# Local Recalibration of Pre-trained Foundation Models

**Anonymous Author(s)**
Affiliation
Address
email

## Abstract

Uncertainty quantification in machine learning is critical for many real-world applications that require interpretable decision-making. In classification setting, measuring the prediction uncertainty of neural networks often relies on softmax probabilities, which may not reliably reflect true confidence levels and thus require recalibration. However, common post-hoc recalibration, such as temperature scaling, assumes a uniform calibration strategy across the whole input space, which can be unrealistic due to complex dependencies between the learned representations and the classifier's confidence scores. In this work, we analyze the pitfalls of recalibration of modern pre-trained foundation models in feature extraction regime. First, we provide theoretical insight into how local recalibration can reduce overall classification loss compared to global recalibration. Second, we propose a simple yet effective method for improving recalibration by performing it locally within clusters in the embedding space. Our experimental results demonstrate that this methodology enhances calibration across a range of foundation models in domains such as time-series, image, natural language processing, and tabular classification.

## 1 Introduction

As machine learning systems become increasingly integrated into high-stakes applications – ranging from healthcare and finance to autonomous systems and scientific discovery – the need for reliable and interpretable decision-making grows in parallel (Doshi-Velez and Kim, 2017; Rudin, 2019; Amodei et al., 2016; Lipton, 2016). In these contexts, it is not sufficient for models to produce accurate predictions; they must also express how confident they are in those predictions. This has motivated a growing body of research on uncertainty quantification (UQ) in machine learning, particularly in classification tasks (Kendall and Gal, 2017; Gal and Ghahramani, 2016; Abdar et al., 2021).

Neural networks, which form the backbone of many state-of-the-art classifiers, typically represent predictive uncertainty using softmax probabilities. However, it is well established that softmax outputs are often poorly calibrated, meaning that the predicted probabilities do not reliably reflect the true likelihood of correctness (Guo et al., 2017). Such overconfidence (or underconfidence) can undermine trust in automated decisions, especially when models are deployed in real-world environments where interpretability and reliability are critical.

To address this, post-hoc calibration techniques – most notably temperature scaling – have become popular due to their simplicity and effectiveness. These methods adjust the confidence scores without altering the model's underlying predictions, thereby improving calibration with minimal additional computation (Guo et al., 2017). However, a key limitation of these techniques is their reliance on a global calibration parameter, which assumes that miscalibration is uniform across the input space. This assumption often fails in practice, particularly with complex models and diverse datasets, where confidence scores may vary in a highly input-dependent manner (Ovadia et al., 2019).

This problem is exacerbated in the context of foundation models – large, pre-trained models that are increasingly used in a feature extraction regime, where a frozen backbone provides representations to

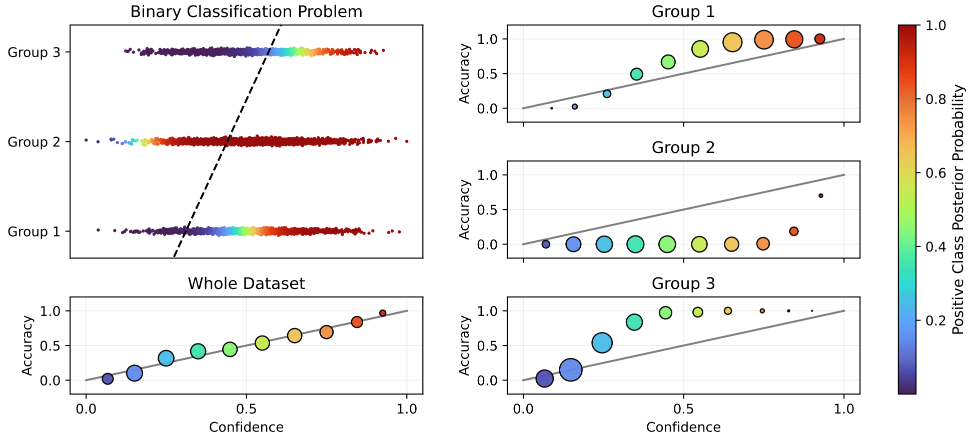

Figure 1: Case where three different groups exhibit different distribution $(g(\mathbf{X}), \mathbb{E}[\mathbf{Y}|g(\mathbf{X})])$. Note that overall calibration is poorly informative of subgroup calibration. Different recalibration needs to be applied per group.

a lightweight task-specific classifier. While these models offer strong performance across multiple domains, their internal representations are high-dimensional and structured in ways that can lead to heterogeneous calibration errors. This situation is illustrated on a toy example in Figure 1 where three groups of samples within a given dataset are all badly calibrated, yet the overall calibration is rather good. In such cases, a single global temperature parameter is unlikely to suffice, for instance, if the model is to be deployed on unbalanced samples of the three groups considered.

In this work, we investigate the limitations of global recalibration strategies and explore an alternative approach: local calibration based on the geometry of the learned representation space. We begin by providing a theoretical justification for local calibration, showing how partitioning the input space into clusters and performing recalibration within each cluster can lead to lower expected classification loss. Building on this insight, we propose a straightforward and model-agnostic method that clusters examples in the embedding space of a frozen foundation model and applies temperature scaling within each cluster.

Our experiments span a wide range of domains – including image, time-series, tabular, and natural language data – and demonstrate that local recalibration consistently improves calibration metrics over standard global methods. These results highlight the importance of incorporating local structure into uncertainty quantification strategies and offer a simple yet effective tool for improving the reliability of predictions made by modern machine learning models.

## 2 Background

In this section, we formulate the problem setup and introduce to the background of the work.

### 2.1 Framework

We consider classification problems of $d$-dimensional vectors from an input space $\mathcal{X} \subset \mathbb{R}^d$ that can be categorized into $c$ classes. We define an output space of one-hot class vectors as $\mathcal{Y} = \{(\mathbb{I}(k = j))_{j=1}^{c}\}_{k=1}^{c}$. Denoting the input and the output random variables respectively by $\mathbf{X}$ and $\mathbf{Y}$, we assume having access to a training set $\mathcal{D}_{\text{train}} = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{n}$, where $(\mathbf{x}_i, \mathbf{y}_i)$ are identically and independently distributed with respect to a joint distribution $P(\mathbf{X}, \mathbf{Y})$ over $\mathcal{X} \times \mathcal{Y}$. We define a prediction model as a function that outputs a label given an input $f : \mathcal{X} \to \mathcal{Y}$, while the associated probability scores for each class are given by a scoring function $g : \mathcal{X} \to \Delta_c$ with $\Delta_c = \{(p_k)_k \in [0, 1]^c | \sum_k p_k = 1\}$ being the probability simplex. Further, we denote the score for class $k \in \{1, \ldots, c\}$ by $g_k(\mathbf{x})$. We then seek to find $f$ that minimizes the generalization error $\mathbb{E}_{(\mathbf{X},\mathbf{Y})}\mathbb{I}(f(\mathbf{x}) \neq \mathbf{y})$ estimated using a hold-out test set. In practice, for modern neural networks, $f$ is defined as $\operatorname{argmax} g(\mathbf{x})$ and training is performed by minimizing a *proper loss* on the training set. In this work, we consider the Negative Log-Likelihood loss (NLL, also known as the cross-entropy loss) defined as

$$\mathcal{L}(\mathbf{Y}, g(\mathbf{X})) = -\log g(\mathbf{X})^\top \mathbf{Y}.$$

### 2.2 Calibrated model

In this work, we perform post-hoc analysis of the predicted probabilities $g(\mathbf{x})$, which we want to be interpretable, i.e., their absolute values trustfully reflect the model confidence. In this sense, the problem goes beyond the classical machine learning that is solely focused on learning the best estimate of $\mathbf{Y}|\mathbf{X} = \mathbf{x}$. Consequently, we want to additionally measure how well calibrated the model is. Below we give definitions of a jointly-calibrated and top-class-calibrated model.

**Definition 2.1** (Calibrated Model)**.** *A model $g : \mathcal{X} \to \Delta_c$ is calibrated iff for every score vector $\mathbf{s} = (s_k)_{k=1}^c \in \Delta_c$, the following holds:*

$$\mathbb{E}[\mathbf{Y}|g(\mathbf{X}) = \mathbf{s}] = \mathbf{s}, \tag{jointly}$$
$$\mathbb{E}[\mathbf{Y}_k|g_k(\mathbf{X}) = s_k] = s_k. \tag{top-class}$$

The joint calibration is a stronger notion and implies top-class calibration. However, in practice, the top-class calibration is usually measured due to it being more tractable and easier to estimate (Guo et al., 2017).

Intuitively, one may think that the model trustfulness can be measured solely by estimating the deviation from the perfectly calibrated model defined by Definition 2.1. In reality, the problem is more intricate, requiring a more thorough analysis (Perez-Lebel et al., 2022). To demonstrate this, let us consider the divergence measure associated to the NLL loss function: $d_{\text{NLL}}(\mathbf{Y}, g(\mathbf{X})) = -\log g(\mathbf{X})^\top \mathbf{Y} + \log g(\mathbf{X})^\top g(\mathbf{X})$. Following Kull and Flach (2015, Section 5.1), these two divergences can be decomposed in the following way:

$$\mathbb{E}[d(\mathbf{Y}, g(\mathbf{X}))] = \underbrace{\mathbb{E}[d(\mathbf{Y}, \mathbb{E}[\mathbf{Y}|\mathbf{X}])]}_{\text{Irreducible Error (IE)}} + \underbrace{\mathbb{E}[d(\mathbb{E}[\mathbf{Y}|\mathbf{X}], \mathbb{E}[\mathbf{Y}|g(\mathbf{X})])]}_{\text{Refinement Error (RE)}} + \underbrace{\mathbb{E}[d(\mathbb{E}[\mathbf{Y}|g(\mathbf{X})], g(\mathbf{X}))]}_{\text{Calibration Error (CE)}}.$$

Thus, we can decompose the classification error into three terms: (a) Irreducible Error that quantifies the inherent noise of the problem, (b) Refinement Error that is the loss of assigning the same score to instances from different classes, and (c) Calibration Error that measures the divergence from the jointly-calibrated model. This decomposition shows us that finding a model with the minimal calibration error may be confusing as minimization of calibration error alone can lead to an arbitrary refinement error. For example, by taking $g(\mathbf{X}) = \mathbb{E}[\mathbf{Y}]$, we nullify the calibration error, while increasing significantly the refinement one. The reverse is also true: (a) for a function $h : \mathcal{X} \to \Delta^c$ that doesn't change the $\sigma$-algebra, i.e., $\sigma(h(\mathbf{X})) = \sigma(\mathbf{X})$, and (b) for $g(\mathbf{X}) = \mathbb{E}[\mathbf{Y}|\mathbf{X}] + 0.1$, we can obtain a reliable model but with a high calibration error. We provide necessary proofs in Appendix **??**.

Based on this reasoning, we formulate the objective of model calibration: we aim to minimize the calibration error while not degrading the proper loss. In this paper, we consider a recalibration approach that consists in rescaling the estimated probabilities (given by $g(\mathbf{x})$) using a remapping function $m : \Delta_c \to \Delta_c$ (Minderer et al., 2021; Blasiok et al., 2023). In practice, the remapping function is found by minimizing the proper loss on a hold-out validation set $\mathcal{D}_{\text{val}}$, i.e., $\hat{m} = \text{argmin}_m \sum_{(x,y) \in \mathcal{D}_{\text{val}}} \mathcal{L}_{\text{NLL}}(\mathbf{y}, m \circ g(\mathbf{x}))$.

### 2.3 Recalibration methods

There exists various approaches to recalibrate probabilities in the binary classification setting including Histogram Binning (Zadrozny and Elkan, 2001) and Platt scaling (Platt et al., 1999). In the multi-class case, Isotonic Regression (Zhang et al., 2020) has been proposed, which is a flexible approach but prone to overfitting. In the deep learning literature, a common approach is Temperature Scaling (TS, Guo et al., 2017) that consists in rescaling the logits of the classifier with some temperature parameter $T \in \mathcal{T} \subset \mathbb{R}_+$. Let us parametrize linear probing by $g(\mathbf{x}) = \text{softmax}(\mathbf{W}^\top \mathbf{z} + \mathbf{b})$, where $\mathbf{z}$ is the input of the last layer, and $(\mathbf{W}, \mathbf{b})$ are classification parameters. Then, the temperature scaling

$m : \Delta_c \times \mathcal{T} \to \Delta_c$ is defined as follows:

$$m(g(\mathbf{x}), T) = \text{softmax}\left((\mathbf{W}^\top \mathbf{z} + \mathbf{b})/T\right).$$

In the case if logits are not directly available (e.g., using Logistic Regression from scikit-learn (Pedregosa et al., 2011)), a common approach is to rescale log-probabilities $\log \mathbf{g}(\mathbf{x})$ instead of logits. As $t$ approaches $\infty$, the probabilities converge to the uniform ones, decreasing the model confidence, while $t \to 0$ increases the confidence approaching the one-hot vector. Temperature Scaling has little risks to overfit to validation data, but as we will motivate later, its simplicity may lead to a suboptimal recalibration.

## 3 Proposed methodology

### 3.1 Recalibration of foundation models

In this work, to study calibration of foundation models, we consider the following framework, where the learning process consists of model pre-training, linear probing and recalibration:

$$\hat{\phi} = \underset{\phi : \mathcal{X} \to \mathbb{R}^h}{\text{argmin}} \sum_{(\mathbf{x},\mathbf{y}) \in \mathcal{D}_{\text{pre}}} \mathcal{L}_{\text{pre}}(\mathbf{y}, \phi(\mathbf{x})), \qquad \text{(pre-training)}$$

$$\hat{g} = \underset{g : \mathbb{R}^h \to \Delta_c}{\text{argmin}} \sum_{(\mathbf{x},\mathbf{y}) \in \mathcal{D}_{\text{train}}} \mathcal{L}_{\text{NLL}}(\mathbf{y}, g \circ \hat{\phi}(\mathbf{x})), \qquad \text{(linear probing)}$$

$$\hat{m} = \underset{m : \Delta_c \to \Delta_c}{\text{argmin}} \sum_{(\mathbf{x},\mathbf{y}) \in \mathcal{D}_{\text{val}}} \mathcal{L}_{\text{NLL}}(\mathbf{y}, m \circ \hat{g} \circ \hat{\phi}(\mathbf{x})), \qquad \text{(recalibration)}$$

where $\mathcal{D}_{\text{pre}}$ denotes a pre-training dataset. Then, the final model estimate posterior probabilities for a given $\mathbf{x}$ according to $\hat{m} \circ \hat{g}(\mathbf{x})$. We have formulated the learning process following a standard approach to apply foundation models for classification: the pre-trained model is used as a feature extractor that projects an input $\mathbf{x} \in \mathcal{X}$ to a compact embedding space $\mathbb{R}^h$, then, for a new downstream task, a linear probing is performed, which consists in learning a simple linear classifier in the embedding space. Finally, as soon as we obtain the feature extractor and the score function, we recalibrate predicted probabilities using a validation set. In the proposed framework, the recalibration approach offers several advantages. First, it is computationally efficient as the model can be recalibrated for every new downstream task without concerning about calibration during pre-training. Second, recalibration after linear probing offers interpretation of the learned embeddings and does not require designing a complex well-calibrated classification rule.

In the rest of the paper, for the sake of simplicity, we omit $\phi$-notation and use $\mathbf{x}$ to directly denote the input's embedding.

### 3.2 Local recalibration

The main drawback of Temperature Scaling is that it performs recalibration globally, assuming that different regions of the embedding space need to be rescaled in a similar way. As we motivated in Figure 1, there can exist different groups that exhibit different behavior around the decision boundary as well as the classification error. The nature of such grouping may vary from non-linearly separable embeddings to clustered data or a presence of sensitive variables. Thus, in this section, we introduce a latent variable $Z \in \mathcal{Z} = \{1, \ldots, k\}$ that model the grouping effect with $k$ being the number of groups.

Global calibration is only focused on improving $g(\mathbf{X})$ overall on $X \times Y \in \mathcal{X} \times \mathcal{Y}$, keeping eyes on calibration loss $\mathbb{E}|\mathbb{E}[\mathbf{Y}|g(\mathbf{X})] - g(\mathbf{X})|$. We introduce $Z-$calibration error as a way to incorporate latent information $Z$ into the analysis:

$$\text{Z-calibration error} = \mathbb{E}[\mathcal{L}(\mathbb{E}[\mathbf{Y}|g(\mathbf{X}), Z], g(\mathbf{X}))])$$

With this error, we have the new loss decomposition:

**Proposition 3.1.** *The proper loss $\mathbb{E}[\mathcal{L}(Y, g(\mathbf{X}))]$ can be decomposed into three terms:*

$$\underbrace{\mathbb{E}[d(Y, \mathbb{E}[\mathbf{Y}|\mathbf{X}, Z])]}_{\textit{Z–Irreducible error}} + \underbrace{\mathbb{E}[d(\mathbb{E}[\mathbf{Y}|\mathbf{X}, Z], \mathbb{E}[\mathbf{Y}|g(\mathbf{X}), Z])]}_{\textit{Z-Refinement error}} + \underbrace{\mathbb{E}[d(\mathbb{E}[\mathbf{Y}|g(\mathbf{X}), Z], g(\mathbf{X}))]}_{\textit{Z-Calibration Error (Z-CE)}} .$$

The proof is deferred to Appendix A.3.

**Local recalibration**

**Definition 3.2** (Local temperature scaling)**.** *Let the score function is given by $g(\mathbf{x}) = \mathit{softmax}(\mathbf{W}^\top \mathbf{x} + \mathbf{b})$. We define the temperature as a deterministic function $t : \mathcal{Z} \to \mathcal{T}$ of the group index $z$. Then, for a given pair $(\mathbf{x}, z)$, the local temperature scaling $m : \Delta_c \times \mathcal{Z} \to \Delta_c$ is defined as*

$$m(g(\mathbf{x}), z) = \mathit{softmax}\left((\mathbf{W}^\top \mathbf{x} + \mathbf{b})/t(z)\right) .$$

Now we would like to find the optimal local recalibration in terms of minimization of the expected negative log-likelihood $\mathbb{E}[\mathcal{L}_{\text{NLL}}(\mathbf{Y}, m(g(\mathbf{X}), Z))]$. Then, we can notice that minimization of this loss is equivalent to the likelihood maximization, i.e., for every probability score $\mathbf{s} \in \Delta_c$ and group index $z$, we have the Optimal Bayes solution given by

$$m^*(\mathbf{s}, z) := \operatorname*{argmin}_m \mathbb{E}\left[\mathcal{L}_{\text{NLL}}(\mathbf{Y}, m(\mathbf{s}, z))|\, g(\mathbf{X}) = \mathbf{s}, Z = z\right] = \mathbb{E}[\mathbf{Y}|g(\mathbf{X}) = \mathbf{s}, Z = z].$$

As before, we would like to approach this recalibration mapping using parametrized function spaces $\{m^\gamma : p \mapsto \phi(p, \gamma), \gamma \in \Gamma\}$. But now we allow $\gamma$ to depend on $Z$ – we note $\psi : z \in Z \mapsto \gamma(z) \in \Gamma$ the function attributing to a latent space $z$ its corresponding parameters. Hence, we speak of $\psi$ as being *local recalibration* parameters. We are looking for optimal local recalibration parameters $\psi^* := \text{arg min}_{\psi \in \Gamma^{\mathcal{Z}}} \mathbb{E}[\mathcal{L}(Y, \phi(g(\mathbf{X}), \psi(Z))]$. From the law of total probability, we obtain that optimal local recalibration parameters $\psi^*$ give, for any latent state $z$, the ones minimizing calibration loss $\mathbb{E}[\mathcal{L}(Y, \phi(g(\mathbf{X}), \gamma)|Z = z]$ conditioned on this latent space

Property $\psi^*(z) = \text{arg min}_{\gamma \in \Gamma} \mathbb{E}[\mathcal{L}(Y, \phi(g(\mathbf{X}), \gamma)|Z = z]$

. Notably, by considering $\psi(z) = \gamma^*$ the parameters learnt for classical recalibration, we obtain that $\mathbb{E}[\mathcal{L}(Y, m_Z^{\psi^*} \circ g(\mathbf{X}))] \leq \mathbb{E}[\mathcal{L}(Y, m^{\gamma^*} \circ g(\mathbf{X}))]$ i.e. that the classification error can only be reduced by considering optimal local recalibration parameters $\psi^*(z)$ w.r.t. optimal global recalibration parameters.

In the case where $Z$ takes discrete values between 1 and $k$, one can just learn the local recalibration parameters $\gamma_1^* \dots, \gamma_k^*$ and recalibrate with the mapping $m_z : p \mapsto m^{\gamma_z}(p)$.

Assume a latent discrete variable $Z$, taking integer values between 1 and k. We define $T_z$ the local optimal temperature obtained through minimization of the classification risk:

$$T_z := \text{arg min }_t \mathbb{E}[\mathcal{L}(Y, m(g(\mathbf{X}), t))|Z = z]$$

As well as $T_0$ the temperature obtained through minimization of global classification risk:

$$T_0 := \text{arg min }_t \mathbb{E}[\mathcal{L}(Y, m(g(\mathbf{X}), t))]$$

### 3.3 Related work on local calibration

Several works have pointed out that model calibration can vary across the input space ((Hébert-Johnson et al., 2018), (Xiong et al., 2023)). For instance, (Xiong et al., 2023) showed that points far from the center of the data distribution tend to be less calibrated. One way to address this issue is to strengthen the notion of calibration itself, notably by conditioning on the true class label or considering more expressive recalibration mappings – See Dirichlet calibration (Kull et al., 2019) for example. A related and influential concept is multicalibration (Hébert-Johnson et al., 2018), which aims to ensure calibration across many subgroups in the input space. However, practical benefits of these methods are sometimes unclear (Hansen et al., 2024). We finally highlight the method in (Luo et al., 2022), which recalibrates predictions based on local neighborhoods in the input space. While similar in spirit to our method, it relies on heavy dimensionality reduction and sensitive hyperparameters. Additional information on related work is given in apprendix.

### 3.4 Algorithm

In this subsection, we put into practice the theoretical developments made here above. We take now $Z$ as discrete and fully deternmined by $\mathbf{X}$ – $Z = \pi(\mathbb{X})$. The process to fit recalibration mapping $m_{\text{loc}} : \mathcal{X} \mapsto \Delta^c$ on validation set is in two stages:

1. Cluster the samples $x \in D_{\text{train}}$ from the train set to learn

To model it, we opt then for a simple, robust and computationally efficient approach of clustering input space $\mathcal{X}$ using classical K-Means (KM) algorithm. Implementation used, from scikit-learn, relies on efficient *kmeans++* seeding (Arthur and Vassilvitskii, 2006). Indeed, local recalibration is designated notably for simple classification heads, and and efficiency expectations for obtaining calibrated classifier $m \circ g$ are high.

Note that, allthough we make use of clustering methods, we are not in the clustering set-up properly speaking, as we do not want to partition samples, but rather find *meaningfull* groupings of samples. Hence, overlapping or incomplete clusterings can also be considered. Moreover, several ensembling runs of clustering can be performed to refine recalibration. When ensembling is considered, recalibrated logits are averaged accross all runs.

**Recalibration function class.** We choose temperature scaling as recalibration method, known for its robustness when few samples are available. Optimal temperatures are found using grid-search of 4000 values equally log-spaced between $10^{-2}$ and $10^2$. We minimize $\mathcal{L}_{\text{NLL}}$, with an additional $L^2$ regularization loss added to the classification loss. The regularization is performed with respect to the prior optimal temperature $t_{0,\text{NLL}} := \text{arg min}_t\ \mathbb{E}[\mathcal{L}_{\text{NLL}}(Y, \phi(g(\mathbf{X}), t))]$, found with the classical optimization method. We obtain the following loss, evaluated on validation set:

$$\mathcal{L}_{z,\text{NLL}}(t) = \mathcal{L}_{\text{NLL}}(Y, \phi(g(\mathbf{X}), t))) + \lambda_{z,\text{NLL}} \cdot ||t - t_0||^2$$

### 3.5 Evaluation and Metrics

**Assessing performance.** Calibration of a model is usually evaluated using Expected Calibration Error (ECE). This metric is built around confidence-calibration property and $L^1$ loss – we have true ECE $= \mathbb{E}|\mathbb{E}[Y_C|g_C(\mathbf{X})] - g_C(\mathbf{X})|$. In practice, ECE is evaluated through binning the distribution $g(\mathbf{X})$ on the test set, and leads to the following empirical estimate:

$$\text{empirical ECE} = \frac{1}{|D_{test}|} \sum_{i=1}^{b} \left| \sum_{x_i \in B_i} s_i - (y_i)_{c_i} \right|$$

As mentioned by several authors, notably most recently (Chidambaram and Ge, 2024), ECE has several pitfalls. Most notably, ECE only assesses calibration error – a classifier outputting marginal distribution $\mathbb{E}[Y]$ has zero ECE – , and ECE obscures subgroup and latent calibration gaps. nVarious attempts in improving ECE – for example ECE with equally-sized bins (Arrieta-Ibarra et al., 2022), or smoothECE proposed recently by (Błasiok and Nakkiran, 2023) – do not clearly address. Therefore, we systematically compute, in addition, negative log likelihood.

## 4 Calibration of foundation models

Foundation Models (FM) are pretrained Deep Learning Models able to transfer their knowledge to unseen tasks, involving for example classification. To our knowledge, no specific study on their calibration has been performed so far, nor specific methodology proposed to improve it. We believe that $Z-$recalibration approach proposed below can be of particular interest for these models, applied on their representation space, rich and polysemic, designed for a wide diversity of task. We briefly introduce them, notably how their representation space is trained, in the following section.

### 4.1 Introduction to considered models

**Computer Vision.** CLIP (Contrastive Language-Image Pretraining) (Radford et al., 2021) is a vision-language model that learns a shared embedding space for images and texts using a contrastive loss. Given image and text embeddings, the objective encourages aligned pairs while pushing apart mismatched ones

**Natural Language Processing.** Transformer-based language models like BERT (Devlin et al., 2019) and GPT2 (Radford et al., 2019) learn contextual token-level embeddings through self-supervised objectives. BERT uses a masked language modeling objective to reconstruct randomly masked words in a sentence, learning bidirectional representations that capture syntactic and semantic context. GPT2, in contrast, is trained with an autoregressive objective to predict the next word given the previous context, resulting in unidirectional, generative representations. In both cases, sentence-level embeddings can be derived by pooling token representations.

**Time Series.** MOMENT (Goswami et al., 2024) and MANTIS (Feofanov et al., 2025) are foundation models for time series that leverage Transformer architectures. These models are pretrained using masked reconstruction tasks – MOMENT – and contrastive learning – MANTIS – to capture temporal patterns and contextual dependencies. Representations are obtained by encoding the entire sequence or segments into fixed-dimensional embeddings that can generalize across datasets and tasks

**Tabular data.** TabPFN (Hollmann et al., 2022) is a transformer-based model trained as a conditional meta-learner to solve small tabular classification tasks in a single forward pass. It learns representations by simulating a wide variety of classification problems during training, capturing priors over data distributions and decision boundaries. The resulting model can produce predictive distributions–and hence informative embeddings–without gradient-based optimization on the new task. Note that, in that case, representations are obtained in a superfised manner.

### 4.2 Strategies to adapt to new classification tasks

When adapting pretrained models to downstream classification tasks, a common strategy is to finetune the entire model on labeled data. However, full finetuning is often computationally expensive and inconvenient in practice. An alternative is to treat the pretrained model as a frozen feature extractor and train a lightweight classifier on its embeddings. This representation-based approach is simple, efficient, and avoids the complexities of end-to-end training. In this work, we adopt this paradigm, focusing on calibrating the predictions of classifiers trained on fixed embeddings, rather than calibrating large models during supervised training.

## 5 Experiments

In this part, we assess the performance of local recalibration on Foundation Models of various modalities fitted on several classification tasks.

### 5.1 Technical details

**Train, Validation and Test.** For each dataset considered, we reshuffle first train and test if existing split is available. For vision datasets, we limit the number of samples to 5000. For NLP datasets, we limit the number of samples to 2000. We then perform stratified random split, with the following proportions: $40\%$ for train set, $10\%$ for validation set, and $50\%$ for test set. Note that test set must be large enough to properly evaluate uncertainty modelling abilities of $m \circ g$.

**Head training.** Classification head is Logistic Regression (LR) (sometimes referred as "Linear Probe"), this method being among the most widely used for such tasks ((Devlin et al., 2019), (Radford et al., 2021)).

We explore $L^2$ regularization ranging from $10^-3$ to $10^4$, and select the one providing the best results in the test set in terms of accuracy. Indeed, we want our recalibration method to work best specifically for the moded trained in the optimal set-up.

Models can output probability scores that are numerically very close to 0 or 1, preventing the computation of appropriate NLL. To cope with this issue, we clip the scores to the range $[10^{-7}, 1 - 10^{-7}]$, to the output of the heads as well as before any $\mathcal{L}_{\text{NLL}}$ computation.

### 5.2 Results and Findings

We display for each modality several datasets which exhibit strong changes when applying Local Recalibration.

#### 5.2.1 Local recalibration improves both classification and calibration errors

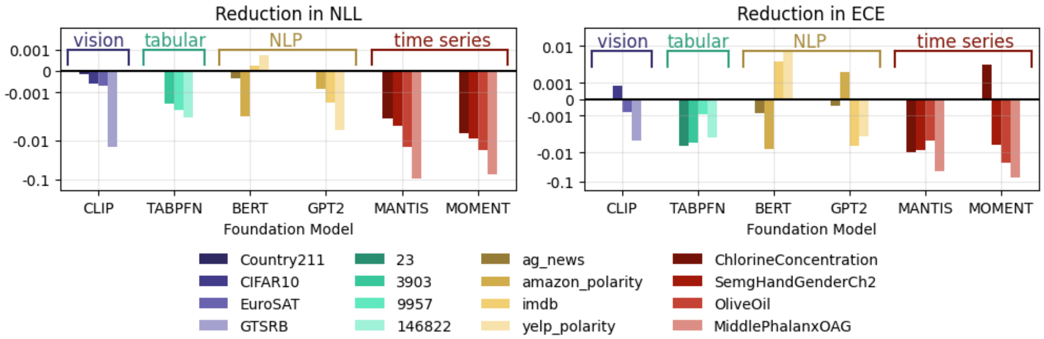

Figure 2: Reduction of both classification error (NLL) and calibration error (ECE), for every classification dataset and foundation model, compar

**General results.** In Table 1, we exhibit several datasets where recalibration, and more proeminently local recalibration, improves NLL. Note that in some cases – OliveOil dataset for example – applying global calibration worsens proper loss, but local loss is able to reverse the trend.

In Figure , we outline several datasets where we observe improvements of classification error when, measured by NLL. These

In Table 2, we exhibit several datasets where recalibration, and more proeminently local recalibration, improves ECE. We outline here after several findings.

First, **improvement of local w.r.t. global recalibration in terms of ECE is more consequent than in terms of NLL**. Indeed, it is easier to improve calibration error than overall error, which incorporates also irreducible error.

Moreover, for the selected datasets, **improvements in terms of ECE, from very small to very large ones, are allways backed by at least no degradation of proper loss** (Negative Log Likelihood). This indicates that we have no catastrophic increase in refinement error due to poor recalibration only focused on calibration error.

Lastly, **improvement in ECE exhibits more variability and less interpretativeness than improvement in NLL**, wich calls once again for a more parcimonious usage of this metric.

#### 5.2.2 Local Temperatures found are interpretable

In Figure 3, we have compared global and local recalibration on IMDB dataset with GPT2 embeddings. Local temperature scaling recalibrates better the predicted probabilities, taking into account the change in behavior of the joint distribution of $\mathbf{Y}$ and $g(\mathbf{X})$ accross a regressor exhibited in Figure 3. Such directions of calibration heterogeneity can be exhibited for several datasets, and correlate well with obtained temperatures.

#### 5.2.3 Impact of regularization

In this part, we study the impact of the regularization $\lambda_z||t - t_{\text{global}}||^2$ applied to searching optimal temperature in each cluster. Note that $\lambda_z$ can depend on the chosen cluster $z$. For example, it can decrease when cluster size increases. Empirically, we find out however that having such dependency does not bring improvements.

We study values of $\lambda$ ranging from 0.003 to 0.3, as well as not applying any regularization. Results are displayed in Figure 4 in terms of NLL – see appendix for results in terms of ECE. It appears

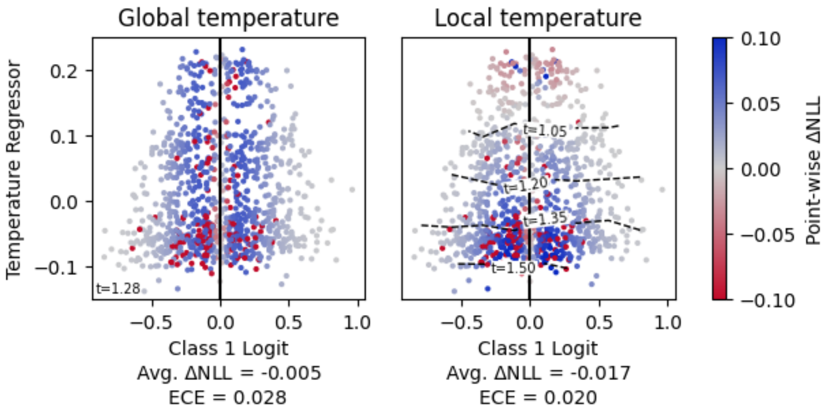

Figure 3: Global vs. Local recalibration for IMDB classification task on GPT2 embeddings. Test samples are projected along a 2D plane whose horizontal axis is the logistic regressor and ordinate axis is a regressor of the computed local temperature. Decision boundary is displayed as a vertical line. Points are coloured according to the improvement of the recalibration method in terms of NLL (denoted by $\Delta \log g(\mathbf{X})^\top \mathbf{Y} := (\log g_{\text{cal}}(\mathbf{X}) - \log g(\mathbf{X}))^\top \mathbf{Y}$). Global temperature is $t = 1.28$ ; Local Temperatures are indicated by isolines and range from $0.8$ to $1.5$. Note that for points above $0.1$ in $y$-axis exhibit confidence increase with local calibration, whereas points below $0.0$ in $y$-axis exhibit strong confidence decrease with local calibration. Global Temperature does not permit this granularity. Overall, Local Recalibration leads to a NLL improvement 3 times more important than Global Recalibration.

clearly that **regularization is of absolute necessity, as no regularization leads to divergence of NLL**. Indeed, regularization helps preventing cases where clusters are poorly represented in validation set, and recalibration mapping is overfitted on these poorly representative samples.

Moreover, a choice of $\lambda$ around $0.03$ or $0.1$, leads to close to optimal improvement in average in terms of NLL for most models. Note that, for BERT, we observe much more fast divergence of NLL as regularization strength diminishes. We hypothesise that embedding space of BERT is less prone to having heterogeneous calibration regions when fitting linear classifiers.

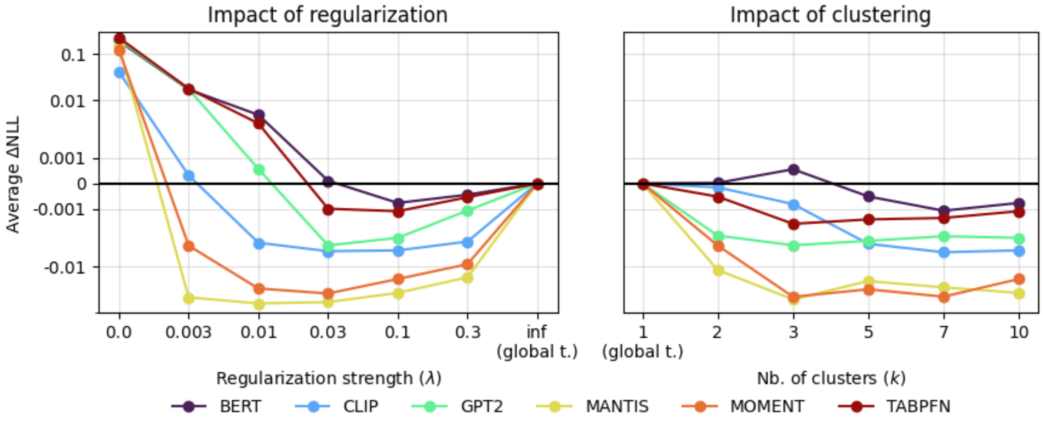

Figure 4: Impact on NLL of the regularization parameter $\lambda$ – on the left – and of the number of clusters $k$ – on the right. *Gobal t.* corresponds to results obtained with global recalibration

### 5.2.4 Choice of clustering method is not critical

In this part, we study the impact of the clustering methodology, by varying the number of clusters $k$. We compute results for values of k in $\{2, 3, 5, 7, 10\}$, and report the results for NLL in Figure 4 – See appendix for results in terms of ECE.

It appears that choosing $k \geq 5$ provides already most expected improvements in terms of NLL. Subsequent results for $k \geq 5$ do not display significant variability in $k$. This, as well as experiments with Gaussian Mixture Models and soft clustering, suggest that **choice of clustering algorithm is not critical provided the obtained clusterings are expressive enough to separate regions of input space exhibiting different calibration behaviors**.

## 6 Conclusion and Future Work

In this work, we introduced Local Temperature Scaling, a novel recalibration method that improves classification and calibration error on linear classifiers fitted on rich embedding spaces of Foundation Models. By allowing the temperature parameter to vary across locally clustered regions of the feature space, our approach effectively adapts to heterogeneous calibration needs that global methods often miss.

Our experimental results show that:

- On several set-ups, local recalibration consistently outperforms global temperature scaling in terms of both general classification error (NLL) and calibration error (ECE). Margins can reach very significant levels (20%) in any modality.
- Local temperatures are interpretable and aligned with meaningful variations in the data, uncaptured by the classifier's scores.
- Proper regularization is critical to prevent overfitting on underrepresented clusters and statistical noise in the validation set.
- Approach is robust to the choice of clustering strategy, provided the partitioning captures calibration-relevant structure.

However, we also want to point out to the following limitations:

- During our experiments, we also encountered many datasets for which impact of local recalibration is very limited and/or pooorly captured by existing metrics. Quantification of possible improvement, as well as appropriate improvement metrics should receive further attention.
- Performance can drop significantly if regularization is not strong enough. Use of more robust regularization-like techniques needs to be investigated – preliminary results with ensembling have displayed more robustness to a wide range of setups.

Overall, our findings underline the importance of fine-grained, localized calibration mechanisms in modern machine learning systems, taking into account latent information priors, and pave the way for more nuanced post-hoc correction techniques.

## 7 Preparing PDF files

Please prepare submission files with paper size "US Letter," and not, for example, "A4."

Fonts were the main cause of problems in the past years. Your PDF file must only contain Type 1 or Embedded TrueType fonts. Here are a few instructions to achieve this.

- You should directly generate PDF files using `pdflatex`.
- You can check which fonts a PDF files uses. In Acrobat Reader, select the menu Files>Document Properties>Fonts and select Show All Fonts. You can also use the program `pdffonts` which comes with `xpdf` and is available out-of-the-box on most Linux machines.
- `xfig` "patterned" shapes are implemented with bitmap fonts. Use "solid" shapes instead.
- The `\bbold` package almost always uses bitmap fonts. You should use the equivalent AMS Fonts:

  ```
  \usepackage{amsfonts}
  ```

  followed by, e.g., `\mathbb{R}`, `\mathbb{N}`, or `\mathbb{C}` for $\mathbb{R}$, $\mathbb{N}$ or $\mathbb{C}$. You can also use the following workaround for reals, natural and complex:

```
        \newcommand{\RR}{I\!\!R} %real numbers
        \newcommand{\Nat}{I\!\!N} %natural numbers
        \newcommand{\CC}{I\!\!\!\!C} %complex numbers
```

Note that `amsfonts` is automatically loaded by the `amssymb` package.

If your file contains type 3 fonts or non embedded TrueType fonts, we will ask you to fix it.

### 7.1 Margins in LaTeX

Most of the margin problems come from figures positioned by hand using `\special` or other commands. We suggest using the command `\includegraphics` from the `graphicx` package. Always specify the figure width as a multiple of the line width as in the example below:

```
   \usepackage[pdftex]{graphicx} ...
   \includegraphics[width=0.8\linewidth]{myfile.pdf}
```

See Section 4.4 in the graphics bundle documentation (`http://mirrors.ctan.org/macros/latex/required/graphics/grfguide.pdf`)

A number of width problems arise when LaTeX cannot properly hyphenate a line. Please give LaTeX hyphenation hints using the `\-` command when necessary.

## References

References follow the acknowledgments in the camera-ready paper. Use unnumbered first-level heading for the references. Any choice of citation style is acceptable as long as you are consistent. It is permissible to reduce the font size to `small` (9 point) when listing the references. Note that the Reference section does not count towards the page limit.

## References

Abdar, M., Pourpanah, F., Hussain, S., Rezazadegan, D., Liu, L., Ghavamzadeh, M., Fieguth, P., Cao, X., Khosravi, A., Acharya, U. R., et al. (2021). A review of uncertainty quantification in deep learning: Techniques, applications and challenges. *Information Fusion*, 76:243–297.

Amodei, D., Olah, C., Steinhardt, J., Christiano, P., Schulman, J., and Mané, D. (2016). Concrete problems in ai safety. *arXiv preprint arXiv:1606.06565*.

Arrieta-Ibarra, I., Gujral, P., Tannen, J., Tygert, M., and Xu, C. (2022). Metrics of calibration for probabilistic predictions. *Journal of Machine Learning Research*, 23(351):1–54.

Arthur, D. and Vassilvitskii, S. (2006). k-means++: The advantages of careful seeding. Technical report, Stanford.

Blasiok, J., Gopalan, P., Hu, L., and Nakkiran, P. (2023). When does optimizing a proper loss yield calibration? *Advances in Neural Information Processing Systems*, 36:72071–72095.

Błasiok, J. and Nakkiran, P. (2023). Smooth ece: Principled reliability diagrams via kernel smoothing. *arXiv preprint arXiv:2309.12236*.

Chidambaram, M. and Ge, R. (2024). Reassessing how to compare and improve the calibration of machine learning models. *arXiv preprint arXiv:2406.04068*.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pages 4171–4186.

Doshi-Velez, F. and Kim, B. (2017). Towards a rigorous science of interpretable machine learning. *arXiv preprint arXiv:1702.08608*.

Feofanov, V., Wen, S., Alonso, M., Ilbert, R., Guo, H., Tiomoko, M., Pan, L., Zhang, J., and Redko, I. (2025). Mantis: Lightweight calibrated foundation model for user-friendly time series classification. *arXiv preprint arXiv:2502.15637*.

Gal, Y. and Ghahramani, Z. (2016). Dropout as a bayesian approximation: Representing model uncertainty in deep learning. *International Conference on Machine Learning (ICML)*, pages 1050–1059.

Goswami, M., Szafer, K., Choudhry, A., Cai, Y., Li, S., and Dubrawski, A. (2024). Moment: A family of open time-series foundation models. *arXiv preprint arXiv:2402.03885*.

Guo, C., Pleiss, G., Sun, Y., and Weinberger, K. Q. (2017). On calibration of modern neural networks. In *International conference on machine learning*, pages 1321–1330. PMLR.

Hansen, D., Devic, S., Nakkiran, P., and Sharan, V. (2024). When is multicalibration post-processing necessary? *arXiv preprint arXiv:2406.06487*.

Hébert-Johnson, U., Kim, M., Reingold, O., and Rothblum, G. (2018). Multicalibration: Calibration for the (computationally-identifiable) masses. In *International Conference on Machine Learning*, pages 1939–1948. PMLR.

Hollmann, N., Müller, S., Eggensperger, K., and Hutter, F. (2022). Tabpfn: A transformer that solves small tabular classification problems in a second. *arXiv preprint arXiv:2207.01848*.

Kendall, A. and Gal, Y. (2017). What uncertainties do we need in bayesian deep learning for computer vision? *Advances in Neural Information Processing Systems*, 30.

Kull, M. and Flach, P. (2015). Novel decompositions of proper scoring rules for classification: Score adjustment as precursor to calibration. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2015, Porto, Portugal, September 7-11, 2015, Proceedings, Part I 15*, pages 68–85. Springer.

Kull, M., Perello Nieto, M., Kängsepp, M., Silva Filho, T., Song, H., and Flach, P. (2019). Beyond temperature scaling: Obtaining well-calibrated multi-class probabilities with dirichlet calibration. *Advances in neural information processing systems*, 32.

Lipton, Z. C. (2016). The mythos of model interpretability. *arXiv preprint arXiv:1606.03490*.

Luo, R., Bhatnagar, A., Bai, Y., Zhao, S., Wang, H., Xiong, C., Savarese, S., Ermon, S., Schmerling, E., and Pavone, M. (2022). Local calibration: metrics and recalibration. In *Uncertainty in Artificial Intelligence*, pages 1286–1295. PMLR.

Minderer, M., Djolonga, J., Romijnders, R., Hubis, F., Zhai, X., Houlsby, N., Tran, D., and Lucic, M. (2021). Revisiting the calibration of modern neural networks. *Advances in neural information processing systems*, 34:15682–15694.

Ovadia, Y., Fertig, E., Ren, J., Nado, Z., Sculley, D., Nowozin, S., Dillon, J. V., Lakshminarayanan, B., and Snoek, J. (2019). Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. *Advances in Neural Information Processing Systems*, 32.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Perez-Lebel, A., Morvan, M. L., and Varoquaux, G. (2022). Beyond calibration: estimating the grouping loss of modern neural networks. *arXiv preprint arXiv:2210.16315*.

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

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al. (2021). Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al. (2019). Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Rudin, C. (2019). Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 1(5):206–215.

Wortsman, M., Ilharco, G., Kim, J., Wightman, R., Weller, A., Hajishirzi, H., Farhadi, A., Schmidt, L., and Kornblith, S. (2022). Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *International Conference on Machine Learning (ICML)*, pages 23965–23998.

Xiong, M., Deng, A., Koh, P. W. W., Wu, J., Li, S., Xu, J., and Hooi, B. (2023). Proximity-informed calibration for deep neural networks. *Advances in Neural Information Processing Systems*, 36:68511–68538.

Zadrozny, B. and Elkan, C. (2001). Obtaining calibrated probability estimates from decision trees and naive bayesian classifiers. In *Icml*, volume 1.

Zhang, J., Kailkhura, B., and Han, T. Y.-J. (2020). Mix-n-match: Ensemble and compositional methods for uncertainty calibration in deep learning. In *International conference on machine learning*, pages 11117–11128. PMLR.

Zhao, S., Ma, T., and Ermon, S. (2020). Individual calibration with randomized forecasting. In *International Conference on Machine Learning*, pages 11387–11397. PMLR.

# A Proofs

## A.1 If $g(\mathbf{X}) = \mathbb{E}[\mathbf{Y}|\mathcal{F}]$, then calibration error is null

Let $\mathcal{F}$ be a $\sigma$-algebra. We will show that if $g(\mathbf{X}) = \mathbb{E}[\mathbf{Y}|\mathcal{F}]$, then we have $\text{CE}(g) = 0$. Indeed, we have in that case:

$$\begin{aligned}\mathbb{E}[\mathbf{Y}|g(\mathbf{X})] &= \mathbb{E}[\mathbf{Y}|\mathbb{E}[\mathbf{Y}|\mathcal{F}]] \\ &= \mathbb{E}[\mathbb{E}[\mathbf{Y}|\mathcal{F}]|\mathbb{E}[\mathbf{Y}|\mathcal{F}]] \text{ using tower property, since } \sigma(\mathbb{E}[\mathbf{Y}|\mathcal{F}]) \subseteq \mathcal{F} \\ &= \mathbb{E}[\mathbf{Y}|\mathcal{F}] \text{ as } \mathbb{E}[U|U] = U \text{ for any r.v. } U \\ &= g(\mathbf{X})\end{aligned}$$

Therefore,

$$\text{CE}(g) = \mathbb{E}[d(\mathbb{E}[\mathbf{Y}|g(\mathbf{X})], g(\mathbf{X}))] = \mathbb{E}[d(g(\mathbf{X}), g(\mathbf{X}))] = 0$$

In particular, we have $\text{CE}(g) = 0$ for $g(\mathbf{X}) = \mathbb{E}[Y]$.

## A.2 If $g(\mathbf{X}) = \mathbb{E}[\mathbf{Y}|\mathbf{X}] + c$, then refinement error is null, and calibration error is non null provided $c \neq 0$

Let $c \neq 0$ be a constant. We have

$$\begin{aligned}\mathbb{E}[\mathbf{Y}|g(\mathbf{X})] &= \mathbb{E}[\mathbf{Y}|\mathbb{E}[\mathbf{Y}|\mathbf{X}] + c] \\ &= \mathbb{E}[Y + c|\mathbb{E}[\mathbf{Y} + c|\mathbf{X}]] - c \\ &= \mathbb{E}[\mathbb{E}[\mathbf{Y} + c|\mathbf{X}]|\mathbb{E}[\mathbf{Y} + c|\mathbf{X}]] - c \\ &= \mathbb{E}[\mathbf{Y} + c|\mathbf{X}] - c \\ &= \mathbb{E}[\mathbf{Y}|\mathbf{X}]\end{aligned}$$

Therefore, refinement error is null. Moreover,

$$\text{CE}(g) = \mathbb{E}[d(\mathbb{E}[\mathbf{Y}|g(\mathbf{X})], g(\mathbf{X}))] = \mathbb{E}[d(\mathbb{E}[\mathbf{Y}|\mathbf{X}] + c, \mathbb{E}[\mathbf{Y}|\mathbf{X}])] \neq 0 \text{ in general, } c^2 \text{ for } d_{\text{Brier}}$$

### A.3 Proposition 3.1

We aim at proving the following for any random variable $Z$:

$$\mathbb{E}[d(\mathbf{Y}, g(\mathbf{X}))] = \underbrace{\mathbb{E}[d(\mathbf{Y}, \mathbb{E}[\mathbf{Y}|\mathbf{X}, Z])]}_{\text{Z–Irreducible error}} + \underbrace{\mathbb{E}[d(\mathbb{E}[\mathbf{Y}|\mathbf{X}, Z], \mathbb{E}[\mathbf{Y}|g(\mathbf{X}), Z])]}_{\text{Z-Refinement error}} + \underbrace{\mathbb{E}[d(\mathbb{E}[\mathbf{Y}|g(\mathbf{X}), Z], g(\mathbf{X}))]}_{\text{Z-Calibration Error (Z-CE)}}$$

From (Kull and Flach, 2015) – Lemma 1, we have for any random variables $\mathbf{V_1}$, $\mathbf{V_2}$, $\mathbf{V_3}$, $\mathbf{W}$, such that $\mathbf{V_2} = \mathbb{E}[\mathbf{V_3}|\mathbf{W}]$ and $V_1 = \delta(\mathbf{W})$ function of $\mathbf{W}$:

$$\mathbb{E}[d(\mathbf{V_1}, \mathbf{V_3})] = \mathbb{E}[d(\mathbf{V_1}, \mathbf{V_2})] + \mathbb{E}[d(\mathbf{V_2}, \mathbf{V_3})]$$

We apply this lemma, first using $\mathbf{W} = (g(\mathbf{X}), Z)$, $\mathbf{V_1} = g(\mathbf{X})$, $\mathbf{V_3} = Y$ and $\mathbf{V_2} = [\mathbf{Y}|g(\mathbf{X}), Z]$. We obtain:

$$\mathbb{E}[d(\mathbf{Y}, g(\mathbf{X}))] = \mathbb{E}[d(\mathbf{Y}, \mathbb{E}[\mathbf{Y}|g(\mathbf{X}), Z])] + \mathbb{E}[d(\mathbb{E}[\mathbf{Y}|g(\mathbf{X}), Z], g(\mathbf{X}))]$$

We apply again the lemma on the first term of the sum, this time using $\mathbf{W} = (\mathbf{X}, Z)$, $\mathbf{V_1} = \mathbb{E}[\mathbf{Y}|g(\mathbf{X}), Z])]$, $\mathbf{V_3} = \mathbf{Y}$ and $\mathbf{V_2} = \mathbb{E}[\mathbf{Y}|\mathbf{X}, Y]$. We obtain the desired decomposition.

## B Technical Details on Local Calibration Methods

### B.1 Dirichlet Calibration

Dirichlet calibration (Kull et al., 2019) models the distribution of predicted probabilities $g(\mathbf{X})$ conditioned on the true class label $Y = y$ using a Dirichlet distribution. This enables more expressive recalibration mappings compared to simpler approaches like Platt scaling or temperature scaling.

### B.2 Multicalibration and $Z$-calibration

Multicalibration, introduced in (Hébert-Johnson et al., 2018), enforces the condition:

$$\mathbb{E}[\mathbf{Y}|g(\mathbf{X}), h(\mathbf{X}) = 1] = g(\mathbf{X}) \quad \text{for all } h \in \mathcal{H} \subset \{0, 1\}^{\mathcal{X}}.$$

This can be seen as a special case of $Z$-calibration, where $Z$ is a random variable indexing subgroups via $Z = h(\mathbf{X})$. While this framework yields strong theoretical guarantees, the practical gains over global recalibration are limited (Hansen et al., 2024).

### B.3 Local Neighborhood Calibration

The method of (Luo et al., 2022) recalibrates each prediction based on the accuracy of nearby samples with similar predicted probabilities. Vicinity is computed using a Laplacian kernel in a reduced-dimensional space. This method, while powerful, depends on sensitive hyperparameters and assumes a deep learning-based $g$.

### B.4 Randomized Calibration via Latent $Z$

By treating $Z$ as a latent variable, one can define a randomized calibrated score:

$$M_{\mathcal{Z}}(x) \sim m_Z(g(x)) \mid \mathbf{X} = \mathbf{x},$$

rather than taking the expected value over $Z|\mathbf{X} = \mathbf{x}$. This opens the door to randomized prediction schemes, as explored by Zhao et al. (2020), though we do not pursue this direction in our work.

## C Experiments

### C.1 Complete results for error reduction

#### C.1.1 Classification error – NLL

See table 1

|  |  |  | No Cal. | Global | Local |
|---|---|---|---|---|---|
| Modality | Dataset | FM |  |  |  |
| vision | Country211 | CLIP | 4.455 | 4.295 | **4.281** |
|  | CIFAR10 | CLIP | 0.240 | **0.205** | **0.205** |
|  | EuroSAT | CLIP | 0.209 | **0.206** | **0.206** |
|  | GTSRB | CLIP | 0.359 | 0.356 | **0.352** |
| tabular | 23 | TabPFN | **0.889** | **0.889** | **0.889** |
|  | 3903 | TabPFN | 0.280 | 0.277 | **0.276** |
|  | 9957 | TabPFN | 0.398 | 0.347 | **0.345** |
|  | 146822 | TabPFN | 0.213 | 0.187 | **0.185** |
| language | ag news | BERT | 0.373 | 0.344 | **0.342** |
|  |  | GPT2 | 0.379 | 0.385 | **0.384** |
|  | amazon polarity | BERT | 0.308 | 0.308 | 0.308 |
|  |  | GPT2 | 0.349 | 0.346 | **0.344** |
|  | imdb | BERT | **0.355** | 0.356 | 0.357 |
|  |  | GPT2 | 0.378 | 0.373 | **0.361** |
|  | yelp polarity | BERT | 0.290 | **0.234** | **0.234** |
|  |  | GPT2 | 0.259 | **0.252** | **0.252** |
| time series | ChlorineConcentration | MANTIS | 0.793 | 0.516 | **0.501** |
|  |  | MOMENT | 0.937 | 0.710 | **0.702** |
|  | SemgHandGenderCh2 | MANTIS | 0.726 | 0.249 | **0.246** |
|  |  | MOMENT | 0.184 | 0.188 | **0.182** |
|  | OliveOil | MANTIS | 0.626 | 0.690 | **0.593** |
|  |  | MOMENT | 0.859 | 0.909 | **0.835** |
|  | MiddlePhalanxOAG | MANTIS | 0.793 | 0.516 | **0.501** |
|  |  | MOMENT | 0.562 | 0.529 | **0.511** |
|  | Change | absolute |  | -0.074 | **-0.088** |
|  |  | relative |  | -10% | **-12%** |

Table 1: Negative Log Likelihood on test set for uncalibrated, globally recalibrated and locally recalibrated probabilities. Change is indicated w.r.t. uncalibrated probabilities

### C.1.2 Calibration error – ECE

See table 2

## C.2 Impact of $\lambda$ and $k$ on ECE

See figure 5

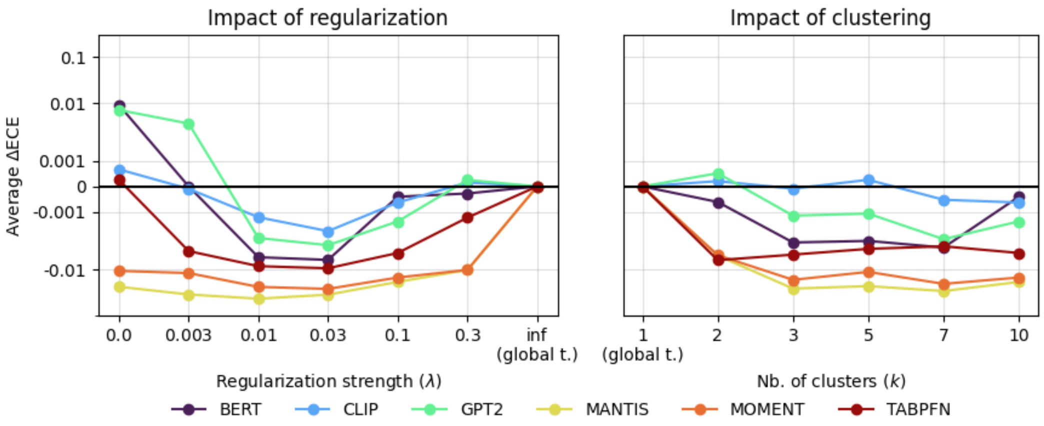

Figure 5: Impact on ECE of the regularization parameter $\lambda$ – on the left – and of the number of clusters $k$ – on the right.

| Modality | Dataset | FM | No Cal. | Global | Local |
|---|---|---|---|---|---|
| vision | Country211 | CLIP | 0.109 | 0.019 | **0.015** |
|  | CIFAR10 | CLIP | 0.058 | 0.011 | **0.010** |
|  | EuroSAT | CLIP | 0.018 | **0.009** | **0.009** |
|  | GTSRB | CLIP | 0.023 | **0.012** | **0.012** |
| tabular | 23 | TabPFN | 0.034 | 0.036 | **0.030** |
|  | 3903 | TabPFN | 0.033 | 0.035 | **0.031** |
|  | 9957 | TabPFN | 0.141 | 0.077 | **0.076** |
|  | 146822 | TabPFN | 0.034 | 0.013 | **0.010** |
| language | ag news | BERT | 0.072 | 0.035 | **0.027** |
|  |  | GPT2 | 0.034 | **0.028** | 0.029 |
|  | amazon polarity | BERT | 0.018 | **0.017** | 0.020 |
|  |  | GPT2 | 0.042 | 0.024 | **0.018** |
|  | imdb | BERT | **0.016** | 0.026 | 0.032 |
|  |  | GPT2 | 0.033 | 0.029 | **0.023** |
|  | yelp polarity | BERT | 0.102 | 0.020 | **0.019** |
|  |  | GPT2 | 0.041 | **0.016** | **0.016** |
| time series | ChlorineConcentration | MANTIS | 0.194 | 0.069 | **0.060** |
|  |  | MOMENT | 0.081 | 0.065 | **0.059** |
|  | SemgHandGenderCh2 | MANTIS | 0.080 | 0.030 | **0.021** |
|  |  | MOMENT | 0.032 | **0.031** | 0.033 |
|  | OliveOil | MANTIS | **0.084** | 0.253 | 0.209 |
|  |  | MOMENT | **0.275** | 0.383 | 0.312 |
|  | MiddlePhalanxOAG | MANTIS | 0.107 | 0.103 | **0.099** |
|  |  | MOMENT | **0.053** | 0.079 | 0.057 |
|  | Change | absolute |  | -0.008 | **-0.017** |
|  |  | relative |  | -10% | **-18%** |

Table 2: Expected Calibration Error on test set for uncalibrated, globally recalibrated and locally recalibrated probabilities. Change is indicated w.r.t. uncalibrated probabilities

## NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes] , [No] , or [NA] .
- [NA] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for NA).

**The checklist answers are an integral part of your paper submission.** They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will be asked to also include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While "[Yes] " is generally preferable to "[No] ", it is perfectly acceptable to answer "[No] " provided a proper justification is given (e.g., "error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering "[No] " or "[NA] " is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading “NeurIPS Paper Checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper’s contributions and scope?

   Answer: **[TODO]**

   Justification: **[TODO]**

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: **[TODO]**

   Justification: **[TODO]**

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren’t acknowledged in the paper. The authors should use their best

judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: **[TODO]**

   Justification: **[TODO]**

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: **[TODO]**

   Justification: **[TODO]**

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: **[TODO]**

Justification: **[TODO]**

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: **[TODO]**

Justification: **[TODO]**

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: **[TODO]**

Justification: **[TODO]**

Guidelines:

- The answer NA means that the paper does not include experiments.

- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: **[TODO]**

   Justification: **[TODO]**

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: **[TODO]**

   Justification: **[TODO]**

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: **[TODO]**

    Justification: **[TODO]**

    Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: **[TODO]**

    Justification: **[TODO]**

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: **[TODO]**

    Justification: **[TODO]**

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: **[TODO]**

    Justification: **[TODO]**

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: **[TODO]**

    Justification: **[TODO]**

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: **[TODO]**

    Justification: **[TODO]**

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: **[TODO]**

    Justification: **[TODO]**

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.