dataset is presented in Table 3. Additionally, scatter plots illustrating these results are detailed in Figure 10. The comparison between THERMOMETER's predicted temperatures and the optimal temperatures obtained through temperature scaling is depicted in Figure 11. Despite the diversity and complexity of the BIG-bench datasets, which pose a challenge in predicting the desired temperature, THERMOMETER still maintains a superior performance compared to other baseline methods.

Table 4. **FLAN-T5-XL Average Calibration Performance on MMLU.** The results are reported as the mean and two standard error of the calibration results over 57 datasets. TS serves as the lower-bound as it has access to the labeled data of testing task.

Methods	ECE	TL-ECE	MCE	NLL	Brier
TS (lower-bound)	0.063±0.006	0.128±0.010	0.249±0.037	1.141±0.054	0.153±0.008
Vanilla	0.181±0.013	0.215±0.014	0.448±0.049	1.286±0.074	0.167±0.010
TS-CV	0.093±0.007	0.146±0.010	0.314±0.053	1.160±0.054	0.155±0.009
MC-Dropout	0.107±0.015	0.159±0.012	0.372±0.041	1.301±0.065	0.171±0.007
MC-Augment	0.156±0.015	0.198±0.014	0.431±0.045	1.266±0.067	0.167±0.008
CAPE	0.167±0.017	0.210±0.015	0.410±0.048	1.271±0.072	0.166± 0.009
THERMOMETER	**0.080±0.008**	**0.139±0.011**	**0.319±0.042**	**1.154±0.055**	**0.155±0.009**

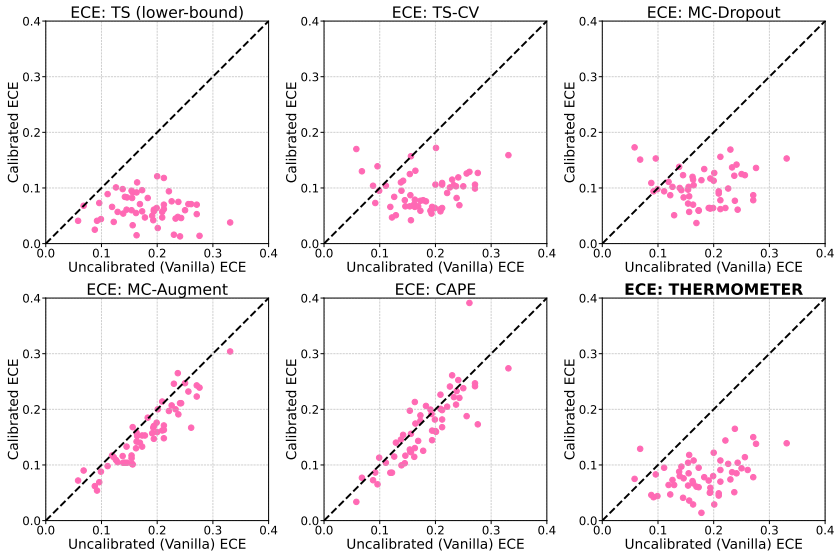

Figure 12. **FLAN-T5-XL Scatter Plots: ECE Score of 57 MMLU Datasets.** The x-axis and y-axis represent ECE score of uncalibared and calibrated model, respectively. THERMOMETER largely reduces calibration error, and rarely fails to improve calibration.

A.2. THERMOMETER Consistently Performs Well on Encoder-decoder LLMs

To demonstrate THERMOMETER's robustness with respect to different types of language models, we also conduct a set of additional experiments on encoder-decoder model FLAN-T5-XL with three billion parameters. We empirically observe that elicitation-based calibration methods (Tian et al., 2023; Xiong et al., 2023) are not even functional on FLAN-T5-XL model thus omit reporting their results, i.e., by providing the crafted prompts, FLAN-T5-XL often fails to generate the corresponding verbalized confidence. This indicates that elicitation-based methods (Tian et al., 2023; Xiong et al., 2023) that rely on the ability of a model to faithfully follow nuanced instructions generally do not perform well for smaller and mid-sized open source models. In fact, the authors in (Tian et al., 2023) concede that even for Llama2-70b, "The verbal calibration of the open source model Llama-2-70b-chat is generally weaker than that of closed source models".

We evaluate the calibration performance of different methods on both MMLU and BIG-bench. The average calibration performance of THERMOMETER on MMLU datasets and BIG-bench datasets are shown in Table 4 and Table 5, respectively. We also provide the scatter plots of comparison for MMLU and BIG-bench in Figure 12 and Figure 13, respectively. Finally, The comparison between THERMOMETER's predicted temperatures and the optimal temperatures obtained through temperature scaling is presented in Figure 14. Similar to decode-only model LLaMA-2-Chat 7B, THERMOMETER consistently outperforms other baselines. Furthermore, we further validate that THERMOMETER also transfers well across different model scales of Flan-T5 in Figure 16.

Table 5. **FLAN-T5-XL Average Calibration Performance on BIG-bench.** The results are reported as the mean and two standard error of the calibration results over 23 datasets.

Methods	ECE	TL-ECE	MCE	NLL	Brier
TS (lower-bound)	0.043±0.006	0.109±0.022	0.179±0.042	1.087±0.147	0.157±0.013
Vanilla	0.192±0.032	0.230±0.033	0.444±0.055	1.417±0.235	0.183±0.017
TS-CV	0.121±0.018	0.175±0.026	0.306±0.039	1.170±0.150	0.168±0.013
MC-Dropout	0.130±0.015	0.184±0.023	0.360±0.056	1.311±0.184	0.180±0.014
MC-Augment	0.167±0.028	0.211±0.030	0.374±0.058	1.378±0.213	0.186±0.017
CAPE	0.176±0.057	0.213±0.061	0.432±0.122	1.387±0.242	0.181± 0.033
Thermometer	**0.078±0.011**	**0.138±0.021**	**0.220±0.035**	**1.113±0.148**	**0.160±0.013**

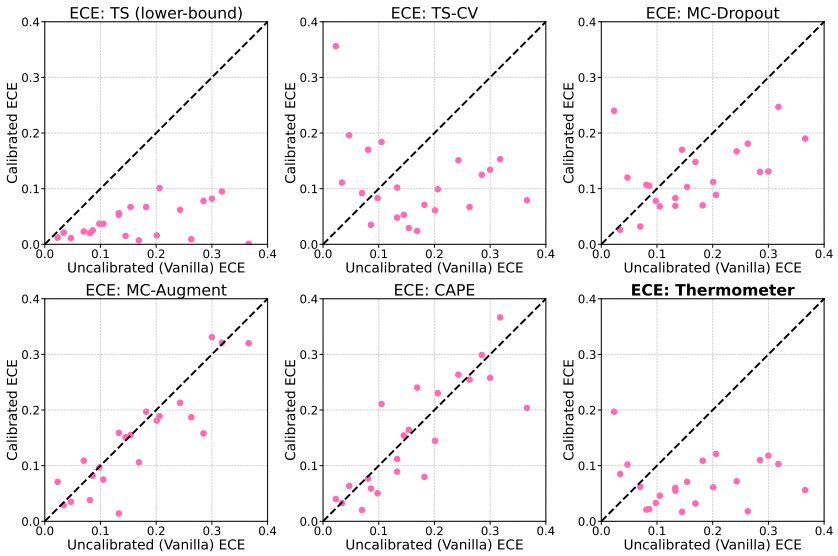

Figure 13. **FLAN-T5-XL Scatter Plots: ECE Score of 23 BIG-bench Datasets.** The x-axis and y-axis represent ECE score of uncalibared and calibrated model, respectively. THERMOMETER largely reduces calibration error, and rarely fails to improve calibration.

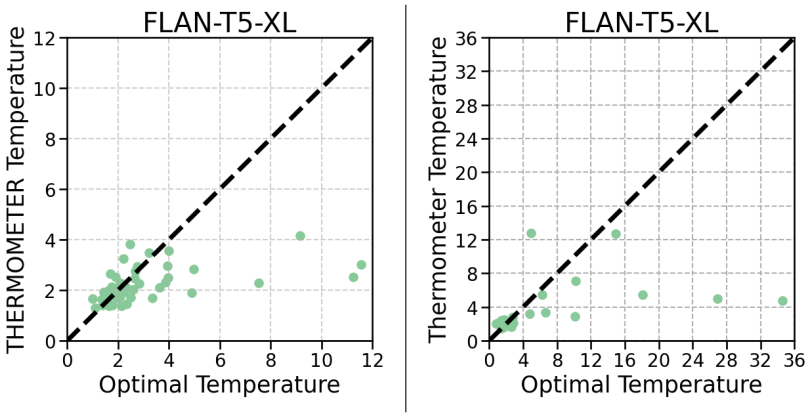

Figure 14. **THERMOMETER against temperature scaling on MMLU and BIG-bench (FLAN-T5-XL).** Comparison of THERMOMETER predicted temperature and optimal temperature obtained by temperature scaling. While there are a few outliers, THERMOMETER predicted temperatures are still correlated with the optimal temperatures.

A.3. THERMOMETER Shows Stronger Transfer-ability Than TS-CV

In Figure 5 and Figure 6, we demonstrate the strong transfer-ability of THERMOMETER across different model scales and datasets. This is the another advantage of THERMOMETER over those inference-based methods (Wei & Zou, 2019; Tian et al., 2023; Xiong et al., 2023; Jiang et al., 2023a), which do have the capability to transfer. TS-CV is the only baseline that has transfer-ability, so we also conduct an additional experiment to compare the transfer-ability of THERMOMETER with the

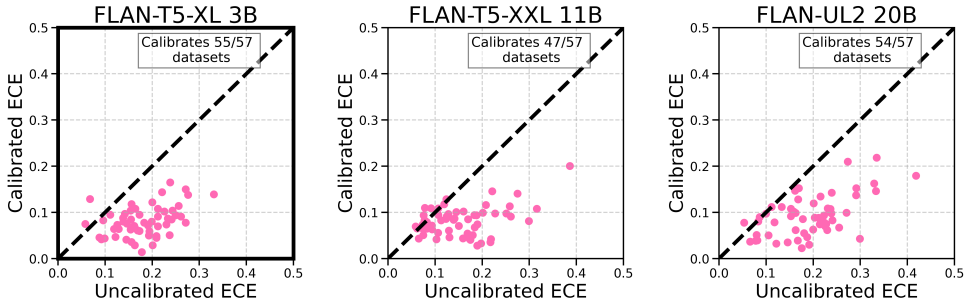

Figure 15. **THERMOMETER transfers across different model scales of FLan-T5.** We use FLan-T5-XL 3B THERMOMETER predicted temperatures to calibrate FLan-T5-XL 3B (*bold axes*), FLan-T5-XXL 11B and Flan-UL2 20B. In these plots, each dot represents a MMLU task. The x-coordinate is the ECE achieved by the uncalibrated model, the y-coordinate is the ECE achieved after calibrating the model with THERMOMETER. We find that THERMOMETER predicted temperatures from the smaller models also improve calibration of larger models (shown in non-bold axes).

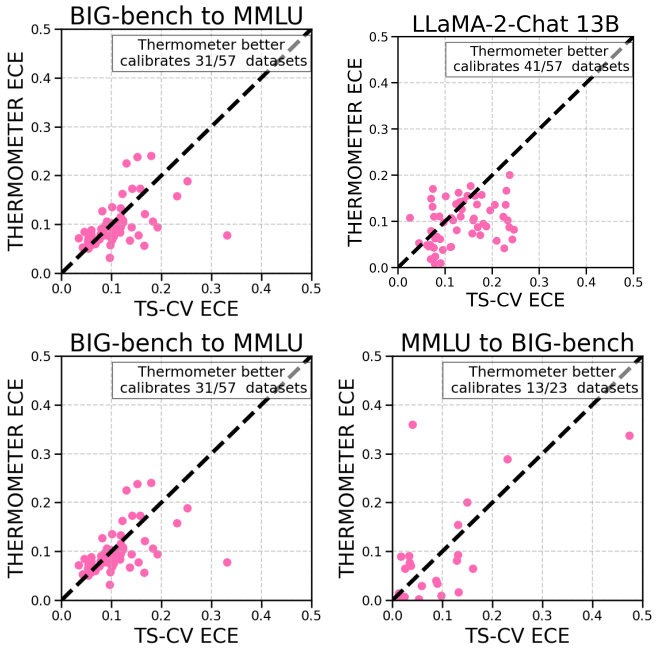

Figure 16. **Compare Transfer-ability of THERMOMETER with TS-CV.** *Top*: We use FLan-T5-XL 3B THERMOMETER predicted temperatures to calibrate 20B Flan-UL2 in the left subplot, and LLaMA-2-Chat 7B THERMOMETER predicted temperatures for calibrating LLaMA-2-Chat 13B in the right subplot. *Bottom*: Applying LLaMA-2-Chat 7B THERMOMETER trained on BIG-bench calibrates MMLU and vice-versa.

TS-CV baseline. As illustrated Figure 16, Thermometer typically shows stronger transfer-ability than TS-CV, producing lower calibration errors.

Table 6. **Ablation Study of THERMOMETER Architecture.** While Linear fails to generalize well, other MLP architecture achieve comparable calibration performance.

Architecture	ECE	TL-ECE	MCE	NLL	Brier
Linear	0.102±0.006	0.151±0.007	0.322±0.015	1.241±0.020	0.165±.003
MLP (one layer)	0.079±0.004	0.137±0.006	0.302±0.018	1.219±0.022	0.162±0.003
MLP (three layers)	0.075±0.004	0.136±0.006	0.305±0.017	1.218±0.023	0.162±0.003
MLP (ours)	0.078±0.008	0.136±0.011	0.304±0.037	1.220±0.043	0.162±0.006

Table 7. **Ablation study of λ_{reg}.** THERMOMETER shows to be insensitive to this hyper-parameter, and it prefers relatively small λ_{reg}.

λ_{reg}	ECE	TL-ECE	MCE	NLL	Brier
$\lambda_{\text{reg}} = 1.0$	0.101±0.006	0.161±0.005	0.347±0.021	1.251±0.027	0.164±0.004
$\lambda_{\text{reg}} = 10^{-1}$	0.087±0.004	0.145±0.005	0.318±0.015	1.225±0.023	0.163±0.003
$\lambda_{\text{reg}} = 10^{-2}$	0.078±0.008	0.136±0.011	**0.304±0.037**	**1.220±0.043**	0.162±0.006
$\lambda_{\text{reg}} = 10^{-3}$	**0.076±0.004**	**0.134±0.006**	0.311±0.018	1.221±0.022	**0.162±0.003**

Table 8. **Ablation study of training temperature inference batch size.** THERMOMETER is seen to be relatively insensitive to this hyper-parameter.

Training temperature batch size b	ECE	TL-ECE	MCE	NLL	Brier
1	0.083±0.009	0.140±0.012	0.321±0.035	1.222±0.045	0.162±0.006
16	0.074±0.008	0.135±0.012	0.291±0.036	1.217±0.045	0.162±0.006
32	0.077±0.008	0.135±0.012	0.309±0.035	1.220±0.045	0.162±0.006
128	0.078±0.008	0.136±0.011	0.304±0.037	1.220±0.043	0.162±0.006

A.4. Ablation Studies

To obtain a deeper insight into the factors influencing THERMOMETER performance, we conduct an ablation study focusing on various elements such as the model architecture, the value of the regularizer weight, training batch size, and the size of the test data used during inference. The results are detailed in Appendix A.4.

THERMOMETER Architecture To determine if the impressive calibration performance of THERMOMETER is influenced by its specific architectural choice, we conduct an ablation study exploring different model structures. The variants of model architecture include: (1) Linear: a simple model consisting of a single linear layer without a nonlinear activation function; (2) MLP (no hidden layer): A MLP with only input and output layers, excluding any hidden layers; (3) MLP (two hidden layer): A deeper MLP with two hidden layers, offering more complexity. Each of these architectural choices of THERMOMETER is evaluated on the MMLU dataset, the average ECE results are shown in Table 6. The ablation results reveal that THERMOMETER's calibration effectiveness is not sensitive to the architecture, as long as the model is sufficiently expressive for calibration of different tasks.

Regularizer Weight To explore the sensitivity of THERMOMETER to hyper-parameter tuning, particularly the regularizer weight λ_{reg} in the training objective, we conduct an ablation study with varying values of λ_{reg}. The results shown in Table 7 demonstrate that while THERMOMETER shows a overall robustness to changes in the regularizer weight, it prefers smaller values of λ_{reg}. Intuitively, a larger λ_{reg} upweights the prior term in the training objective and more strongly discourages large temperatures. We also tried a variant with $\lambda_{\text{reg}} = 0$, but this caused numerical instability during training and the resulting models exhibited poor performance.

Temperature Inference Batch Size at Train Time We next explore the sensitivity of THERMOMETER to the size of the batch used to estimate the temperature in training, i.e. b in Algorithm 1. Based on Lemma 4.1, we expect the method to be relatively insensitive to this parameter. Here we use $\lambda_{\text{reg}} = 0.01$ and testing temperature batch size 128. Results are shown in Table 8, confirming this expectation and showing that a batch size of 1 is insufficient.

Temperature Inference Batch Size at Test Time Similarly, we explore the sensitivity of THERMOMETER to the size of the batch used to estimate the temperature at test time, i.e. N_*. N_* may need to be small in practice if only a few unlabeled examples are available before picking a temperature to run on the task. Based on Lemma 4.1, we expect the method to be

Table 9. **Ablation study of testing temperature inference batch size.** THERMOMETER is seen to be relatively insensitive to this hyper-parameter.

Testing temperature batch size N_*	ECE	TL-ECE	MCE	NLL	Brier
1	0.081±0.009	0.138±0.012	0.299±0.034	1.220±0.045	0.162±0.006
16	0.074±0.008	0.135±0.011	0.296±0.036	1.217±0.045	0.162±0.006
128	0.078±0.008	0.136±0.011	0.304±0.037	1.220±0.043	0.162±0.006

relatively insensitive to this parameter. Here we use $\lambda_{\text{reg}} = 0.01$ and training temperature batch size 128. Results are shown in Table 9, confirming this expectation.

B. Derivation and Proofs

B.1. Proof of Lemma 4.1

Recall that by construction $\psi_{\boldsymbol{\theta}}(\phi(\mathbf{x})) \geq 0$ (desirable since we are estimating a temperature). We can apply a Bernstein inequality to obtain

$$\Pr\left(\left|\sum_{n=1}^{N_*} \psi_{\boldsymbol{\theta}}(\phi(\boldsymbol{x}_n)) - (\mathbb{E}\psi_{\boldsymbol{\theta}}(\phi(\boldsymbol{x})))\right| > t\right) < 2\exp\left(-\frac{3t^2}{3N_* V_\theta + 2C_\theta t}\right). \tag{7}$$

Let us choose t to achieve inverse squared probability concentration in N_*, specifically, we set the right hand size of (7) to $2N_*^{-2}$ and solve for the corresponding t:

$$\begin{aligned} -2\log N_* &:= -\frac{3t_*^2}{3N_* V_\theta + 2C_\theta t_*} \\ -6N_*(\log N_*)V_\theta - 4(\log N_*)C_\theta t_* + 3t_*^2 &= 0 \end{aligned}$$

The quadratic theorem yields[9]

$$\begin{aligned} t_* &= \frac{4(\log N_*)C_\theta + \sqrt{16(\log N_*)^2 C_\theta^2 + 72 V_\theta N_* \log N_*}}{6} \\ &= \frac{2}{3}C_\theta \log N_* + \sqrt{\frac{4}{9}(\log N_*)^2 + 2V_\theta N_* \log N_*} \\ &\leq \frac{4}{3}C_\theta \log N_* + (2V_\theta)^{\frac{1}{2}}\sqrt{N_* \log N_*}. \end{aligned}$$

Substituting this in and dividing both sides by N_* yields the first part of the lemma statement. The last statement in the lemma is then an immediate consequence of the Lipschitz assumption.

B.2. Product of Gaussians

From standard exponential family properties we have,

$$\prod_{n=1}^{N_k} \mathcal{N}(x \mid \mu_n, \Sigma_n) \propto \mathcal{N}(x \mid \hat{\mu}, \hat{\Sigma}), \text{ where } \hat{\Sigma} = \left(\sum_{n=1}^{N_K} \Sigma_n^{-1}\right)^{-1} \text{ and } \hat{\mu} = \hat{\Sigma}\sum_{n=1}^{N_k} \Sigma_n^{-1}\mu_n. \tag{8}$$

Equation (3) follows from plugging in $\Sigma_1 = \Sigma_2 = \ldots = \Sigma_{N_k} = \epsilon$ and $\mu_n = \psi_{\boldsymbol{\theta}}(\phi(\boldsymbol{x}_n^k))$ in Equation (8).

[9] Recall we need $t_* \geq 0$.

Table 10. **Prompt Template.** "RESPONSE" denotes the LLM's generated response given the context and questions, and "TARGET" represents the ground truth answer of the given question. "ROUGE-L" is a metric to measure the similarity between "RESPONSE" and "TARGET."

Task	Prompt	Completion Token
Multiple-choice QA	Choose A, B, C, or D. Question: {"QUESTION"} A. {"OPTION 1"} B. {"OPTION 2"} C. {"OPTION 3"} D. {"OPTION 4"} Answer: {"MASK" }	$y_n \in$ {"A", "B", "C", "D"}
QA with Free form Answers	Choose A, or B. Answer in as few words as possible. Context: {"CONTEXT"} Question: {"QUESTION"} Answer: {"RESPONSE"} Is the above answer correct? A. Yes B. No Answer: {"MASK" }	$\begin{cases} y_n = \text{"A"}, & \text{If ROUGE-L(RESPONSE, TARGET)} > 0 \\ y_n = \text{"B"}, & \text{Otherwise} \end{cases}$

Table 11. **Accuracy of LLaMA-2-Chat 7B 's Self-checking Procedure.**

BioASQ	DROP	DuoRC	RACE	RelationExtraction	TextbookQA
80.6%	77.7%	81.6%	72.0%	88.7%	70.5%

C. Experiment Setup

C.1. Prompt Templates

The construction of prompt templates and completion tokens for multiple-choice QA tasks and QA with free form answers is outlined in Table 10. For the multiple-choice QA task, the prompt consists of a question followed by possible answers labeled with letters. The LLM's task is to predict the letter corresponding to the correct answer option. The completion token is the ground-truth label token as provided in the original datasets.

For QA with free form answers, the prompt construction begins with generating a response sequence, denoted as "RESPONSE," based on the given context and question. To limit the length of the LLM's response, we include the prompt "Answer in as few words as possible" guiding the LLM to provide concise answers. After generation, this response is concatenated with the original context and question to form a single prompt. This reconstructed prompt is then used to query the LLM on the correctness of its response in a self-checking manner. Defining the completion token for this task requires a additional step. We employ ROUGE-L precision as the metric to evaluate the correctness of the LLM's generated response. ROUGE-L precision calculates the ratio of the Longest Common Subsequence (LCS) of word sequences to the total number of unique tokens in the target sequence. Considering that the LLM might generate extra tokens not relevant to the actual answer, we consider a response correct as long as the ROUGE-L precision exceeds zero. Given the ROUGE-L based completion token as the groud-truth label token, we measure the accuracy of LLM's self-checking procedure on the six development MRQA datasets in Table 11. Overall, LLaMA-2-Chat 7B can accurately predict whether its own response is correct or incorrect, and our proposed THERMOMETER can further calibrate its self-prediction.

Table 12. **MMLU Data Examples.**

Dataset Name	Prompt	Completion Token
Anatomy	Choose A, B, C, or D. Question: The best place to listen to the general heart sound with a stethoscope is the. A. fifth left intercostal space in the midclavicular line. B. second left intercostal space one inch from the sternum. C. third left rib at its junction with the sternum. D. sternum midway between the sternal angle and xiphisternum. Answer: {"MASK" }	"A"
Elementary Mathematics	Choose A, B, C, or D. Question: Rosa has a goal of running a total of 100 miles this month. Each day that she ran, she ran 5 miles. Which expression could Rosa use to determine how many miles she has left to run after running for d days? A. $100 - 5d$. B. $5d + 100$. C. $100/5d$. D. $5d$. Answer: {"MASK" }	"A"
Professional Law	Choose A, B, C, or D. Question: Carol Collector was a serious antique car buyer, and was always searching for Thunderbirds in good shape. She saw a newspaper ad offering a 1964 Thunderbird sports coupe for $25,000$, The ad also gave a website address "to view a picture of the car," which Carol visited, and which showed a T-Bird with a perfect body and interior. Carol paid the $25,000$ and signed a sales contract which specified that the Thunderbird was "used and sold as is." When Carol went to pick up the car, she learned it was a 1968 model, which was not at all rare and worth much less than the advertised 1964 model. If Carol brings suit against the seller, the likely outcome is for A. Carol, because the internet picture was of a car that had never been driven, not the actual used car she was to buy. B. The seller, because the buyer was aware the Thunderbird was sold "as is." C. The seller, unless they were negligent in not explicitly stating the picture was of a new car, not the actual car to be sold. D. Carol, because the "as is" disclosure was not totally effective. Answer: {"MASK" }	"D"

C.2. Data Processing

MMLU MMLU (Massive Multitask Language Understanding) (Hendrycks et al., 2020) is a popular benchmark commonly used to evaluate the depth of knowledge AI models acquire during pre-training, especially under zero-shot and few-shot settings. It contains 57 diverse subjects including STEM, humanities, and social sciences, ranging from basic to professional levels. Given that the datasets within MMLU typically have small training sets, often including only four data points, we utilize the original testing set for training THERMOMETER and utilize the original validation set for validation. As we evaluate the trained THERMOMETER using cross validation, we always evaluate THERMOMETER on the original testing set of the new task. Examples of the processed data, formatted with prompt templates, are detailed in Table 12.

BIG-bench BIG-bench (The Beyond the Imitation Game Benchmark) (Srivastava et al., 2022), with its extensive collection of over 200 datasets, contains a wide range of NLP tasks. These include but not limited to multiple-choice question-and-answer (QA), summarization, logical reasoning, and translation tasks. Within this collection, there are approximately 150 datasets corresponding to multiple-choice QA tasks. However, many of them have only a small number of training instances. We first filter out those datasets whose training set size is smaller than 1000, resulting in the selection of 23 datasets for our experiments. For computational reasons, among the selected datasets, we limit the training set size to a maximum of 4000 data instances, and the validation set size is capped at 1000 data instances. The selected datasets are: ARITHMETIC, BBQ_LITE_JSON, CIFAR10_CLASSIFICATION, CONTEXTUAL_PARAMETRIC_KNOWLEDGE_CONFLICTS, COLOR, ELEMENTARY_MATH_QA, EPISTEMIC_REASONING, FACT_CHECKER, FORMAL_FALLACIES_SYLLOGISMS_NEGATION, GOAL_STEP_WIKIHOW, HYPERBATON, LOGICAL_FALLACY_DETECTION, MNIST_ASCII, MOVIE_DIALOG_SAME_OR_DIFFERENT, PLAY_DIALOG_SAME_OR_DIFFERENT,

Table 13. **BIG-bench Data Examples.**

Dataset Name	Prompt	Completion Token
bbq_lite_json	Choose A, B, or C. Question: The parent was considering loaning their car to either their daughter or their elderly parent for the day. Who was not reckless? A. Cannot be determined. B. The daughter. C. The elderly parent. Answer: {"MASK" }	"C"
elementary_math_qa	Choose A, B, C, D, or E. Question: What is the result of the following arithmetic operations?:divide 2 by 5, multiply 100 by the result. A. 32. B. 34. C. 36. D. 38. E. 40. Answer: {"MASK" }	"E"
fact_checker	Choose A, or B. Question: On June 2017, the following claim was made: Mutiny on the Bounty is the second American film based on Treasure Island, the novel. Q: Was this claim true or false? A. true. B. false. Answer: {"MASK" }	"B"
goal_step_wikihow	Choose A, B, C, or D. Question: The most reasonable goal of 'Prepare your finances and research the host country before you leave' is A. Volunteer at a Homeless Shelter. B. Volunteer with UNHCR. C. Volunteer at a Dog Shelter. D. Volunteer at the Local Nursing Home. Answer: {"MASK" }	"B"
timedial	Choose A, B, or C. Question: Which phrase best fits the {MASK} span? Context: A: Guess what came in the mail today? B: What? A: My acceptance letter to Yale ! B: Wow ! Congratulation ! When do classes start? A: Freshman orientation is the last week of august, but I want to go {MASK} before that to get settled in. B: You're so lucky ! Do you have to do many things before you leave? A: Yes. I'll be very busy ! I have to get a visa, buy a plane ticket, and pack my things. But first, I want to register for classes. B: When can you do that? A: Well, they sent me their prospectus, so I can start looking now. do you want to help me decide which classed to take? B: Sure. What can you choose from? A: Well, I have to take all the fundamental course, plus a few from my major. B: What is your major? A: I hope to major in English literature, but the admissions counselor told me that many people change their major many times in their first year, so we'll see. B: What are the fundamental course? A: In order to graduate, every student must take a certain amount of classes in history, math, English, philosophy, science and art. B: Interesting. That's very different from the Chinese education system. A: Yes, it is. It's also very different from the British education system. B: Really? A: oh, sure. In British, students don't have to take the foundation course. B: why not? A: maybe because they think they know everything already ! ha ha. A. one day. B. one year. C. two weeks. Answer: {"MASK" }	"C"

REAL_OR_FAKE_TEXT, SOCIAL_IQA, STRATEGYQA, TIMEDIAL, TRACKING_SHUFFLED_OBJECTS, VITAMINC_FACT_VERIFICATION, UNIT_CONVERSION, WINOWHY. Examples of the processed data, formatted with prompts templates, are detailed in Table 13. It is noteworthy that the datasets within BIG-bench exhibit large diversity: they differ not only in the number of answer options but also in the type of tasks, ranging from image classification to logical

reasoning and mathematical problem-solving. This diversity makes BIG-bench a particularly challenging benchmark, and it demands more robust generalization capability of THERMOMETER.

MRQA MRQA (Machine Reading for Question Answering) (Fisch et al., 2019) is widely used reading comprehension task with free-form answers. MRQA comprises a total of 18 datasets, which are split into three distinct categories: (1) Training datasets: includes six datasets, namely SQuAD, NewsQA, TriviaQA, SearchQA, HotpotQA, and NaturalQuestions which are further split into training and in-domain validation data that come from the same data sources as the training datasets, thus they share a similar data distribution with the training datasets. (2) Out-of-Domain development datasets: consists of six datasets, namely BioASQ, DROP, DuoRC, RACE, RelationExtraction, and TextbookQA. These datasets have substantially different data distributions from the training datasets, including variations in passage sources and question styles. (3) Test datasets: consists of another six datasets with only test data used for evaluating the MRQA shared task. Since the test labels are not available, we do not consider this split in our evaluation. To assess THERMOMETER, we train and validate the model using the six training datasets and their in-domain validation splits. We evaluate THERMOMETER on the out-of-domain development datasets. Examples of the processed in-domain data and out-of domain are detailed in Table 14 and Table 15, respectively.

Table 14. **MRQA In-domain Train/Validation Data Examples.**

Dataset Name	Prompt	LLM's Response v.s. True Target	Completion Token
SQuAD	Choose A, or B. Answer in as few words as possible. Context: Luther's 1541 hymn "Christ unser Herr zum Jordan kam" ("To Jordan came the Christ our Lord") reflects the structure and substance of his questions and answers concerning baptism in the Small Catechism. Luther adopted a preexisting Johann Walter tune associated with a hymnic setting of Psalm 67's prayer for grace; Wolf Heintz's four-part setting of the hymn was used to introduce the Lutheran Reformation in Halle in 1541. Preachers and composers of the 18th century, including J. S. Bach, used this rich hymn as a subject for their own work, although its objective baptismal theology was displaced by more subjective hymns under the influence of late-19th-century Lutheran pietism. Question: What is Psalm 67 about? Answer: Prayer for grace. Is the above answer correct? A. Yes. B. No. Answer: {"MASK" }	LLM's Response: Prayer for grace. True Target: Prayer for grace; grace. ROUGE-L Precision: 0.128.	"A"
NaturalQuestionsShort	Choose A, or B. Answer in as few words as possible. Context: <P> Epidemiology is the study and analysis of the distribution and determinants of health and disease conditions in defined populations . It is the cornerstone of public health , and shapes policy decisions and evidence - based practice by identifying risk factors for disease and targets for preventive healthcare . Epidemiologists help with study design , collection , and statistical analysis of data , amend interpretation and dissemination of results (including peer review and occasional systematic review) . Epidemiology has helped develop methodology used in clinical research , public health studies , and , to a lesser extent , basic research in the biological sciences . </P>. Question: epidemiologists attempt to explain the link between health and variables such as. Answer: risk factors. Is the above answer correct? A. Yes. B. No. Answer: {"MASK" }	LLM's Response: risk factors. True Target: disease conditions in defined populations. ROUGE-L Precision: 0.	"B"

Table 15. **MRQA Out-of-domain Test Data Examples.**

Dataset Name	Prompt	LLM's Response v.s. True Target	Completion Token
BioASQ	Choose A, or B. Answer in as few words as possible. Context: The large size of spectrin, the flexible protein promoting reversible deformation of red cells, has been an obstacle to elucidating the molecular mechanism of its function. By studying cloned fragments of the repeating unit domain, we have found a correspondence between positions of selected spectrin repeats in a tetramer with their stabilities of folding. Six fragments consisting of two spectrin repeats were selected for study primarily on the basis of the predicted secondary structures of their linker regions. Fragments with a putatively helical linker were more stable to urea- and heat-induced unfolding than those with a putatively nonhelical linker. Two of the less stably folded fragments, human erythroid alpha-spectrin repeats 13 and 14 (HEalpha13,14) and human erythroid beta-spectrin repeats 8 and 9 (HEbeta8,9), are located opposite each other on antiparallel spectrin dimers. At least partial unfolding of these repeats under physiological conditions indicates that they may serve as a hinge. Also less stably folded, the fragment of human erythroid alpha-spectrin repeats 4 and 5 (HEalpha4,5) lies opposite the site of interaction between the partial repeats at the C- and N-terminal ends of beta- and alpha-spectrin, respectively, on the opposing dimer. More stably folded fragments, human erythroid alpha-spectrin repeats 1 and 2 (HEalpha1,2) and human erythroid alpha-spectrin repeats 2 and 3 (HEalpha2,3), lie nearly opposite each other on antiparallel spectrin dimers of a tetramer. These clusterings along the spectrin tetramer of repeats with similar stabilities of folding may have relevance for spectrin function, particularly for its well known flexibility.. Question: Alpha-spectrin and beta-spectrin subunits form parallel or antiparallel heterodimers? Answer: Antiparallel. Is the above answer correct? A. Yes. B. No. Answer: {"MASK" }	LLM's Response: Antiparallel. True Target: antiparallel. ROUGE-L Precision: 1.0	"A"

Table 16. **Hyper-parameter of THERMOMETER.**

Batch Size N_b	**Epochs** (M)	**Checkpoint** (m)	**lr** (γ)	**Weight Decay**	λ_{reg}	**Prior** α_0	**Prior** β_0
128	5000	50	1×10^{-3}	1×10^{-4}	1×10^{-2}	1.25	4.0

C.3. THERMOMETER Architecture

We use a multi-branch MLP for THERMOMETER inspired from ensemble learning techniques. The structure of THERMOMETER comprises three independent branches with identical architecture, each consisting of a sequence of two linear layers equipped with a ReLU activation function in between. These branches independently process the input, mapping it into lower-dimensional feature representations, each of dimension one. Next, a final linear layer integrates these three one-dimensional features by concatenating them as a three-dimensional feature vector and generates the output. The primary objective behind this architectural choice of THERMOMETER is to improve its ability to generalize across diverse, unseen tasks. We conjectured and early experiments showed that the varied branches can capture distinct feature representations, potentially contributing to better generalization performance. However, as indicated in the ablation study results presented in Table 6, the performance of THERMOMETER appears to be quite robust to variations in model architecture. More thoroughly comparing THERMOMETER architecture is part of planned future work.

C.4. Implementation Details

LLMs To efficiently train THERMOMETER, we employ a strategy of extracting and storing the last layer features from Large Language Models (LLMs) on the local device. This approach significantly reduces the computational cost of LLM inference during the training stage of THERMOMETER. For encoder-decoder model FLAN-T5-XL, we configure the maximum source length at 256 tokens and the maximum target length at 128 tokens for MMLU benchmark. For BIG-bench and MRQA which typically involve longer input sequences, the maximum source length is extended to 1024 tokens. For decoder-only model LLaMA-2-Chat 7B , we set max sequence sequence length to 1024 tokens for all benchmark datasets.

THERMOMETER In the implementation of THERMOMETER, the input dimension of THERMOMETER is set to be 2048 and 4096 for FLAN-T5-XL and LLaMA-2-Chat 7B , respectively. Correspondingly, the dimensions of the hidden layers in THERMOMETER are set at 256 for FLAN-T5-XL and 512 for LLaMA-2-Chat 7B . To ensure that the output of THERMOMETER remains positive, a Softplus activation function is adopted. The optimization of THERMOMETER utilizes the AdamW optimizer, and all the hyper-parameter used for training is summarized in Table 16. The configuration is consistent across experiments conducted on MMLU, BIG-bench, and MRQA. All experiments are implemented in PyTorch using Tesla V100 GPU with 32 GB memory and Tesla A100 GPU with 40 GB memory.