CALIBRATING LLMS BY REDUCING OVERCONFIDENCE VIA INSTANCE-ADAPTIVE SMOOTHING

Anonymous authors
Paper under double-blind review

ABSTRACT

Large language models (LLMs) play a growing role in real world decisions, emphasizing a need to ensure accurate reflection of their uncertainty to minimize risk. While the uncertainty calibration of neural networks precedes LLMs, the uniqueness of the setting leads to observations that cannot directly be addressed through existing methods, such as the emergence of severe overconfidence past the base LLM pre-training phase. While label smoothing (LS) have been suggested as a possible remedy due to its ability to mitigate overconfidence, approaches targeted to traditional machine learning settings can be ill-suited to the demands of LLMs, for example multiple training and alignment stages. We identify why this may fail to consider specific nuances unique to LLMs, in particular how the decomposition of the traditional label smoothing loss fails to eliminate overconfidence during misclassifications, leading to possible degradations in the generative abilities of models that are essential for further alignment and tuning. To this end, we introduce ***Ho**listic **S**moothing **Hi**nges (***HOSHI*), an instance-adaptive smoothing approach during supervised fine-tuning to enable calibration while retaining strong generative capability. HOSHI acts as an adjustment to the learning objective that can mitigate previous concerns and enable the stable training of language models that remain calibrated at each stage. Results after different stages demonstrates the effectiveness of our approach, highlighting possible avenues towards better uncertainty-aware LLMs that are more interpretable and effective for public use.

1 INTRODUCTION

Progress in building models that follow natural language instructions (Ouyang et al., 2022; Sanh et al., 2022; Chung et al., 2024) stems through the use of LLMs pre-trained on large amounts of data as well as high-quality datasets that enable them to learn to interact in a human-like manner (Bach et al., 2022; Wang et al., 2022; 2023). However, such models have demonstrated a propensity for over-confidence in their predictions (Figure 1), eliciting concerns over their use in more high-stakes settings (Zhao et al., 2021; Jiang et al., 2021; Xiong et al., 2024; Xiao et al., 2022; Chen et al., 2023). Such observations are not new with respect to neural networks, which have consistently been shown to suffer from overconfident predictions and over-estimate the likelihood of their correctness (Guo et al., 2017; Naeini et al., 2015; Minderer et al., 2021; Joy et al., 2023; Zhang et al., 2018). To improve this, methods such as temperature scaling (Guo et al., 2017) and label smoothing (Szegedy et al., 2016; Müller et al., 2019) have been proposed as solutions with varying effectiveness, spurring additional work in ensuring that predictions and confidence remain matching (Lin et al., 2017; Mukhoti et al., 2020; Pereyra et al., 2017; Liu et al., 2022; Kull et al., 2019). More recently, ensuring the calibration of large foundation models (Huang et al., 2022) has earned increasing attention due to the wide-spread diffusion of such models in the general public (Lin et al., 2022a; Tian et al., 2023; Xu et al., 2024; Han et al., 2024a; Abbas et al., 2024; Han et al., 2023; Jiang et al., 2023c).

Calibrating such models has often relied on post-hoc tuning and adaptation of models through the determination of new hyper-parameters that can be used to adjust model outputs (Jiang et al., 2023b; Luo et al., 2025). However, such methods require the use of additional training datasets that may not necessarily determine parameters that are universally usable (across multiple downstream tasks). This raises a question of whether calibration can be accounted for apriori to evaluation. On this front, label smoothing has in particular been popular due to its easy integration to standard training pipelines (Huang et al., 2025; 2026) as well as general interpretability from an optimization perspec-

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

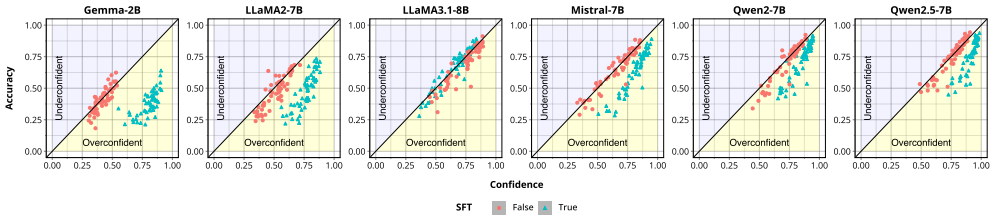


Figure 1: Reliability diagrams of open-sourced models with (blue) and without supervised fine-tuning (red) on MMLU. Confidence (x-axis) and accuracy (y-axis) within pre-defined groups of questions, with evidence of supervised fine-tuning leading to over-confidence.

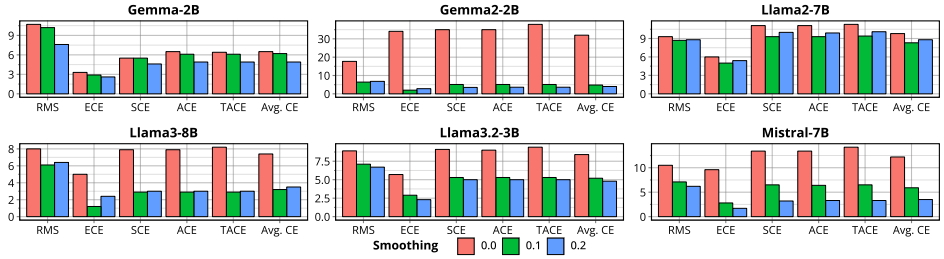


Figure 2: Calibration of different models fine-tuned on `Tulu3Mixture` and tested on MMLU. LS reduces calibration error, i.e the gap between predicted probability and correctness (y-axis where ↓ is always better), across various different binning and sample grouping strategies (x-axis).

tive, with its use during the *supervised fine-tuning* (SFT) process (Wei et al., 2022b; Ouyang et al., 2022; Chung et al., 2024; Dubois et al., 2023) leading to large improvements in calibration with minimal accuracy degradation (Figure 2). However, language model tuning often encompasses multiple stages of adaptation which are interlinked. Some, including *direct preference optimization* (DPO) and *reinforcement learning from verifiable rewards* (RLVR), have emerged as important to ensure models are aligned with human preferences or capable of solving more complex problems that require greater reflection/reasoning. These differ considerably from pre-training and SFT, which rely on a traditional language modeling objective, and thus even small deviations in what is learnt during these preceding stages can adversely impact the models' learning during DPO/RLVR, leading to downstream effects that can be difficult to correct post-hoc.

In this work, we expose this to be a salient issue that has failed to be considered in existing work; through a more explicit analysis of label smoothing, we demonstrate how it incorporates an undesirable error attenuation term that can adversely impact the ability of models to learn after supervised fine-tuning, something that is undetectable using standard uncertainty quantification metrics. Empirical support validates this as problematic, particularly during alignment or reinforcement tuning scenarios that rely heavily on the generative capabilities of models. Understanding this concern, we propose a modified smoothing technique applied during SFT, ***Ho**listic **S**moothing **Hi**nges (***HoSHi***)*, whose suitability is particularly motivated through an entropy analysis of predictions. More specifically, HoSHi uses the logit of the target class to separate the logit vector into two regions demonstrating over-confidence and under-confidence in predictions and applying a different regularization to each.

To validate this, we compare against traditional methods as well as comparable approaches during both the supervised fine-tuning and alignment stages of LLM tuning. Effectiveness is validated on many ends: our models remain better calibrated while retaining performance relative to traditionally trained models, with visible gains against comparable approaches. Overall, we contribute: **(1)** a deeper analysis linking the limitations of label smoothing usage during LLM training and how it can limit effectiveness during later tuning stages, **(2)** a novel smoothing-inspired approach that mitigates these concerns, with its use justified through entropy analysis of predictions, and **(3)** large-scale empirical validation on LLMs, highlighting real-world applicability.

2 Background

Model Calibration (Brier, 1950; Murphy, 1972; DeGroot and Fienberg, 1983) tries to match prediction probabilities yielded for different inputs to expected accuracy. For K-way classification,

let $\mathcal{X} \in \mathbb{R}^D$ and $\mathcal{Y} \in [K]$ indicate the input and label space. Let f be a classifier and $f(\hat{y}|\boldsymbol{x}) = \hat{c}$ be the confidence on $\boldsymbol{x}$, i.e., the maximum of probabilities among K dimensions corresponding to the prediction of f on an example, denoted as $\hat{y}$. A model is *perfectly-calibrated* when

$$P(\hat{y} = y|\hat{c} = c) = c \;\; \forall c \in [0, 1]. \tag{1}$$

Qualitatively, model calibration can be derived as $\mathbb{E}\left[|P(\hat{y} = y|\hat{c} = c) - c|\right]$. One metric that has been widely used for measuring calibration is the expected calibration error (ECE) (Naeini et al., 2015), which is a weighted average of bin-wise mis-calibration. The ECE divides the confidence score of N samples into M uniform confidence bins $\{B_m\}_{m=1}^{M}$ and takes the mean of the gap between accuracy (acc) and confidence (conf) over the bins weighted by the number of samples in the bins.

$$\mathrm{ECE} = \textstyle\sum_{m=1}^{M} \frac{|B_m|}{N} \left|\mathrm{acc}(B_m) - \mathrm{conf}(B_m)\right|. \tag{2}$$

Additional metrics that have been proposed to offer complementary perspectives to standard calibration error measures, enabling a more comprehensive assessment of model confidence alignment.

Calibration has been further adopted to modern generative models by adapting the notion of confidence. One such method, which this work follows, is through using answer probabilities over candidate responses (under the assumption that a ground truth exists) (Jiang et al., 2023b; Luo et al., 2025). Given an input $\boldsymbol{x}$ and candidate responses $\{\boldsymbol{y}^a\}_{a=1}^{A}$, we score each response by its length-normalized log-likelihood under the model,

$$s_a = \tfrac{1}{|\boldsymbol{y}^a|} \textstyle\sum_{t=1}^{|\boldsymbol{y}^a|} \log q(y_t^a|\boldsymbol{x}, \boldsymbol{y}_{<t}^a; \theta)\,,$$

where q is the probability of y_t^a being predicted given $\boldsymbol{x}, \boldsymbol{y}_{<t}^a$, or equivalently by the negative length-normalized perplexity. These candidate-level scores are normalized with a $\mathrm{softmax}$, $\pi_a = \exp(s_a)/\sum_{b=1}^{A}\exp(s_b)$, using the largest π_a as the response and $\max_a \pi_a$ as the confidence. Therefore the resulting sequence likelihoods induce the same confidence/prediction pair used in standard classification calibration metrics.

Label Smoothing (LS) has been previously shown useful for preventing overconfident models (Lukasik et al., 2020; Wei et al., 2022c). Given a model parameterized by θ that applies a function $f(\cdot;\theta)$ on $\boldsymbol{x}$ and $\sigma(\boldsymbol{x};\theta) \in \Delta^K$ be the post-$\mathrm{softmax}$ output on the $(K-1)$-simplex with $\boldsymbol{\ell} \in \mathbb{R}^K$ the pre-$\mathrm{softmax}$ output of the model, i.e. the logits. Models are usually trained by minimizing a cross-entropy (CE) loss on a dataset $\mathcal{D} = \{\boldsymbol{x}_n, y_n\}_{n=1}^{N}$ sampled from an unknown distribution $p(\boldsymbol{x}, y)$ in order to learn the true conditional distribution $p\,(y|\boldsymbol{x})$, computed as

$$\begin{aligned}\mathcal{L}^{\mathrm{CE}}(\theta) &= -\tfrac{1}{N}\textstyle\sum_{n=1}^{N}\sum_{j=1}^{K}\delta_{y_n}^{j}\log q(j|\boldsymbol{x}_n;\theta) \approx -\mathbb{E}_{p(\boldsymbol{x},y)}\left[\sum_{j=1}^{K} p(j|\boldsymbol{x})\log q(j|\boldsymbol{x};\theta)\right] \\ &= \mathbb{E}_{p(\boldsymbol{x},y)}\left[D_{\mathrm{KL}}[p(\cdot|\boldsymbol{x})\|\sigma(\boldsymbol{x};\theta)]\right] + \text{constant} = \mathcal{L}^{\mathrm{CE}}_{p(\boldsymbol{x},y)}(\theta)\,,\end{aligned} \tag{3}$$

where $\delta_i^j = \mathbb{1}[i = j]$. LS mixes with $\boldsymbol{u} = [1/K]^K \in \mathbb{R}^K$ using a smoothing rate $\beta \in [0, 1]$, giving

$$\begin{aligned}\mathcal{L}^{\mathrm{LS}}(\theta) &= -\tfrac{1}{N}\textstyle\sum_{n=1}^{N}\sum_{j=1}^{K}\left[(1-\beta)\,\delta_{y_n}^{j} + \tfrac{\beta}{K}\right]\log q(j|\boldsymbol{x}_n;\theta) \\ &= (1-\beta)\,\mathcal{L}^{\mathrm{CE}}(\theta) + \tfrac{\beta}{N}\textstyle\sum_{n=1}^{N} D_{\mathrm{KL}}[\boldsymbol{u}\|\sigma(\boldsymbol{x}_n;\theta)]\end{aligned} \tag{4}$$

Thus, label smoothing can be understood as a regularization term that encourages a uniform distribution over the output labels, hence preventing it from over-fitting to the training data and encouraging a model to be less confident on all samples by smoothing the true conditional being learned. Application of label smoothing to large models has been previously shown to be beneficial during pre-training of large models (Vaswani et al., 2017), showing downstream task improvements.

LLM supervised fine-tuning (SFT) directly relates to each of these through its underlying training objective, which serves to minimize a cross-entropy objective over (instruction, response) data pairs. This is done by distinguishing a given sequence $\boldsymbol{X}$ of length L into the tokens corresponding to to the instruction and the response, then minimizing a CE objective over the only the response portion. In essence, the learning problem is a V-way classification problem for each element in the target portion of the input sequence (V is the vocabulary size of the model), where the prediction for any specific position is influenced by all previous elements in the sequence. Thus the application of techniques for model calibration and label smoothing are often directly applicable for SFT.

However, this relationship is no longer the case at stages such as DPO and RLHF, which no longer act as direct classification problems, motivating a need to better study the effects that calibration during SFT can bring about and how this may have downstream consequences on training at later stages.

3 On the Benefits and Limitations of Label Smoothing

3.1 The Effects of Label Smoothing

To understand LS from a model calibration perspective, we adopt an optimization-based viewpoint to analyze the implicit constraints imposed by the regularization (Bertsekas, 2005). This perspective reveals how LS modifies the solution space of the model's predictions, encouraging not only smoother output distributions but also a reduction in the model's confidence on any single class.

Definition 3.1 (Logit distance). The **logit distance** vector for $\boldsymbol{x}$, $\boldsymbol{d} \in \mathbb{R}^K$, is defined by

$$\boldsymbol{d}_i = \max_{1 \le j \le K} \boldsymbol{\ell}_j - \boldsymbol{\ell}_i.$$

One way of ensuring that a model does not over-estimate a specific class is to enforce this as a hard constraint, which results in equal logits among all classes and therefore a $\mathrm{softmax}$ output of $\sigma(\boldsymbol{x};\theta) = \boldsymbol{u}$. As such, it is often preferable to enforce this as a soft-penalty function $\mathcal{P} : \mathbb{R}^K \to \mathbb{R}$ into the objective function minimized during training. Using Equation 4, we can relate this soft-penalty to the additional KL-divergence introduced by the label smoothing objective.

Proposition 3.2 (LS Penalty). *A linear penalty (or a Lagrangian term) for the hard constraint $\boldsymbol{d} = \mathbf{0}$ is bounded from above and below by $D_{\mathrm{KL}}[\boldsymbol{u} \| \sigma(\boldsymbol{x};\theta)]$, up to additive constants*

$$D_{\mathrm{KL}}[\boldsymbol{u} \| \sigma(\boldsymbol{x};\theta)] \le \textstyle\sum_{i=1}^K \frac{\boldsymbol{d}_i}{K} \le D_{\mathrm{KL}}[\boldsymbol{u} \| \sigma(\boldsymbol{x};\theta)] + \log K.$$

Thus LS approximately minimizes, for a linear penalty, the constraint $\boldsymbol{d} = \mathbf{0}$, encouraging similarity among the logits for each class to ensure that over-confidence is penalized. Moreover,

Proposition 3.3. *Let $\boldsymbol{z} = \sigma(\boldsymbol{x};\theta) = \mathrm{softmax}(f(\boldsymbol{x};\theta)) \in \Delta^{K-1}$, and suppose that $p(y|\boldsymbol{x},\boldsymbol{z}) = \mathrm{Cat}(\boldsymbol{z})$. Under the independence assumption $p(\boldsymbol{z}|\boldsymbol{x}) = p(\boldsymbol{z})$ and the symmetric Dirichlet prior*

$$p(\boldsymbol{z}) = \mathrm{Dir}(\boldsymbol{\alpha}), \qquad [\boldsymbol{\alpha}]_k = 1 + \tfrac{\beta}{K(1-\beta)} \quad \forall k \in \{1,\dots,K\},$$

minimizing the label smoothing objective with smoothing rate $\beta \in [0,1)$ is equivalent, up to multiplication by the positive constant $(1-\beta)$, to Maximum A Posteriori (MAP) estimation of $\boldsymbol{z}$.

The log-prior term of the above symmetric Dirichlet prior, $\sum_{k=1}^K ([\boldsymbol{\alpha}]_k - 1) \log [\boldsymbol{z}]_k$, favors interior points of the probability simplex, shrinking overly peaked predictions away from the vertices and toward the uniform distribution, providing an interpretation of why LS can suppress over-confidence.

3.2 Identifying the Failure Modes of Label Smoothing

We previously observed how LS can improve general uncertainty calibration by acting as a regularization that penalizes for over-confidence. However, a more fundamental question arises: how does the general nature of LS influence the performance of models? We can first reformulate Equation 4 more explicitly as a composition of multiple cross-entropy losses.

Lemma 3.4 (Cross-Entropy Decomposition). *Denote $\mathrm{CE}(\boldsymbol{q}, \boldsymbol{e}^{\mathsf{t}})$ as the cross-entropy between a probability vector $\boldsymbol{q}$ and a one-hot encoding of a target, $\boldsymbol{e}^{\mathsf{t}}$. Under label smoothing rate β,*

$$\mathrm{CE}_{\mathrm{LS}}(\boldsymbol{q}, \boldsymbol{e}^{\mathsf{t}}; \beta) = \mathrm{CE}(\boldsymbol{q}, (1-\beta)\,\boldsymbol{e}^{\mathsf{t}} + \beta \boldsymbol{u}) = \underbrace{\mathrm{CE}(\boldsymbol{q}, \boldsymbol{e}^{\mathsf{t}})}_{\textit{Target Loss}} + \underbrace{\beta\big(\mathrm{CE}(\boldsymbol{q}, \boldsymbol{u}) - \mathrm{CE}(\boldsymbol{q}, \boldsymbol{e}^{\mathsf{t}})\big)}_{\textit{Smoothing Regularization}}. \quad (5)$$

From here, we can then explicitly express the regularization term in terms of the logits.

Corollary 3.5 (Logit Formulation). *The cross-entropy regularization term can be re-written as*

$$\beta\big(\mathrm{CE}(\boldsymbol{q}, \boldsymbol{u}) - \mathrm{CE}(\boldsymbol{q}, \boldsymbol{e}^{\mathsf{t}})\big) = \beta\Big(\boldsymbol{\ell}_{\mathsf{t}} - \textstyle\sum_{i=1}^K \frac{\boldsymbol{\ell}_i}{K}\Big) = \frac{\beta}{K}\sum_{i:\boldsymbol{\ell}_i < \boldsymbol{\ell}_{\mathsf{t}}} (\boldsymbol{\ell}_{\mathsf{t}} - \boldsymbol{\ell}_i) + \frac{\beta}{K}\sum_{i:\boldsymbol{\ell}_i > \boldsymbol{\ell}_{\mathsf{t}}} (\boldsymbol{\ell}_{\mathsf{t}} - \boldsymbol{\ell}_i). \quad (6)$$

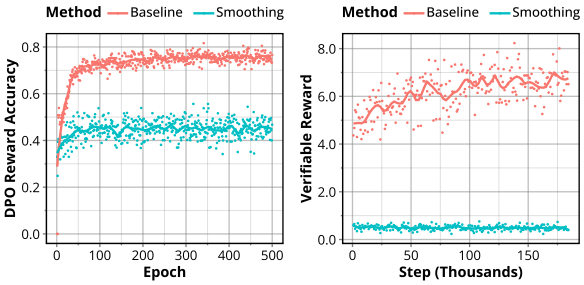
LS affects two distinct regions: (1) a moderation region containing the terms corresponding to classes which are less likely than the ground truth, and (2) a misclassified area, classes where the probability exceeds that of the ground truth. Comparing this to the gradient for LS losses

$$\partial\, \mathrm{CE}_{\mathrm{LS}} / \partial \boldsymbol{\ell}_i = \boldsymbol{q}_i - (1-\beta)\,\mathbb{1}[i = \mathsf{t}] - \beta/K.$$

Relative to standard cross-entropy, every class receives the additional term $-\frac{\beta}{K}$, and the target class also loses part of the original -1 correction through the factor $(1-\beta)$. This presents an issue for (2): when an incorrect class is already too confident, LS weakens the corrective downward update, so mistakes can persist and even be amplified over training. Thus the gap between the ground truth and prioritized incorrect prediction is widened with traditional LS, potentially impeding learning while not being directly detectable by traditional metrics (Xia et al., 2025; Zhou et al., 2025a). Thus despite benefits on general calibration metrics, actual learning may still be compromised.

Remark 3.6. The previous decomposition allows us to understand that LS is precisely beneficial for over-confidence reduction when the misclassified area is generally non-existent; this allows the LS regularization to regulate the weight placed on the target.

This can become an issue during reinforcement fine-tuning of language models (Leike et al., 2018; Batmanghelich et al., 2014; Ouyang et al., 2022; Lambert, 2025; Rafailov et al., 2023; Shao et al., 2024), which rely on the models' generative ability achieved during prior tuning. In Figure 3a, we attempt DPO tuning on instruction-tuned `Llama3-8Bs` and observe clear compromise of the ability to learn. In particular, the DPO reward being observed during training fails to improve with the LS model, highlighting how the smoothing process has significantly degraded the generative abilities of the model to the point that it is unable to learn. Figure 3b presents an additional setting where we further train using RLVR (Lambert et al., 2025), where despite achieving better uncertainty calibration, the model tuned with LS is incapable of learning during the subsequent reinforcement fine-tuning phase, which can critically degrade the ability of a model to function as an agent or interactive tool. Figure 4 further compares accuracy on MMLU as well as target logit rank on a validation set from the SFT dataset. Despite accuracy not suffering from the use of LS, there is a steep decline in the true token appearing in the top Top-5% and Top-10% of logits is observed (this is measured using where the logit for the target token falls in the distribution of the models' prediction). This decrease demonstrates that the predictive or generative ability of the model changes, a signal that later tuning or downstream performance can suffer.



(a) DPO Reward Accuracy. **(b)** RLVR Verifiable Rewards.

Figure 3: `Llama3-8B` RL training following `Tülu3`.

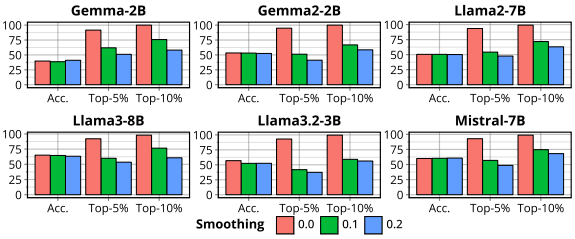


Figure 4: Accuracy and target logit rank after tuning.

4 Overcoming Smoothing-Based Learning Degradation

The above downside of traditional LS demands an alternative to limit this magnification of error. Doing so, however, requires a better understanding of the behavior within the region in question.

4.1 Analysis

We therefore analyze the misclassified region and how the distribution fares within it, starting with an analysis of the logits and its relation to entropy.

Lemma 4.1. *Let $\boldsymbol{C} \in \mathbb{R}^{D\times K}$ be a matrix with $\|\boldsymbol{C}\|_2 = \sigma_C$, and let $\boldsymbol{h} \in \mathbb{R}^D$ be a vector such that each entry of $\boldsymbol{h}$ satisfies $|\boldsymbol{h}_i| \leq \sigma_h$. The upper bound for $\left\|\boldsymbol{C}^\top \boldsymbol{h}\right\|_2$ is:*

$$\left\|\boldsymbol{C}^\top \boldsymbol{h}\right\|_2 \leq \sigma_C \cdot \sigma_h \cdot \sqrt{D}. \tag{7}$$

Thus, the norm of the final logit vector is bounded and depends on both the hidden dimension and number of classes. This enables an entropy-based analysis of the predictions as a function of D and K from which we can build a lower bound on the entropy of the language modeling head/classifier.

Theorem 4.2. *Let* $\rho = \sigma_C \sigma_h$, $\boldsymbol{z} = \boldsymbol{C}^\top \boldsymbol{h}$, $\boldsymbol{p} = \mathrm{softmax}(\boldsymbol{z})$, $\gamma = \exp\left(-\rho\sqrt{\frac{D \cdot K}{K-1}}\right)$, *and* H *be the Shannon entropy function. Then*

$$H(\boldsymbol{p}) \geq \log(1 + (K-1)\,\gamma) + \frac{\rho \cdot \gamma \sqrt{D \cdot K \cdot (K-1)}}{1 + (K-1) \cdot \gamma}. \tag{8}$$

The above indicates that for fixed ρ and D, the minimum entropy lower bound increases with K through the term $(K-1)\,\gamma$, and approaches a logarithmic-in-K scale when $(K-1)\,\gamma$ is large. Thus, when K is large, as is often the case for modern language models with large vocabulary sizes, the prediction cannot become arbitrarily concentrated unless the logit norm also grows sufficiently. This allows us to conjecture that the entropy of the language modeling head output is generally large. This is observed in Figure 5, where we plot the entropy of the normalized misclassified region for 1000 incorrect predictions made by the model. Relative to the maximum entropy possible from the region size, we see that in most cases, entropy remains near the maximum.

The high entropy provides us an avenue to approach this problem; by using the target class as a barrier, we can attempt to directly regularize the two regions separately. This is made possible because the high entropy suggests a relatively flat distribution in the misclassified region. Thus without distinguishing between the different classes in said region, we can apply a more general strategy to regularize the entire region in tandem, while maintaining a more traditional approach for the moderation region, as opposed to selectively penalizing a single logit (the basis of traditional LS).

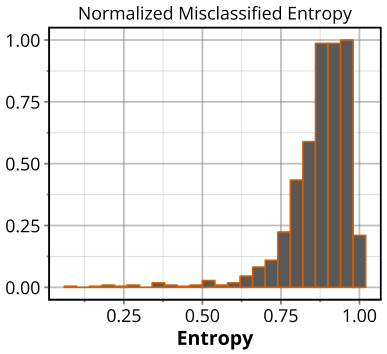


Figure 5: Mis-classification Entropy.

4.2 Overcoming Overconfidence Magnification with Instance-Adaptive Label Smoothing

These observations motivate us to introduce an alternative smoothing approach. In particular, our motivation is to reduce the additional over-confidence error that is added to the logits greater than the target. Instead, we modify LS to reverse the direction of this regularization, defined as

$$\mathrm{CE}_{\mathrm{H}}(\boldsymbol{q}, \boldsymbol{e}^{\mathsf{t}}; \beta) = \mathrm{CE}(\boldsymbol{q}, \boldsymbol{e}^{\mathsf{t}} - \beta\boldsymbol{s} + \beta\boldsymbol{u}) = \mathrm{CE}(\boldsymbol{q}, \boldsymbol{e}^{\mathsf{t}}) + \underbrace{\beta(\mathrm{CE}(\boldsymbol{q}, \boldsymbol{u}) - \mathrm{CE}(\boldsymbol{q}, \boldsymbol{s}))}_{\mathcal{R}_{\mathrm{H}}}, \tag{9}$$

such that $\boldsymbol{s}_i = \frac{1}{Z}$ if $\boldsymbol{\ell}_{\mathsf{t}} \leq \boldsymbol{\ell}_i$ else 0 with $Z = |\{i : \boldsymbol{\ell}_{\mathsf{t}} \leq \boldsymbol{\ell}_i\}|$ the number of logits greater than that of the target class. We call this ***Ho**listic **S**moothing **Hi**nges (***HoSHi***)*, whose goal is to better moderate the $\boldsymbol{\ell}_{\mathsf{t}}$ logit when the model makes mis-classifications by using the target logit as a *hinge* during LS, selectively optimizing the two regions differently based on the target logit and size of each region.

For logits greater than target, the gradients are thus penalized: this change in design enables for a more uniform regularization across both the correct and misclassified samples, effectively eliminating the issue of an error intensification stemming from incorrect samples, while moderating the confidence on the target and classes with smaller logits. Unlike traditional LS, this approach places a penalizing force across all logits which show over-confidence in an incorrect prediction, allowing for the model to balance out greater subtleties simultaneously. Similarly, gradients can be computed (Appendix A.7) for the modified regularization terms as

$$\partial \mathcal{R}_{\mathrm{H}} / \partial \boldsymbol{\ell}_i = \beta\left(\tfrac{1}{Z} - \tfrac{1}{K}\right) \text{ if } \boldsymbol{\ell}_{\mathsf{t}} < \boldsymbol{\ell}_i \text{ else } -\tfrac{\beta}{K} \tag{10}$$

Remark 4.3. Equation 10 shows that only for a mis-classification, i.e. $\max(\boldsymbol{\ell}) \neq \boldsymbol{\ell}_{\mathsf{t}}$, does the new objective differ from traditional LS. Instead of the traditional $-\frac{\beta}{K}$ factor given to each class, those with greater logits will have a positive factor added to the gradient, reducing the error intensification.

A similar approach is MaxSup (Zhou et al., 2025a), which adapt the traditional label smoothing penalty to remove mass from the largest logit (irrespective of correctness) rather than the target logit. However that can be inefficient; in high-entropy settings, the amount of steps that are needed to reduce the logits larger than the target logit remains bounded by the size of the misclassified region Z, as each class in the misclassified region may need to be individually reduced. Alternatively, our approach adapts directly to Z, thus reducing approximately to the MaxSup setting when Z

is small but when Z is large applies an even penalty to all elements, while also ensuring that the under-confident region remains well regulated throughout.

This further bears resemblance to Ren and Sutherland (2025), who demonstrate how SFT induces a squeezing effect during gradient descent and shift confidence between regions, but when uncontrolled naturally shifts to further expand the more confident logits. By spreading out this effect, we can ensure a smoother shift that enables the confidence to be evenly moved from the misclassified region to the target and classes whose logits are being moderated, providing a more desirable learning effect.

5 Experiments and Results

Setup. To validate our approach, we train models across the different language modeling stages. First, we use publicly available pre-trained base model checkpoints for `Mistral-7B` (Jiang et al., 2023a), `Llama3-8B` (Llama Team, 2024) and `Gemma2-2B` (Gemma Team, 2024) and tune on different SFT datasets (Teknium, 2023; Wang et al., 2023). We compare our approach against traditional SFT without LS, with LS and MaxSup, using $\beta = 0.1$ for all approaches requiring it.

We then focus on `Llama3` and conduct two stages of alignment tuning: preference tuning with DPO followed by RLVR. Based on our previous analysis, we only compare our reference model (no LS) against our approaches. We again evaluate models in terms of accuracy and calibration: as an additional step, we also provide results for temperature scaling (Guo et al., 2017) models after both the DPO and RLVR step. We also provide raw performance results on numerous downstream tasks to validate the effectiveness of the introduced approach.

Table 1: Model reliability using expected calibration error (ECE) and root mean square calibration error (RMS) after SFT. We highlight the best scoring method in green and the second best in yellow.

Data	Model	Setting	MMLU			HellaSwag			Arc-Easy			Average		
			Acc.↑	ECE↓	RMS↓	Acc.↑	ECE↓	RMS↓	Acc.↑	ECE↓	RMS↓	Acc.↑	ECE↓	RMS↓
Tülu3Mixture	Mistral	Reference	60.0	0.096	0.105	36.9	0.044	0.085	84.3	0.078	0.135	60.4	0.073	0.108
		Smoothing	60.0	0.034	0.073	37.2	0.025	0.074	83.7	0.041	0.098	60.3	0.033	0.082
		MaxSup	59.8	0.031	0.076	36.8	0.026	0.073	84.1	0.033	0.096	60.2	0.030	0.082
		HoSHi	60.1	0.028	0.071	37.1	0.021	0.067	84.0	0.030	0.094	60.4	**0.026**	**0.077**
	Llama3	Reference	65.1	0.050	0.080	36.1	0.049	0.091	85.7	0.058	0.114	**62.3**	0.052	0.095
		Smoothing	64.6	0.017	0.067	35.6	0.026	0.071	85.6	0.041	0.106	62.0	0.028	0.081
		MaxSup	64.8	0.014	0.064	35.4	0.012	0.061	85.3	0.039	0.102	61.8	0.027	0.076
		HoSHi	64.8	0.012	0.061	35.7	0.025	0.064	85.6	0.035	0.097	62.1	**0.024**	**0.074**
	Gemma2	Reference	53.3	0.341	0.177	27.3	0.082	0.128	75.8	0.086	0.142	**52.1**	0.170	0.149
		Smoothing	53.1	0.029	0.077	27.1	0.048	0.092	75.5	0.031	0.105	51.9	0.036	0.091
		MaxSup	53.2	0.023	0.064	27.1	0.043	0.093	75.2	0.027	0.104	51.8	0.031	0.087
		HoSHi	53.1	0.020	0.064	27.0	0.041	0.087	75.5	0.029	0.101	51.9	**0.030**	**0.084**
OpenHermes	Mistral	Reference	60.2	0.071	0.094	54.6	0.041	0.071	86.7	0.066	0.100	**67.2**	0.059	0.088
		Smoothing	60.2	0.015	0.065	54.9	0.024	0.045	85.7	0.039	0.081	66.9	0.026	0.064
		MaxSup	59.9	0.013	0.062	54.5	0.023	0.045	86.4	0.036	0.080	66.9	0.024	0.062
		HoSHi	60.1	0.014	0.059	55.0	0.021	0.042	85.8	0.036	0.076	67.0	0.024	**0.059**
	Llama3	Reference	65.4	0.038	0.077	55.2	0.063	0.074	88.0	0.065	0.112	69.5	0.055	0.088
		Smoothing	64.6	0.022	0.062	55.4	0.041	0.040	88.0	0.045	0.090	69.3	0.036	0.064
		MaxSup	64.3	0.019	0.060	55.1	0.042	0.040	87.8	0.046	0.092	69.1	0.036	0.064
		HoSHi	65.3	0.016	0.059	55.3	0.038	0.037	88.2	0.041	0.089	**69.6**	**0.032**	**0.062**
	Gemma2	Reference	54.1	0.353	0.180	36.4	0.125	0.143	81.6	0.131	0.175	57.4	0.203	0.166
		Smoothing	54.2	0.019	0.064	36.2	0.080	0.098	81.3	0.039	0.100	57.2	0.046	0.087
		MaxSup	54.1	0.018	0.063	36.4	0.079	0.096	81.2	0.039	0.097	57.2	0.045	0.085
		HoSHi	54.4	0.016	0.063	36.2	0.077	0.096	81.5	0.038	0.096	57.4	**0.044**	0.085

Results. We first evaluate our methods against traditional supervised fine-tuning without and with LS. Table 1 shows these results on MMLU, HellaSwag (Zellers et al., 2019) and Arc-Easy (Bhakthavatsalam et al., 2021). Our results show two important points. First, HoSHi is consistent with other approaches in terms of accuracy, indicating a consistent ability to learn. Second, calibration metrics match those from LS with minimal differences. Thus our methods are able to maintain performance without sacrificing calibration, presenting an ideal scenario for such methods as the SFT process is not impacted through the changed objective. We can further see how HoSHi is the only method to consistently (on average) be either best or second-best in terms of either accuracy or calibration, highlighting consistency across models, tuning datasets and evaluation tasks.

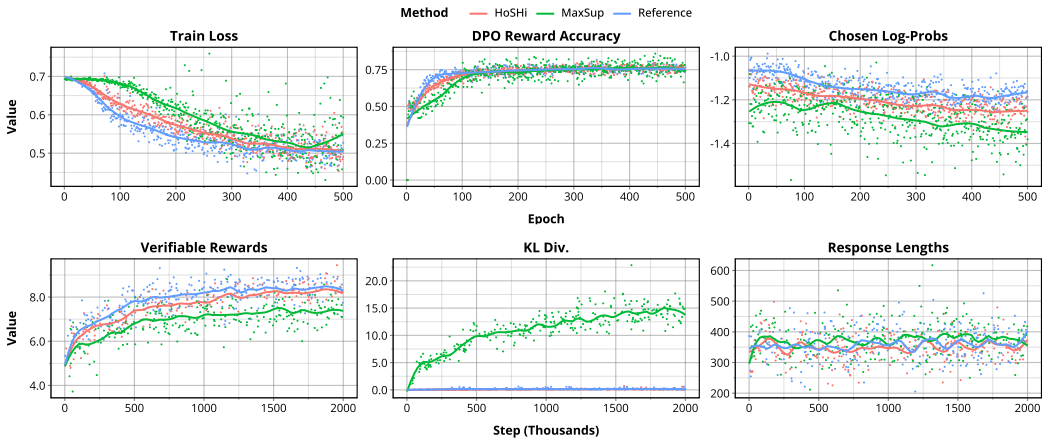


Figure 6: Training plots of `Llama3` during DPO (top) and RLVR (bottom); colors represent the different approaches used for SFT-ing the model on which DPO and RLVR are conducted.

Table 2: Calibration results of models (based on `Llama3-8B`) after alignment stages. Temperature scaling (`TS`) is calibrated on the training set of each dataset prior to evaluation. We **bold** the best method and underline when HoSHi performs second-best.

SFT	Stage	MMLU			HellaSwag			Arc-Easy		
		Acc.↑	ECE↓	RMS↓	Acc.↑	ECE↓	RMS↓	Acc.↑	ECE↓	RMS↓
Reference	DPO	**68.7**	0.065	0.094	37.4	0.061	0.108	88.7	0.066	0.131
	+ TS	**68.6**	0.056	0.081	**37.5**	0.053	0.092	88.4	0.055	0.119
	+ RLVR	68.2	0.081	0.112	37.2	0.075	0.114	**88.9**	0.083	0.145
	+ TS	68.2	0.062	0.085	**37.1**	0.054	0.100	88.7	0.062	0.123
MaxSup	DPO	66.4	0.036	0.076	34.5	0.048	0.088	87.1	0.051	**0.109**
	+ TS	66.2	0.032	0.071	34.4	0.043	0.079	87.3	**0.042**	**0.091**
	+ RLVR	67.5	0.052	0.094	34.7	0.061	0.102	86.9	**0.064**	0.121
	+ TS	67.4	0.043	0.086	34.8	0.060	0.094	87.0	0.059	**0.110**
HoSHi	DPO	68.4	**0.034**	**0.071**	**37.5**	**0.042**	**0.083**	**89.1**	**0.050**	0.112
	+ TS	68.2	**0.031**	**0.066**	37.4	**0.039**	**0.072**	**89.3**	0.043	0.096
	+ RLVR	**68.5**	**0.043**	**0.087**	**37.2**	**0.051**	**0.096**	**88.9**	**0.064**	**0.119**
	+ TS	**68.4**	**0.038**	**0.077**	37.0	**0.044**	**0.087**	**89.0**	**0.053**	0.112

We next verify the ability to learn during DPO and RLVR. Figure 6 plots the learning trajectories during both of these stages from the instruction-tuned checkpoints from the previous step. Both MaxSup as well as HoSHi are able to overcome the pure limitation of LS discussed in §3; in particular, we observe that during both DPO and RLVR, the instruction-tuned models are in fact capable of learning from the alignment process. We next look at how each of these alignment stages can impact calibration; using the same evaluation scheme as we did to measure instruction-tuned model performance, Table 2 demonstrates that both stages of alignment lead to some worsening calibration relative to the instruction-tuned model. While MaxSup sees less performance increases from these alignment stages, HoSHi is generally able to maintain pace relative to the reference while being able to reduce calibration error at nearly all stages. We further observe that adding temperature scaling permits some reversion back to a better calibrated state without accuracy drops, though this does require a small tuning set which may not be feasible in all settings.

6 Additional Discussion and Supporting Evaluations

Biasing towards Answers. One prior observation that has also existed in LLM calibration literature (Zhao et al., 2021) is the bias of models toward specific answers, for example those in specific positions within prompts. To estimate this effect in our models, we provide a synthetic experiment for binary classification, where a model is prompted with a randomly generated 256-character string and asked to answer True or False. Since the input is random, the model should ideally exhibit complete uncertainty about its response, i.e. the probability of both

Table 3: Bias evaluation using post-SFT `Llama3-8B` models.

Method	Acc.↑	ECE↓	Entropy↑
Reference	0.502	0.738	0.523
MaxSup	0.498	0.553	0.895
HoSHi	0.501	**0.512**	**0.974**

responses should be near identical. Table 3 shows how our method is able to retain lower ECE as well as near complete uncertainty as measured by entropy in the distribution over True/False, compared to the reference and MAXSUP which still retain some level of bias.

Open-Ended QA. Previous works have estimated confidence scores in open-ended QA by reformulating free-form QA into a multiple-choice format (Shen et al., 2024a; Kapoor et al., 2024). Specifically, they pose a binary "Yes/No" question to a language model f, asking whether its own generated answer x is correct or incorrect. This approach, commonly referred to as $P(\text{True})$, serves as a well-known baseline and we use it to obtain confidence scores to experiment with open-ended generation on TRUTHFULQA (Lin et al., 2022b). Formally, the confidence score of model f on sample x is defined as $c = p(\text{Yes}|x, f)$. This can then be directly used in standard metrics to obtain calibration errors. Figure 7 presents performance of trained `Llama3` models, with evidence of consistent improvements.

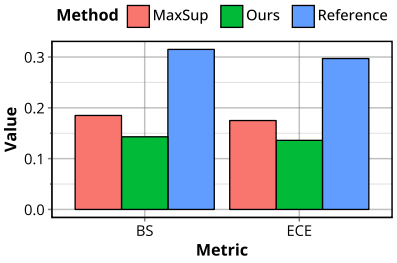


Figure 7: Brier Score and ECE on TRUTHFULQA, with lower better.

Calibration and Performance. Many tasks are not directly measured through classification, making it necessary to understand the relationship with varied performance on different tasks. Table 4 provides a more direct picture of this; across a number of different tasks, we see that a slight drop is observed with both alternatives. However, HOSHI keeps performance generally consistent with the baseline, unlike MAXSUP, highlighting the benefits of this adaptable smoothing approach. This is important, as despite not being directly measurable during tuning, downstream performance remains strong and the final models to remain quite usable in practice. Moreover, calibration of generative models remains important beyond accuracy as false distributional characteristics can lead to significant downstream concerns (Smith et al., 2026) while also posing difficulties in post-hoc processes meant to address similar concerns such as steering (Turner et al., 2023; Zhou et al., 2025b).

Table 4: Downstream task comparison (See Appendix B for details).

Benchmark	Reference		MAXSUP		HOSHI	
	DPO	RLVR	DPO	RLVR	DPO	RLVR
MMLU	68.7	68.3	66.4	67.5	68.4	68.5
PopQA	29.7	30.1	27.4	30.0	28.9	29.8
TruthfulQA	54.3	55.2	52.7	54.1	53.2	54.4
BBH	65.8	66.4	51.5	55.4	62.3	65.5
DROP	62.6	62.2	57.3	59.2	61.4	62.0
MATH	42.6	42.2	37.3	39.7	40.6	42.0
GSM8K	64.7	88.2	53.5	78.3	61.7	83.2
HE	83.2	83.0	71.4	76.4	80.6	81.6
HE+	78.4	79.1	65.4	72.2	72.3	76.4
IFEval	81.6	82.4	74.5	78.3	77.4	81.7
AE	33.7	35.5	30.5	33.5	32.5	34.7
Safety	86.8	84.4	82.4	83.5	83.4	84.1
AVERAGE	62.7	64.8	55.9	60.7	60.3	63.6

Computational Considerations. The large vocabulary sizes of LLMs introduces a number of novel challenges. The final cross-entropy begins to consume a significant amount of memory, as the logits must be materialized to compute the loss. While optimizations have been implemented to manage other model components (Dao et al., 2022; Dao, 2024), managing the footprint of the logit matrix (stacked vectors for each token in the sequence) has motivated efficient CE implementations (Wijmans et al., 2025; Hsu et al., 2024; Meta PyTorch, 2024) which remove unnecessary computations in the non-smoothed loss. However smoothing is more complex; Huang et al. (2025) minimize redundant computations and compute logits in an online manner but for instance-dependent methods this integration is more complex and leaves open an area for bridging effectiveness and efficiency.

7 CONCLUSION

We present **HOSHI**, an adaptive label smoothing (LS) inspired approach that overcomes a limitation of the more traditional variant, namely a persistence of over-ranked or misclassified logits. As motivation, we demonstrate how despite being able to improve measurable uncertainty calibration of LLM, LS constrains models during the stages of alignment, which often requires a high capacity for generating human-readable text, a property difficult to capture directly in calibration metrics. We then demonstrate how the over-ranked logits issue can serve as a root cause of this failure, thus we attempt to mitigate this concern through a modified loss objective. By further analyzing the exact properties of language model prediction heads, we justify how our approach can serve as more effective compared to alternatives, which is then evidenced through empirical results that show improved calibration with minimal changes in performance, all at various stages of LLM training/tuning. Altogether these results highlight possible ways to ensure well-calibrated LLMs throughout training, opening paths to producing models that can be more interpretable and reliable for direct usage in the open world.

8 Limitations, Ethics And Impact

This work focuses on developing a method to maintain calibrated large language models. While our method improves the general confidence calibration of Transformer-based LLMs, the ethical considerations are largely consistent with those of general-purpose language modeling. Potential risks include misuse in generating harmful or misleading content, reinforcement of biases present in training corpora, and environmental concerns arising from large-scale training.

As previously mentioned, computational overhead relative to traditional cross-entropy losses remains a possible area to investigate, especially given the existing large footprint of LLMs. Better understanding how to integrate these within hardware-aware techniques can serve an important way to realize real-world gains in effectiveness and efficiency. Another important direction is also how to enable generative models to sample candidate responses that enable more accurate estimates of uncertainty calibration error, given the intractable output space that remains on this front. Finally, directly training using a calibration-focused loss is also an open direction which this work does not directly explore, but acknowledges as a potential way of overcoming the issues revealed in the preceding sections.

In terms of impact, HOSHI shows promise as a solution for ensuring LLM calibration throughout training, offering a clearer understanding of the sensitivity of models to individual processes playing out at each stage. By offering a better picture of existing limitations and possible alternatives, this approach can help guide the development of more trustworthy and reliable LLMs for open usage. Nevertheless, it remains itself decoupled from the calibration metrics that are used and there remains open space for improved methods on this end, which we believe is open work to be explored both by ourselves as well as others interested in statistical uncertainty and trustworthy machine learning.

9 Reproducibility and AI Use Statement

We include details of our direct implementation in Section 5 and Appendix G. We commit to ensuring that the code will be fully released upon acceptance of this work alongside a camera-ready version. Though some small variability may exist due to differences in the exact computational environment (GPUs, CPUs, etc.), we will further clarify the exact devices and environment used alongside the publicly available code.

The use of large language models was limited to LaTeX formatting as well as anonymization of possibly revealing details within the supplementary material (primarily code). Research design, execution, results collection and presentation were done by authors themselves. All technical content and claims are the responsibility of the authors.

References

M. Abbas, Y. Zhou, P. Ram, N. Baracaldo, H. Samulowitz, T. Salonidis, and T. Chen. Enhancing in-context learning via linear probe calibration. In S. Dasgupta, S. Mandt, and Y. Li, editors, *International Conference on Artificial Intelligence and Statistics, 2-4 May 2024, Palau de Congressos, Valencia, Spain*, volume 238 of *Proceedings of Machine Learning Research*, pages 307–315. PMLR, 2024. URL https://proceedings.mlr.press/v238/abbas24a.html.

S. H. Bach, V. Sanh, Z. X. Yong, A. Webson, C. Raffel, N. V. Nayak, A. Sharma, T. Kim, M. S. Bari, T. Févry, Z. Alyafeai, M. Dey, A. Santilli, Z. Sun, S. Ben-David, C. Xu, G. Chhablani, H. Wang, J. A. Fries, M. S. AlShaibani, S. Sharma, U. Thakker, K. Almubarak, X. Tang, D. R. Radev, M. T. Jiang, and A. M. Rush. Promptsource: An integrated development environment and repository for natural language prompts. In V. Basile, Z. Kozareva, and S. Stajner, editors, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics, ACL 2022 - System Demonstrations, Dublin, Ireland, May 22-27, 2022*, pages 93–104. Association for Computational Linguistics, 2022. doi: 10.18653/V1/2022.ACL-DEMO.9. URL https://doi.org/10.18653/v1/2022.acl-demo.9.

N. K. Batmanghelich, G. T. Quon, A. Kulesza, M. Kellis, P. Golland, and L. Bornn. Diversifying sparsity using variational determinantal point processes, 2014. URL http://arxiv.org/abs/1411.6307.

D. P. Bertsekas. *Dynamic programming and optimal control, 3rd Edition*. Athena Scientific, 2005. ISBN 1886529264. URL https://www.worldcat.org/oclc/314894080.

S. Bhakthavatsalam, D. Khashabi, T. Khot, B. D. Mishra, K. Richardson, A. Sabharwal, C. Schoenick, O. Tafjord, and P. Clark. Think you have solved direct-answer question answering? try arc-da, the direct-answer AI2 reasoning challenge, 2021. URL https://arxiv.org/abs/2102.03315.

G. W. Brier. Verification of forecasts expressed in terms of probability. *Monthly Weather Review*, 78(1): 1–3, 1950. URL https://journals.ametsoc.org/view/journals/mwre/78/1/1520-0493_1950_078_0001_vofeit_2_0_co_2.xml.

S. Chakraborty, J. Qiu, H. Yuan, A. Koppel, D. Manocha, F. Huang, A. S. Bedi, and M. Wang. Maxmin-rlhf: Alignment with diverse human preferences. In R. Salakhutdinov, Z. Kolter, K. A. Heller, A. Weller, N. Oliver, J. Scarlett, and F. Berkenkamp, editors, *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*, volume 235 of *Proceedings of Machine Learning Research*, pages 6116–6135. PMLR / OpenReview.net, 2024. URL https://proceedings.mlr.press/v235/chakraborty24b.html.

M. Chen, J. Tworek, H. Jun, Q. Yuan, H. P. de Oliveira Pinto, J. Kaplan, H. Edwards, Y. Burda, N. Joseph, G. Brockman, A. Ray, R. Puri, G. Krueger, M. Petrov, H. Khlaaf, G. Sastry, P. Mishkin, B. Chan, S. Gray, N. Ryder, M. Pavlov, A. Power, L. Kaiser, M. Bavarian, C. Winter, P. Tillet, F. P. Such, D. Cummings, M. Plappert, F. Chantzis, E. Barnes, A. Herbert-Voss, W. H. Guss, A. Nichol, A. Paino, N. Tezak, J. Tang, I. Babuschkin, S. Balaji, S. Jain, W. Saunders, C. Hesse, A. N. Carr, J. Leike, J. Achiam, V. Misra, E. Morikawa, A. Radford, M. Knight, M. Brundage, M. Murati, K. Mayer, P. Welinder, B. McGrew, D. Amodei, S. McCandlish, I. Sutskever, and W. Zaremba. Evaluating large language models trained on code, 2021. URL https://arxiv.org/abs/2107.03374.

Y. Chen, L. Yuan, G. Cui, Z. Liu, and H. Ji. A close look into the calibration of pre-trained language models. In A. Rogers, J. L. Boyd-Graber, and N. Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 1343–1367. Association for Computational Linguistics, 2023. doi: 10.18653/V1/2023.ACL-LONG.75. URL https://doi.org/10.18653/v1/2023.acl-long.75.

H. W. Chung, L. Hou, S. Longpre, B. Zoph, Y. Tay, W. Fedus, Y. Li, X. Wang, M. Dehghani, S. Brahma, A. Webson, S. S. Gu, Z. Dai, M. Suzgun, X. Chen, A. Chowdhery, A. Castro-Ros, M. Pellat, K. Robinson, D. Valter, S. Narang, G. Mishra, A. Yu, V. Y. Zhao, Y. Huang, A. M. Dai, H. Yu, S. Petrov, E. H. Chi, J. Dean, J. Devlin, A. Roberts, D. Zhou, Q. V. Le, and J. Wei. Scaling instruction-finetuned language models. *J. Mach. Learn. Res.*, 25:70:1–70:53, 2024. URL https://jmlr.org/papers/v25/23-0870.html.

K. Cobbe, V. Kosaraju, M. Bavarian, M. Chen, H. Jun, L. Kaiser, M. Plappert, J. Tworek, J. Hilton, R. Nakano, C. Hesse, and J. Schulman. Training verifiers to solve math word problems, 2021. URL https://arxiv.org/abs/2110.14168.

T. Dao. Flashattention-2: Faster attention with better parallelism and work partitioning. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=mZn2Xyh9Ec.

T. Dao, D. Y. Fu, S. Ermon, A. Rudra, and C. Ré. Flashattention: Fast and memory-efficient exact attention with io-awareness. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022. URL https://openreview.net/forum?id=H4DqfPSibmx.

M. H. DeGroot and S. E. Fienberg. The comparison and evaluation of forecasters. *Journal of the Royal Statistical Society. Series D (The Statistician)*, 32(1):12–22, 1983. URL https://rss.onlinelibrary.wiley.com/doi/abs/10.2307/2987588.

J. Deng, W. Dong, R. Socher, L. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Computer Society Conference on Computer Vision and Pattern*

Recognition (CVPR 2009), 20-25 June 2009, Miami, Florida, USA, pages 248–255. IEEE Computer Society, 2009. doi: 10.1109/CVPR.2009.5206848. URL https://doi.org/10.1109/CVPR.2009.5206848.

D. Dua, Y. Wang, P. Dasigi, G. Stanovsky, S. Singh, and M. Gardner. DROP: A reading comprehension benchmark requiring discrete reasoning over paragraphs. In J. Burstein, C. Doran, and T. Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 2368–2378. Association for Computational Linguistics, 2019. doi: 10.18653/V1/N19-1246. URL https://doi.org/10.18653/v1/n19-1246.

Y. Dubois, C. X. Li, R. Taori, T. Zhang, I. Gulrajani, J. Ba, C. Guestrin, P. Liang, and T. B. Hashimoto. Alpacafarm: A simulation framework for methods that learn from human feedback. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. URL https://openreview.net/forum?id=4hturzLcKX.

Y. Dubois, B. Galambosi, P. Liang, et al. Length-controlled alpacaeval: A simple way to debias automatic evaluators. In *The First Conference on Language Modeling, COLM 2024, Philadelpha, PA, USA, October 7-9, 2024*, 2024. URL https://openreview.net/forum?id=CybBmzWBX0.

Gemma Team. Gemma 2: Improving open language models at a practical size, 2024. URL https://arxiv.org/abs/2408.00118.

C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger. On calibration of modern neural networks. In D. Precup and Y. W. Teh, editors, *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 1321–1330. PMLR, 2017. URL http://proceedings.mlr.press/v70/guo17a.html.

H. Han, T. Li, S. Chen, J. Shi, C. Du, Y. Xiao, J. Liang, and X. Lin. Enhancing confidence expression in large language models through learning from past experience, 2024a. URL https://arxiv.org/abs/2404.10315.

S. Han, K. Rao, A. Ettinger, L. Jiang, B. Y. Lin, N. Lambert, Y. Choi, and N. Dziri. Wildguard: Open one-stop moderation tools for safety risks, jailbreaks, and refusals of llms. In A. Globersons, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. M. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems 37: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024b. URL https://openreview.net/forum?id=Ich4tv4202.

Z. Han, Y. Hao, L. Dong, Y. Sun, and F. Wei. Prototypical calibration for few-shot learning of language models. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL https://openreview.net/forum?id=nUsP9lFADUF.

K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 770–778. IEEE Computer Society, 2016. doi: 10.1109/CVPR.2016.90. URL https://doi.org/10.1109/CVPR.2016.90.

D. Hendrycks, C. Burns, S. Basart, A. Zou, M. Mazeika, D. Song, and J. Steinhardt. Measuring massive multitask language understanding. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021a. URL https://openreview.net/forum?id=d7KBjmI3GmQ.

D. Hendrycks, C. Burns, S. Kadavath, A. Arora, S. Basart, E. Tang, D. Song, and J. Steinhardt. Measuring mathematical problem solving with the MATH dataset. In J. Vanschoren and S. Yeung, editors, *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021, December 2021, virtual*, 2021b. URL https://openreview.net/forum?id=7Bywt2mQsCe.

P. Hsu, Y. Dai, V. Kothapalli, Q. Song, S. Tang, S. Zhu, S. Shimizu, S. Sahni, H. Ning, and Y. Chen. Liger kernel: Efficient triton kernels for LLM training, 2024. URL https://arxiv.org/abs/2410.10989.

J. Huang, J. Xu, X. Shi, et al. Revisiting confidence calibration for misclassification detection in VLMs. In *The Fourteenth International Conference on Learning Representations, ICLR 2026, Rio de Janeiro, Brazil, April 23-27, 2026*. OpenReview.net, 2022. URL https://openreview.net/forum?id=d8WMoi571f.

J. Huang, P. Lu, and Q. Zeng. Calibrated language models and how to find them with label smoothing. In *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*, 2025.

J. Huang, P. Lu, Q. Zeng, Y. Iwasawa, Y. Matsuo, S. Chandar, E. Marrese-Taylor, and I. Li. Investigating the multilingual calibration effects of language model instruction tuning. In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2026 - Volume 2: Short Papers, Rabat, Morocco, March 24-29, 2026*, pages 1–59, 2026.

Y. Huang, L. Sun, H. Wang, S. Wu, Q. Zhang, Y. Li, C. Gao, Y. Huang, W. Lyu, Y. Zhang, X. Li, H. Sun, Z. Liu, Y. Liu, Y. Wang, Z. Zhang, B. Vidgen, B. Kailkhura, C. Xiong, C. Xiao, C. Li, E. P. Xing, F. Huang, H. Liu, H. Ji, H. Wang, H. Zhang, H. Yao, M. Kellis, M. Zitnik, M. Jiang, M. Bansal, J. Zou, J. Pei, J. Liu, J. Gao, J. Han, J. Zhao, J. Tang, J. Wang, J. Vanschoren, J. C. Mitchell, K. Shu, K. Xu, K. Chang, L. He, L. Huang, M. Backes, N. Z. Gong, P. S. Yu, P. Chen, Q. Gu, R. Xu, R. Ying, S. Ji, S. Jana, T. Chen, T. Liu, T. Zhou, W. Wang, X. Li, X. Zhang, X. Wang, X. Xie, X. Chen, X. Wang, Y. Liu, Y. Ye, Y. Cao, Y. Chen, and Y. Zhao. Position: Trustllm: Trustworthiness in large language models. In R. Salakhutdinov, Z. Kolter, K. A. Heller, A. Weller, N. Oliver, J. Scarlett, and F. Berkenkamp, editors, *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*, volume 235 of *Proceedings of Machine Learning Research*, pages 20166–20270. PMLR / OpenReview.net, 2024. URL https://proceedings.mlr.press/v235/huang24x.html.

A. Q. Jiang, A. Sablayrolles, A. Mensch, C. Bamford, D. S. Chaplot, D. de Las Casas, F. Bressand, G. Lengyel, G. Lample, L. Saulnier, L. R. Lavaud, M. Lachaux, P. Stock, T. L. Scao, T. Lavril, T. Wang, T. Lacroix, and W. E. Sayed. Mistral 7b, 2023a. URL https://arxiv.org/abs/2310.06825.

L. Jiang, K. Rao, S. Han, A. Ettinger, F. Brahman, S. Kumar, N. Mireshghallah, X. Lu, M. Sap, Y. Choi, and N. Dziri. Wildteaming at scale: From in-the-wild jailbreaks to (adversarially) safer language models. In A. Globersons, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. M. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems 37: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024. URL https://openreview.net/forum?id=n5R6TvBVcX.

M. Jiang, Y. Ruan, S. Huang, S. Liao, S. Pitis, R. B. Grosse, and J. Ba. Calibrating language models via augmented prompt ensembles, 2023b. URL https://openreview.net/forum?id=L0dc4wqbNs.

Z. Jiang, J. Araki, H. Ding, and G. Neubig. How can we know *When* language models know? on the calibration of language models for question answering. *Trans. Assoc. Comput. Linguistics*, 9:962–977, 2021. doi: 10.1162/TACL_A_00407. URL https://doi.org/10.1162/tacl_a_00407.

Z. Jiang, Y. Zhang, C. Liu, J. Zhao, and K. Liu. Generative calibration for in-context learning. In H. Bouamor, J. Pino, and K. Bali, editors, *Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023*, volume EMNLP 2023 of *Findings of ACL*, pages 2312–2333. Association for Computational Linguistics, 2023c. doi: 10.18653/V1/2023.FINDINGS-EMNLP.152. URL https://doi.org/10.18653/v1/2023.findings-emnlp.152.

T. Joy, F. Pinto, S. Lim, P. H. S. Torr, and P. K. Dokania. Sample-dependent adaptive temperature scaling for improved calibration. In B. Williams, Y. Chen, and J. Neville, editors, *Thirty-Seventh AAAI Conference on Artificial Intelligence, AAAI 2023, Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence, IAAI 2023, Thirteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2023, Washington, DC, USA, February 7-14, 2023*, pages 14919–14926. AAAI Press, 2023. doi: 10.1609/AAAI.V37I12.26742. URL https://doi.org/10.1609/aaai.v37i12.26742.

S. Kapoor, N. Gruver, M. Roberts, K. Collins, A. Pal, U. Bhatt, A. Weller, S. Dooley, M. Goldblum, and A. G. Wilson. Large language models must be taught to know what they don't know. In A. Globersons, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. M. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems 37: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024. URL https://openreview.net/forum?id=QzvWyggrYB.

A. Krizhevsky. Learning multiple layers of features from tiny images. Master's thesis, University of Toronto, 2009. URL https://cave.cs.toronto.edu/kriz/learning-features-2009-TR.pdf.

M. Kull, M. Perelló-Nieto, M. Kängsepp, T. de Menezes e Silva Filho, H. Song, and P. A. Flach. Beyond temperature scaling: Obtaining well-calibrated multi-class probabilities with dirichlet calibration. In H. M. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. B. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 12295–12305, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/8ca01ea920679a0fe3728441494041b9-Abstract.html.

N. Lambert. Reinforcement learning from human feedback, 2025. URL https://arxiv.org/abs/2504.12501.

N. Lambert, J. Morrison, V. Pyatkin, et al. Tülu 3: Pushing frontiers in open language model post-training. In *The Second Conference on Language Modeling, COLM 2025, Montreal, QC, Canada, October 7-10, 2025*, 2025. URL https://openreview.net/forum?id=i1uGbfHHpH.

J. Leike, D. Krueger, T. Everitt, M. Martic, V. Maini, and S. Legg. Scalable agent alignment via reward modeling: a research direction, 2018. URL http://arxiv.org/abs/1811.07871.

J. Leng, C. Huang, B. Zhu, and J. Huang. Taming overconfidence in llms: Reward calibration in RLHF. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. URL https://openreview.net/forum?id=l0tg0jzsdL.

J. Liang, L. Li, Z. Bing, B. Zhao, Y. Tang, B. Lin, and H. Fan. Efficient one pass self-distillation with zipf's label smoothing. In S. Avidan, G. J. Brostow, M. Cissé, G. M. Farinella, and T. Hassner, editors, *Computer Vision - ECCV 2022 - 17th European Conference, Tel Aviv, Israel, October 23-27, 2022, Proceedings, Part XI*, volume 13671 of *Lecture Notes in Computer Science*, pages 104–119. Springer, 2022. doi: 10.1007/978-3-031-20083-0_7. URL https://doi.org/10.1007/978-3-031-20083-0_7.

S. Lin, J. Hilton, and O. Evans. Teaching models to express their uncertainty in words. *Trans. Mach. Learn. Res.*, 2022, 2022a. URL https://openreview.net/forum?id=8s8K2UZGTZ.

S. Lin, J. Hilton, and O. Evans. Truthfulqa: Measuring how models mimic human falsehoods. In S. Muresan, P. Nakov, and A. Villavicencio, editors, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 3214–3252. Association for Computational Linguistics, 2022b. doi: 10.18653/V1/2022.ACL-LONG.229. URL https://doi.org/10.18653/v1/2022.acl-long.229.

T. Lin, P. Goyal, R. B. Girshick, K. He, and P. Dollár. Focal loss for dense object detection. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, pages 2999–3007. IEEE Computer Society, 2017. doi: 10.1109/ICCV.2017.324. URL https://doi.org/10.1109/ICCV.2017.324.

B. Liu, I. B. Ayed, A. Galdran, and J. Dolz. The devil is in the margin: Margin-based label smoothing for network calibration. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pages 80–88. IEEE, 2022. doi: 10.1109/CVPR52688.2022.00018. URL https://doi.org/10.1109/CVPR52688.2022.00018.

J. Liu, C. S. Xia, Y. Wang, and L. Zhang. Is your code generated by chatgpt really correct? rigorous evaluation of large language models for code generation. In A. Oh, T. Naumann, A. Globerson,

K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. URL https://openreview.net/forum?id=1qvx610Cu7.

Llama Team. The llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

M. Lukasik, S. Bhojanapalli, A. K. Menon, and S. Kumar. Does label smoothing mitigate label noise? In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 6448–6458. PMLR, 2020. URL http://proceedings.mlr.press/v119/lukasik20a.html.

B. Luo, S. Wang, S. Li, and H. Wei. Your pre-trained LLM is secretly an unsupervised confidence calibrator. In D. Belgrave, C. Zhang, L. N. Montoya, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, N. Chen, I. V. M. Ruíz, and A. Loaiza-Bonilla, editors, *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2025, NeurIPS 2025, San Diego, CA, USA, December 2-7, 2025 / Mexico City, Mexico, November 30 - December 5, 2025*, 2025. URL https://openreview.net/forum?id=I4PJYZvfW5.

A. Mallen, A. Asai, V. Zhong, R. Das, D. Khashabi, and H. Hajishirzi. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. In A. Rogers, J. L. Boyd-Graber, and N. Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 9802–9822. Association for Computational Linguistics, 2023. doi: 10.18653/V1/2023.ACL-LONG.546. URL https://doi.org/10.18653/v1/2023.acl-long.546.

M. Mazeika, L. Phan, X. Yin, A. Zou, Z. Wang, N. Mu, E. Sakhaee, N. Li, S. Basart, B. Li, D. A. Forsyth, and D. Hendrycks. Harmbench: A standardized evaluation framework for automated red teaming and robust refusal. In R. Salakhutdinov, Z. Kolter, K. A. Heller, A. Weller, N. Oliver, J. Scarlett, and F. Berkenkamp, editors, *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*, volume 235 of *Proceedings of Machine Learning Research*, pages 35181–35224. PMLR / OpenReview.net, 2024. URL https://proceedings.mlr.press/v235/mazeika24a.html.

Meta PyTorch. torchtune: Pytorch's finetuning library, Apr. 2024. URL https//github.com/pytorch/torchtune.

M. Minderer, J. Djolonga, R. Romijnders, F. Hubis, X. Zhai, N. Houlsby, D. Tran, and M. Lucic. Revisiting the calibration of modern neural networks. In M. Ranzato, A. Beygelzimer, Y. N. Dauphin, P. Liang, and J. W. Vaughan, editors, *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 15682–15694, 2021. URL https://openreview.net/forum?id=QRBvLayFXI.

J. Mukhoti, V. Kulharia, A. Sanyal, S. Golodetz, P. H. S. Torr, and P. K. Dokania. Calibrating deep neural networks using focal loss. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/aeb7b30ef1d024a76f21a1d40e30c302-Abstract.html.

R. Müller, S. Kornblith, and G. E. Hinton. When does label smoothing help? In H. M. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. B. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 4696–4705, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/f1748d6b0fd9d439f71450117eba2725-Abstract.html.

A. H. Murphy. Scalar and vector partitions of the probability score: Part i. two-state situation. *Journal of Applied Meteorology and Climatology*, 11(2):273–282, 1972. URL https://journals.ametsoc.org/view/journals/apme/11/2/1520-0450_1972_011_0273_savpot_2_0_co_2.xml.

M. P. Naeini, G. F. Cooper, and M. Hauskrecht. Obtaining well calibrated probabilities using bayesian binning. In B. Bonet and S. Koenig, editors, *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA*, pages 2901–2907. AAAI Press, 2015. doi: 10.1609/AAAI.V29I1.9602. URL https://doi.org/10.1609/aaai.v29i1.9602.

L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. L. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, J. Schulman, J. Hilton, F. Kelton, L. Miller, M. Simens, A. Askell, P. Welinder, P. F. Christiano, J. Leike, and R. Lowe. Training language models to follow instructions with human feedback. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022. URL https://openreview.net/forum?id=TG8KACxEON.

G. Pereyra, G. Tucker, J. Chorowski, L. Kaiser, and G. E. Hinton. Regularizing neural networks by penalizing confident output distributions. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Workshop Track Proceedings*. OpenReview.net, 2017. URL https://openreview.net/forum?id=HyhbYrGYe.

R. Rafailov, A. Sharma, E. Mitchell, C. D. Manning, S. Ermon, and C. Finn. Direct preference optimization: Your language model is secretly a reward model. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. URL https://openreview.net/forum?id=HPuSIXJaa9.

Y. Ren and D. J. Sutherland. Learning dynamics of LLM finetuning. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. URL https://openreview.net/forum?id=tPNHOoZFl9.

P. Röttger, H. Kirk, B. Vidgen, G. Attanasio, F. Bianchi, and D. Hovy. Xstest: A test suite for identifying exaggerated safety behaviours in large language models. In K. Duh, H. Gómez-Adorno, and S. Bethard, editors, *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), NAACL 2024, Mexico City, Mexico, June 16-21, 2024*, pages 5377–5400. Association for Computational Linguistics, 2024. doi: 10.18653/V1/2024.NAACL-LONG.301. URL https://doi.org/10.18653/v1/2024.naacl-long.301.

V. Sanh, A. Webson, C. Raffel, S. H. Bach, L. Sutawika, Z. Alyafeai, A. Chaffin, A. Stiegler, A. Raja, M. Dey, M. S. Bari, C. Xu, U. Thakker, S. S. Sharma, E. Szczechla, T. Kim, G. Chhablani, N. V. Nayak, D. Datta, J. Chang, M. T. Jiang, H. Wang, M. Manica, S. Shen, Z. X. Yong, H. Pandey, R. Bawden, T. Wang, T. Neeraj, J. Rozen, A. Sharma, A. Santilli, T. Févry, J. A. Fries, R. Teehan, T. L. Scao, S. Biderman, L. Gao, T. Wolf, and A. M. Rush. Multitask prompted training enables zero-shot task generalization. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022. URL https://openreview.net/forum?id=9Vrb9D0WI4.

Z. Shao, P. Wang, Q. Zhu, R. Xu, J. Song, M. Zhang, Y. K. Li, Y. Wu, and D. Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models, 2024. URL https://arxiv.org/abs/2402.03300.

M. Shen, S. Das, K. H. Greenewald, P. Sattigeri, G. W. Wornell, and S. Ghosh. Thermometer: Towards universal calibration for large language models. In R. Salakhutdinov, Z. Kolter, K. A. Heller, A. Weller, N. Oliver, J. Scarlett, and F. Berkenkamp, editors, *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*, volume 235 of *Proceedings of Machine Learning Research*, pages 44687–44711. PMLR / OpenReview.net, 2024a. URL https://proceedings.mlr.press/v235/shen24c.html.

X. Shen, Z. Chen, M. Backes, Y. Shen, and Y. Zhang. "do anything now": Characterizing and evaluating in-the-wild jailbreak prompts on large language models. In B. Luo, X. Liao, J. Xu, E. Kirda, and D. Lie, editors, *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security, CCS 2024, Salt Lake City, UT, USA, October 14-18, 2024*, pages

1671–1685. ACM, 2024b. doi: 10.1145/3658644.3670388. URL https://doi.org/10.1145/3658644.3670388.

H. Smith, N. L. Diamant, and B. L. Trippe. Calibrating generative models to distributional constraints. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=5yXKRaNI2R.

M. Suzgun, N. Scales, N. Schärli, S. Gehrmann, Y. Tay, H. W. Chung, A. Chowdhery, Q. V. Le, E. H. Chi, D. Zhou, and J. Wei. Challenging big-bench tasks and whether chain-of-thought can solve them. In A. Rogers, J. L. Boyd-Graber, and N. Okazaki, editors, *Findings of the Association for Computational Linguistics: ACL 2023, Toronto, Canada, July 9-14, 2023*, volume ACL 2023 of *Findings of ACL*, pages 13003–13051. Association for Computational Linguistics, 2023. doi: 10.18653/V1/2023.FINDINGS-ACL.824. URL https://doi.org/10.18653/v1/2023.findings-acl.824.

C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 2818–2826. IEEE Computer Society, 2016. doi: 10.1109/CVPR.2016.308. URL https://doi.org/10.1109/CVPR.2016.308.

S. Tao, L. Yao, H. Ding, Y. Xie, Q. Cao, F. Sun, J. Gao, H. Shen, and B. Ding. When to trust llms: Aligning confidence with response quality. In L. Ku, A. Martins, and V. Srikumar, editors, *Findings of the Association for Computational Linguistics, ACL 2024, Bangkok, Thailand and virtual meeting, August 11-16, 2024*, volume ACL 2024 of *Findings of ACL*, pages 5984–5996. Association for Computational Linguistics, 2024. doi: 10.18653/V1/2024.FINDINGS-ACL.357. URL https://doi.org/10.18653/v1/2024.findings-acl.357.

Teknium. Openhermes 2.5: An open dataset of synthetic data for generalist llm assistants, 2023. URL https://huggingface.co/datasets/teknium/OpenHermes-2.5.

K. Tian, E. Mitchell, A. Zhou, A. Sharma, R. Rafailov, H. Yao, C. Finn, and C. D. Manning. Just ask for calibration: Strategies for eliciting calibrated confidence scores from language models fine-tuned with human feedback. In H. Bouamor, J. Pino, and K. Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 5433–5442. Association for Computational Linguistics, 2023. doi: 10.18653/V1/2023.EMNLP-MAIN.330. URL https://doi.org/10.18653/v1/2023.emnlp-main.330.

A. M. Turner, L. Thiergart, D. Udell, G. Leech, U. Mini, and M. MacDiarmid. Activation addition: Steering language models without optimization, 2023. URL https://arxiv.org/abs/2308.10248.

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. In I. Guyon, U. von Luxburg, S. Bengio, H. M. Wallach, R. Fergus, S. V. N. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

V. Wang and E. Stengel-Eskin. Calibrating verbalized confidence with self-generated distractors, 2025. URL https://arxiv.org/abs/2509.25532.

Y. Wang, S. Mishra, P. Alipoormolabashi, Y. Kordi, A. Mirzaei, A. Naik, A. Ashok, A. S. Dhanasekaran, A. Arunkumar, D. Stap, E. Pathak, G. Karamanolakis, H. G. Lai, I. Purohit, I. Mondal, J. Anderson, K. Kuznia, K. Doshi, K. K. Pal, M. Patel, M. Moradshahi, M. Parmar, M. Purohit, N. Varshney, P. R. Kaza, P. Verma, R. S. Puri, R. Karia, S. Doshi, S. K. Sampat, S. Mishra, S. R. A, S. Patro, T. Dixit, and X. Shen. Super-naturalinstructions: Generalization via declarative instructions on 1600+ NLP tasks. In Y. Goldberg, Z. Kozareva, and Y. Zhang, editors, *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pages 5085–5109. Association for Computational Linguistics, 2022. doi: 10.18653/V1/2022.EMNLP-MAIN.340. URL https://doi.org/10.18653/v1/2022.emnlp-main.340.

Y. Wang, H. Ivison, P. Dasigi, J. Hessel, T. Khot, K. R. Chandu, D. Wadden, K. MacMillan, N. A. Smith, I. Beltagy, and H. Hajishirzi. How far can camels go? exploring the state of instruction tuning on open resources. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. URL https://openreview.net/forum?id=w4zZNC4ZaV.

H. Wei, R. Xie, H. Cheng, L. Feng, B. An, and Y. Li. Mitigating neural network overconfidence with logit normalization. In K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvári, G. Niu, and S. Sabato, editors, *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pages 23631–23644. PMLR, 2022a. URL https://proceedings.mlr.press/v162/wei22d.html.

J. Wei, M. Bosma, V. Y. Zhao, K. Guu, A. W. Yu, B. Lester, N. Du, A. M. Dai, and Q. V. Le. Finetuned language models are zero-shot learners. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022b. URL https://openreview.net/forum?id=gEZrGCozdqR.

J. Wei, H. Liu, T. Liu, G. Niu, M. Sugiyama, and Y. Liu. To smooth or not? when label smoothing meets noisy labels. In K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvári, G. Niu, and S. Sabato, editors, *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pages 23589–23614. PMLR, 2022c. URL https://proceedings.mlr.press/v162/wei22b.html.

E. Wijmans, B. Huval, A. Hertzberg, V. Koltun, and P. Krähenbühl. Cut your losses in large-vocabulary language models. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. URL https://openreview.net/forum?id=E4Fk3YuG56.

G. Xia, O. Laurent, G. Franchi, and C. Bouganis. Towards understanding why label smoothing degrades selective classification and how to fix it. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. URL https://openreview.net/forum?id=6oWFn6fY4A.

J. Xiao, B. Hou, Z. Wang, R. Jin, Q. Long, W. J. Su, and L. Shen. Restoring calibration for aligned large language models: A calibration-aware fine-tuning approach. In A. Singh, M. Fazel, D. Hsu, S. Lacoste-Julien, F. Berkenkamp, T. Maharaj, K. Wagstaff, and J. Zhu, editors, *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*, volume 267 of *Proceedings of Machine Learning Research*. PMLR / OpenReview.net, 2025a. URL https://proceedings.mlr.press/v267/xiao25b.html.

J. Xiao, Z. Li, X. Xie, et al. 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. URL https://www.tandfonline.com/doi/full/10.1080/01621459.2025.2555067.

Y. Xiao, P. P. Liang, U. Bhatt, W. Neiswanger, R. Salakhutdinov, and L. Morency. Uncertainty quantification with pre-trained language models: A large-scale empirical analysis. In Y. Goldberg, Z. Kozareva, and Y. Zhang, editors, *Findings of the Association for Computational Linguistics: EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, volume EMNLP 2022 of *Findings of ACL*, pages 7273–7284. Association for Computational Linguistics, 2022. doi: 10.18653/V1/2022.FINDINGS-EMNLP.538. URL https://doi.org/10.18653/v1/2022.findings-emnlp.538.

M. Xiong, Z. Hu, X. Lu, Y. Li, J. Fu, J. He, and B. Hooi. Can llms express their uncertainty? an empirical evaluation of confidence elicitation in llms. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=gjeQKFxFpZ.

T. Xu, S. Wu, S. Diao, X. Liu, X. Wang, Y. Chen, and J. Gao. Sayself: Teaching llms to express confidence with self-reflective rationales. In Y. Al-Onaizan, M. Bansal, and Y. Chen, editors, *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing,*

EMNLP 2024, Miami, FL, USA, November 12-16, 2024, pages 5985–5998. Association for Computational Linguistics, 2024. doi: 10.18653/V1/2024.EMNLP-MAIN.343. URL https://doi.org/10.18653/v1/2024.emnlp-main.343.

R. Zellers, A. Holtzman, Y. Bisk, A. Farhadi, and Y. Choi. Hellaswag: Can a machine really finish your sentence? In A. Korhonen, D. R. Traum, and L. Màrquez, editors, *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 4791–4800. Association for Computational Linguistics, 2019. doi: 10.18653/V1/P19-1472. URL https://doi.org/10.18653/v1/p19-1472.

S. Zhai, T. Likhomanenko, E. Littwin, D. Busbridge, J. Ramapuram, Y. Zhang, J. Gu, and J. M. Susskind. Stabilizing transformer training by preventing attention entropy collapse. In A. Krause, E. Brunskill, K. Cho, B. Engelhardt, S. Sabato, and J. Scarlett, editors, *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 40770–40803. PMLR, 2023. URL https://proceedings.mlr.press/v202/zhai23a.html.

C. Zhang, P. Jiang, Q. Hou, Y. Wei, Q. Han, Z. Li, and M. Cheng. Delving deep into label smoothing. *IEEE Trans. Image Process.*, 30:5984–5996, 2021. doi: 10.1109/TIP.2021.3089942. URL https://doi.org/10.1109/TIP.2021.3089942.

H. Zhang, M. Cissé, Y. N. Dauphin, and D. Lopez-Paz. mixup: Beyond empirical risk minimization. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. URL https://openreview.net/forum?id=r1Ddp1-Rb.

Z. Zhang and M. R. Sabuncu. Self-distillation as instance-specific label smoothing. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/1731592aca5fb4d789c4119c65c10b4b-Abstract.html.

Z. Zhao, E. Wallace, S. Feng, D. Klein, and S. Singh. Calibrate before use: Improving few-shot performance of language models. In M. Meila and T. Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 12697–12706. PMLR, 2021. URL http://proceedings.mlr.press/v139/zhao21c.html.

J. Zhou, T. Lu, S. Mishra, S. Brahma, S. Basu, Y. Luan, D. Zhou, and L. Hou. Instruction-following evaluation for large language models, 2023. URL https://arxiv.org/abs/2311.07911.

Y. Zhou, H. Li, Z. Cheng, X. Yan, Y. Dong, M. Fritz, and M. Keuper. Maxsup: Overcoming representation collapse in label smoothing. In D. Belgrave, C. Zhang, L. N. Montoya, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, N. Chen, I. V. M. Ruíz, and A. Loaiza-Bonilla, editors, *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2025, NeurIPS 2025, San Diego, CA, USA, December 2-7, 2025 / Mexico City, Mexico, November 30 - December 5, 2025*, 2025a. URL https://openreview.net/forum?id=efOq8wHH9o.

Z. Zhou, T. Jin, J. Shi, and Q. Li. Calibrating LLM confidence with semantic steering: A multi-prompt aggregation framework, 2025b. URL https://arxiv.org/abs/2503.02863.

Appendix

A Mathematical Details **21**
A.1 Proof of Proposition 3.2 . . . 21
A.2 Proof of Proposition 3.3 . . . 21
A.3 Proof of Lemma 3.4 . . . 22
A.4 Proof of Corollary 3.5 . . . 22
A.5 Proof of Lemma 4.1 . . . 23
A.6 Proof of Theorem 4.2 . . . 24
A.7 Derivations of Gradient . . . 26

B Experimental Details **27**
B.1 Evaluation Tasks . . . 27
B.1.1 Safety Evaluation . . . 28

C Design Ablations **30**
C.1 Hyper-parameter Sensitivity . . . 30
C.2 Regularization on Regions . . . 30
C.3 Margins for Logits . . . 30

D Computational Efficiency Testing **31**

E Extended Details on Calibration Metrics **32**

F Additional Related Works **33**

G Additional Experimental Results **34**
G.1 Additional Baseline Comparisons . . . 34
G.2 Additional Preference-Tuning Calibration Comparisons . . . 34
G.3 Additional Post-Hoc Calibration Comparisons . . . 35
G.4 Additional Uncertainty/Bias Validation . . . 35
G.5 Additional Free-Form Generation Results . . . 36

H Additional Image Classification Results **37**

A Mathematical Details

A.1 Proof of Proposition 3.2

Proposition 3.2. A linear penalty (or a Lagrangian term) for the hard constraint $\boldsymbol{d} = \mathbf{0}$ is bounded from above and below by $D_{\mathrm{KL}}[\boldsymbol{u}\|\sigma(\boldsymbol{x};\theta)]$, up to additive constants

$$D_{\mathrm{KL}}[\boldsymbol{u}\|\sigma(\boldsymbol{x};\theta)] \leq \frac{1}{K}\sum_{i=1}^{K}\boldsymbol{d}_i \leq D_{\mathrm{KL}}[\boldsymbol{u}\|\sigma(\boldsymbol{x};\theta)] + \log K. \tag{11}$$

Proof. Using $\boldsymbol{u}_i = \frac{1}{K}$ and $q(i|\boldsymbol{x};\theta) = \sigma(\boldsymbol{x};\theta)_i$, we can write

$$D_{\mathrm{KL}}[\boldsymbol{u}\|\sigma(\boldsymbol{x};\theta)] = \sum_{i=1}^{K}\frac{1}{K}\log\frac{1/K}{q(i|\boldsymbol{x};\theta)} = -\log K - \frac{1}{K}\sum_{i=1}^{K}\log q(i|\boldsymbol{x};\theta)\,.$$

Substituting the softmax form $q(i|\boldsymbol{x};\theta) = \frac{e^{\boldsymbol{\ell}_i}}{\sum_{j=1}^{K}e^{\boldsymbol{\ell}_j}}$ gives

$$D_{\mathrm{KL}}[\boldsymbol{u}\|\sigma(\boldsymbol{x};\theta)] = -\log K + \log\sum_{j=1}^{K}e^{\boldsymbol{\ell}_j} - \frac{1}{K}\sum_{i=1}^{K}\boldsymbol{\ell}_i.$$

Now apply the standard LogSumExp bound

$$\max_j \boldsymbol{\ell}_j \leq \log\sum_{j=1}^{K}e^{\boldsymbol{\ell}_j} \leq \max_j \boldsymbol{\ell}_j + \log K.$$

Since $\boldsymbol{d}_i = \max_j \boldsymbol{\ell}_j - \boldsymbol{\ell}_i$, it follows that

$$D_{\mathrm{KL}}[\boldsymbol{u}\|\sigma(\boldsymbol{x};\theta)] \leq \frac{1}{K}\sum_{i=1}^{K}\boldsymbol{d}_i \leq D_{\mathrm{KL}}[\boldsymbol{u}\|\sigma(\boldsymbol{x};\theta)] + \log K.$$

Hence the additional penalty $D_{\mathrm{KL}}[\boldsymbol{u}\|\sigma(\boldsymbol{x};\theta)]$ imposed by LS is, up to an additive constant, optimizing a linear penalty for the constraint $\boldsymbol{d} = \mathbf{0}$ and therefore encouraging similarity among logits.

A.2 Proof of Proposition 3.3

Proposition 3.3. Let $\boldsymbol{z} = \sigma(\boldsymbol{x};\theta) = \mathrm{softmax}(f(\boldsymbol{x};\theta)) \in \Delta^{K-1}$, and suppose that $p(y|\boldsymbol{x},\boldsymbol{z}) = \mathrm{Cat}(\boldsymbol{z})$. Under the independence assumption $p(\boldsymbol{z}|\boldsymbol{x}) = p(\boldsymbol{z})$ and the symmetric Dirichlet prior

$$p(\boldsymbol{z}) = \mathrm{Dir}(\boldsymbol{\alpha})\,, \qquad [\boldsymbol{\alpha}]_k = 1 + \frac{\beta}{K(1-\beta)} \quad \forall k \in \{1,\dots,K\},$$

minimizing the label smoothing objective with smoothing rate $\beta \in [0,1)$ is equivalent, up to multiplication by the positive constant $(1-\beta)$, to Maximum A Posteriori (MAP) estimation of $\boldsymbol{z}$.

Proof. This proof follows the MAP interpretation in Zhang and Sabuncu (2020). Let $\mathcal{D} = \{(\boldsymbol{x}_n, y_n)\}_{n=1}^{N}$ and denote $\boldsymbol{z}_n = \sigma(\boldsymbol{x}_n;\theta)$. Under the likelihood $p(y_n|\boldsymbol{x}_n,\boldsymbol{z}_n) = \mathrm{Cat}(\boldsymbol{z}_n)$ and the symmetric prior $p(\boldsymbol{z}_n) = \mathrm{Dir}(\boldsymbol{\alpha})$, the negative log posterior is, up to constants independent of θ,

$$\begin{aligned}\mathcal{L}_{\mathrm{MAP}}(\theta) &= -\sum_{n=1}^{N}\log p(y_n|\boldsymbol{x}_n,\boldsymbol{z}_n) - \sum_{n=1}^{N}\log p(\boldsymbol{z}_n) \\ &= -\sum_{n=1}^{N}\log[\boldsymbol{z}_n]_{y_n} - \sum_{n=1}^{N}\sum_{k=1}^{K}([\boldsymbol{\alpha}]_k - 1)\log[\boldsymbol{z}_n]_k + c.\end{aligned} \tag{12}$$

Choosing $[\boldsymbol{\alpha}]_k = 1 + \frac{\beta}{K(1-\beta)}$ for every k yields

$$\mathcal{L}_{\mathrm{MAP}}(\theta) = -\sum_{n=1}^{N}\log[\boldsymbol{z}_n]_{y_n} - \frac{\beta}{K(1-\beta)}\sum_{n=1}^{N}\sum_{k=1}^{K}\log[\boldsymbol{z}_n]_k + c. \tag{13}$$

Multiplying by $(1-\beta)$, which does not change the minimizer, gives

$$\begin{aligned}(1-\beta)\mathcal{L}_{\text{MAP}}(\theta) &= -\sum_{n=1}^{N}\left[(1-\beta)\log[\boldsymbol{z}_n]_{y_n} + \frac{\beta}{K}\sum_{k=1}^{K}\log[\boldsymbol{z}_n]_k\right] + c' \\ &= -\sum_{n=1}^{N}\sum_{k=1}^{K}\left[(1-\beta)\,\delta^k_{y_n} + \frac{\beta}{K}\right]\log[\boldsymbol{z}_n]_k + c'.\end{aligned} \tag{14}$$

The right-hand side is exactly the label smoothing objective with smoothing rate β. Hence minimizing the label smoothing loss is equivalent, up to the positive multiplicative factor $(1-\beta)$, to MAP estimation under the stated symmetric Dirichlet prior.

A.3 Proof of Lemma 3.4

Lemma 3.4. Denote $\text{CE}(\boldsymbol{q}, \boldsymbol{e}^{\mathsf{t}})$ as the CE between a predicted probability vector $\boldsymbol{q}$ and a one-hot encoding of the target class, $\boldsymbol{e}^{\mathsf{t}}$. Under label smoothing rate β,

$$\text{CE}_{\text{LS}}\big(\boldsymbol{q}, \boldsymbol{e}^{\mathsf{t}}; \beta\big) = \underbrace{\text{CE}\big(\boldsymbol{q}, \boldsymbol{e}^{\mathsf{t}}\big)}_{\text{Target Loss}} + \underbrace{\beta\big(\text{CE}(\boldsymbol{q}, \boldsymbol{u}) - \text{CE}\big(\boldsymbol{q}, \boldsymbol{e}^{\mathsf{t}}\big)\big)}_{\text{Smoothing Regularization}}$$

where $\boldsymbol{u}$ is the K-dimensional vector defined as $\boldsymbol{u} = \left[\frac{1}{K}, \ldots, \frac{1}{K}\right]$ such that $\sum_{i=1}^{K} \boldsymbol{u}_i = 1$.

Proof. Begin by expressing the cross-entropy between an arbitrary probability vector $\boldsymbol{q}$ and a target vector $\boldsymbol{e}$ as

$$\text{CE}(\boldsymbol{q}, \boldsymbol{e}) = -\sum_{i=1}^{K} e_i \log \boldsymbol{q}_i.$$

In the context of label smoothing, the target vector is transformed such that

$$e_i = (1-\beta)\cdot \mathbb{1}[i = \mathsf{t}] + \frac{\beta}{K},$$

where β is the smoothing parameter, t is the target class index, and K is the number of classes. Substituting the definition of e_i yields

$$\text{CE}(\boldsymbol{q}, \boldsymbol{e}) = -\sum_{i=1}^{K}\left((1-\beta)\cdot \mathbb{1}[i = \mathsf{t}] + \frac{\beta}{K}\right)\log \boldsymbol{q}_i.$$

We can then expand this to get

$$\text{CE}(\boldsymbol{q}, \boldsymbol{e}) = -(1-\beta)\sum_{i=1}^{K}\mathbb{1}[i = \mathsf{t}]\log \boldsymbol{q}_i - \frac{\beta}{K}\sum_{i=1}^{K}\log \boldsymbol{q}_i.$$

The first term is directly equivalent to $(1-\beta)\,\text{CE}(\boldsymbol{q}, \boldsymbol{e}^{\mathsf{t}})$, where $\boldsymbol{e}^{\mathsf{t}}$ is the one-hot encoding of the target class. Similarly, the second term is equivalent to $\beta \cdot \text{CE}(\boldsymbol{q}, \boldsymbol{u})$ if we define $\boldsymbol{u}$ as the K-dimensional uniform distribution. By rearranging, we get

$$\text{CE}_{\text{LS}}\big(\boldsymbol{q}, \boldsymbol{e}^{\mathsf{t}}; \beta\big) = \text{CE}\big(\boldsymbol{q}, \boldsymbol{e}^{\mathsf{t}}\big) + \beta\big(\text{CE}(\boldsymbol{q}, \boldsymbol{u}) - \text{CE}\big(\boldsymbol{q}, \boldsymbol{e}^{\mathsf{t}}\big)\big)\,.$$

A.4 Proof of Corollary 3.5

Corollary 3.5. Define $\boldsymbol{\ell}_i$ as the logit prediction of the model for the i-th class. Then the cross-entropy regularization term can be re-written as

$$\begin{aligned}\beta\big(\text{CE}(\boldsymbol{q}, \boldsymbol{u}) - \text{CE}\big(\boldsymbol{q}, \boldsymbol{e}^{\mathsf{t}}\big)\big) &= \beta\left(\boldsymbol{\ell}_{\mathsf{t}} - \frac{1}{K}\sum_{i=1}^{K}\boldsymbol{\ell}_i\right) \\ &= \frac{\beta}{K}\sum_{i:\boldsymbol{\ell}_i < \boldsymbol{\ell}_{\mathsf{t}}}(\boldsymbol{\ell}_{\mathsf{t}} - \boldsymbol{\ell}_i) + \frac{\beta}{K}\sum_{i:\boldsymbol{\ell}_i > \boldsymbol{\ell}_{\mathsf{t}}}(\boldsymbol{\ell}_{\mathsf{t}} - \boldsymbol{\ell}_i)\,,\end{aligned}$$

Proof. From Lemma 3.4, we have

$$\mathrm{CE}_{\mathrm{LS}}\left(\boldsymbol{q}, \boldsymbol{e}^{\mathsf{t}}; \beta\right) = -\left(1-\beta\right) \log \boldsymbol{q}_{\mathsf{t}} - \frac{\beta}{K} \sum_{i=1}^{K} \log \boldsymbol{q}_i.$$

After the $\mathrm{softmax}$, $\boldsymbol{q}_i = \frac{\exp(\boldsymbol{\ell}_i)}{\sum_{j=1}^{K} \exp(\boldsymbol{\ell}_j)}$ and we can express $\log \boldsymbol{q}_i$ in terms of the logits as

$$\log \boldsymbol{q}_i = \boldsymbol{\ell}_i - \log \left(\sum_{j=1}^{K} e^{\boldsymbol{\ell}_j} \right),$$

which we can substitute to get

$$\begin{aligned}
\mathrm{CE}_{\mathrm{LS}}\left(\boldsymbol{q}, \boldsymbol{e}^{\mathsf{t}}; \beta\right) &= -\left(1-\beta\right) \left[\boldsymbol{\ell}_{\mathsf{t}} - \log \left(\sum_{j=1}^{K} e^{\boldsymbol{\ell}_j} \right) \right] - \frac{\beta}{K} \sum_{i=1}^{K} \left[\boldsymbol{\ell}_i - \log \left(\sum_{j=1}^{K} e^{\boldsymbol{\ell}_j} \right) \right] \\
&= -\left(1-\beta\right) \boldsymbol{\ell}_{\mathsf{t}} + \left(1-\beta\right) \log \left(\sum_{j=1}^{K} e^{\boldsymbol{\ell}_j} \right) - \frac{\beta}{K} \sum_{i=1}^{K} \boldsymbol{\ell}_i + \beta \log \left(\sum_{j=1}^{K} e^{\boldsymbol{\ell}_j} \right) \qquad (15) \\
&= -\boldsymbol{\ell}_{\mathsf{t}} + \log \left(\sum_{j=1}^{K} e^{\boldsymbol{\ell}_j} \right) + \beta \left[\boldsymbol{\ell}_{\mathsf{t}} - \frac{1}{K} \sum_{j=1}^{K} \boldsymbol{\ell}_j \right].
\end{aligned}$$

The term in brackets is exactly the smoothing regularization. We can rewrite it as

$$\beta \left[\boldsymbol{\ell}_{\mathsf{t}} - \frac{1}{K} \sum_{j=1}^{K} \boldsymbol{\ell}_j \right] = \frac{\beta}{K} \left[K \cdot \boldsymbol{\ell}_{\mathsf{t}} - \sum_{j=1}^{K} \boldsymbol{\ell}_j \right] = \frac{\beta}{K} \sum_{i:\boldsymbol{\ell}_i < \boldsymbol{\ell}_{\mathsf{t}}} \left(\boldsymbol{\ell}_{\mathsf{t}} - \boldsymbol{\ell}_i \right) + \frac{\beta}{K} \sum_{i:\boldsymbol{\ell}_i > \boldsymbol{\ell}_{\mathsf{t}}} \left(\boldsymbol{\ell}_{\mathsf{t}} - \boldsymbol{\ell}_i \right), \qquad (16)$$

which proves the claim.

A.5 Proof of Lemma 4.1

Lemma 4.1. Let $\boldsymbol{C} \in \mathbb{R}^{D \times K}$ be a matrix with $\|\boldsymbol{C}\|_2 = \sigma_C$, and let $\boldsymbol{h} \in \mathbb{R}^D$ be a vector such that each entry of $\boldsymbol{h}$ satisfies $|\boldsymbol{h}_i| \leq \sigma_h$. The upper bound for $\left\|\boldsymbol{C}^\top \boldsymbol{h}\right\|_2$ is:

$$\left\|\boldsymbol{C}^\top \boldsymbol{h}\right\|_2 \leq \sigma_C \cdot \sigma_h \cdot \sqrt{D}. \qquad (17)$$

Proof. For any vector $\boldsymbol{h} \in \mathbb{R}^D$, it follows that:

$$\left\|\boldsymbol{C}^\top \boldsymbol{h}\right\|_2 \leq \|\boldsymbol{C}\|_2 \cdot \|\boldsymbol{h}\|_2 .$$

Substituting $\|\boldsymbol{C}\|_2 = \sigma_C$, we obtain:

$$\left\|\boldsymbol{C}^\top \boldsymbol{h}\right\|_2 \leq \sigma_C \cdot \|\boldsymbol{h}\|_2$$

and

$$\|\boldsymbol{h}\|_2 \leq \sqrt{\sum_{i=1}^{D} \sigma_h^2} = \sqrt{D} \cdot \sigma_h.$$

Substituting the bound on $\|\boldsymbol{h}\|_2$ into the inequality for $\left\|\boldsymbol{C}^\top \boldsymbol{h}\right\|_2$, we have the norm of the logit vector $\boldsymbol{z} \in \mathbb{R}^K$:

$$\|\boldsymbol{z}\|_2 = \left\|\boldsymbol{C}^\top \boldsymbol{h}\right\|_2 \leq \sigma_C \cdot \|\boldsymbol{h}\|_2 \leq \sigma_C \cdot \sqrt{D} \cdot \sigma_h.$$

A.6 Proof of Theorem 4.2

Theorem 4.2. Let $\rho = \sigma_C \sigma_h$, $\boldsymbol{z} = \boldsymbol{C}^\top \boldsymbol{h}$, $\boldsymbol{p} = \mathrm{softmax}(\boldsymbol{z})$ and $\gamma = \exp\left(-\rho\sqrt{\frac{D \cdot K}{K-1}}\right)$, then the entropy of the prediction holds that:

$$H(\boldsymbol{p}) \geq \log\left(1 + (K-1)\,\gamma\right) + \frac{\rho \cdot \gamma\sqrt{D \cdot K \cdot (K-1)}}{1 + (K-1) \cdot \gamma}. \tag{18}$$

Proof. Let $R = \rho\sqrt{D}$. From Lemma 4.1, $\|\boldsymbol{z}\|_2 \leq R$. Since the softmax is invariant to adding a constant to all logits, define the centered logits

$$\tilde{\boldsymbol{z}} = \boldsymbol{z} - \frac{1}{K}\left(\sum_{i=1}^{K} z_i\right)\mathbf{1}.$$

Then $\mathrm{softmax}(\tilde{\boldsymbol{z}}) = \mathrm{softmax}(\boldsymbol{z})$ and $\|\tilde{\boldsymbol{z}}\|_2 \leq \|\boldsymbol{z}\|_2 \leq R$. Hence it is sufficient to lower bound the entropy over the centered set

$$\mathcal{B}_0(R) = \{\boldsymbol{u} \in \mathbb{R}^K : \sum_{i=1}^{K} u_i = 0,\ \|\boldsymbol{u}\|_2 \leq R\}.$$

For any non-constant direction $\boldsymbol{u}$, entropy decreases as the logits are scaled away from zero:

$$\frac{d}{d\alpha} H(\mathrm{softmax}(\alpha\boldsymbol{u})) = -\alpha\ \mathrm{Var}_{\mathrm{softmax}(\alpha\boldsymbol{u})}\left(\boldsymbol{u}\right) \leq 0.$$

Thus the minimum over $\mathcal{B}_0(R)$ is attained on the boundary $\|\boldsymbol{u}\|_2 = R$.

We next characterize the boundary minimizer directly by following Zhai et al. (2023). Write $p_i = \mathrm{softmax}(\boldsymbol{u})_i$ and $\mu = \sum_{j=1}^{K} p_j u_j$. Since

$$H(\boldsymbol{p}) = \log \sum_{j=1}^{K} e^{u_j} - \sum_{j=1}^{K} p_j u_j, \qquad \frac{\partial H(\boldsymbol{p})}{\partial u_i} = p_i\left(\mu - u_i\right),$$

the Karush–Kuhn–Tucker conditions for minimizing $H(\mathrm{softmax}(\boldsymbol{u}))$ subject to $\sum_i u_i = 0$ and $\|\boldsymbol{u}\|_2 = R$ are

$$p_i\left(\mu - u_i\right) + 2\lambda u_i + \nu = 0, \qquad p_i = \mathrm{softmax}(\boldsymbol{u})_i, \quad \mu = \sum_{j=1}^{K} p_j u_j.$$

Equivalently, each coordinate u_i must be a root of the same scalar equation

$$\frac{e^x}{\sum_j e^{u_j}}\left(\mu - x\right) + 2\lambda x + \nu = 0.$$

At an entropy-minimizing extremum, these roots can be grouped into two values. To see why, suppose two coordinates on the same side of μ take different values. Replacing them by their average preserves the centering constraint locally and reduces the squared norm allocated to that side; the remaining norm can then be moved along the boundary direction that increases the largest logit and decreases the remaining logits, which lowers entropy by the monotonic scaling argument above. Hence such unequal same-side coordinates provide a descent direction unless they are equal. Repeating this argument leaves a two-level vector. Thus any minimizing candidate can be written, up to permutation, with m coordinates equal to a larger value a and $K - m$ coordinates equal to a smaller value b:

$$\boldsymbol{u}_m = (\underbrace{a, \ldots, a}_{m}, \underbrace{b, \ldots, b}_{K-m}), \qquad 1 \leq m \leq K-1,$$

where the constraints give

$$ma + (K-m)\,b = 0, \qquad ma^2 + (K-m)\,b^2 = R^2.$$

Solving these two equations yields

$$a_m = R\sqrt{\frac{K-m}{mK}}, \qquad b_m = -R\sqrt{\frac{m}{K\,(K-m)}}, \qquad \delta_m = a_m - b_m = R\sqrt{\frac{K}{m\,(K-m)}}.$$

The entropy of this two-level candidate is

$$H_m = \log\big(m + (K-m)\,e^{-\delta_m}\big) + \frac{(K-m)\,e^{-\delta_m}}{m + (K-m)\,e^{-\delta_m}}\delta_m.$$

The candidate family is now one-dimensional and discrete. The expression above is minimized at $m = 1$: for $m > 1$, the top probability mass is split across at least two equally likely classes, contributing the non-negative within-group term $\alpha_m \log m$ in the decomposition

$$H_m = H(\alpha_m) + \alpha_m \log m + (1-\alpha_m)\log(K-m), \qquad \alpha_m = \frac{m}{m + (K-m)\,e^{-\delta_m}},$$

while $\delta_m \le \delta_1$ for $1 < m < K-1$, so no $m > 1$ candidate can be more concentrated than the $m = 1$ one-vs-rest candidate. The $m = K-1$ endpoint places the larger logit on $K-1$ classes and has entropy at least $\log(K-1)$ up to a non-negative binary entropy term, so it is also not smaller. Therefore the entropy minimizer can be written as

$$\boldsymbol{u}^\star = (a, b, \ldots, b), \qquad a + (K-1)\,b = 0, \qquad a^2 + (K-1)\,b^2 = R^2.$$

Solving these two constraints gives

$$a = R\sqrt{\frac{K-1}{K}}, \qquad b = -R\sqrt{\frac{1}{K(K-1)}}, \qquad a - b = R\sqrt{\frac{K}{K-1}}.$$

Let

$$\gamma = \exp(-(a-b)) = \exp\left(-\rho\sqrt{\frac{DK}{K-1}}\right).$$

Using $\mathrm{softmax}$ shift-invariance again, the same probability vector is obtained from logits $(a - b, 0, \ldots, 0)$, so

$$p_1^\star = \frac{1}{1 + (K-1)\,\gamma}, \qquad p_i^\star = \frac{\gamma}{1 + (K-1)\,\gamma} \quad (i \neq 1).$$

The entropy of this (lowest-entropy) feasible distribution is

$$\begin{aligned} H(\boldsymbol{p}^\star) &= -p_1^\star \log p_1^\star - \sum_{i=2}^{K} p_i^\star \log p_i^\star \\ &= \log(1 + (K-1)\,\gamma) + \frac{(K-1)\,\gamma}{1 + (K-1)\,\gamma}\,(a-b) \\ &= \log(1 + (K-1)\,\gamma) + \frac{\rho\gamma\sqrt{DK(K-1)}}{1 + (K-1)\,\gamma}. \end{aligned}$$

Since $\boldsymbol{p}^\star$ minimizes entropy among all logits satisfying the norm bound, every $\boldsymbol{p} = \mathrm{softmax}(\boldsymbol{z})$ satisfies the stated lower bound.

A.7 Derivations of Gradient

For HoSHi, we define our new objective as

$$\mathrm{CE}_{\mathrm{H}}\big(\boldsymbol{q}, \boldsymbol{e}^{\mathtt{t}}; \beta\big) = \mathrm{CE}\big(\boldsymbol{q}, \boldsymbol{e}^{\mathtt{t}}\big) + \underbrace{\beta(\mathrm{CE}(\boldsymbol{q}, \boldsymbol{u}) - \mathrm{CE}(\boldsymbol{q}, \boldsymbol{s}))}_{\mathcal{R}_{\mathrm{H}}}$$

with

$$\boldsymbol{s}_i = \frac{1}{Z} \text{ if } \boldsymbol{\ell}_{\mathtt{t}} < \boldsymbol{\ell}_i \text{ else } 0$$

and

$$Z = \left|\{i : \boldsymbol{\ell}_{\mathtt{t}} < \boldsymbol{\ell}_i\}\right|.$$

The gradient here is formulated as

$$\frac{\partial\, \mathrm{CE}_{\mathrm{H}}}{\partial\, \boldsymbol{\ell}_i} = \boldsymbol{q}_i - \mathbb{1}[i = \mathtt{t}] - \frac{\beta}{K} + \frac{\beta}{Z} \cdot \mathbb{1}[\boldsymbol{q}_i > \boldsymbol{q}_{\mathtt{t}}]\,,$$

since

$$\boldsymbol{\ell}_i > \boldsymbol{\ell}_{\mathtt{t}} \iff \boldsymbol{q}_i > \boldsymbol{q}_{\mathtt{t}}$$

by definition of the $\mathrm{softmax}$ operator.

Under this formulation, there are two cases to consider: (1) the model makes the correct prediction, i.e. $\mathrm{argmax}(\boldsymbol{q}) = \mathtt{t}$, and (2) the prediction is incorrect.

- **Case 1:** If the model prediction is correct, then this objective remains the same as standard label smoothing, with gradients
  - For the ground truth class: $\frac{\partial\, \mathrm{CE}_{\mathrm{H}}}{\partial\, \boldsymbol{\ell}_i} = \boldsymbol{q}_i - 1 - \frac{\beta}{K} + \beta$.
  - For all other classes: $\frac{\partial\, \mathrm{CE}_{\mathrm{H}}}{\partial\, \boldsymbol{\ell}_i} = \boldsymbol{q}_i - \frac{\beta}{K}$.
- **Case 2:** If the prediction is incorrect, then the gradients are separated for the predicted class, the target class, and the other classes:
  - For the ground truth class: $\frac{\partial\, \mathrm{CE}_{\mathrm{H}}}{\partial\, \boldsymbol{\ell}_i} = \boldsymbol{q}_i - 1 - \frac{\beta}{K}$.
  - For any class with logit greater than the ground truth: $\frac{\partial\, \mathrm{CE}_{\mathrm{H}}}{\partial\, \boldsymbol{\ell}_i} = \boldsymbol{q}_i - \frac{\beta}{K} + \frac{\beta}{Z}$.
  - For all other classes: $\frac{\partial\, \mathrm{CE}_{\mathrm{H}}}{\partial\, \boldsymbol{\ell}_i} = \boldsymbol{q}_i - \frac{\beta}{K}$.

Thus for any class that is incorrectly predicted as more likely than the ground truth, a positive factor is added to the gradient, evenly spreading out the controlled adjustment among all cases of over-confidence made by the model.

B EXPERIMENTAL DETAILS

Technical Details For Reproducibility. All experiments were conducted on a single machine with 8 NVIDIA H100 80GB GPUs and 2 x Intel 8570 @ 2.1 GHz, 300MB cache L3 CPUs. Each machine is equipped with 112 cores and 2000GB of memory. For supervised fine-tuning and DPO training, we utilize all GPUs and cores on the machine for training. For RLVR training, we use 7 GPUs for training and 1 GPU for inference.

Training. We used the `open-instruct` repository[1] for our training setup. Our supervised fine-tuning uses the code at commit `e363290` while our DPO/RLVR training uses commit `3835f56`. We follow the training recipes of Wang et al. (2023) and Lambert et al. (2025) for tuning our models and determining specific experimental hyper-parameter settings and baselines.

Evaluation. For evaluating calibration metrics, our implementation is based on the MMLU official repository[2]. We first evaluate our models on MMLU and then modify their files to directly adapt the evaluation dataset to the other tasks at hand (HELLASWAG and ARC-EASY). We follow MMLU and use the following prompt for all tasks:

```
The following is a multiple choice question (with answers). Respond
with only the answer.\n\n{[query]}\n{[answers]}\nAnswer:
```

MMLU Abstract Algebra Prompt Example

```
The following is a multiple choice question (with answers). Respond with
only the answer.

Find the degree for the given field extension Q(sqrt(2), sqrt(3),
sqrt(18)) over Q.

(A): 0 (B): 4 (C): 2 (D): 6

Answer:
```

B.1 EVALUATION TASKS

For results in Table 4, our evaluation is based on evaluation code from the `olmes`[3] repository.

`MMLU` (Hendrycks et al., 2021a) is heterogeneous with regard to the type of reasoning skills required to answer the questions, containing instances that require basic factual recall as well as those that demand logical reasoning and problem-solving skills. Models are prompted in a 5-shot manner and the reported result is a macro average over all the subjects in `MMLU` as the final task metric.

`PopQA` (Mallen et al., 2023) is an entity-centric question-answering benchmark that evaluates language models' tendency to *forget* information about long-tail entities. Models are prompted in a 15-shot setting (as recommended in the dataset paper) without any additional instructions, with each QA demonstration presented in a different conversation turn, a setting that we refer to as *Multiturn ICL* henceforth, and use greedy sampling to obtain model predictions.

`TruthfulQA` (Lin et al., 2022b) contains questions that humans tend to answer incorrectly due to common misconceptions. We use the multiple-choice version of this benchmark where the models being evaluated are presented with questions and options containing multiple correct answers (the `MC2` setting).

`HumanEval` (`HE`) (Chen et al., 2021) and `HumanEval+` (`HE+`) (Liu et al., 2023) evaluate models' ability to complete Python code given doc-strings. `HumanEval+` uses a more rigorous evaluation procedure than the original `HumanEval` benchmark with additional tests. Models are sampled at a temperature of 0.8 and use `pass@10` as the evaluation metric.

[1] https://github.com/allenai/open-instruct
[2] https://github.com/hendrycks/test
[3] https://github.com/allenai/olmes

GSM8K (Cobbe et al., 2021) contains grade school math word problems. Models are prompted using a 8-shot chain-of-thought prompt, formatted for a multi-turn in-context learning evaluation. Model responses are obtained using greedy sampling and extract the last number in the model response as the predicted answer.

MATH (Hendrycks et al., 2021b) contains problems from mathematics competitions spanning various categories such as algebra and calculus. A 4-shot multi-turn setup with chain-of-thought is formatted for multi-turn in-context learning and greedy sampling is used for model completions.

BigBench-Hard (BBH) (Suzgun et al., 2023) contains challenging reasoning problems for which models benefit from step-by-step reasoning. The setup follows the original paper and uses 3-shot chain-of-through prompts, formatted for multi-turn in-context learning. Greedy sampling is used for obtaining model predictions.

DROP (Dua et al., 2019) is a reading comprehension task that requires discrete reasoning. 3 random few-shot examples are drawn from the train split following the setup used for Llama 3 (Llama Team, 2024), and use greedy sampling to get model predictions.

IFEval (Zhou et al., 2023) evaluates instruction following ability of models in a setting where each instruction corresponds to constraints such that the it can be programmatically verified whether the outputs satisfy those constraints. Greedy decoding is used to generate model outputs give the instructions in the dataset, and the prompt-level accuracy of satisfying constraints in the loose evaluation setting is measured.

AlpacaEval (AE) (Dubois et al., 2024) contains a set of prompts sourced that reflect real human usages of LMs, and compares model outputs to GPT-4 turbo responses with an additional length control to avoid longer answers being unfairly favored. Responses are generated using greedy decoding up to 8192 tokens in length.

B.1.1 Safety Evaluation

XSTest (Röttger et al., 2024) consists of 200 unsafe prompts and 250 prompts which are safe but superficially resemble unsafe prompts: these prompts use vocabulary similar to that of unsafe prompts. Categories include homonyms, figurative language, safe targets, safe contexts, definitions, real discrimination/nonsense group, nonsense discrimination/real group, historical events, public privacy, and fictional privacy. We report the overall accuracy score based on whether `WildGuard` (Han et al., 2024b) classifies the response as a refusal or compliance.

HarmBench (Mazeika et al., 2024). A subset of harmful prompts which consists of 321 harmful prompts[4] is evaluated and categorized into Functional and Semantic categories. Functional category includes two types of behavior: Standard behaviors, which are modeled after existing datasets of harmful behaviors such as `AdvBench` and the TDC 2023 Red Teaming Track dataset, and Copyright behaviors, which test the handling of copyrighted content. The semantic category comprises seven types of harmful behaviors: cybercrime, unauthorized intrusion, chemical/biological weapons or drugs, copyright violations, misinformation/disinformation, harassment/bullying, illegal activities, and general harm. `WildGuard` is used to evaluate the models' refusal to assist with harmful prompts.

Do-Anything-Now (Shen et al., 2024b) consists of jailbreak prompts that were created by combining the jailbreak templates from DAN with harmful behaviors from HarmBench, and subsample 300 of them for testing. Accuracy is reported using the `WildGuard` classifier.

JailbreakTrigger (Huang et al., 2024) incorporates prompts based on 13 distinct jailbreak attack methods. In total, the dataset[5] consists of 400 examples, split evenly between two categories: "Questions about Bad Behaviors" and "Instructions to generate Toxic Content". This dataset serves to evaluate the effectiveness of LLMs' defenses and measures the toxicity of responses under jailbreak scenarios. The reported metric is `RTA` measured by `WildGuard`.

WildJailbreakTest (Jiang et al., 2024) is an adversarial evaluation set which contains a subset for adversarial benign queries (210 examples) and a subset for adversarial harmful queries (2000

[4] https://github.com/centerforaisafety/HarmBench/blob/main/data/behavior_datasets/harmbench_behaviors_text_test.csv

[5] https://huggingface.co/datasets/TrustLLM/TrustLLM-dataset

examples). The adversarial benign queries are used to measure models' exaggerated safety behaviors and the adversarial harmful queries are used to measure models' safeguards regarding adversarial attacks. We report RTA using WildGuard for both categories. For benign queries, RTA is expected to be (↓) and for harmful queries, the RTA is expected to be (↑).

WildGuardTest (Han et al., 2024b) contains 1725 items for prompt harm, response harm, and response refusal classification tasks. 55% are vanilla prompts, and 45% are adversarial. The prompts are collected based on adversarial synthetic data and in-the-wild user-LLM (In-the-wild) interactions. RTA is reported using WildGuard.

C Design Ablations

C.1 Hyper-parameter Sensitivity

Given the use of the smoothing hyper-parameter β, we first verify sensitivity of the results to this hyper-parameter. Table 5 evaluates this on MMLU and shows that for reasonably small β, the model is able to retain accuracy scores relative to the non-smoothed reference while also lowering calibration error significantly. For extremely large β, we can see that calibration error improves but accuracy begins to degrade, likely due to over-focusing on the regularization while the learning signal is decreased.

Table 5: `Llama-3-8B` evaluation on after the SFT stage on MMLU using HoSHi.

β	Acc.↑	ECE↓	RMS↓
Reference	65.1	0.050	0.080
0.05	64.6	0.016	0.071
0.1	64.7	0.013	0.064
0.2	64.5	0.018	0.066
0.5	44.6	0.015	0.062

C.2 Regularization on Regions

Recalling our analysis of how the label smoothing regularization term can be broken down into two distinct regions where the applied effect differs

$$\beta\big(\mathrm{CE}(\boldsymbol{q},\boldsymbol{u})-\mathrm{CE}\big(\boldsymbol{q},\boldsymbol{e}^{\mathsf{t}}\big)\big)=\beta\left(\boldsymbol{\ell}_{\mathsf{t}}-\sum_{i=1}^{K}\frac{\boldsymbol{\ell}_i}{K}\right)=\underbrace{\frac{\beta}{K}\sum_{i:\boldsymbol{\ell}_i<\boldsymbol{\ell}_{\mathsf{t}}}(\boldsymbol{\ell}_{\mathsf{t}}-\boldsymbol{\ell}_i)}_{(1)\text{ moderation},>0}+\underbrace{\frac{\beta}{K}\sum_{i:\boldsymbol{\ell}_i>\boldsymbol{\ell}_{\mathsf{t}}}(\boldsymbol{\ell}_{\mathsf{t}}-\boldsymbol{\ell}_i)}_{(2)\text{ misclassified},<0}, \tag{19}$$

we can further isolate the direct effect of our approach by verifying how performance may change by optimizing only for the misclassified region. We provide this result in Table 6.

Additionally, we can further observe the effect of only regularizing on the misclassified region only.

Table 6: Calibration results of models (based on `Llama3-8B`) after alignment stages. Temperature scaling is calibrated on the training set of each dataset prior to evaluation.

SFT	Stage	MMLU			HellaSwag			Arc-Easy		
		Acc.↑	ECE↓	RMS↓	Acc.↑	ECE↓	RMS↓	Acc.↑	ECE↓	RMS↓
HoSHi	DPO	68.4	**0.034**	**0.071**	**37.5**	**0.042**	**0.083**	**89.1**	**0.050**	0.112
	+ TS	68.2	**0.031**	**0.066**	37.4	**0.039**	**0.072**	**89.3**	0.043	0.096
	+ RLVR	**68.5**	**0.043**	**0.087**	**37.2**	**0.051**	**0.096**	**88.9**	**0.064**	**0.119**
	+ TS	**68.4**	**0.038**	**0.077**	37.0	**0.044**	**0.087**	**89.0**	**0.053**	0.112
Label Smooth Moderation Region	DPO	66.5	0.040	0.094	34.5	0.051	0.092	87.2	0.056	0.114
	+ TS	66.3	0.039	0.088	34.3	0.051	0.090	87.5	0.044	0.103
	+ RLVR	67.4	0.055	0.099	34.5	0.067	0.112	86.6	0.066	0.120
	+ TS	67.7	0.046	0.086	34.3	0.062	0.098	87.7	0.062	0.115
HoSHi on Misclassified Region	DPO	66.8	0.058	0.083	34.9	0.055	0.105	87.8	0.063	0.122
	+ TS	66.7	0.051	0.079	35.0	0.050	0.089	88.0	0.052	0.115
	+ RLVR	67.9	0.077	0.106	35.4	0.071	0.105	88.2	0.075	0.133
	+ TS	67.9	0.056	0.079	35.2	0.049	0.089	88.1	0.059	0.120

C.3 Margins for Logits

Another major consideration of our method is how to choose the logits on which to apply the updated regularization. To verify the specific design choice we use (use the target class logit as a separating

bound), we compare against some alternative versions. One of these is to expand the region on which the new regularization is applied to all logits within a specific range of the target logit; for example, up to 95% the value of ℓ_t or only up to 105% the value of ℓ_t (leaving some of the misclassified region untouched). We also compare with applying the new regularization to up to the top 25% and 50% of the logits or all elements greater than ℓ_t, whichever range is larger. We only run these up to the DPO stage. On this front (Table 7), we see that neither or these approaches are particularly better than simply using the target logit as the delimiting factor for the two regions.

Table 7: Calibration results of models (based on `Llama3-8B`) after DPO. We vary the range of the logits on which the new regularization is applied.

Region	Stage	MMLU			HellaSwag			Arc-Easy		
		Acc.↑	ECE↓	RMS↓	Acc.↑	ECE↓	RMS↓	Acc.↑	ECE↓	RMS↓
HoSHi	DPO	**68.4**	0.034	**0.071**	**37.5**	**0.042**	**0.083**	**89.1**	**0.050**	**0.112**
105% ℓ_t	DPO	68.4	**0.033**	0.075	37.4	0.043	0.084	89.1	0.053	0.116
95% ℓ_t	DPO	68.1	0.035	0.075	37.3	0.043	**0.083**	88.7	0.052	0.114
Maximum Top 50%	DPO	68.3	0.058	0.085	37.1	0.057	0.099	88.8	0.064	0.130
Maximum Top 25%	DPO	68.4	0.043	0.082	37.4	0.051	0.094	89.0	0.059	0.128

D Computational Efficiency Testing

Beyond the standard cross-entropy operation, our approach only requires

- An additional access to determine the target logit.
- A boolean masking operation over the K-dimensional logit vector.
- A sum of the values of this boolean mask.

In total, this remains an $O(K)$ operation, thus leads to minimal computational overhead.

Table 8: Time to compute gradients on a `Gemma-2B` model, using a sequence of length 8192.

Method	CE (with `torch.compile`)	CE+LS (with `torch.compile`)	MaxSup	HoSHi
Time	22.8 ms (20.6 ms)	24.2 ms (22.8 ms)	25.2 ms	25.4 ms

As seen above, the measured run times are generally negligible per forward pass; however we do note that due to possible ways of compiling `torch` provided functions, the standard cross-entropy implementations do remain slightly faster than both HoSHi and MaxSup, which have negligible difference in speed.

Computational Considerations. The large vocabulary sizes of LLMs introduces a number of novel challenges. The final cross-entropy begins to consume a significant amount of memory, as the logits must be materialized to compute the loss. While optimizations have been implemented to manage other model components (Dao et al., 2022; Dao, 2024), managing the footprint of the logit matrix (stacked vectors for each token in the sequence) has motivated efficient CE implementations (Wijmans et al., 2025; Hsu et al., 2024; Meta PyTorch, 2024) which remove unnecessary computations in the non-smoothed loss. However smoothed losses are become more complex; Huang et al. (2025) minimize redundant computations and compute logits in an online manner but for instance-dependent methods such as MaxSup and HoSHi, this integration is more complex and leaves open an important area for bridging technique effectiveness and real-world hardware efficiency.

E Extended Details on Calibration Metrics

Brier Score. Traditional calibration has been measured using the Brier Score (Brier, 1950), which is defined as the gap between the accuracy of a model (0/1 score) and the model's confidence on it's answer.

$$\text{BS} = \frac{1}{N}\sum_{n=1}^{N} \left|\text{acc}(x_n) - \text{conf}(x_n)\right|^2 = \frac{1}{N}\sum_{n=1}^{N} \left| \mathbb{1}\left\{ \underset{i}{\text{argmax}}\, f(x_n)_i = y_n \right\} - \underset{i}{\text{argmax}}\, f(x_n)_i \right|^2 . \tag{20}$$

Expected Calibration Error. Another widely used calibration metric is the expected calibration error (ECE) (Naeini et al., 2015), which divides the confidence scores of N samples into M uniform confidence bins $\{B_m\}_{m=1}^{M}$ and takes a weighted sum over the bin-wise errors.

$$\text{ECE} = \sum_{m=1}^{M} \frac{|B_m|}{N} \left|\text{acc}(B_m) - \text{conf}(B_m)\right| . \tag{21}$$

Multiclass & Static Calibration Error. Static Calibration Error (SCE) extends ECE to every probability in the multiclass setting. SCE bins predictions separately for each class probability, computes the calibration error within the bin, and averages across bins:

$$\text{SCE} = \frac{1}{K}\sum_{k=1}^{K}\sum_{m=1}^{M} \frac{|B_{mk}|}{N} \left|\text{acc}(m,k) - \text{conf}(m,k)\right| . \tag{22}$$

Here, $\text{acc}(m,k)$ and $\text{conf}(m,k)$ are the accuracy and confidence of bin m for class label k, respectively; $|B_{mk}|$ is the number of predictions in bin b for class label k.

Adaptivity & Adaptive Calibration Error. Adaptive calibration ranges are motivated by the bias-variance tradeoff in the choice of ranges, suggesting that to get the best estimate of the overall calibration error, the metric should focus on regions where the predictions are made (and focus less on regions with few predictions). This leads to the Adaptive Calibration Error (ACE), which uses an adaptive scheme which spaces bin intervals so that each contains an equal number of predictions.

ACE takes the predictions (usually out of a softmax), correct labels, and a number of ranges R.

$$\text{ACE} = \frac{1}{K \cdot R}\sum_{k=1}^{K}\sum_{r=1}^{R} \left|\text{acc}(r,k) - \text{conf}(r,k)\right| . \tag{23}$$

Here, $\text{acc}(r,k)$ and $\text{conf}(r,k)$ are the accuracy/confidence of adaptive calibration range r for class label k. Range r is defined by the $\lfloor N/R \rfloor$-th index of the sorted and thresholded predictions.

RMS and MAD Calibration Error. The Root Mean Square Calibration Error measures the square root of the expected squared difference between confidence and accuracy at a confidence level. A similar formulation which less severely penalizes large confidence-accuracy deviations is the Mean Absolute Value (MAD), which is a lower bound of the RMS Calibration Error.

To empirically estimate these mis-calibration measures, the N samples of are partitioned again into M bins. Here, bins are not equally spaced since the distribution of confidence values is not uniform but dynamic. Concretely, the RMS Calibration Error is estimated with the numerically stable formula

$$\text{RMSCE} = \sqrt{\sum_{m=1}^{M} \frac{|B_m|}{N} \left(\frac{1}{|B_m|}\sum_{k\in B_m} \mathbb{1}[y_k = \widehat{y}_k] - \frac{1}{|B_m|}\sum_{k\in B_m} c_k \right)^2} . \tag{24}$$

Along similar lines, the MAD Calibration Error — which is an improper scoring rule due to its use of absolute differences rather than squared differences — is estimated with

$$\text{MAD} = \sum_{m=1}^{M} \frac{|B_m|}{N} \left| \frac{1}{|B_m|}\sum_{k\in B_m} \mathbb{1}[y_k = \widehat{y}_k] - \frac{1}{|B_m|}\sum_{k\in B_m} c_k \right| . \tag{25}$$

F Additional Related Works

Calibrated Preference Tuning. Despite revelations relating to how calibrating large language models can be improved through modifications during supervised fine-tuning, the latter stages of alignment remain unmodified. Indeed, we observe some evidence of worsening calibration within these stages, leaving open a question of whether calibration techniques can be extended here despite the different objectives. Previous works (Tao et al., 2024; Leng et al., 2025; Chakraborty et al., 2024; Xiao et al., 2025b) have linked this phenomena during DPO as a tendency to excessively favor certain responses; temperature scaling has often emerged as the most effective calibration method but more complex calibration-aware tuning approaches (Xiao et al., 2025a) have also emerged. Nevertheless the changing nature of alignment techniques highlights a need to explore methods that are more universal techniques that can be applied as the training pipelines of models evolve. Furthermore, such methods can be perceived as supplementary to our method, which aims to modify the supervised fine-tuning phase but has no limitations on further techniques that can be applied post-hoc.

G Additional Experimental Results

G.1 Additional Baseline Comparisons

We further compare additional methods that can possibly lead to better calibration and train-ability. In particular, we look at logit normalization (Wei et al., 2022a) and the use of temperature scaling with a fixed temperature directly during the supervised fine-tuning process. While temperature scaling can improve learning on some fronts, the volatility of this approach leads to a potential lack of transferability and thus falls below other methods. Similarly, logit normalization can reduce overconfidence but leads to as significant reduction in performance.

Table 9: Ablation using other methods during supervised fine-tuning. We use the `Alpaca` dataset and fix a temperature to use during training for temperature scaling.

Model	MMLU			HellaSwag			Arc-Easy		
	Acc.↑	ECE↓	RMS↓	Acc.↑	ECE↓	RMS↓	Acc.↑	ECE↓	RMS↓
Reference	63.8	0.113	0.113	37.5	0.162	0.085	86.3	0.070	0.127
LS ($\beta = 0.1$)	63.6	0.073	0.094	37.4	0.087	0.037	86.4	0.037	0.098
MaxSup	64.2	0.096	0.099	37.4	0.144	0.067	86.2	0.054	0.110
HoSHi	63.7	0.093	0.092	37.8	0.134	0.053	86.4	0.057	0.108
TS ($\tau = 1.5$)	63.5	0.116	0.106	36.6	0.283	0.182	85.9	0.072	0.129
TS ($\tau = 2.0$)	63.2	0.109	0.107	37.5	0.221	0.165	86.3	0.070	0.129
LogitNorm (Wei et al., 2022a)	59.3	0.095	0.106	33.5	0.123	0.076	79.5	0.064	0.110

G.2 Additional Preference-Tuning Calibration Comparisons

In addition to previous results, Xiao et al. (2025a) suggest that a preference collapse issue exists in DPO training and leads to growing overconfidence. They suggest that to calibrate models requires a calibration-aware fine-tuning approach that requires learning from in-domain examples. To compare against their method, we aggregate the trainings sets of our evaluation tasks (MMLU, HellaSwag, Arc-Easy) and use this in their framework during RLHF. Table 10 shows that while there are improvements in accuracy (which is expected due to training on the tasks themselves), calibration error remains relatively similar to MaxSup and HoSHi, which do not require any training on the task at hand and are unlikely to degrade performance on out-of-domain tasks, which Xiao et al. (2025a) observe.

Table 10: Calibration results of models (based on `Llama3-8B`) after alignment stages. Temperature scaling is calibrated on the training set of each dataset prior to evaluation.

SFT	Stage	MMLU			HellaSwag			Arc-Easy		
		Acc.↑	ECE↓	RMS↓	Acc.↑	ECE↓	RMS↓	Acc.↑	ECE↓	RMS↓
Baseline	DPO	68.7	0.065	0.094	37.4	0.061	0.108	88.5	0.066	0.131
	+ TS	68.6	0.056	0.081	37.5	0.053	0.092	88.2	0.055	0.119
	+ RLVR	68.2	0.081	0.112	37.2	0.075	0.114	88.7	0.083	0.145
	+ TS	68.2	0.062	0.085	37.1	0.054	0.100	88.5	0.062	0.123
	CFT (Xiao et al., 2025a)	69.4	0.037	0.084	38.5	0.046	0.088	89.1	0.053	0.094
	RCFT (Xiao et al., 2025a)	69.4	0.037	0.081	38.4	0.048	0.093	89.3	0.047	0.096
MaxSup	DPO	66.4	0.036	0.076	34.5	0.048	0.088	87.1	0.051	0.109
	+ TS	66.2	0.032	0.071	34.4	0.043	0.079	87.3	0.042	0.091
	+ RLVR	67.5	0.052	0.094	34.7	0.061	0.102	86.9	0.064	0.121
	+ TS	67.4	0.043	0.086	34.8	0.060	0.094	87.0	0.059	0.110
HoSHi	DPO	68.4	0.034	0.071	37.5	0.042	0.083	89.1	0.050	0.112
	+ TS	68.2	0.031	0.066	37.4	0.039	0.072	89.3	0.043	0.096
	+ RLVR	68.5	0.043	0.087	37.2	0.051	0.096	88.9	0.064	0.119
	+ TS	68.4	0.038	0.077	37.0	0.044	0.087	89.0	0.053	0.112

G.3 Additional Post-Hoc Calibration Comparisons

In addition to calibration methods applied directly during training, we further compare against methods that are post-hoc, i.e. they are applied after the model is directly trained. On this font, we compare the use of DACA (Luo et al., 2025), which requires the use of a calibration set for tuning on the task of interest.

Table 11: `Llama-3-8B` evaluation on MMLU against post-hoc calibration methods at different training stages. We follow Lambert et al. (2025) again for all training stages.

Stage	Method	Tuning	MMLU		
			Acc.↑	ECE↓	RMS↓
SFT	Reference	✗	63.8	0.113	0.113
	+ DACA	✔	63.7	0.096	0.093
	HoSHi	✗	63.7	0.073	0.094
	+ DACA	✔	63.8	0.062	0.081
DPO	Reference	✗	68.7	0.065	0.094
	+ DACA	✔	68.5	0.041	0.084
	HoSHi	✗	68.4	0.034	0.071
	+ DACA	✔	68.5	0.029	0.062
RLVR	Reference	✗	68.2	0.081	0.112
	+ DACA	✔	68.4	0.056	0.092
	HoSHi	✗	68.5	0.043	0.087
	+ DACA	✔	68.5	0.040	0.082

We can observe that even without any additional tuning on a task of interest, we can match performance in terms of accuracy and calibration metrics, highlighting how HoSHi can serve as an adaptive approach rather than a competing method.

Calibrated Preference Tuning. Despite revelations relating to how calibrating large language models can be improved through modifications during supervised fine-tuning, the latter stages of alignment remain unmodified. Indeed, we observe some evidence of worsening calibration within these stages, leaving open a question of whether calibration techniques can be extended here despite the different objectives. Previous works (Tao et al., 2024; Leng et al., 2025; Chakraborty et al., 2024; Xiao et al., 2025b) have linked this phenomena during DPO as a tendency to excessively favor certain responses; temperature scaling has often emerged as the most effective calibration method but more complex calibration-aware tuning approaches (Xiao et al., 2025a) have also emerged. Nevertheless the changing nature of alignment techniques highlights a need to explore methods that are more universal techniques that can be applied as the training pipelines of models evolve. Furthermore, such methods can be perceived as supplementary to our method, which aims to modify the supervised fine-tuning phase but has no limitations on further techniques that can be applied post-hoc.

G.4 Additional Uncertainty/Bias Validation

At a high-level, calibration is important when the decision that is made can have some adverse effect (high risk) and thus ensuring that the model can quantify this risk in some manner when providing some advice or decision is relevant.

Synthetic True/False Bias Validation Task

```
Read the following statement and state whether it is True or False. Respond
with only the answer.

{string}

Answer:
```

Since such datasets or settings are difficult to truly apply models on for the time being (due to a lack of public data), we attempt a synthetic evaluation to show that the issue can exist. To further supplement our initial ablation, we use a synthetic version of MMLU where for any example, we prompt the model with an empty context and ask the model to predict whether the model response is correct. This is relevant because the model has no context and thus should ideally exhibit more randomness in its response; by providing something far from the random distribution, the model can have some inherent biases that can make trusting the model very dangerous if the setting were actually changed to more sensitive data. These results are presented in Table 12.

MMLU Abstract Algebra Prompt Bias Validation Example

```
The following is a multiple choice question (with answers). Respond with
only the answer.

Question:

(A): 0 (B): 4 (C): 2 (D): 6

Answer:
```

Table 12: `Llama-3-8B` evaluation on after the SFT stage on our synthetic setup.

Method	Acc.↑	ECE↓	RMS↓	Entropy↑
Reference	0.253	0.689	0.789	0.43
MaxSup	0.248	0.462	0.478	1.43
HoSHi	0.251	0.378	0.387	1.76

Our method is able to reduce calibration error as well as demonstrate increased entropy over the prediction, which is a good signal that the model is now more uncertain given the nature of the problem. Meanwhile, the reference retains low entropy, indicating an uneven output distribution in terms of the options, alongside higher calibration. MaxSup resolves these issues somewhat, however, remains inferior to our method on this setup. Meanwhile, all three methods indicate near-random accuracy (25%), again showing how accuracy fails to be a significant signal of calibration issues.

G.5 Additional Free-Form Generation Results

A major challenge in applying traditional uncertainty and confidence calibration techniques to generative models is the inability to extract precise confidence estimates from the model, in part due to difficulties in measuring the underlying output probability distribution. For such tasks, alternative *evaluation techniques* are often used in tandem with standard metrics to approximate calibration error on such tasks. As such, we seek to evaluate whether or not our approach can in fact provide some benefits using these downstream evaluation as well. As such, we use a number of recent approaches to see how the application of HoSHi influences the downstream metrics when used in tandem with these evaluation strategies. We follow the setup introduced by Wang and Stengel-Eskin (2025) and evaluate on the same settings as their work, where our method generally is still better after their post-hoc approach (Table 13).

Table 13: Expected Calibration Error (ECE), Brier score (BS), and area under the ROC curve (AUC) on generative tasks.

Method	Tuning	TriviaQA			SimpleQA			FactScore		
		ECE↓	BS↓	AUC↑	ECE↓	BS↓	AUC↑	ECE↓	BS↓	AUC↑
DiNCo	Reference	0.172	0.277	0.853	0.094	0.213	0.623	0.088	0.213	0.735
	MaxSup	0.103	0.224	0.869	0.088	0.195	0.682	0.076	0.202	0.767
	HoSHi	0.089	0.201	0.892	0.078	0.178	0.704	0.072	0.195	0.781

H ADDITIONAL IMAGE CLASSIFICATION RESULTS

While we motivate our method primarily for the language model settings, our technique is directly applicable to different models and tasks. On this front, we show image classification results on IMAGENET (Deng et al., 2009) and CIFAR100 (Krizhevsky, 2009) using a RESNET-18 (He et al., 2016).

Table 14: Performance comparison (accuracy) of classical CNNs on IMAGENET and CIFAR100. Best performance is bolded and second best is underlined.

Method	CIFAR100	IMAGENET
Reference	76.16	69.09
Label Smoothing	75.05	69.54
Zipf-LS (Liang et al., 2022)	72.08	69.96
OLS (Zhang et al., 2021)	77.33	69.45
MAXSUP	**77.82**	<u>69.96</u>
Logit Penalty	76.41	68.48
HOSHI	<u>77.79</u>	**70.01**

We further investigate on different RESNET to investigate the robustness of this approach for different models on standard classification tasks.

Table 15: Performance comparison (accuracy) of different-sized networks on IMAGENET. Best performance is bolded and second best is underlined.

Method	RESNET-18	RESNET-50	RESNET-101	MOBILENETV2
Reference	69.11	76.44	76.00	71.42
Label Smoothing	69.38	76.65	77.01	71.40
Zipf-LS	69.43	76.89	76.91	71.24
OLS	69.45	76.81	77.12	71.29
MAXSUP	<u>69.96</u>	**77.08**	<u>77.33</u>	**71.59**
Logit Penalty	66.97	74.21	75.17	70.39
HOSHI	**70.01**	<u>77.03</u>	**77.41**	<u>71.48</u>

Overall these results show that even in settings where the number of classes is significantly smaller, our approach is effective and comparable with other SOTA methods.