Beyond Post-Hoc Temperature Scaling: Bilevel Optimization for LLM Calibration

Ruochen Jin[1,*], **Zhanliang Wang**[2,*], **Zongyu Dai**[2], **Jiancong Xiao**[3,†], **and Bojian Hou**[2,†]
[1]Dartmouth College, Hanover, NH; [2]University of Pennsylvania, Philadelphia, PA;
[3]National University of Singapore, Singapore
ruochen.jin.gr@dartmouth.edu, aaronwzl@upenn.edu, daizy@sas.upenn.edu
jiancongxiao@nus.edu.sg, bojianh@upenn.edu

Abstract

Preference alignment often makes large language models (LLMs) overconfident and poorly calibrated. Traditional post-hoc temperature scaling is inherently domain-dependent: a temperature fitted on one domain does not generalize across domains. This motivates us to modify model parameters during training to improve calibration. We propose maximizing the entropy of predictive distributions as the calibration objective, which directly targets overconfidence by discouraging overly concentrated predictions. Inspired by temperature scaling, we realize this through a bilevel optimization formulation, where the lower level trains the model under a parametric loss and the upper level selects loss hyperparameters to maximize entropy. To make the framework practical at LLM scale, we adopt an efficient first-order approximation that avoids explicit second-order computation. Across both multiple-choice and open-ended generative question answering, experiments demonstrate that our method yields well-calibrated LLMs with particular advantages in out-of-domain generalization[1].

1 Introduction

Large Language Models (LLMs) (OpenAI, 2023; Anthropic, 2024) have become a foundation of modern Natural Language Processing (NLP), achieving strong performance across a wide range of tasks (Bubeck et al., 2023; Chowdhery et al., 2023; Touvron et al., 2023; Team et al., 2023). Beyond task performance, practical success also requires reliability. *Preference alignment*, notably through Reinforcement Learning from Human Feedback (RLHF) (Ouyang et al., 2022) and Direct Preference Optimization (DPO) (Rafailov et al., 2023), steers models toward human-preferred behaviors but degrades calibration as a side effect. While pre-trained LLMs are often reasonably calibrated, aligned models frequently become overconfident (OpenAI, 2023; He et al., 2023). This matters because confidence is widely used as a proxy for reliability; in high-stakes settings such as healthcare or legal analysis, miscalibrated confidence can lead to poor decisions and misplaced trust (Savage et al., 2025).

Most existing calibration methods are post-hoc, with Temperature Scaling (TS) as the canonical example. TS is two-stage and *test-time*: the model is first trained, then a temperature is selected on a validation set to improve the calibration of predicted probabilities. Although simple and effective, TS is inherently dataset-dependent—a temperature fitted on one dataset often does not transfer to another domain—so it is difficult to find a single "general temperature" that calibrates well across diverse datasets.

This motivates a training-time view of calibration. Instead of searching for a dataset-specific temperature only after training, we fine-tune the model itself so that a temperature-like adjustment generalizes across datasets, treating the model parameters, rather than a

*Equal contribution. †Corresponding author.
[1]Code is available at https://github.com/BojianHou/calm.

purely post-hoc correction, as the target. This requires a training objective that explicitly encourages calibrated predictions during optimization.

A second challenge lies in selecting an appropriate calibration-oriented objective. Our empirical observations in Figure 1, together with prior findings (Xiao et al., 2025a; Leng et al., 2025), suggest that overconfidence is the predominant calibration failure mode in aligned LLMs. This observation motivates the use of entropy maximization, which provides a controllable direction for improving calibration without prescribing a specific target solution. The maximum-entropy solution represents the opposite extreme of underconfidence. Increasing the predictive entropy therefore moves the model continuously between these two regimes, potentially allowing it to reach a point at which its confidence better matches its predictive accuracy.

Inspired by the two-stage nature of temperature scaling, we formulate calibration-oriented training as a bilevel optimization problem. Temperature scaling can be viewed as a special two-stage procedure: the lower level learns the model parameters, and the upper level selects a temperature to improve calibration. We generalize this idea from post-hoc test-time adjustment to training-time loss design. In our formulation, the lower level fine-tunes the LLM under a parametric training loss, while the upper level optimizes temperature-like loss hyperparameters using the entropy-based calibration objective on held-out inputs. From this perspective, post-hoc TS becomes a special case of a broader training-time framework for calibration-oriented optimization.

A key practical challenge, however, is that bilevel optimization is computationally expensive for LLMs. Exact hypergradient computation typically involves differentiating through the lower-level optimization, which requires second-order derivatives and, in implicit formulations, may further involve Hessian-inverse terms (Tarzanagh et al., 2024; Nazari et al., 2025). These operations are prohibitively expensive for modern LLM fine-tuning. To address this issue, we develop an efficient first-order bilevel optimization framework inspired by the algorithm introduced in Bilevel Optimization Made Easy (BOME; Liu et al. 2022) and propose Calibration for Large Models via Bilevel Optimization (CALM). Our method avoids explicit hypergradient computation and instead uses first-order updates to approximate the upper-level optimization, making calibration-oriented bilevel training practical at scale.

Our contributions are as follows:

- We identify the dataset dependence of post-hoc temperature scaling and propose a training-time perspective that aims to learn models admitting a more general calibration adjustment across datasets.
- We formulate calibration-oriented loss design as a bilevel optimization problem, where the lower level trains the model and the upper level optimizes temperature-like loss hyperparameters using an entropy-based objective.
- We propose an entropy-maximization upper-level objective to mitigate overconfidence in aligned LLMs without requiring additional hard labels.
- We empirically demonstrate that our method provides strong calibration performance across both multiple-choice and open-ended generative question answering, with especially clear advantages in out-of-domain settings, against strong training-time baselines including calibration-aware fine-tuning.

2 Related Work

Calibration for LLMs. Recent work shows that calibration remains a central reliability issue for large language models, especially after instruction tuning (Jiang et al., 2021; Xiao et al., 2022; Chen et al., 2023) or preference alignment (Ouyang et al., 2022; Xiao et al., 2025b; Liu et al., 2025). Prior studies on LLM calibration have mainly followed three directions. First, several works adapt *post-hoc* calibration ideas to LLMs, with temperature scaling often emerging as a strong baseline (Jiang et al., 2021; Chen et al., 2023; Shen et al., 2024). Second, some methods introduce auxiliary models, additional calibration heads, or

external confidence predictors to recalibrate model outputs (Zhang et al., 2021; Kadavath et al., 2022; Shen et al., 2024), including lightweight calibration layers trained on top of a frozen LM (Liu et al., 2024) and probes that read a model's hidden states to recover latent correctness signals (Burns et al., 2023). Third, another line of work teaches LLMs to explicitly express their uncertainty, either through prompting or through dedicated supervision on confidence estimation (Lin et al., 2022; Tian et al., 2023; Xu et al., 2024). Although related, work on debiasing in-context learning (Abbas et al., 2024; Han et al., 2023; Jiang et al., 2023b) addresses prompt-induced prediction bias rather than statistical calibration in the strict sense. In addition, Wang et al. (2026) proposed a semantic-sampling framework for evaluating calibration in open-ended question answering.

Training-time optimization for LLM calibration. More recent studies move beyond post-hoc correction and directly optimize calibration during fine-tuning or alignment. Prior work has proposed uncertainty-aware or calibration-aware objectives that modify the training loss so as to improve calibration while retaining task performance (Han et al., 2024; Xiao et al., 2025a). Related work also studies how the interaction between fine-tuning data and a model's prior knowledge can systematically induce overconfidence, and proposes cognition-aware training strategies to mitigate this effect (Wang et al., 2025b). A complementary line intervenes during alignment itself, calibrating the RLHF reward model to curb the overconfidence introduced by preference optimization (Leng et al., 2025). In addition, recent evidence suggests that label smoothing can be an effective training-time regularizer for preserving calibration during LLM fine-tuning, although its effectiveness may depend on model scale and vocabulary size (Huang et al., 2025).

Positioning of our work. Our work belongs to this emerging line of *training-time optimization for calibration*: rather than post-hoc scaling or auxiliary confidence estimators, we directly optimize calibration during supervised fine-tuning, motivated by evidence that miscalibration in aligned LLMs is induced by training itself. Unlike auxiliary-head calibrators (Liu et al., 2024), hidden-state probes (Burns et al., 2023), or RLHF-time reward calibration (Leng et al., 2025) which add a module, read activations, or reshape rewards, CALM optimizes the base model's own predictive distribution during fine-tuning and is thus complementary to them.

3 Problem Setup

Let $\mathcal{S}_{\mathcal{T}} = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$ be a training dataset of size n, and let $\mathcal{S}_{\mathcal{V}} = \{(\mathbf{x}_i^v, y_i^v)\}_{i=1}^m$ be a validation dataset of size m, where each sample $(\mathbf{x}, y)$ is drawn i.i.d. from a distribution $\mathcal{D}$ over $\mathcal{X} \times [K]$, with input space $\mathcal{X}$ and label space $[K] := \{1, \ldots, K\}$. We use $(X, Y) \sim \mathcal{D}$ to denote a generic test sample drawn from the same distribution.

Let $z_\theta : \mathcal{X} \to \mathbb{R}^K$ denote a model parameterized by θ, where the corresponding logit vector is given by $z_\theta(\mathbf{x}) = (z_{\theta,1}(\mathbf{x}), \ldots, z_{\theta,K}(\mathbf{x}))$. Its predictive distribution is $f_\theta(\cdot \mid \mathbf{x}) = \mathrm{softmax}(z_\theta(\mathbf{x})) \in \Delta_K$, where $\Delta_K := \{\mathbf{p} \in [0,1]^K : \sum_{i=1}^K p_i = 1\}$ is the K-dimensional probability simplex. For each class $i \in [K]$, the predicted probability is

$$f_\theta(i \mid \mathbf{x}) = \frac{\exp(z_{\theta,i}(\mathbf{x}))}{\sum_{j=1}^K \exp(z_{\theta,j}(\mathbf{x}))},$$

and the predicted label is $\hat{y}_\theta(\mathbf{x}) = \arg\max_{i \in [K]} z_{\theta,i}(\mathbf{x})$.

Multiple-choice LLM setting. To extend the classification formulation to a multiple-choice setting with four answer candidates $\{A, B, C, D\}$[2] in LLMs, we extract the corresponding logits $z_{\theta,A}(\mathbf{x}), z_{\theta,B}(\mathbf{x}), z_{\theta,C}(\mathbf{x})$, and $z_{\theta,D}(\mathbf{x})$. The definitions of predicted probabilities and labels then follow analogously from the classification setting.

[2]We adopt the four-option format $\{A, B, C, D\}$ for readability; the formulation naturally generalizes to a K-choice setting.

Open-ended generation setting. The multiple-choice view above makes calibration easy to define over a fixed option set, but CALM itself is not tied to it: the entropy-based upper-level objective operates on the model's next-token predictive distribution and requires no fixed label set (Section 4). We therefore also evaluate CALM on free-form generation, where the answer is an arbitrary token sequence rather than one of K options. In this setting a token-level confidence no longer corresponds to a single categorical prediction, so at evaluation time we follow the semantic-sampling protocol of Wang et al. (2026): for a query we draw multiple samples, cluster semantically equivalent generations, and treat each cluster's empirical frequency as the model's confidence in that answer, which yields a well-defined calibration target (Sem-ECE) for open-ended outputs.

4 Method

Let us start from the temperature scaling approach. Given a fixed model, temperature scaling is a post-processing (or test-time) calibration technique that rescales logits by a positive scalar temperature parameter $\tau > 0$. Specifically, given logits $z_\theta(\mathbf{x}) \in \mathbb{R}^K$, the temperature-scaled predictive distribution is defined as

$$p_i^{(\tau)}(\mathbf{x}) = \frac{\exp\left(z_{\theta,i}(\mathbf{x})/\tau\right)}{\sum_{j=1}^{K} \exp\left(z_{\theta,j}(\mathbf{x})/\tau\right)}, \qquad i \in [K].$$

The temperature τ controls the confidence of the prediction: $\tau < 1$ sharpens the distribution, while $\tau > 1$ produces a more uniform (less confident) distribution. The temperature τ is estimated by minimizing the empirical cross-entropy on a dataset:

$$\min_{\tau > 0} \; -\frac{1}{n} \sum_{i=1}^{n} \log p_{y_i}^{(\tau)}(\mathbf{x}_i). \tag{1}$$

The optimized temperature often improves the agreement between predictive confidence and empirical accuracy on the calibration dataset. A significant drawback of temperature scaling is that the learned temperature τ is highly specific to the current dataset and generalizes poorly to others. Consequently, achieving proper model calibration requires identifying a distinct temperature for each unique dataset (Yu et al., 2022).

In the era of LLMs, this has become a major drawback compared to the previous era of specialized classification models. A truly robust LLM should be well-calibrated across a diverse range of tasks without requiring task-specific tuning.

To address this, we identify three potential causes for poor generalization and tackle them simultaneously. First, one possible source of the lack of generalizability stems from the scalar nature of τ. A single-dimension value may be too restrictive to capture complex data distributions. Consequently, we consider a higher-dimensional temperature parameter:

$$\boldsymbol{\alpha} = (\mathbf{l}, \boldsymbol{\tau}), \quad \mathbf{l} \in \mathbb{R}^K, \; \boldsymbol{\tau} \in \mathbb{R}^K_{>0}.$$

Here, $\mathbf{l}$ introduces additive logit shifts, while $\boldsymbol{\tau}$ serves as a vector of class-dependent temperatures. In the experiments and appendices we equivalently write the multiplicative factor as $\mathbf{d}_y := \tau_y^{-1}$, so $(\mathbf{d}_y, \mathbf{l}_y)$ are the per-vocabulary adjustments; a single global temperature T is the tied special case ($\mathbf{d}_y$ all equal, $\mathbf{l} = \mathbf{0}$). Given an input-label pair $(\mathbf{x}, y)$ and the model's logits $z_\theta(\mathbf{x}) \in \mathbb{R}^K$, the predicted probabilities are defined as:

$$f_{\boldsymbol{\alpha},\theta}(y \mid \mathbf{x}) = \frac{\exp(\tau_y^{-1} z_{\theta,y}(\mathbf{x}) + l_y)}{\sum_{i=1}^{K} \exp(\tau_i^{-1} z_{\theta,i}(\mathbf{x}) + l_i)}. \tag{2}$$

Notably, when $l_i = 0$ and $\tau_i = \tau$ for all $i \in [K]$, Equation (2) reduces to standard temperature scaling. By applying the higher-dimensional temperature to Equation (1), the loss becomes:

$$\min_{\boldsymbol{\alpha}} \; -\frac{1}{n} \sum_{i=1}^{n} \log f_{\boldsymbol{\alpha},\theta}(y_i \mid \mathbf{x}_i). \tag{3}$$

Second, we posit that the lack of generalizability may be an inherent limitation of post-processing, where the model parameters remain fixed. While a universal temperature may not exist for a specific given model, there likely exists a model with comparable performance that is more amenable to a universal temperature. Consequently, we extend temperature scaling to training time, aiming to optimize both the temperature and the model parameters jointly. Under this framework, Equation (3) is reformulated as:

$$\min_{\boldsymbol{\alpha},\theta} \; -\frac{1}{n}\sum_{i=1}^{n} \log f_{\boldsymbol{\alpha},\theta}(y_i \mid \mathbf{x}_i). \tag{4}$$

This step moves beyond post-hoc calibration and should be viewed as a training-time generalization of temperature scaling rather than standard temperature scaling itself.

Third, the poor generalization may stem from overfitting to the specific characteristics of the validation dataset used for calibration. To address this, we reformulate Equation (4) into a more generalized loss function that reduces dependence on ground-truth labels. A common failure mode of aligned LLMs is *overconfidence*: the predictive distribution is often excessively concentrated even when the model is uncertain. We therefore employ entropy maximization as a direct mechanism to discourage overly sharp predictions.

Given the predictive distribution $f_\theta(\cdot \mid \mathbf{x}) \in \Delta_K$, its Shannon entropy is defined as:

$$H(f_{\boldsymbol{\alpha},\theta}(\cdot \mid \mathbf{x})) = -\sum_{c=1}^{K} f_{\boldsymbol{\alpha},\theta}(c \mid \mathbf{x}) \log f_{\boldsymbol{\alpha},\theta}(c \mid \mathbf{x}). \tag{5}$$

By replacing the cross-entropy in Equation (4) with the entropy term, we obtain the following objective:

$$\min_{\boldsymbol{\alpha},\theta} \; -\frac{1}{n}\sum_{i=1}^{n} H(f_{\boldsymbol{\alpha},\theta}(\cdot \mid \mathbf{x}))\,. \tag{6}$$

This objective is appealing because entropy maximization provides a controllable direction for calibration rather than prescribing the final solution. Modern neural networks are typically overconfident, whereas the maximum-entropy solution represents extreme underconfidence. As entropy increases, the model moves continuously between these regimes and may pass through a point where its confidence better matches its predictive accuracy.

Entropy maximization alone could overshoot this point and impair discrimination. Without further constraints, optimizing solely for entropy could lead to a degradation in task performance. To address this, we reintroduce the cross-entropy loss on the training set as a constraint, requiring the model parameters to be optimal with respect to $\boldsymbol{\alpha}$:

$$\theta \in \arg\min_{\theta'} \; -\frac{1}{n}\sum_{i=1}^{n} \log f_{\boldsymbol{\alpha},\theta'}(y_i \mid \mathbf{x}_i). \tag{7}$$

Finally, by applying the training set $\mathcal{S}_\mathcal{T}$ and the calibration set $\mathcal{S}_\mathcal{V}$ to the objectives in Equation (6) and Equation (7), we arrive at the bilevel optimization formulation. For notational convenience, we denote the upper-level entropy objective as $\mathcal{L}_{\text{ent}}$ and the lower-level training loss as $\mathcal{L}_{\text{tr}}$:

$$\min_{\boldsymbol{\alpha},\theta} \; \underbrace{-\frac{1}{|\mathcal{S}_\mathcal{V}|}\sum_{\mathbf{x}_i \in \mathcal{S}_\mathcal{V}} H(f_{\boldsymbol{\alpha},\theta}(\cdot \mid \mathbf{x}_i))}_{\mathcal{L}_{\text{ent}}^{\mathcal{S}_\mathcal{V}}(\boldsymbol{\alpha},\theta)} \text{ s.t. } \quad \theta \in \arg\min_{\theta'} \; \underbrace{-\frac{1}{|\mathcal{S}_\mathcal{T}|}\sum_{(\mathbf{x}_i,y_i) \in \mathcal{S}_\mathcal{T}} \log f_{\boldsymbol{\alpha},\theta'}(y_i \mid \mathbf{x}_i)}_{\mathcal{L}_{\text{tr}}^{\mathcal{S}_\mathcal{T}}(\theta';\boldsymbol{\alpha})}\,. \tag{8}$$

The upper level minimizes $\mathcal{L}_{\text{ent}}$ to discourage overconfident predictions, while the lower level ensures that θ remains optimal for $\mathcal{L}_{\text{tr}}$ under the given $\boldsymbol{\alpha}$, preserving discriminative ability. An equivalent formulation that makes the implicit dependence $\theta(\boldsymbol{\alpha})$ explicit is:

$$\begin{aligned} \min_{\boldsymbol{\alpha}} \quad & \mathcal{L}_{\text{ent}}^{\mathcal{S}_\mathcal{V}}(\boldsymbol{\alpha}, \theta(\boldsymbol{\alpha})) \\ \text{s.t.} \quad & \theta(\boldsymbol{\alpha}) \in \arg\min_{\theta'} \; \mathcal{L}_{\text{tr}}^{\mathcal{S}_\mathcal{T}}(\theta';\boldsymbol{\alpha}). \end{aligned} \tag{9}$$

First-order bilevel optimization. To optimize $\boldsymbol{\alpha}$ in Equation (9), one must compute the hypergradient $\nabla_{\boldsymbol{\alpha}} \mathcal{L}_{\text{ent}}(\boldsymbol{\alpha}, \theta(\boldsymbol{\alpha}))$. By the chain rule, this requires the implicit derivative $\nabla_{\boldsymbol{\alpha}} \theta(\boldsymbol{\alpha})$, which standard approaches obtain via the implicit function theorem:

$$\nabla_{\boldsymbol{\alpha}} \theta(\boldsymbol{\alpha}) = -\left[\nabla_{\theta}^{2} \mathcal{L}_{\text{tr}}(\theta; \boldsymbol{\alpha})\right]^{-1} \nabla_{\boldsymbol{\alpha}} \nabla_{\theta} \mathcal{L}_{\text{tr}}(\theta; \boldsymbol{\alpha}).$$

This involves second-order derivatives and a Hessian inversion of $\nabla_{\theta}^{2} \mathcal{L}_{\text{tr}}$, which is computationally infeasible for LLMs with billions of parameters.

To circumvent this, we adopt a first-order approximation inspired by BOME (Liu et al., 2022). The key idea is to reformulate the bilevel problem as a constrained optimization over the joint variable $u = (\boldsymbol{\alpha}, \theta)$, where the lower-level optimality is enforced through a residual constraint $q(\boldsymbol{\alpha}, \theta) := \mathcal{L}_{\text{tr}}(\theta; \boldsymbol{\alpha}) - \mathcal{L}_{\text{tr}}^{*}(\boldsymbol{\alpha}) \leq 0$, with $\mathcal{L}_{\text{tr}}^{*}(\boldsymbol{\alpha}) := \min_{\theta'} \mathcal{L}_{\text{tr}}(\theta'; \boldsymbol{\alpha})$ denoting the value function. Since the exact value function is intractable, we approximate it by running T inner gradient steps from the current θ to obtain a surrogate $\tilde{\theta}$, and construct a plug-in residual $\hat{q} = \mathcal{L}_{\text{tr}}(\theta; \boldsymbol{\alpha}) - \mathcal{L}_{\text{tr}}(\tilde{\theta}; \boldsymbol{\alpha})$. The joint update direction combines the upper-level entropy gradient $\nabla_{u} \mathcal{L}_{\text{ent}}$ with the residual gradient $\nabla_{u} \hat{q}$, requiring only first-order derivatives throughout. The complete algorithm is given in Appendix A.

5 Experiment

5.1 Experimental Setup

Models. We evaluate our methods using four open-source large language models: Llama-3.1-Tulu-8B (Lambert et al., 2024), Vicuna-7B-v1.5 (Chiang et al., 2023), Olmo 2-7B (OLMo et al., 2024), Mistral-7B (Jiang et al., 2023a), with detailed information provided in Appendix B. We deliberately focus on these four models because each has undergone alignment using either RLHF (Ouyang et al., 2022) or DPO (Rafailov et al., 2023) and exhibits notably poor calibration. Specifically, Vicuna-7B is aligned via RLHF, while Mistral-7B, Olmo 2-7B, and Llama-3.1-Tulu-8B are aligned via DPO.

Baselines and ablations. We compare CALM against four external calibration methods and two ablations of CALM itself. The external baselines are:

- **DPO/RLHF baseline**: The original preference-aligned model without calibration.
- **Temperature scaling (TS)** (Guo et al., 2017): Post-hoc scaling of logits by a single temperature T fit on a validation set; the post-hoc method CALM generalizes.
- **Label smoothing (LS)** (Müller et al., 2019): Soft labels assigning $1 - \varepsilon$ to the correct class and spreading ε over the rest.
- **Calibration-aware fine-tuning (CFT)** (Xiao et al., 2025a): Fine-tunes with a calibration-aware objective to restore calibration lost during alignment; as the closest training-time baseline, it receives a per-model tuning budget comparable to CALM's (Section 5.3).

Two ablations isolate the contribution of CALM's bilevel structure:

- **Regularization**: The entropy objective applied as a flat single-level penalty with a grid-tuned fixed weight; an ablation of CALM that removes the bilevel constraint.
- **Iterate**: Alternates task and calibration updates without the coupling (implicit/residual) term; an ablation of CALM that removes the coupling.

Evaluation settings. We evaluate all methods across two task families, multiple-choice QA (MCQA) and general open-ended QA, each under a domain-shifted (OOD) and a matched in-domain (ID) condition. In both families the OOD setting is our *primary* evaluation, as it reflects the more realistic and challenging scenario, while the ID setting serves as a complementary reference.

Multiple-choice QA (MCQA).

- **Out-of-domain (OOD)**: The model is trained on the Alpaca dataset (Taori et al., 2023) (a general question-answering dataset) and tested on MCQA[3], a multiple-

[3] https://huggingface.co/datasets/aaronwzl/mcqa_calibration_dataset.

choice benchmark consisting of MMLU, MedMCQA, OpenBookQA, and ARC-Challenge (3,000 test samples). This setting tests the model's ability to *transfer* calibration from one domain to another, highlighting zero-shot generalization.
- **In-domain (ID)**: Both training and evaluation use the MCQA dataset (2,000 training / 1,000 Calibration / 2,000 test samples). This setting represents the best-case scenario where calibration and evaluation distributions are matched.

General open-ended QA.

- **Out-of-domain (OOD)**: Each method is trained on one open-ended QA dataset and evaluated on the other, using PopQA (Mallen et al., 2023) and TriviaQA (Joshi et al., 2017). The generative answer format is held fixed across training and evaluation, so the only source of shift is the underlying question distribution rather than the output format. This is a direct test of cross-domain calibration transfer under free-form generation.

Note that Label Smoothing is evaluated only in the ID setting of the MCQA family, since MCQA-OOD trains on Alpaca, which lacks the ground-truth MCQA labels that LS requires. Both open-ended QA benchmarks provide answer supervision, so LS is evaluated in the general-QA family as well.

Evaluation metrics. For the MCQA family we report two calibration metrics: confidence Expected Calibration Error (conf-ECE) (Guo et al., 2017) and class-wise Expected Calibration Error (cw-ECE). For the general open-ended QA family, where answers are free-form and the label set is unbounded so the ECE variants above no longer apply, we report Sem-ECE (Wang et al., 2026), which draws multiple samples from the model, clusters semantically equivalent generations, and treats each cluster's empirical frequency as the model's confidence, thereby overcoming the strict-format limitations of standard ECE in open-ended settings. All calibration metrics are lower-is-better. Across both families we additionally report Accuracy (higher is better) to confirm that calibration gains do not come at the expense of language understanding. Formal definitions are provided in Appendix B.

Implementation details. We employ the Quantized Low Rank (QLoRA) technique (Dettmers et al., 2023) to fine-tune all models with rank = 64, QLoRA scaling parameter $\alpha = 32$, and `float16` precision on NVIDIA A100 (40G) GPUs. Training is conducted for 5 epochs with a batch size of 16 and a weight decay of 0.02. Hyperparameters for each method are selected via grid search based on the lowest ECE while maintaining reasonable accuracy ($> 80\%$ of baseline); detailed configurations, grids, and training-cost comparisons are provided in Appendix C.

5.2 Multiple-Choice QA Results

OOD calibration (Primary Evaluation). Table 1 summarizes calibration quality and task accuracy across four aligned models in the OOD setting.

CALM achieves the best conf-ECE on three of four models while preserving accuracy. On Llama-3.1 it reduces conf-ECE by 41.1% (0.1784 $\rightarrow$ 0.1050, -2.35 pp accuracy); on Vicuna-7B it attains the lowest conf-ECE (0.0380) and cw-ECE (0.0576) while marginally *improving* accuracy; on OLMo-2-7B the best cw-ECE (0.0912, -0.75 pp); and on Mistral-7B the best conf-ECE (0.0822) with accuracy (51.60%) above Regularization (44.20%) and Iterate (25.80%).

Competing methods exhibit severe accuracy–calibration trade-offs. Iterate occasionally achieves strong ECE (e.g., Mistral cw-ECE 0.0518) but suffers catastrophic accuracy collapse (59.80%$\rightarrow$25.80%). Regularization preserves accuracy on some models but severely degrades it on others (Mistral: 44.20%). CFT restores calibration relative to the base model on some backbones but leaves conf-ECE elevated ($\geq$ 0.17 on all four). Temperature Scaling preserves accuracy by construction but its OOD calibration is inconsistent—it *worsens* conf-ECE on LLaMA (0.1784$\rightarrow$0.2513). Notably, Regularization is exactly CALM's entropy objective as a grid-tuned single-level penalty, so its failure to match CALM at any grid point shows that no fixed penalty weight recovers what the bilevel constraint finds automatically—the coupling, not entropy alone, is necessary.

Table 1: Performance comparison across four aligned models in the OOD MCQA setting. Best results (per model, per metric) are in **bold**. Lower is better for conf-ECE/cw-ECE (↓), and higher is better for Accuracy (↑).

Method	Llama-3.1			Vicuna-7B			OLMo-2-7B			Mistral-7B		
	conf-ECE↓	cw-ECE↓	Acc.↑	conf-ECE↓	cw-ECE↓	Acc.↑	conf-ECE↓	cw-ECE↓	Acc.↑	conf-ECE↓	cw-ECE↓	Acc.↑
DPO / RLHF	0.1784	0.0931	**66.80%**	0.0581	0.0646	44.70%	0.1115	0.0901	**59.70%**	0.3351	0.1717	**59.80%**
Temp. Scale.	0.2513	0.1291	**66.80%**	0.0953	0.0765	44.70%	**0.0608**	0.0812	**59.70%**	0.2735	0.1418	**59.80%**
Regularization	0.1431	0.0900	66.50%	0.0477	0.0591	**45.70%**	0.1394	0.1036	58.00%	0.2100	0.1247	44.20%
Iterate	0.2438	0.1317	39.55%	0.3270	0.1754	31.50%	0.2612	0.1420	58.90%	0.0923	**0.0518**	25.80%
CFT	0.1707	0.0925	64.05%	0.1750	0.1130	42.15%	0.1813	0.1033	62.50%	0.2043	0.1372	49.30%
CALM (Ours)	**0.1050**	**0.0621**	64.45%	**0.0380**	**0.0576**	44.95%	0.0878	**0.0912**	58.95%	**0.0822**	0.0881	51.60%

Table 2: Performance comparison across four aligned models in the ID MCQA setting. Best results (per model, per metric) are in **bold**.

Method	Llama-3.1			Vicuna-7B			OLMo-2-7B			Mistral-7B		
	conf-ECE↓	cw-ECE↓	Acc.↑	conf-ECE↓	cw-ECE↓	Acc.↑	conf-ECE↓	cw-ECE↓	Acc.↑	conf-ECE↓	cw-ECE↓	Acc.↑
DPO / RLHF	0.1784	0.0931	66.80%	0.0581	0.0646	44.70%	0.1115	0.0901	59.70%	0.3351	0.1717	59.80%
Temp. Scale.	**0.0224**	**0.0389**	66.80%	**0.0294**	**0.0559**	44.70%	0.0587	0.0878	59.70%	**0.0560**	**0.0474**	59.80%
Label Smooth.	0.0495	0.0550	65.90%	0.0649	0.0646	42.65%	**0.0510**	**0.0517**	62.10%	0.1898	0.1198	53.45%
Regularization	0.2132	0.1095	**74.20%**	0.2470	0.1378	53.35%	0.2355	0.1212	**69.90%**	0.2730	0.1393	**67.00%**
Iterate	0.2681	0.1348	67.50%	0.4041	0.2110	48.95%	0.2970	0.1494	66.15%	0.3129	0.1627	58.85%
CFT	0.2408	0.1213	71.70%	0.2389	0.1351	**56.40%**	0.2374	0.1225	69.35%	0.2450	0.1272	**67.15%**
CALM (Ours)	0.1752	0.0910	**68.10%**	0.0376	0.0536	45.65%	0.0862	0.0765	62.20%	0.2525	0.1328	60.85%

CALM provides the strongest accuracy–calibration trade-off. Unlike methods that achieve low ECE only via utility collapse (Iterate) or fail to generalize across domains (TS-OOD), CALM reduces calibration error while retaining the highest accuracy among training-based methods. Two design choices explain this: the bilevel constraint prevents calibration from degrading task performance, and the entropy objective does not fit a particular test distribution, suiting the OOD setting.

Robustness across seeds. A three-seed study (Appendix D.1, Table 11) confirms that CALM's mean conf-ECE improves over the uncalibrated baseline on all four models. Vicuna-7B and OLMo-2-7B are highly stable (conf-ECE std ≤ 0.002); Llama-3.1 and Mistral-7B show larger run-to-run variance, so their single-run margins in Table 1 should be read with more caution.

ID calibration (Complementary Reference). Table 2 presents the in-domain results, where both training and evaluation use MCQA data. Temperature Scaling achieves the best ECE on most models in the ID setting, as expected since it directly optimizes calibration on the matched distribution, but it cannot improve accuracy. Label Smoothing provides strong calibration on Olmo (ECE 0.0510, cw-ECE 0.0517) but degrades both calibration and accuracy on Mistral. CFT attains high accuracy on Vicuna (56.40%) and Mistral (67.15%) but its conf-ECE stays near 0.24 on all four models. CALM (ID) is the only method that *consistently improves accuracy* over the baseline across all four models (+1.3, +1.0, +2.5, +1.1 percentage points), while simultaneously improving calibration. Regularization (ID) achieves the highest accuracy gains (+5 to +10 percentage points) but at the cost of worsened calibration.

Scalar vs. vector parameterization. An ablation on Mistral-7B (OOD) shows CALM's per-vocabulary $\mathbf{d}_y, \mathbf{l}_y$ are far better calibrated than a collapsed scalar (learned-TS) variant (conf-ECE 0.0822 vs. 0.3140); details in Appendix D.2.

Table 3: Cross-domain calibration under generative distribution shift (train PopQA, test TriviaQA). Values are Sem-ECE (lower is better); best per column in **bold**. CFT is shown at its original fixed $\alpha = 0.5$ and at the best per-model $\alpha \in \{0.1, 0.5, 0.9\}$ (Llama-3.1 0.9, Vicuna-7B 0.1, OLMo-2-7B 0.1, Mistral-7B 0.5).

Method	Llama-3.1	Vicuna-7B	OLMo-2-7B	Mistral-7B	Mean
DPO / RLHF	0.0512	0.1421	**0.0716**	0.0736	0.0846
Reg	0.0745	**0.0762**	0.1444	0.1110	0.1015
CFT ($\alpha = 0.5$)	0.1117	0.1532	0.1800	0.0870	0.1330
CFT (best-α)	0.0754	0.1215	0.1597	0.0870	0.1109
LS (collapsed)	0.9167	1.0000	0.3049	0.3633	0.6462
CALM (Ours)	**0.0490**	0.1105	0.0724	**0.0366**	**0.0671**

5.3 General Open-Ended QA Results

We now move beyond the multiple-choice format to free-form generation, where LLM calibration matters most in practice. Table 3 reports the cross-domain generative test: each method is trained on one QA dataset and evaluated on the other (train PopQA, test TriviaQA), with the generative answer format held constant so that question-distribution shift is the only variable. Calibration is measured with Sem-ECE (Wang et al., 2026) (defined in the metrics paragraph above and Appendix B).

CALM transfers best under generative domain shift. CALM achieves the lowest mean Sem-ECE, 0.0671, improving on DPO/RLHF (0.0846) and every training-time baseline. It is best on Llama-3.1 and Mistral-7B and within 0.001 of the best on OLMo-2-7B; no baseline stays competitive across all four backbones. Label smoothing collapses under the shift (Sem-ECE up to 1.0) and is reported only as a cautionary baseline. These gains preserve answer quality: CALM matches or exceeds DPO/RLHF accuracy on TriviaQA, so lower Sem-ECE reflects better-calibrated confidence over predictions of equal or higher quality.

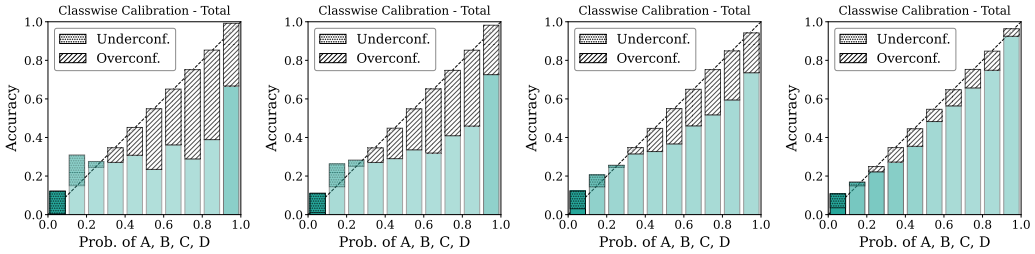
A tuning-budget-matched comparison with CFT. In the generative setting the most relevant training-time baseline, CFT, must be adapted from its native MCQA design; to avoid under-serving it, we give CFT a per-model sweep over its calibration weight $\alpha \in \{0.1, 0.5, 0.9\}$, comparable in size to CALM's learning-rate grid (Appendix C). Tuning helps: the best-α CFT improves mean Sem-ECE from 0.1330 (fixed $\alpha = 0.5$) to 0.1109. However, even with a matched tuning budget, best-α CFT still trails both CALM (0.0671) and the uncalibrated baseline (0.0846) in this setting. The comparison is therefore fair and the conclusion is unchanged: CALM is the strongest method under generative domain shift.

5.4 Plot Analysis

Figure 1 shows classwise reliability diagrams on Mistral-7B (OOD) for the DPO baseline, Temperature Scaling, CFT, and CALM. The DPO baseline is clearly overconfident in the medium- and high-confidence regions, where empirical correctness falls well below the predicted probability; Temperature Scaling and CFT reduce this only partially, with high-confidence bins still below the diagonal; and CALM aligns the bins closely with the diagonal without distorting the overall confidence profile. This confirms that overconfidence is the dominant failure mode and that CALM provides the most robust correction under distribution shift.

5.5 Language Ability Retention

A practical calibration method must also preserve language understanding. We evaluate all calibrated models on five benchmarks—HellaSwag (Zellers et al., 2019), ARC-Easy, ARC-Challenge (Clark et al., 2018), WinoGrande (Sakaguchi et al., 2020), and PIQA (Bisk et al., 2020)—with the LM Evaluation Harness (Gao et al., 2024) (zero-shot, log-likelihood scoring); Table 4 reports average accuracy under both settings. In the ID setting, CALM



(a) DPO baseline (b) Temp. Scale. (c) CFT (d) CALM (Ours)

Figure 1: Classwise reliability diagrams on Mistral-7B (OOD): (a) DPO baseline, (b) Temperature Scaling, (c) CFT, (d) CALM. Each panel plots predicted probability (x-axis) against observed accuracy (y-axis) over ten bins; the diagonal is perfect calibration and bars below it indicate overconfidence. Diagrams for all methods and all four models (OOD and ID) are in Appendix D.

Table 4: Language ability retention: average accuracy (%) over five standard benchmarks. Values in parentheses denote absolute changes relative to the baseline. Best accuracy-preserving result per model is **bold**.

Setting	Method	Llama-3.1	Vicuna-7B	OLMo-2-7B	Mistral-7B
—	Baseline	72.82	64.87	71.33	69.02
ID	CALM (Ours)	**72.97** (+0.2)	64.92 (+0.1)	71.34 (+0.0)	**69.60** (+0.6)
	Iterate	72.96 (+0.1)	64.39 (-0.5)	**71.76** (+0.4)	69.79 (+0.8)
	Regularization	73.03 (+0.2)	**64.98** (+0.1)	**71.76** (+0.4)	71.72 (+2.7)
	Label Smooth.	58.72 (-14.1)	51.30 (-13.6)	55.83 (-15.5)	55.40 (-13.6)
OOD	CALM (Ours)	**66.99** (-5.8)	**64.59** (-0.3)	**71.29** (-0.0)	**66.94** (-2.1)
	Iterate	56.44 (-16.4)	44.31 (-20.6)	57.65 (-13.7)	34.64 (-34.4)
	Regularization	70.13 (-2.7)	65.27 (+0.4)	69.33 (-2.0)	60.61 (-8.4)
	Label Smooth.	67.13 (-5.7)	62.94 (-1.9)	61.12 (-10.2)	69.43 (+0.4)

($+0.2$ avg.), Iterate ($+0.2$), and Regularization ($+0.9$) all maintain or slightly improve over the baseline, confirming that when calibration and evaluation distributions are aligned, calibration training does not degrade general capabilities. Under the harder OOD setting, methods diverge: CALM is the most stable training-based method (near-zero change on OLMo-2-7B and Vicuna-7B, minor drops on Llama-3.1 and Mistral-7B), whereas Iterate regresses catastrophically (-21.3 avg.) and Label Smoothing degrades severely. Consequently, CALM has the smallest OOD-to-ID gap (2.3 on average, vs. Iterate 21.5 and Regularization 4.0), reflecting the bilevel framework's robustness to distribution shift.

6 Conclusion

In this work, we proposed a bilevel framework for calibration-oriented loss design in aligned LLMs. Unlike post-hoc temperature scaling, which is dataset-dependent, our method improves calibration at training time by optimizing temperature-like loss hyperparameters with an entropy-based upper-level objective, together with a BOME-inspired first-order optimization scheme that makes the framework practical for LLM fine-tuning. Empirically, across both multiple-choice and open-ended generative question answering, our method achieves strong calibration gains across multiple aligned models while maintaining a better calibration–utility trade-off than competing baselines with especially clear advantages in out-of-domain settings where alternative methods often become unstable or suffer substantial utility loss.

Acknowledgments

We thank the area chair and the anonymous reviewers for their comments and suggestions. This work was supported in part by the NUS School of Computing Start-up Grant to Jiancong Xiao.

References

Momin Abbas, Yi Zhou, Parikshit Ram, Nathalie Baracaldo, Horst Samulowitz, Theodoros Salonidis, and Tianyi Chen. Enhancing in-context learning via linear probe calibration. In *Proceedings of The 27th International Conference on Artificial Intelligence and Statistics*, pp. 307–315, 2024.

Anthropic. The claude 3 model family: Opus, sonnet, haiku. *Claude-3 Model Card*, 2024.

Mohammad Gheshlaghi Azar, Zhaohan Daniel Guo, Bilal Piot, Remi Munos, Mark Rowland, Michal Valko, and Daniele Calandriello. A general theoretical paradigm to understand learning from human preferences. In *International Conference on Artificial Intelligence and Statistics*, pp. 4447–4455. PMLR, 2024.

Yonatan Bisk, Rowan Zellers, Jianfeng Gao, Yejin Choi, et al. Piqa: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 7432–7439, 2020.

Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, et al. Sparks of artificial general intelligence: Early experiments with gpt-4. *arXiv preprint arXiv:2303.12712*, 2023.

Collin Burns, Haotian Ye, Dan Klein, and Jacob Steinhardt. Discovering latent knowledge in language models without supervision. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=ETKGuby0hcs.

Jonathan D Chang, Wenhao Zhan, Owen Oertell, Kianté Brantley, Dipendra Misra, Jason D Lee, and Wen Sun. Dataset reset policy optimization for rlhf. *arXiv preprint arXiv:2404.08495*, 2024.

Yangyi Chen, Lifan Yuan, Ganqu Cui, Zhiyuan Liu, and Heng Ji. A close look into the calibration of pre-trained language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1343–1367, 2023.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E Gonzalez, et al. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality. *See https://vicuna. lmsys. org*, 2023.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113, 2023.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. Qlora: Efficient finetuning of quantized llms. *Advances in neural information processing systems*, 36:10088–10115, 2023.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang,

and Andy Zou. The language model evaluation harness, July 2024. URL https://zenodo.org/records/12608602.

Alexey Gorbatovski, Boris Shaposhnikov, Alexey Malakhov, Nikita Surnachev, Yaroslav Aksenov, Ian Maksimov, Nikita Balagansky, and Daniil Gavrilov. Learn your reference model for real good alignment. *arXiv preprint arXiv:2404.09656*, 2024.

Sebastian Gruber and Florian Buettner. Better uncertainty calibration via proper scores for classification and beyond. *Advances in Neural Information Processing Systems*, 35:8618–8632, 2022.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International conference on machine learning*, pp. 1321–1330. PMLR, 2017.

Chirag Gupta and Aaditya Ramdas. Distribution-free calibration guarantees for histogram binning without sample splitting. In *International conference on machine learning*, pp. 3942–3952. PMLR, 2021.

Chirag Gupta, Aleksandr Podkopaev, and Aaditya Ramdas. Distribution-free binary classification: prediction sets, confidence intervals and calibration. *Advances in Neural Information Processing Systems*, 33:3711–3723, 2020.

Haixia Han, Tingyun Li, Shisong Chen, Jie Shi, Chengyu Du, Yanghua Xiao, Jiaqing Liang, and Xin Lin. Enhancing confidence expression in large language models through learning from past experience. *arXiv preprint arXiv:2404.10315*, 2024.

Zhixiong Han, Yaru Hao, Li Dong, Yutao Sun, and Furu Wei. Prototypical calibration for few-shot learning of language models. In *The Eleventh International Conference on Learning Representations*, 2023.

Guande He, Peng Cui, Jianfei Chen, Wenbo Hu, and Jun Zhu. Investigating uncertainty calibration of aligned language models under the multiple-choice setting. *arXiv preprint arXiv:2310.11732*, 2023.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021.

Jerry Huang, Peng Lu, and Qiuhao Zeng. Calibrated language models and how to find them with label smoothing. In *International Conference on Machine Learning*, pp. 25593–25612. PMLR, 2025.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. Mistral 7b. *arXiv preprint arXiv:2310.06825*, 2023a.

Zhengbao Jiang, Jun Araki, Haibo Ding, and Graham Neubig. 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.

Zhongtao Jiang, Yuanzhe Zhang, Cao Liu, Jun Zhao, and Kang Liu. Generative calibration for in-context learning. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 2312–2333, 2023b.

Ruochen Jin, Bojian Hou, Jiancong Xiao, Weijie J Su, and Li Shen. Fine-tuning attention modules only: Enhancing weight disentanglement in task arithmetic. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=dj0TktJcVI.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In Regina Barzilay and Min-Yen Kan (eds.), *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1601–1611, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-1147. URL https://aclanthology.org/P17-1147/.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, et al. Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*, 2022.

Archit Karandikar, Nicholas Cain, Dustin Tran, Balaji Lakshminarayanan, Jonathon Shlens, Michael C Mozer, and Becca Roelofs. Soft calibration objectives for neural networks. *Advances in Neural Information Processing Systems*, 34:29768–29779, 2021.

Ranganath Krishnan and Omesh Tickoo. Improving model calibration with accuracy versus uncertainty optimization. *Advances in Neural Information Processing Systems*, 33: 18237–18248, 2020.

Aviral Kumar, Sunita Sarawagi, and Ujjwal Jain. Trainable calibration measures for neural networks from kernel mean embeddings. In *International Conference on Machine Learning*, pp. 2805–2814. PMLR, 2018.

Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, et al. Tulu 3: Pushing frontiers in open language model post-training. *arXiv preprint arXiv:2411.15124*, 2024.

Jixuan Leng, Chengsong Huang, Banghua Zhu, and Jiaxin Huang. Taming overconfidence in LLMs: Reward calibration in RLHF. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=l0tg0jzsdL.

Ziniu Li, Tian Xu, and Yang Yu. Policy optimization in rlhf: The impact of out-of-preference data. *arXiv preprint arXiv:2312.10584*, 2023.

Ziniu Li, Tian Xu, Yushun Zhang, Zhihang Lin, Yang Yu, Ruoyu Sun, and Zhi-Quan Luo. ReMax: A simple, effective, and efficient reinforcement learning method for aligning large language models. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 29128–29163. PMLR, 21–27 Jul 2024.

Ziniu Li, Congliang Chen, Tian Xu, Zeyu Qin, Jiancong Xiao, Zhi-Quan Luo, and Ruoyu Sun. Preserving diversity in supervised fine-tuning of large language models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=NQEe7B7bSw.

Stephanie Lin, Jacob Hilton, and Owain Evans. Teaching models to express their uncertainty in words. *Transactions on Machine Learning Research*, 2022.

Bo Liu, Mao Ye, Stephen Wright, Peter Stone, and Qiang Liu. Bome! bilevel optimization made easy: A simple first-order approach. *Advances in neural information processing systems*, 35:17248–17262, 2022.

Kaizhao Liu, Qi Long, Zhekun Shi, Weijie J Su, and Jiancong Xiao. Statistical impossibility and possibility of aligning llms with human preferences: From condorcet paradox to nash equilibrium. *arXiv preprint arXiv:2503.10990*, 2025.

Xin Liu, Muhammad Khalifa, and Lu Wang. LitCab: Lightweight language model calibration over short- and long-form responses. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=jH67LHVOIO.

Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. In *Proceedings of the 61st annual meeting of the association for computational linguistics (volume 1: Long papers)*, pp. 9802–9822, 2023.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2018.

Rafael Müller, Simon Kornblith, and Geoffrey E Hinton. When does label smoothing help? *Advances in neural information processing systems*, 32, 2019.

Remi Munos, Michal Valko, Daniele Calandriello, Mohammad Gheshlaghi Azar, Mark Rowland, Zhaohan Daniel Guo, Yunhao Tang, Matthieu Geist, Thomas Mesnard, Côme Fiegel, Andrea Michi, Marco Selvi, Sertan Girgin, Nikola Momchev, Olivier Bachem, Daniel J Mankowitz, Doina Precup, and Bilal Piot. Nash learning from human feedback. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 36743–36768. PMLR, 21–27 Jul 2024.

Parvin Nazari, Bojian Hou, Davoud Ataee Tarzanagh, Li Shen, and George Michailidis. Stochastic regret guarantees for online zeroth-and first-order bilevel optimization. *Advances in Neural Information Processing Systems*, 38:147406–147493, 2025.

Team OLMo, Pete Walsh, Luca Soldaini, Dirk Groeneveld, Kyle Lo, Shane Arora, Akshita Bhagia, Yuling Gu, Shengyi Huang, Matt Jordan, et al. 2 olmo 2 furious. *arXiv preprint arXiv:2501.00656*, 2024.

OpenAI. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.

Ankit Pal, Logesh Kumar Umapathi, and Malaikannan Sankarasubbu. Medmcqa: A large-scale multi-subject multi-choice dataset for medical domain question answering. In *Proceedings of the Conference on Health, Inference, and Learning (CHIL)*, 2022.

Teodora Popordanoska, Raphael Sayer, and Matthew Blaschko. A consistent and differentiable lp canonical calibration error estimator. *Advances in Neural Information Processing Systems*, 35:7933–7946, 2022.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36, 2023.

Rafael Rafailov, Joey Hejna, Ryan Park, and Chelsea Finn. From r to q^*: Your language model is secretly a q-function. *arXiv preprint arXiv:2404.12358*, 2024.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Winogrande: An adversarial winograd schema challenge at scale. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, New York, NY, USA, February 7–12, 2020*, pp. 8732–8740. AAAI Press, 2020.

Thomas Savage, John Wang, Robert Gallo, Abdessalem Boukil, Vishwesh Patel, Seyed Amir Ahmad Safavi-Naini, Ali Soroush, and Jonathan H Chen. Large language model uncertainty proxies: discrimination and calibration for medical diagnosis and treatment. *Journal of the American Medical Informatics Association*, 32(1):139–149, 2025.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *ArXiv preprint*, abs/1707.06347, 2017. URL https://arxiv.org/abs/1707.06347.

Maohao Shen, Subhro Das, Kristjan Greenewald, Prasanna Sattigeri, Gregory W Wornell, and Soumya Ghosh. Thermometer: Towards universal calibration for large language models. In *International Conference on Machine Learning*, pp. 44687–44711. PMLR, 2024.

Zhekun Shi, Kaizhao Liu, Qi Long, Weijie J Su, and Jiancong Xiao. Fundamental limits of game-theoretic llm alignment: Smith consistency and preference matching. *arXiv preprint arXiv:2505.20627*, 2025.

Zeyu Sun, Dogyoon Song, and Alfred Hero. Minimum-risk recalibration of classifiers. *Advances in Neural Information Processing Systems*, 36:69505–69531, 2023.

Yunhao Tang, Daniel Zhaohan Guo, Zeyu Zheng, Daniele Calandriello, Yuan Cao, Eugene Tarassov, Rémi Munos, Bernardo Ávila Pires, Michal Valko, Yong Cheng, et al. Understanding the performance gap between online and offline alignment algorithms. *arXiv preprint arXiv:2405.08448*, 2024.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. Stanford alpaca: An instruction-following llama model. https://github.com/tatsu-lab/stanford_alpaca, 2023.

Davoud Ataee Tarzanagh, Parvin Nazari, Bojian Hou, Li Shen, and Laura Balzano. Online bilevel optimization: Regret analysis of online alternating gradient methods. In *International Conference on Artificial Intelligence and Statistics*, pp. 2854–2862. PMLR, 2024.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.

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

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

Mingzhi Wang, Chengdong Ma, Qizhi Chen, Linjian Meng, Yang Han, Jiancong Xiao, Zhaowei Zhang, Jing Huo, Weijie J Su, and Yaodong Yang. Magnetic preference optimization: Achieving last-iterate convergence for language model alignment. In *The Thirteenth International Conference on Learning Representations*, 2025a. URL https://openreview.net/forum?id=PDnEDS244P.

Zhanliang Wang, Jiancong Xiao, Ruochen Jin, Shu Yang, Bojian Hou, and Li Shen. A semantic-sampling framework for evaluating calibration in open-ended question answering. *arXiv preprint arXiv:2605.08432*, 2026.

Ziming Wang, Zeyu Shi, Haoyi Zhou, Shiqi Gao, Qingyun Sun, and Jianxin Li. Towards objective fine-tuning: How llms' prior knowledge causes potential poor calibration? In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 14830–14853, 2025b.

Jiancong Xiao, Bojian Hou, Zhanliang Wang, Ruochen Jin, Qi Long, Weijie J Su, and Li Shen. Restoring calibration for aligned large language models: A calibration-aware fine-tuning approach. In *International Conference on Machine Learning*, pp. 68364–68390. PMLR, 2025a.

Jiancong Xiao, Ziniu Li, Xingyu Xie, Emily Getzen, Cong Fang, Qi Long, and Weijie J Su. On the algorithmic bias of aligning large language models with rlhf: Preference collapse and matching regularization. *Journal of the American Statistical Association*, 120(552):2154–2164, 2025b.

Jiancong Xiao, Zhekun Shi, Kaizhao Liu, Qi Long, and Weijie J Su. Theoretical tensions in rlhf: Reconciling empirical success with inconsistencies in social choice theory. *arXiv preprint arXiv:2506.12350*, 2025c.

Yuxin Xiao, Paul Pu Liang, Umang Bhatt, Willie Neiswanger, Ruslan Salakhutdinov, and Louis-Philippe Morency. Uncertainty quantification with pre-trained language models: A large-scale empirical analysis. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pp. 7273–7284, 2022.

Tianyang Xu, Shujin Wu, Shizhe Diao, Xiaoze Liu, Xingyao Wang, Yangyi Chen, and Jing Gao. Sayself: Teaching llms to express confidence with self-reflective rationales. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 5985–5998, 2024.

Joy Qiping Yang, Salman Salamatian, Ziteng Sun, Ananda Theertha Suresh, and Ahmad Beirami. Asymptotics of language model alignment. *arXiv preprint arXiv:2404.01730*, 2024.

Chenlu Ye, Wei Xiong, Yuheng Zhang, Nan Jiang, and Tong Zhang. A theoretical analysis of nash learning from human feedback under general kl-regularized preference. *arXiv preprint arXiv:2402.07314*, 2024.

Yaodong Yu, Stephen Bates, Yi Ma, and Michael Jordan. Robust calibration with multi-domain temperature scaling. *Advances in Neural Information Processing Systems*, 35:27510–27523, 2022.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a machine really finish your sentence? In *Proceedings of the 57th annual meeting of the association for computational linguistics*, pp. 4791–4800, 2019.

Shujian Zhang, Chengyue Gong, and Eunsol Choi. Knowing more about questions can help: Improving calibration in question answering. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pp. 1958–1970, 2021.

Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. Calibrate before use: Improving few-shot performance of language models. In *International conference on machine learning*, pp. 12697–12706. PMLR, 2021.

A Algorithm Details

This section provides the detailed derivation and the complete algorithm for the first-order bilevel optimization scheme introduced in Section 4. We use the notation $\mathcal{L}_{\text{tr}}^{\mathcal{S}_\mathcal{T}}(\theta;\boldsymbol{\alpha})$ and $\mathcal{L}_{\text{ent}}^{\mathcal{S}_\mathcal{V}}(\boldsymbol{\alpha},\theta)$ as defined in Equation (8).

A.1 Value Function Reformulation

As discussed in the main text, the standard implicit-differentiation approach to bilevel optimization requires computing $\nabla_{\boldsymbol{\alpha}}\theta(\boldsymbol{\alpha})$ via the implicit function theorem, which involves the Hessian $\nabla_\theta^2\mathcal{L}_{\text{tr}}$ and its inverse. To avoid this, we adopt a value-function-based reformulation (Liu et al., 2022).

We define the lower-level value function

$$\mathcal{L}_{\text{tr}}^\star(\boldsymbol{\alpha}) := \min_\theta \mathcal{L}_{\text{tr}}^{\mathcal{S}_\mathcal{T}}(\theta;\boldsymbol{\alpha}),$$

and the associated optimality residual

$$q(\boldsymbol{\alpha},\theta) := \mathcal{L}_{\text{tr}}^{\mathcal{S}_\mathcal{T}}(\theta;\boldsymbol{\alpha}) - \mathcal{L}_{\text{tr}}^\star(\boldsymbol{\alpha}). \tag{10}$$

Note that $q(\boldsymbol{\alpha},\theta) \geq 0$ by construction, and $q(\boldsymbol{\alpha},\theta) = 0$ if and only if θ is a minimizer of $\mathcal{L}_{\text{tr}}$ for the given $\boldsymbol{\alpha}$. Therefore, the bilevel problem in Equation (8) can be equivalently rewritten as the following constrained optimization over the joint variable $u = (\boldsymbol{\alpha},\theta)$:

$$\min_u \ \mathcal{L}_{\text{ent}}^{\mathcal{S}_\mathcal{V}}(u) \qquad \text{subject to} \qquad q(\boldsymbol{\alpha},\theta) \leq 0. \tag{11}$$

This constrained form makes explicit that each update should both improve the upper-level entropy objective and move the current iterate toward lower-level optimality. Crucially, by Danskin's theorem, $\nabla_{\boldsymbol{\alpha}}\mathcal{L}_{\text{tr}}^\star(\boldsymbol{\alpha}) = \nabla_{\boldsymbol{\alpha}}\mathcal{L}_{\text{tr}}(\theta^*(\boldsymbol{\alpha});\boldsymbol{\alpha})$, which does *not* require computing $\nabla_{\boldsymbol{\alpha}}\theta^*(\boldsymbol{\alpha})$. Hence, the gradient of the residual q with respect to u involves only first-order derivatives.

A.2 Plug-in Residual Approximation

In practice, the exact value function $\mathcal{L}_{\text{tr}}^\star(\boldsymbol{\alpha})$ is intractable because it requires solving the lower-level problem to optimality. At outer iteration k, we instead approximate the lower-level optimum by running T gradient descent steps on $\mathcal{L}_{\text{tr}}$ starting from the current parameter θ_k, while keeping $\boldsymbol{\alpha}_k$ fixed:

$$\theta_k^{(t+1)} = \theta_k^{(t)} - \rho\,\nabla_\theta\mathcal{L}_{\text{tr}}^{\mathcal{S}_\mathcal{T}}(\theta_k^{(t)};\boldsymbol{\alpha}_k), \qquad t = 0,\ldots,T-1,$$

with $\theta_k^{(0)} = \theta_k$. We denote the resulting approximate optimum by $\tilde{\theta}_k := \theta_k^{(T)}$ and define the plug-in residual surrogate

$$\hat{q}_k(\boldsymbol{\alpha},\theta) = \mathcal{L}_{\text{tr}}^{\mathcal{S}_\mathcal{T}}(\theta;\boldsymbol{\alpha}) - \mathcal{L}_{\text{tr}}^{\mathcal{S}_\mathcal{T}}(\tilde{\theta}_k;\boldsymbol{\alpha}_k), \tag{12}$$

where $\tilde{\theta}_k$ is treated as a constant when differentiating. This surrogate satisfies $\hat{q}_k \geq q$ (since $\tilde{\theta}_k$ is not an exact minimizer), providing a relaxation of the exact constraint.

A.3 Joint Update Direction

Using the plug-in residual, we compute two gradient directions:

$$g_{\text{ent}} = \nabla_u\mathcal{L}_{\text{ent}}^{\mathcal{S}_\mathcal{V}}(u_k), \qquad g_q = \nabla_u\hat{q}_k(u_k).$$

The first is the gradient of the upper-level entropy objective, and the second is the gradient of the approximate residual. Following the BOME framework, the joint update direction is constructed to simultaneously decrease $\mathcal{L}_{\text{ent}}$ and the residual $\hat{q}$. Specifically, we set the dynamic barrier

$$\phi_k = \gamma\|g_q\|^2,$$

Algorithm 1 CALM: First-Order Bilevel Optimization for LLM Calibration

Require: Training set $\mathcal{S}_{\mathcal{T}}$, validation inputs $\mathcal{S}_{\mathcal{V}}$, initial parameters $(\boldsymbol{\alpha}_0, \theta_0)$, outer iterations K, inner steps T, learning rates $\eta_{\boldsymbol{\alpha}}, \eta_\theta, \rho$, balancing coefficient γ

for $k = 0, 1, \ldots, K-1$ **do**

  (Inner optimization) Starting from $\theta^{(0)} \leftarrow \theta_k$, run T steps:

$$\theta^{(t+1)} = \theta^{(t)} - \rho \, \nabla_\theta \mathcal{L}_{\text{tr}}^{\mathcal{S}_{\mathcal{T}}}(\theta^{(t)}; \boldsymbol{\alpha}_k), \quad t = 0, \ldots, T-1$$

  Set $\tilde{\theta}_k \leftarrow \theta^{(T)}$

  (Residual surrogate) $\hat{q}_k(\boldsymbol{\alpha}, \theta) = \mathcal{L}_{\text{tr}}^{\mathcal{S}_{\mathcal{T}}}(\theta; \boldsymbol{\alpha}) - \mathcal{L}_{\text{tr}}^{\mathcal{S}_{\mathcal{T}}}(\tilde{\theta}_k; \boldsymbol{\alpha}_k)$

  (Gradients) $g_{\text{ent}} = \nabla_u \mathcal{L}_{\text{ent}}^{\mathcal{S}_{\mathcal{V}}}(u_k), \quad g_q = \nabla_u \hat{q}_k(u_k)$

  (Adaptive weight) $\lambda_k = \max\left(\frac{\gamma \|g_q\|^2 - \langle g_q, g_{\text{ent}} \rangle}{\|g_q\|^2}, \, 0 \right)$

  (Joint direction) $d_k = g_{\text{ent}} + \lambda_k \, g_q$; split $d_k = (d_k^{\boldsymbol{\alpha}}, d_k^{\theta})$

  (Update) $\boldsymbol{\alpha}_{k+1} \leftarrow \boldsymbol{\alpha}_k - \eta_{\boldsymbol{\alpha}} \, d_k^{\boldsymbol{\alpha}}, \quad \theta_{k+1} \leftarrow \theta_k - \eta_\theta \, d_k^{\theta}$

end for

return $(\boldsymbol{\alpha}_K, \theta_K)$

where $\gamma > 0$ is a balancing coefficient, and compute the adaptive weight

$$\lambda_k = \max\left(\frac{\phi_k - \langle g_q, g_{\text{ent}} \rangle}{\|g_q\|^2}, \, 0 \right),$$

with $\lambda_k = 0$ if $\|g_q\| = 0$. The joint search direction is then

$$d_k = g_{\text{ent}} + \lambda_k \, g_q.$$

Intuitively, when the entropy gradient g_{ent} already makes sufficient progress toward reducing the residual (i.e., $\langle g_q, g_{\text{ent}} \rangle \geq \phi_k$), we have $\lambda_k = 0$ and the update follows the entropy gradient alone. Otherwise, $\lambda_k > 0$ adds a correction that steers the update toward lower-level feasibility.

Writing $d_k = (d_k^{\boldsymbol{\alpha}}, d_k^{\theta})$, the parameters are updated as

$$\boldsymbol{\alpha}_{k+1} = \boldsymbol{\alpha}_k - \eta_{\boldsymbol{\alpha}} \, d_k^{\boldsymbol{\alpha}}, \qquad \theta_{k+1} = \theta_k - \eta_\theta \, d_k^{\theta},$$

where $\eta_{\boldsymbol{\alpha}}$ and η_θ are the learning rates for the calibration hyperparameters and model parameters, respectively.

A.4 Complete Algorithm

Algorithm 1 summarizes the complete procedure. Each outer iteration consists of two phases: (i) a short inner optimization that runs T gradient steps on $\mathcal{L}_{\text{tr}}$ to construct the plug-in residual surrogate $\hat{q}_k$, and (ii) a joint update of $\boldsymbol{\alpha}$ and θ along the combined direction d_k that balances entropy reduction with approximate lower-level optimality.

B Models, Datasets, and Metrics

This section provides descriptions of the models, datasets, and evaluation metrics used in this paper. For training configurations and hyperparameter details, see Appendix C.

Models. We evaluate on four open-source LLMs that have undergone preference alignment:

- **LLaMA-3.1-Tulu-8B-DPO** (Lambert et al., 2024)[4]: An instruction-following model from Allen Institute for AI, trained using supervised fine-tuning and DPO. Part of the Tulu 3 family, designed for diverse tasks.

[4] https://huggingface.co/allenai/Llama-3.1-Tulu-3-8B

- **Vicuna-7B-v1.5** (Chiang et al., 2023)[5]: A chat assistant from LMSYS, fine-tuned from LLaMA 2 on approximately 125,000 user-shared conversations via RLHF.
- **OLMo 2-7B-DPO** (OLMo et al., 2024)[6]: A fully open-source model from Allen Institute for AI, trained on the Dolma dataset and fine-tuned using DPO.
- **Mistral 7B-DPO** (Jiang et al., 2023a)[7]: A high-performance model fine-tuned using DPO, known for strong efficiency and benchmark results.

All models are aligned via either RLHF (Vicuna) or DPO (the others), and exhibit notable overconfidence after alignment, making them suitable targets for calibration evaluation.

Datasets. Our calibration experiments use two datasets:

- **Alpaca** (Taori et al., 2023): A question-answering dataset of 52,002 instruction-response pairs. Used for out-of-domain (OOD) training.
- **MCQA**: A multiple-choice question-answering benchmark[8] aggregating questions from four sources:
  - *MMLU* (Hendrycks et al., 2021): 57-subject knowledge and reasoning benchmark covering STEM, humanities, and social sciences.
  - *MedMCQA* (Pal et al., 2022): Medical multiple-choice questions from AIIMS and NEET PG entrance exams.
  - *OpenBookQA* (Mihaylov et al., 2018): Elementary-level science questions requiring common knowledge beyond provided facts.
  - *ARC-Challenge* (Clark et al., 2018): Advanced science questions from grades 3–9, requiring multi-step reasoning.

  The MCQA dataset contains 3,000 training samples, from which we extract 1,000 validation samples, and 2,000 test samples. It is used for both in-domain (ID) training and for evaluating all methods.
- **PopQA** (Mallen et al., 2023): An open-domain, entity-centric question-answering benchmark with short free-form answers. Used in the general open-ended QA family.
- **TriviaQA** (Joshi et al., 2017): A large-scale open-domain trivia question-answering benchmark with free-form answers. Used in the general open-ended QA family.

For the general open-ended QA family, each method is trained on one of {PopQA, TriviaQA} and evaluated on the other, holding the free-form answer format fixed so that only the question distribution shifts, yielding a pure cross-domain calibration-transfer test under generation.

Metrics. We use several evaluation metrics. Accuracy measures the fraction of correctly answered questions (higher is better). For multiple-choice QA we use the two calibration metrics defined below (conf-ECE and cw-ECE); for open-ended generation we use Sem-ECE. All calibration metrics are lower-is-better.

Confidence expected calibration error (conf-ECE) (Guo et al., 2017) measures the alignment between a model's maximum predicted probability and its empirical accuracy:

$$\text{conf-ECE} = \sum_{m=1}^{M} \frac{|B_m|}{N} \left|\text{acc}(B_m) - \text{conf}(B_m)\right|, \tag{13}$$

where $\{B_m\}_{m=1}^{M}$ partitions the predictions into M bins based on the maximum predicted probability, $|B_m|$ is the number of samples in bin m, $\text{acc}(B_m)$ is the empirical accuracy in

[5] https://huggingface.co/lmsys/vicuna-7b-v1.5
[6] https://huggingface.co/allenai/OLMo-2-1124-7B-DPO
[7] https://huggingface.co/princeton-nlp/Mistral-7B-Base-SFT-DPO
[8] https://huggingface.co/datasets/aaronwzl/mcqa_calibration_dataset

that bin, and $\mathrm{conf}(B_m)$ is the average confidence (maximum predicted probability) in that bin.

Class-wise expected calibration error (cw-ECE) extends conf-ECE by evaluating calibration separately for each class:

$$\text{cw-ECE} = \frac{1}{K} \sum_{j=1}^{K} \sum_{m=1}^{M} \frac{|B_{m,j}|}{N} \left| \mathrm{acc}_j(B_{m,j}) - \mathrm{conf}_j(B_{m,j}) \right|, \tag{14}$$

where $B_{m,j}$ is the m-th bin of the j-th class based on the predicted probability for class j, $\mathrm{acc}_j(B_{m,j})$ is the fraction of samples in that bin whose true label is j, and $\mathrm{conf}_j(B_{m,j})$ is the average predicted probability for class j in that bin. While conf-ECE measures calibration only for the most confident class, cw-ECE provides a more comprehensive assessment by checking whether the predicted probability for *every* class matches its empirical frequency.

Semantic expected calibration error (Sem-ECE) (Wang et al., 2026) extends ECE to open-ended generation, where the answer is free-form and the label set is unbounded so conf-ECE/cw-ECE no longer apply. For each query, the model generates multiple samples; semantically equivalent generations are clustered, and the empirical frequency of the selected answer's cluster is taken as the model's confidence. Sem-ECE then bins these cluster-frequency confidences against empirical correctness exactly as in conf-ECE:

$$\text{Sem-ECE} = \sum_{m=1}^{M} \frac{|B_m|}{N} \left| \mathrm{acc}(B_m) - \bar{p}(B_m) \right|, \tag{15}$$

where $\bar{p}(B_m)$ is the average cluster-frequency confidence in bin m and $\mathrm{acc}(B_m)$ is the fraction of samples in that bin whose sampled answer is semantically correct. This overcomes the strict-format limitation of standard ECE, which cannot bin free-form outputs.

B.1 Other Related Work

Post-training and model confidence. Preference-based post-training methods, including PPO-based RLHF (Schulman et al., 2017), DPO and its variants (Azar et al., 2024; Chang et al., 2024; Gorbatovski et al., 2024; Rafailov et al., 2024; Yang et al., 2024), and Nash learning from human feedback (Munos et al., 2024; Shi et al., 2025; Wang et al., 2025a; Xiao et al., 2025c), can substantially alter the output distributions of language models and hence their confidence. Prior work has studied the limitations of PPO (Li et al., 2024), the gap between online and offline alignment algorithms (Tang et al., 2024), and the effects of preference-model specification and distribution shift (Li et al., 2023; Ye et al., 2024). Related studies have also examined diversity during supervised fine-tuning (Li et al., 2025) and model editing as an alternative mechanism for adapting model behavior (Jin et al., 2025). Our work is complementary to these approaches: rather than proposing a new alignment objective, we study how to calibrate models after their predictive distributions have been modified by post-training.

Calibration and recalibration. Temperature scaling is a widely used post-hoc calibration method and is often more effective than label smoothing and related training-time techniques (Guo et al., 2017; Müller et al., 2019). For language models, Zhao et al. (2021) introduced contextual calibration to mitigate biases in few-shot prediction. A broader line of work has investigated the interaction between predictive accuracy and calibration (Kumar et al., 2018; Krishnan & Tickoo, 2020; Karandikar et al., 2021; Popordanoska et al., 2022), as well as proper calibration errors and minimum-risk recalibration (Gruber & Buettner, 2022; Sun et al., 2023). The reliability of calibration evaluation has also received theoretical attention, particularly regarding bias in ECE estimators based on uniform-mass binning (Gupta et al., 2020; Gupta & Ramdas, 2021). In contrast to methods that optimize a single calibration loss or fit a fixed post-hoc transformation, our approach formulates calibration as a bilevel problem: an entropy-based objective suppresses overconfidence, while the task objective preserves predictive discrimination. This interaction allows the calibration parameters and model adaptation to be optimized jointly rather than treating calibration as an isolated post-processing step.

C Implementation Details

This section provides comprehensive implementation details for all calibration methods evaluated in this paper.

C.1 Common Training Configuration

All training-based methods (CALM, Iterate, Regularization, Label Smoothing, CFT) share the following base configuration:

- **Parameter-efficient fine-tuning**: QLoRA (Dettmers et al., 2023) with rank $r = 64$, scaling factor $\alpha = 32$, and dropout 0.01.
- **QLoRA target modules**: All linear layers in the transformer (excluding the `lm_head` output projection), identified automatically via module introspection.
- **Precision**: `float16` mixed precision.
- **Optimizer**: AdamW with $(\beta_1, \beta_2) = (0.9, 0.95)$.
- **Learning rate schedule**: Cosine annealing with linear warmup (2 epochs for bilevel methods, 1 epoch for label smoothing).
- **Hardware**: NVIDIA A100 (40G) GPUs.

C.2 Method-Specific Configurations

CALM (Ours). CALM uses the first-order BOME (Bilevel Optimization Made Easy) (Liu et al., 2022) scheme described in Algorithm 1, which avoids explicit hypergradient, implicit-gradient, and Hessian-inverse computation. The key training parameters are:

- **Bilevel structure**: The lower-level optimizes model parameters (QLoRA weights) to minimize cross-entropy loss, while the upper-level optimizes the per-vocabulary calibration parameters of Section 4—the multiplicative logit adjustment $\mathbf{d}_y \in \mathbb{R}^V$ (equivalently τ_y^{-1}) and the additive logit adjustment $\mathbf{l}_y \in \mathbb{R}^V$—to maximize predictive entropy on held-out inputs subject to a lower-level task-optimality constraint. Together these generalize the single global temperature T of standard temperature scaling, which is recovered when the $\mathbf{d}_y$ are tied.
- **Inner optimization steps**: 20 batches per epoch for the lower-level surrogate $\hat{x}$.
- **BOME coefficient**: $\eta = 0.5$ for the bilevel descent direction $d = \nabla f + \mathrm{ReLU}\left(\eta - \frac{\nabla f \cdot \nabla g}{\|\nabla g\|^2}\right) \nabla g$.
- **Epochs**: 5 (OOD) / 5 (ID).
- **Batch size**: 16.
- **Weight decay**: 0.02.
- **Parameter clipping**: All trainable parameters clipped to $[-10, 10]$.

Temperature scaling. Temperature scaling is a post-hoc method that does not modify any model weights. A single scalar temperature T is optimized to minimize the negative log-likelihood on a validation set:

- **Optimization**: Bounded scalar minimization via `scipy.optimize.minimize_scalar` with L-BFGS-B method.
- **Search bounds**: $T \in [0.1, 10.0]$.
- **Tolerance**: 10^{-4}.
- **Data split**: 50% of the calibration data for finding T, 50% for evaluation.

Label smoothing. Label smoothing uses adaptive soft labels based on per-bin accuracy statistics:

- **Smoothing strategy**: For each sample, the smoothing parameter ε is determined by the model's confidence bin: $\varepsilon = 1 - \text{bin_accuracy}(\text{confidence})$. The correct class receives probability bin_accuracy, and the remaining probability is distributed uniformly.
- **Learning rate**: 10^{-4} (fixed, no grid search).

- **Epochs**: 5.
- **Weight decay**: 0.02.
- **Note**: Only evaluated in the ID setting, since it requires ground-truth MCQA labels for computing bin accuracies. The OOD variant using synthetic MCQA from Alpaca is invalid, as the correct answer is always option A with trivial distractors, leading to accuracy collapse.

Calibration-aware fine-tuning (CFT). CFT (Xiao et al., 2025a) augments the supervised loss with a calibration-aware term of the form $\mathcal{L} = (1-\alpha)\,\mathcal{L}_{\text{CE}} + \alpha \cdot c \cdot \mathcal{L}_{\text{cal}}$, where α is the calibration weight, c a fixed scale, and $\mathcal{L}_{\text{cal}}$ a per-bin objective that drives predicted confidence toward empirical bin accuracy.

- **Calibration weight**: α, swept in the general open-ended QA setting (Table 5); fixed at $\alpha = 0.5$ for the MCQA experiments.
- **Calibration scale**: $c = 10^{-3}$ (fixed).
- **Learning rate**: 10^{-4}. **Epochs**: 5. **Weight decay**: 0.02.
- **Architecture**: Standard QLoRA fine-tuning (no `Logit_2level` wrapper).

Regularization. The regularization baseline augments the standard cross-entropy loss with a confidence penalty term:

- **Loss**: $\mathcal{L} = \mathcal{L}_{\text{CE}}(\mathbf{y}, \hat{\mathbf{y}}) - \alpha \cdot \mathcal{L}_{\text{CE}}(\hat{\mathbf{y}}, \hat{\mathbf{y}})$, where $\hat{\mathbf{y}}$ denotes the model's predicted labels. The second term encourages higher entropy (lower confidence) in predictions.
- **Epochs**: 30 (OOD) / 5 (ID).
- **Weight decay**: 0.02.
- **Architecture**: Standard QLoRA fine-tuning without the `Logit_2level` wrapper (no multiplicative/additive logit adjustments).

Iterate. The Iterate method alternates between lower-level (task) and upper-level (calibration) optimization without computing implicit gradients:

- **Lower level**: 20 gradient steps per epoch on the cross-entropy loss with hyperparameters fixed.
- **Upper level**: 1 gradient step per epoch on the calibration loss $(0.9 \cdot \mathcal{L}_{\text{CE}}(\mathbf{y}, \hat{\mathbf{y}}) - \mathcal{L}_{\text{CE}}(\hat{\mathbf{y}}_{\text{pred}}, \hat{\mathbf{y}}))$ with model parameters frozen.
- **Epochs**: 5 (OOD and ID).
- **Weight decay**: 0.02.
- **Key difference from CALM**: No implicit gradient computation; hyperparameters are updated using direct gradients only.

C.3 Hyperparameter Search Strategy

Table 5 summarizes the hyperparameter search grid for each method. The best configuration per model is selected based on the lowest conf-ECE while maintaining accuracy above 80% of the baseline.

Table 5: Hyperparameter search grids for each method.

Method	Hyperparameters	Search Grid
CALM (OOD)	Lower LR × Upper LR	$\{10^{-3}, 10^{-2}, 10^{-1}\} \times \{10^{-4}, 10^{-5}, 10^{-6}, 10^{-7}\}$
CALM (ID)	Lower LR × Upper LR	$\{10^{-3}, 10^{-2}\} \times \{10^{-5}, 10^{-6}, 10^{-7}\}$
Iterate	Lower LR × Upper LR	$\{10^{-3}, 10^{-2}\} \times \{10^{-4}, 10^{-5}, 10^{-6}, 10^{-7}\}$
Regularization	LR × α	$\{10^{-3}, 10^{-4}\} \times \{10^{-5}, 10^{-6}\}$
Temp. Scaling	Temperature T	$T \in [0.1, 10.0]$ (continuous optimization)
Label Smoothing	Learning rate	10^{-4} (fixed)
CFT (generative)	Calibration weight α	$\{0.1, 0.5, 0.9\}$ (lr 10^{-4}, scale 10^{-3} fixed)

C.4 Best Hyperparameters Per Model

Tables 6–8 report the best hyperparameters selected for each model under each method.

Table 6: Best CALM hyperparameters per model (selected by lowest conf-ECE with accuracy $> 80\%$ of baseline).

Setting	Model	Lower LR	Upper LR	conf-ECE	Accuracy
OOD	Llama-3.1	0.01	10^{-4}	0.1050	64.45%
	Vicuna-7B	0.001	10^{-5}	0.0380	44.95%
	OLMo-2-7B	0.001	10^{-4}	0.0878	58.95%
	Mistral-7B	0.01	10^{-6}	0.0822	51.60%
ID	Llama-3.1	0.01	10^{-5}	0.1752	68.10%
	Vicuna-7B	0.01	10^{-5}	0.0376	45.65%
	OLMo-2-7B	0.01	10^{-6}	0.0862	62.20%
	Mistral-7B	0.01	10^{-6}	0.2525	60.85%

Table 7: Best Regularization hyperparameters per model.

Setting	Model	LR	α	conf-ECE	Accuracy
OOD	Llama-3.1	10^{-4}	10^{-5}	0.1431	66.50%
	Vicuna-7B	10^{-4}	10^{-6}	0.0477	45.70%
	OLMo-2-7B	10^{-4}	10^{-6}	0.1394	58.00%
	Mistral-7B	10^{-3}	10^{-5}	0.2100	44.20%
ID	Llama-3.1	10^{-4}	10^{-5}	0.2132	74.20%
	Vicuna-7B	10^{-4}	10^{-5}	0.2470	53.35%
	OLMo-2-7B	10^{-4}	10^{-5}	0.2355	69.90%
	Mistral-7B	10^{-4}	10^{-5}	0.2730	67.00%

Table 8: Optimal temperature values for Temperature Scaling.

Setting	Model	Optimal T	conf-ECE	Accuracy
ID	Llama-3.1	2.1828	0.0224	66.80%
	Vicuna-7B	1.4170	0.0294	44.70%
	OLMo-2-7B	1.9804	0.0587	59.70%
	Mistral-7B	6.4410	0.0560	59.80%
OOD	Llama-3.1	0.5316	0.2513	66.80%
	Vicuna-7B	0.8468	0.0953	44.70%
	OLMo-2-7B	1.4835	0.0608	59.70%
	Mistral-7B	2.0007	0.2735	59.80%

C.5 Robustness and Ablation Protocols

Multi-seed runs. For the robustness study (Table 11), we retrain CALM in the OOD MCQA setting with three random seeds at each model's selected hyperparameters (Table 6) and report the mean and standard deviation of conf-ECE, cw-ECE, and accuracy. All other settings are held fixed.

Scalar vs. vector ablation. To isolate the effect of the per-vocabulary parameterization, we re-run CALM on Mistral (OOD) with $\mathbf{d}_y$ and $\mathbf{l}_y$ collapsed to scalars (size 1), recovering a learned temperature-scaling-style adjustment; all other settings match the full CALM run.

C.6 Language Ability Evaluation Details

To evaluate language ability retention (Section 5.5), we use the LM Evaluation Harness (Gao et al., 2024) (v0.4.12) with the following configuration:

- **Benchmarks**: HellaSwag (Zellers et al., 2019), ARC-Easy, ARC-Challenge (Clark et al., 2018), WinoGrande (Sakaguchi et al., 2020), PIQA (Bisk et al., 2020).
- **Evaluation mode**: Zero-shot, log-likelihood scoring. For each example, the model scores candidate answers by computing the log-probability of the answer text conditioned on the context. The candidate with the highest log-likelihood is selected.
- **Metric**: Accuracy (fraction of correctly answered examples). For HellaSwag, ARC-Easy, ARC-Challenge, and PIQA, we use length-normalized accuracy (`acc_norm`); for WinoGrande, we use standard accuracy (`acc`).
- **Precision**: `float16`.
- **Batch size**: Automatically determined based on available GPU memory.

C.7 Dataset Details

Training Data.

- **OOD setting**: Alpaca (Taori et al., 2023) dataset (52,002 instruction-response pairs). We use the first 48,002 samples for training, the next 2,000 for validation, and the last 2,000 for testing.
- **ID setting**: MCQA calibration dataset[9] containing 4-way multiple choice questions drawn from MMLU, MedMCQA, OpenBookQA, and ARC-Challenge. We use the `calibration_train` split (3,000 training / 1,000 validation) and the `test_1` split (2,000 samples) for testing.
- **Open-ended QA**: PopQA (Mallen et al., 2023) and TriviaQA (Joshi et al., 2017). Each method is trained on one dataset and evaluated on the other under a fixed free-form answer format. For Sem-ECE we draw 20 samples per question over 300 evaluation questions, cluster semantically equivalent generations, and use cluster frequency as confidence, following Wang et al. (2026).

Model Details. Table 9 summarizes the four models used in our experiments.

Table 9: Model specifications.

Model	Parameters	Vocabulary Size	Alignment
Llama-3.1-Tulu-3-8B-DPO	8B	128,256	DPO
Vicuna-7B-v1.5	7B	32,000	RLHF
OLMo-2-1124-7B-DPO	7B	100,352	DPO
Mistral-7B-Instruct-DPO	7B	32,000	DPO

C.8 Training Cost Comparison

Table 10 reports the approximate training time and peak GPU memory for each calibration method on a 7–8B parameter model using a single NVIDIA A100 (40G) GPU. Temperature Scaling requires no training and has negligible cost. Among training-based methods, CALM is the most expensive due to the bilevel structure (inner optimization loop and model surrogate), but remains practical: a full training run completes within a few hours on a single GPU. Regularization and Label Smoothing use standard single-level fine-tuning and are correspondingly cheaper. The Iterate method is the fastest training-based method since it performs only a few alternating gradient steps per epoch without maintaining a model copy.

The higher cost of CALM relative to single-level methods is attributable to two factors: (i) at each outer iteration, $T = 20$ inner gradient steps are performed on a surrogate model

[9] https://huggingface.co/datasets/aaronwzl/mcqa_calibration_dataset

Table 10: Approximate training cost per model on a single NVIDIA A100 (40G) GPU. Training time is for a complete run with the default number of epochs. Peak memory includes the quantized model, optimizer states, and any surrogate copies.

Method	Epochs	Training Time	Peak GPU Memory
Temperature Scaling	—	<1 min	∼15 GB
Iterate	5	∼5 min	∼20 GB
Label Smoothing	5	∼10 min	∼20 GB
Regularization	5	∼35 min	∼20 GB
CALM (Ours)	5	∼2 hr	∼38 GB

to approximate the lower-level solution, and (ii) the surrogate model $\tilde{\theta}$ must be maintained in memory alongside the primary model θ. We note that this overhead is a one-time training cost; at inference time, CALM introduces no additional latency compared to standard models.

D Additional Results

D.1 Multi-Seed Robustness (OOD MCQA)

Because the single-run margins in Table 1 could in principle be affected by random variation, we repeat OOD-MCQA training for CALM with three random seeds at each model's selected configuration (Table 6) and report the mean $\pm$ standard deviation in Table 11. Vicuna-7B and OLMo-2-7B are highly stable (conf-ECE std ≤ 0.002), confirming that their single-run margins are reliable. Llama-3.1 and Mistral-7B exhibit larger run-to-run variance; the single-run values in Table 1 fall within the observed seed range, but we report the full distribution for transparency and caution that these two margins are less tight than the point estimates suggest. Even under this conservative accounting, the mean CALM conf-ECE improves over the uncalibrated baseline on all four models (e.g., Mistral-7B $0.335 \rightarrow 0.149$, Llama-3.1 $0.178 \rightarrow 0.110$).

Table 11: Multi-seed robustness of CALM in the OOD MCQA setting (mean $\pm$ std over 3 seeds). Vicuna-7B/OLMo-2-7B are highly stable; Llama-3.1/Mistral-7B show larger variance. Baseline conf-ECE is shown for reference.

Model	conf-ECE ↓	cw-ECE ↓	Accuracy (%) ↑	Baseline conf-ECE
Llama-3.1	0.110 ± 0.085	0.089 ± 0.023	61.9 ± 3.2	0.1784
Vicuna-7B	0.040 ± 0.002	0.058 ± 0.001	45.0 ± 0.1	0.0581
OLMo-2-7B	0.089 ± 0.002	0.091 ± 0.001	59.0 ± 0.1	0.1115
Mistral-7B	0.149 ± 0.121	0.105 ± 0.044	51.1 ± 0.7	0.3351

D.2 Scalar vs. Vector Calibration Parameters

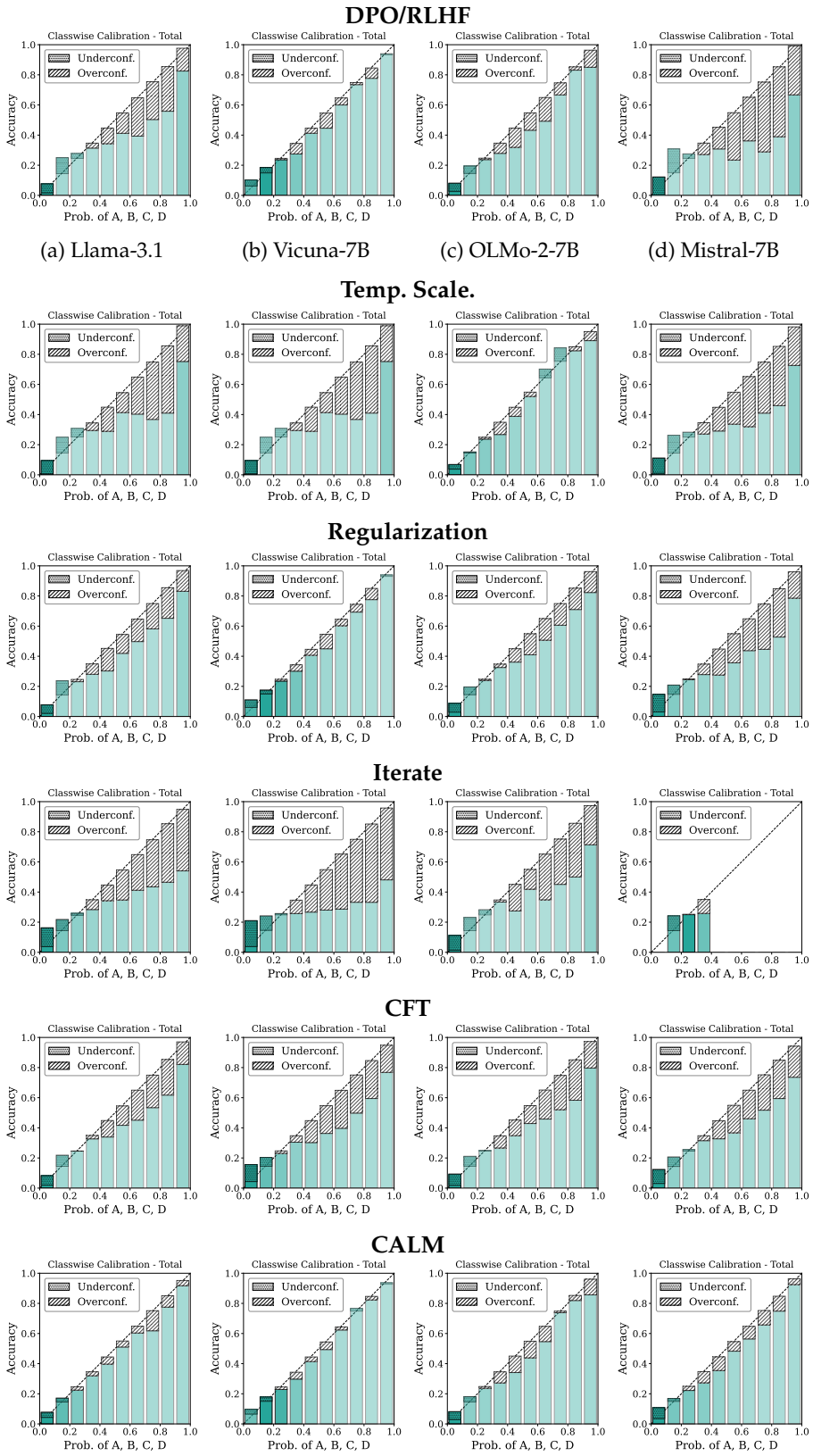
CALM generalizes the single scalar temperature of TS to per-vocabulary multiplicative ($\mathbf{d}_y$) and additive ($\mathbf{l}_y$) logit adjustments. To isolate the contribution of this vector parameterization, we re-run CALM on Mistral-7B (OOD) with $\mathbf{d}_y, \mathbf{l}_y$ collapsed to scalars, which recovers a learned temperature-scaling-style adjustment. As shown in Table 12, the vector form is dramatically better calibrated (conf-ECE 0.0822 vs. 0.3140; cw-ECE 0.0881 vs. 0.1618) at a small accuracy cost (51.6% vs. 59.9%). Miscalibration in aligned LLMs is asymmetric across answer options: a scalar can only rescale overall sharpness, whereas per-vocabulary parameters can redistribute confidence between options.

This section presents the complete set of classwise (Total) reliability diagrams for all four models and all methods. Figure 2 covers the OOD MCQA setting; Figures 3 and 4 cover the ID MCQA setting (split across methods for readability). Across settings, the DPO/RLHF baselines are clearly overconfident; Temperature Scaling and CFT reduce miscalibration

Table 12: Scalar vs. vector calibration parameters for CALM on Mistral-7B (OOD). Lower is better for conf-ECE/cw-ECE.

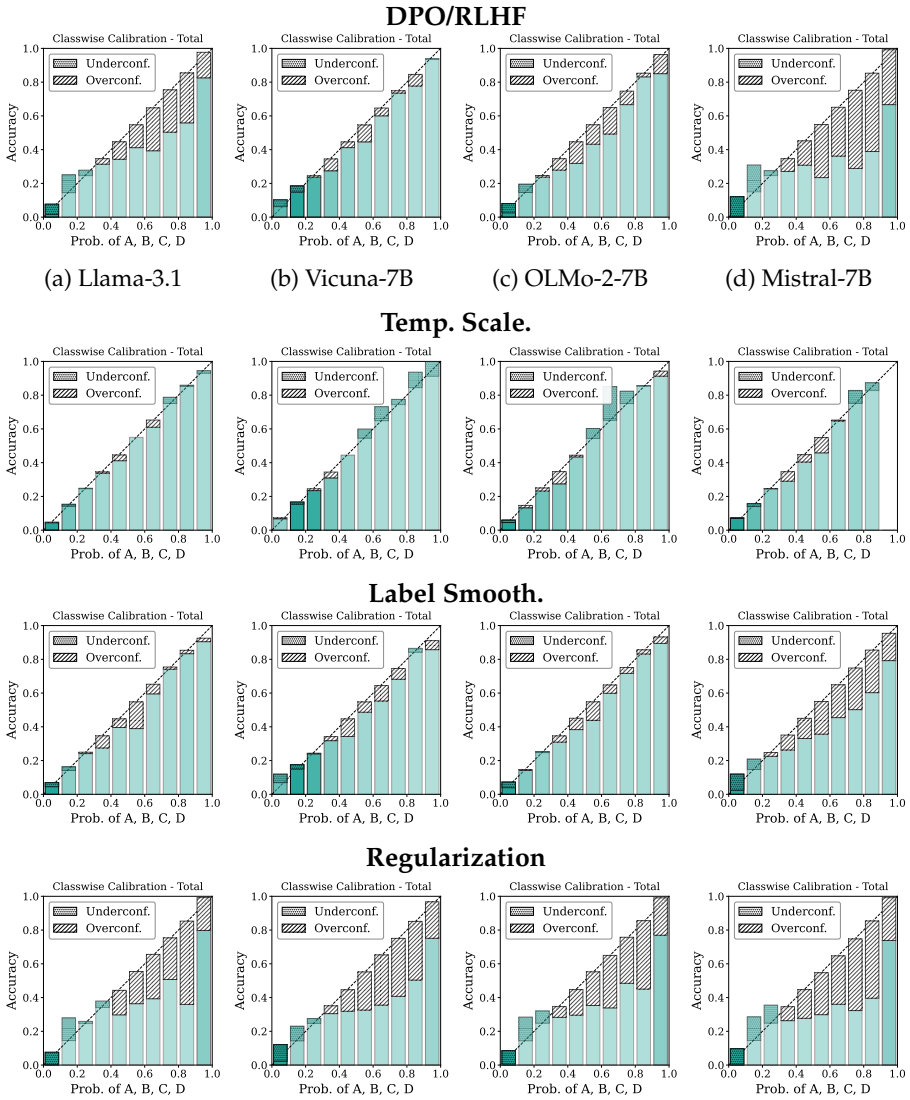
Parameterization	conf-ECE ↓	cw-ECE ↓	Accuracy (%) ↑
Scalar (learned TS)	0.3140	0.1618	59.9
Vector $\mathbf{d}_y, \mathbf{l}_y$ (CALM)	**0.0822**	**0.0881**	51.6

only partially and inconsistently (for Llama-3.1, TS even increases it); Regularization and Iterate help to varying degrees but can distort the curve or collapse accuracy (e.g., Iterate on Vicuna-7B); and CALM produces the most consistent alignment with the diagonal across models and settings.



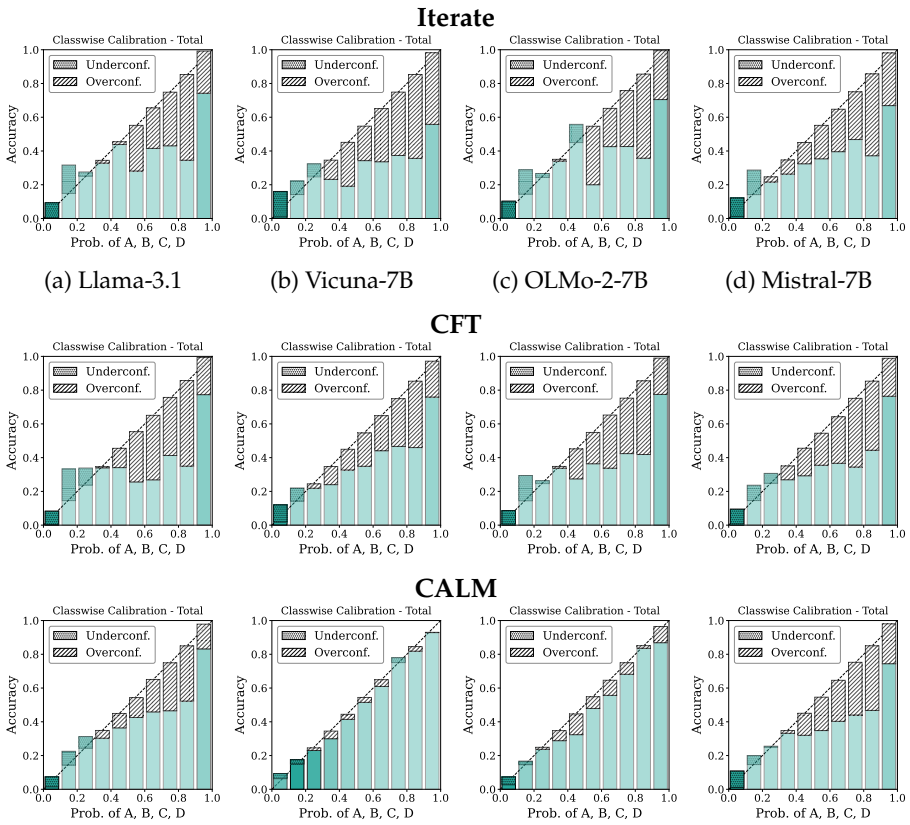
(a) Llama-3.1 (b) Vicuna-7B (c) OLMo-2-7B (d) Mistral-7B

Figure 2: Classwise (Total) reliability diagrams in the **OOD MCQA setting**. Each block is a method (bold header); within a block the four panels are Llama-3.1, Vicuna-7B, OLMo-2-7B, and Mistral-7B (left to right, labeled in the top block). The dashed diagonal is perfect calibration; bars below it indicate overconfidence.



(a) Llama-3.1 (b) Vicuna-7B (c) OLMo-2-7B (d) Mistral-7B

Figure 3: Classwise (Total) reliability diagrams in the **ID MCQA setting**, part 1: DPO/RLHF, Temperature Scaling, Label Smoothing, and Regularization. Within each block the four panels are Llama-3.1, Vicuna-7B, OLMo-2-7B, Mistral-7B (left to right). Remaining methods in Figure 4.



(a) Llama-3.1 (b) Vicuna-7B (c) OLMo-2-7B (d) Mistral-7B

Figure 4: Classwise (Total) reliability diagrams in the **ID MCQA setting**, part 2: Iterate, CFT, and CALM. Panels within each block are ordered as in Figure 3.