MANO: Exploiting Matrix Norm for Unsupervised Accuracy Estimation Under Distribution Shifts

Renchunzi Xie[*]
Nanyang Technological University
Singapore
renchunzi.xie@ntu.edu.sg

Ambroise Odonnat[*]
Huawei Noah's Ark Lab, Inria[◇]
Paris, France
ambroise.odonnat@gmail.com

Vasilii Feofanov[*]
Huawei Noah's Ark Lab
Paris, France
vasilii.feofanov@gmail.com

Weijian Deng
Australian National University
Canberra, Australia
weijian.deng@anu.edu.au

Jianfeng Zhang
Huawei Noah's Ark Lab
Shenzhen, China
zhangjianfeng3@huawei.com

Bo An
Skywork AI
Nanyang Technological University
Singapore
boan@ntu.edu.sg

Abstract

Leveraging the model's outputs, specifically the logits, is a common approach to estimating the test accuracy of a pre-trained neural network on out-of-distribution (OOD) samples without requiring access to the corresponding ground-truth labels. Despite their ease of implementation and computational efficiency, current logit-based methods are vulnerable to overconfidence issues, leading to prediction bias, especially under the natural shift. In this work, we first study the relationship between logits and generalization performance from the view of low-density separation assumption. Our findings motivate our proposed method MANO that **(1)** applies a data-dependent normalization on the logits to reduce prediction bias, and **(2)** takes the L_p norm of the matrix of normalized logits as the estimation score. Our theoretical analysis highlights the connection between the provided score and the model's uncertainty. We conduct an extensive empirical study on common unsupervised accuracy estimation benchmarks and demonstrate that MANO achieves state-of-the-art performance across various architectures in the presence of synthetic, natural, or subpopulation shifts. The code is available at https://github.com/Renchunzi-Xie/MaNo.

1 Introduction

The deployment of machine learning models in real-world scenarios is frequently challenged by distribution shifts between the training and test data. These shifts can substantially deteriorate the model's performance during testing (Geirhos et al., 2018; Koh et al., 2021; Quionero-Candela et al., 2009) and introduce significant risks related to AI safety (Deng and Zheng, 2021; Hendrycks and Mazeika, 2022). To alleviate this issue, it is common to monitor model performance by periodically collecting the ground truth labels for a subset of the current test dataset (Lu et al., 2023). However,

[*]Equal contribution. Correspondence to: Bo An - boan@ntu.edu.sg. [◇]Univ. Rennes 2, CNRS, IRISA

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

this approach is often resource-intensive and time-consuming, which motivates the importance of estimating the model's performance on out-of-distribution (OOD) data in an unsupervised manner, also known as *Unsupervised Accuracy Estimation* (Donmez et al., 2010).

Due to privacy constraints and computational efficiency, one of the most popular ways to estimate accuracy without labels is to rely on the model's outputs, called logits, as a source of confidence in the model's predictions (Deng et al., 2023; Garg et al., 2022; Guillory et al., 2021; Hendrycks and Gimpel, 2016). For instance, *ConfScore* (Hendrycks and Gimpel, 2016) leverages the average maximum `softmax` probability as the test accuracy estimator, while Deng et al. (2023) has recently proposed to estimate the accuracy via the nuclear norm of the `softmax` probability matrix. These approaches, however, tend to underperform on the natural shift applications while the intuition behind the use of logits remains unclear. This motivates us to ask:

Question 1: *What explains the correlation between logits and generalization performance?*

In Section 3, we show that logits are connected to the model's margins, *i.e.*, the distances between the learned embeddings, and the decision boundaries. Inspired by the low-density separation (LDS) assumption (Chapelle and Zien, 2005; Feofanov et al., 2023) that optimal decision boundaries should lie in low-density regions, we propose MANO, an estimation score that aggregates the margins at a dataset level by taking the L_p-norm of the normalized model's prediction matrix to evaluate the density around decision boundaries. Nevertheless, logit-based approaches are known to suffer from overconfidence (Odonnat et al., 2024; Wei et al., 2022a), resulting in high prediction bias, especially under poorly-calibrated scenarios. This leads us to another critical question:

Question 2: *How to alleviate the overconfidence issues of logits-based methods?*

In Section 4, we reveal that this question is connected to the normalization of logits and show that the widely-used `softmax` normalization accumulates errors in the presence of prediction bias, which can lead to overconfidence and significantly degrade the performance of existing accuracy estimation methods in poorly-calibrated scenarios. To mitigate this issue, we propose a novel normalization strategy called `softrun` that takes into account the empirical distribution of logits and aims to find a trade-off between information completeness of ground-truth logits and error accumulation.

Summary of our contributions. **(1)** We show that logits are informative of generalization performance through the lens of the low-density separation assumption by reflecting the distances to decision boundaries. **(2)** We identify the failure of the commonly-used `softmax` normalization that accumulates errors under poorly calibrated because of its overconfidence, leading to biased estimation. **(3)** We propose MANO, a training-free estimation method that quantifies the global distances to decision boundaries by taking the L_p norm of the logits matrix. MANO relies on `softrun`, a novel normalization technique that makes a trade-off between information completeness and error accumulation and is robust to different calibration scenarios. In addition, we demonstrate its connection to the model's uncertainty. **(4)** We demonstrate the superiority of MANO compared to 11 competitors with a large-scale empirical evaluation including 12 benchmarks across diverse distribution shifts. Results show that MANO consistently improves over the state-of-the-art baselines, including on the challenging natural shift.

2 Problem Statement

Setting. Consider a classification task with input space $\mathcal{X} \subset \mathbb{R}^D$ and label space $\mathcal{Y} = \{1, \dots, K\}$. Let p_S and p_T be the source and target distributions on $\mathcal{X} \times \mathcal{Y}$, respectively, with $p_S \neq p_T$. We parameterize a neural network $f \colon \mathcal{X} \to \mathbb{R}^K$ as $f = f_{\mathbf{W}} \circ f_{\boldsymbol{\varphi}}$, where $f_{\boldsymbol{\varphi}} \colon \mathcal{X} \to \mathbb{R}^q$ is a feature extractor and $f_{\mathbf{W}} \colon \mathbb{R}^q \to \mathbb{R}^K$ is a linear classifier with parameters $\mathbf{W} = (\boldsymbol{\omega}_k)_{k=1}^K \in \mathbb{R}^{q \times K}$. Further, we denote an input by $\mathbf{x}$, its corresponding label by y, its representation by $\mathbf{z} = f_{\boldsymbol{\varphi}}(\mathbf{x})$ and logits by $\mathbf{q} = f(\mathbf{x}) = (\boldsymbol{\omega}_k^\top \mathbf{z})_k \in \mathbb{R}^K$. The accuracy of f on $\mathcal{D}$ is defined as $\mathrm{Acc}(f, \mathcal{D}) \coloneqq \frac{1}{|\mathcal{D}|} \sum_{(\mathbf{x},y) \in \mathcal{D}} \mathbb{1}_{\hat{y}=y}$ with predicted labels $\hat{y}$. The probability simplex is denoted by $\Delta_K = \{\mathbf{p} \in [0,1]^K | \mathbb{1}_K^\top \mathbf{p} = 1\}$.

Unsupervised accuracy estimation. Given a model f pre-trained on a training set $\mathcal{D}_{\text{train}}$ with samples drawn i.i.d. from p_S, the goal of unsupervised accuracy estimation is to assess its generalization performance on a given unlabeled test set $\mathcal{D}_{\text{test}} = \{\mathbf{x}_i\}_{i=1}^N$ with N samples drawn i.i.d. from p_T. More specifically, we assume (i) a source-free regime (no direct access to $\mathcal{D}_{\text{train}}$), (ii) no access to test labels, and (iii) a distribution shift, *i.e.* $p_S \neq p_T$, . In this challenging setup, which often occurs in real-world scenarios when ground-truth labels are inaccessible at a test time, we aim to design an estimation score $\mathcal{S}(f, \mathcal{D}_{\text{test}})$ that exhibits a linear correlation with the true OOD accuracy $\text{Acc}(f, \mathcal{D}_{\text{test}})$. Following the standard closed-set setting, both p_T and p_S involve the same K classes. For an extended discussion of related work on unsupervised accuracy estimation, we refer the reader to Appendix B.

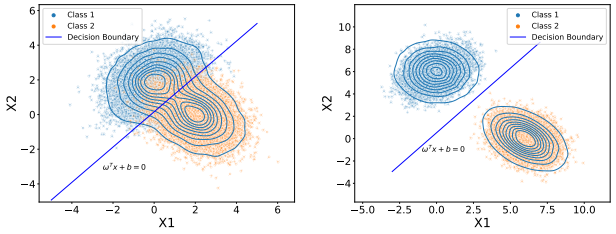
3 What Explains the Correlation between Logits and Test Accuracy?

Although existing literature has shown the feasibility of unsupervised accuracy prediction under distribution shift by utilizing the model's logits (Deng et al., 2023; Garg et al., 2022; Guillory et al., 2021), the reason behind this empirical success remains unclear. In this section, we seek to understand when and why logits can be informative for analyzing generalization performance. Based on the derived understanding, we propose our approach, MANO, for estimating generalization performance.

3.1 Motivation

Logits reflect the distances to decision boundaries. We analyze logits from a linear classification perspective in the embedding space, where the decision boundary of class k is the hyperplane $\{\boldsymbol{z}' \in \mathbb{R}^q | \boldsymbol{\omega}_k^\top \boldsymbol{z}' = 0\}$. In Appendix D.2, we remind that the distance from a point $\mathbf{z}$ to hyperplane $\boldsymbol{\omega}_k$ is given by $\mathrm{d}(\boldsymbol{\omega}_k, \mathbf{z}) = |\boldsymbol{\omega}_k^\top \boldsymbol{z}| / \|\boldsymbol{\omega}_k\|$. As the pre-trained model is fixed and $\boldsymbol{\omega}_k$ can be normalized, we derive that the logits in absolute values are proportional to the distance from the learned embeddings to the decision boundaries, *i.e.*, $|\mathbf{q}_k| = |\boldsymbol{\omega}_k^\top \mathbf{z}| \propto d(\boldsymbol{\omega}_k, \mathbf{z}), \forall k$. This indicates that the magnitude of logits reflects how close the corresponding embedding is from each decision boundary.

Low-density separation assumption. When dealing with unlabeled data, it is required to make assumptions on the relationship between the distance to decision boundaries and generalization performance. The low-density separation assumption (LDS, Chapelle and Zien, 2005) states that optimal decision boundaries should lie in low-density regions (Figure 1) so that unlabeled margin $|\boldsymbol{\omega}_k^\top \mathbf{z}|$ reflects reliable confidence in predicting $\mathbf{x}$ to the class k. The assumption is often empirically supported as the misclassified samples tend to be significantly closer to the decision boundary than the correctly classified ones (Mickisch et al., 2020). This might indicate that **the absolute values of the logits are positively correlated to its generalization performance**.



(a) High-density region (b) Low-density region.

Figure 1: **Illustration of the LDS assumption.** When the boundary passes through dense regions **(a)**, margins have little predictive power and cannot be used without labels. On the contrary, margins are informative in sparse regions **(b)**.

Assumptions on the prediction bias. It is important to note that the LDS assumption has been initially proposed for semi-supervised learning where labeled and unlabeled data are assumed to come from the same distribution, which is not the case in our setting. This leads to logits writing $f(\mathbf{x}) = \mathbf{q}^* + \boldsymbol{\varepsilon}$ in the general case[◇], *i.e.*, subject to a potentially non-negligible prediction bias $\boldsymbol{\varepsilon} = (\varepsilon_k)_k$ with respect to the ground-truth logits $\mathbf{q}^* \in \mathbb{R}^K$. The following proposition shows the impact of the prediction bias on the divergence between the true class posterior probabilities, assumed modeled as $\mathbf{p} = \text{softmax}(\mathbf{q}^*) \in \Delta_K$, and the estimated ones $\mathbf{s} = \text{softmax}(f(\mathbf{x})) \in \Delta_K$.

[◇]We write this decomposition without loss of generality as no restrictions are imposed on $\boldsymbol{\varepsilon}$.

Proposition 3.1. *Let* $\boldsymbol{\varepsilon}_+ = (\max_l\{\varepsilon_l\} - \varepsilon_k)_k$. *Then, the KL divergence between* $\mathbf{p}$ *and* $\mathbf{s}$ *verifies*

$$0 \leq \mathrm{KL}(\mathbf{p}||\mathbf{s}) \leq \boldsymbol{\varepsilon}_+^T \mathbf{p}.$$

The proposition indicates that a large approximation error of the posterior may be caused by prediction bias that has a large norm and / or bad alignment with the true probabilities. Thus, the logit-based methods assume that the magnitude of the bias is reasonably bounded while the direction of bias does not drastically harm the ranking of classes by probabilities. We elaborate on this discussion and present the proof of Proposition 3.1 in Appendix D.1.

3.2 MANO: Predicting Generalization Performance With Matrix Norm of Logits

We have shown a connection between the feature-to-boundary distances and generalization performance as well as the impact of the prediction bias. Based on the derived intuition, we introduce MANO that leverages the model margins at the dataset level performing two steps: normalization and aggregation. The pseudo-code of MANO is provided in Appendix A.

Step 1: Normalization. Given that logits can exhibit significant variations in their scale depending on the input $\mathbf{x}$, it is crucial to normalize the logits within a standardized range to prevent outliers from exerting disproportionate influence on the estimation. A natural range stems from the fact that most deep classifiers have outputs in Δ_K, which amounts to applying a normalization function $\sigma\colon \mathbb{R}^K \to \Delta_K$ on top of the pre-trained neural network (Mensch et al., 2019), where Δ_K refers to probability simplex. This ensures having logits entries in $[0, 1]$. For each test sample $\mathbf{x}_i$, we first extract its learned feature representation $\mathbf{z}_i = f_{\boldsymbol{\varphi}}(\mathbf{x}_i)$. Then, logits corresponding to this representation are computed as $\mathbf{q}_i = f_{\boldsymbol{W}}(\mathbf{z}_i) \in \mathbb{R}^K$. The normalization procedure results in a prediction matrix $\mathbf{Q} \in \mathbb{R}^{N\times K}$ with each row $\mathbf{Q}_i$ containing the normalized logits of an input sample:

$$\mathbf{Q}_i = \sigma(\mathbf{q}_i) \in \Delta_K, \tag{1}$$

where σ denotes the normalization function for the logits values. It is worth noting that not all normalization methods are appropriate candidates. The selection of a suitable normalization function σ based on different calibration scenarios will be discussed in detail in Section 4.

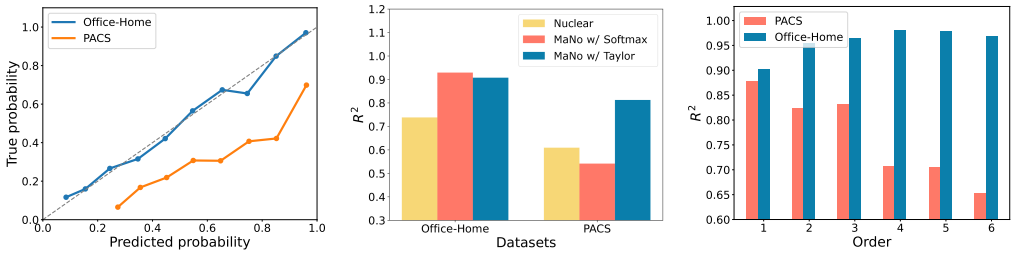
Step 2: Aggregation. Once the logits are scaled, we aggregate the dataset-level information on feature-to-boundary distances by taking the entry-wise L_p norm of the prediction matrix $\mathbf{Q}$, which can be expressed as:

$$\mathcal{S}(f, \mathcal{D}_{\text{test}}) = \frac{1}{\sqrt[p]{NK}}\|\mathbf{Q}\|_p = \left(\frac{1}{NK}\sum_{i=1}^{N}\sum_{k=1}^{K}|\sigma(\boldsymbol{q}_i)_k|^p\right)^{\frac{1}{p}}, \tag{2}$$

As we have $||\mathbf{Q}||_p \leq \sqrt[p]{NK}\max(\mathbf{Q}_{ij}) = \sqrt[p]{NK}$ ($\mathbf{Q}_{ij} \in [0, 1]$), the scaling by $\sqrt[p]{NK}$ leads to $\mathcal{S}(f, \mathcal{D}_{\text{test}}) \in [0, 1]$, providing a standardized metric regardless of variations in the size of the test dataset N and the number of classes K. As p increases, MANO puts greater emphasis on high-margin terms, focusing on confident classification hyperplanes. In the extreme case where $p \to \infty$, we have $|\mathbf{Q}||_p \to \max(\mathbf{Q}_{ij})$. In practice, we choose $p = 4$ in all experiments and provide an ablation study on p in Appendix G.1. As the L_p norm is straightforward to compute, our approach is scalable and efficient compared to the current state-of-the-art method Nuclear (Deng et al., 2023) that requires performing a singular value decomposition.

3.3 Theoretical Analysis of MANO

In this section, we provide the theoretical support for the positive correlation between MANO and test accuracy. More specifically, we reveal that our proposed score is connected with the uncertainty of the neural network's predictions in Theorem 3.3. Before presenting this result, we recall below the definition of Tsallis α-entropies introduced in Tsallis (1988).



(a) Calibration curves. (b) Type of normalization. (c) Order n in Eq. (4).

Figure 2: **Empirical evidence with Resnet18. (a)** The model is well-calibrated on Office-Home and miscalibrated on PACS. (b) `softrun` is superior to the state-of-the-art Nuclear (Deng et al., 2023) in all scenarios while the `softmax` heavily fails on PACS. **(c)** Increasing the approximation order n in Eq. (4) is detrimental on PACS and beneficial on Office-Home. The optimal trade-off in all calibration scenarios is taking $n \in \{2, 3\}$.

Definition 3.2 (Tsallis α-entropies (Tsallis, 1988))**.** *Let $\alpha > 1$ and $k > 0$. The Tsallis α-entropy is defined as:*

$$\mathbf{H}_\alpha^{\mathrm{T}}(\mathbf{p}) = k(\alpha - 1)^{-1}(1 - \|\mathbf{p}\|_\alpha^\alpha).$$

In this work, we choose $k = \frac{1}{\alpha}$ following Blondel et al. (2019). The Tsallis entropies generalize the Shannon entropy (limit case $\alpha \to 1$) and have been used in various applications (Blondel et al., 2019, 2020; Muzellec et al., 2017). More details can be found in Appendix C. The following theorem, whose proof is deferred to Appendix D.3, states that the estimation score obtained with MANO is a function of the average Tsallis entropy of the normalized neural network's logits.

Theorem 3.3 (Connection to uncertainty)**.** *Let $p > 1$, $a = \frac{p(p-1)}{K}$ and $b = \frac{1}{K}$. Given a test set $\mathcal{D}_{\text{test}} = \{\mathbf{x}_i\}_{i=1}^N$, corresponding logits $\mathbf{q}_i = f(\mathbf{x}_i)$, a normalization function $\sigma \colon \mathbb{R}^K \to \Delta_K$ and $p > 1$, the estimation score $\mathcal{S}(f, \mathcal{D}_{\text{test}})$ provided by* MANO *(Algorithm 1) verifies*

$$\mathcal{S}(f, \mathcal{D}_{\text{test}})^p = -a\left(\frac{1}{N}\sum_{i=1}^{N} \mathbf{H}_p^{\mathrm{T}}(\sigma(\mathbf{q}_i))\right) + b. \tag{3}$$

As $a > 0$, Theorem 3.3 implies that the estimation score provided by MANO is negatively correlated with the average Tsallis-entropy on the test set. In particular, the less certain the model is on test data, the lower the test accuracy is and the higher the entropy term is in Eq. (3), resulting in a lower score $\mathcal{S}(f, \mathcal{D}_{\text{test}})$. As the converse sense holds, MANO provides a score positively correlated to the test accuracy. This follows the findings of Guillory et al. (2021); Wang et al. (2021) and empirically confirmed in Section 5 for various architectures, datasets, and types of shift.

4 How to Alleviate Overconfidence Issues of Logit-Based Methods?

The most common normalization technique of existing logit-based approaches is the `softmax` normalization. In this section, we show that the widely used `softmax` is sensitive to prediction bias, which hinders the quality of the estimation in poorly calibrated scenarios. To alleviate this issue, we propose a novel normalization strategy, `softrun`, which balances the information completeness and overconfidence accumulation based on calibration.

4.1 The Failure of Softmax Normalization Under Poorly-Calibrated Scenarios

It is widely known that the `softmax` normalization can suffer from overconfidence issues (Odonnat et al., 2024; Wei et al., 2022b) and saturation of its outputs (Chen et al., 2017), with one entry close to one while the others are close to zero.

Analysis. To alleviate those issues, we first notice that the `softmax` can be decomposed as $\text{softmax}(\mathbf{q}) = \exp(\mathbf{q})/\sum_{k=1}^{K}\exp(\mathbf{q}_k) = (\phi \circ \exp)(\mathbf{q})$, where $\phi\colon \mathbb{R}_+^K \to \Delta_K$ writes $\phi(\mathbf{u}) = \mathbf{u}/\sum_{k=1}^{K}\mathbf{u}_k = \mathbf{u}/\|\mathbf{u}\|_1$. While ϕ has appealing property for normalization (see Proposition D.2), the exponential can accumulate prediction errors, leading to the `softmax` overconfidence and a biased accuracy estimation. In particular, assume that the k-th entry of the output of the neural network on a test sample $\mathbf{x}_i$ writes $\mathbf{q}_{i,k}^* + \varepsilon_k$, where $\mathbf{q}_{i,k}^*$ are the ground-truth logits and ε_k is the prediction error. Then, the n^{th}-order Taylor polynomial of the exponential writes

$$\exp(\mathbf{q}_{i,k}^* + \varepsilon_k) \approx 1 + (\mathbf{q}_{i,k}^* + \varepsilon_k) + \frac{(\mathbf{q}_{i,k}^* + \varepsilon_k)^2}{2!} + \ldots + \frac{(\mathbf{q}_{i,k}^* + \varepsilon_k)^n}{n!}. \tag{4}$$

Consequently, logit-based accuracy estimation methods using `softmax` are sensitive to prediction bias, leading to low-quality estimations in poorly calibrated scenarios.

Empirical evidence. We illustrate this phenomenon in Figure 2(a) on two datasets, where a pre-trained ResNet18 exhibits more pronounced calibration issues on PACS (Li et al., 2017) compared to Office Home (Venkateswara et al., 2017). Figure 2(b) shows that using MANO with `softmax` normalization is appropriate on Office-Home where the model is well calibrated but not in a mis-calibrated scenario on PACS. Conversely, using MANO with 2^{nd}-order Taylor approximation is appropriate under miscalibration on PACS but not on Office-Home. In both cases, we see that MANO can surpass the state-of-the-art method Nuclear (Deng et al., 2023) provided it uses the appropriate normalization σ. Figure 2(c) illustrates the impact of truncating Eq. (4) up to the n-th order. We conclude that a trade-off is needed between *information completeness on true logits and error accumulation* depending on the type of calibration scenario. Specifically, when the model is poorly calibrated on a given dataset (*i.e.*, ε_k large in absolute value), the normalization should focus on avoiding error accumulation, and when the model is well calibrated (*i.e.*, ε_k small in absolute value), the normalization should focus on information completeness.

4.2 Softrun: The Proposed Normalization Strategy

The above analysis shows that different calibration scenarios emphasize different information during normalization. Therefore, we propose a normalization strategy called `softrun` that normalizes the model outputs based on the calibration scenario. Given logits $\mathbf{q}_i \in \mathbb{R}^K$ and reusing the function ϕ previously introduced, it takes the general form:

$$\sigma(\mathbf{q}_i) = (\phi \circ v)(\mathbf{q}_i) = \frac{v(\mathbf{q}_i)}{\sum_{k=1}^{K} v(\mathbf{q}_i)_k} \in \Delta_K. \tag{5}$$

where $v\colon \mathbb{R}^K \to \mathbb{R}_+^K$ is designed to avoid error accumulation under poorly-calibrated scenarios by truncating the exponential (Taylor $n = 2$ in Eq. (4)) and using complete logits information under well-calibrated scenarios (`softmax`). As in practice, the calibration of the model on test data is unknown, `softrun` employs a simple yet effective strategy reminiscent of pseudo-labeling (Lee, 2013; Sohn et al., 2020). More specifically, given a test dataset $\mathcal{D}_{\text{test}} = \{\mathbf{x}_i\}_{i=1}^N$ and corresponding logits $\mathbf{q}_i = f(\mathbf{x}_i)$, a criterion $\Phi(\mathcal{D}_{\text{test}})$ is computed at the dataset level and the normalized logits are defined as[1]

$$v(\mathbf{q}_i) = \begin{cases} 1 + \mathbf{q}_i + \frac{\mathbf{q}_i^2}{2}, & \text{if } \Phi(\mathcal{D}_{\text{test}}) \leq \eta \qquad \text{(Taylor)} \\ \exp(\mathbf{q}_i), & \text{if } \Phi(\mathcal{D}_{\text{test}}) > \eta \qquad \texttt{(softmax)} \end{cases}. \tag{6}$$

We define $\Phi(\mathcal{D}_{\text{test}}) = -\frac{1}{NK}\sum_{j=1}^{N}\sum_{k=1}^{K}\log(\frac{\exp(\mathbf{q}_j)_k}{\sum_{j=1}^{K}\exp(\mathbf{q}_i)_j})$, which is equal, up to a constant, to the average KL divergence between the uniform distribution and the predicted `softmax` probabilities. It follows from Tian et al. (2021) that showed that this KL divergence was small when the uncertainty of the model was high and large for confident models. Hence, when the model is uncertain, *i.e.*, $\Phi(\mathcal{D}_{\text{test}}) \leq \eta$, we truncate the exponential to reduce error accumulation, and when the model is certain, *i.e.*, $\Phi(\mathcal{D}_{\text{test}}) > \eta$, complete information is used with the exact exponential (and we recover the `softmax`). Thus, `softrun` is designed to treat the problem with an additional level of complexity often overlooked by previous methods. While this comes at the cost of introducing the hyperparameter η, we fix $\eta = 5$ across all our experiments. This, along with the design of `softrun`, is justified both theoretically in Appendix E and experimentally in Appendix G.3.

[1] In practice if Taylor is applied, we replace $v(\mathbf{q}_i)$ by $v(\mathbf{q}_i) - \min v(\mathbf{q}_i)$ to make sure the final outputs have nonnegative entries. This is especially needed for approximation orders $n \geq 3$.

Table 1: Method comparison on four benchmarks using ResNet18, ResNet50, and WRN-50-2 under the **synthetic shift**, where R^2 refers to coefficients of determination, and ρ refers to the absolute value of Spearman correlation coefficients (higher is better). The best results for each metric are in **bold**. Overall, MANO achieves the highest R^2 and ρ values across different datasets and architectures, indicating its superior performance.

		Synthetic Shift																							
Dataset	Model	Rotation		ConfScore		Entropy		AgreeScore		ATC		Fréchet		Dispersion		ProjNorm		MDE		COT		Nuclear		MANO	
		R^2	ρ	R^2	ρ	R^2	ρ	R^2	ρ	R^2	ρ	R^2	ρ	R^2	ρ	R^2	ρ	R^2	ρ	R^2	ρ	R^2	ρ	R^2	ρ
CIFAR 10	ResNet18	0.822	0.951	0.869	0.985	0.899	0.987	0.663	0.929	0.884	0.985	0.950	0.971	0.968	0.990	0.936	0.982	0.957	0.987	0.989	0.995	**0.995**	**0.997**	**0.995**	**0.997**
	ResNet50	0.835	0.961	0.935	0.993	0.945	0.994	0.835	0.985	0.946	0.994	0.858	0.964	0.987	0.990	0.944	0.989	0.978	0.963	0.984	0.996	0.994	0.996	**0.996**	**0.997**
	WRN-50-2	0.862	0.976	0.943	0.994	0.942	0.994	0.856	0.986	0.947	0.994	0.814	0.973	0.962	0.988	0.961	0.989	0.930	0.809	0.988	0.994	0.994	0.995	**0.996**	**0.992**
	Average	0.840	0.963	0.916	0.991	0.930	0.992	0.785	0.967	0.926	0.991	0.874	0.970	**0.972**	0.990	0.947	0.987	0.955	0.920	0.987	0.995	0.995	**0.996**	**0.996**	0.995
CIFAR 100	ResNet18	0.860	0.936	0.916	0.985	0.891	0.979	0.902	0.973	0.938	0.986	0.888	0.968	0.952	0.988	0.979	0.980	0.975	0.994	0.991	0.995	0.989	0.995	**0.996**	**0.996**
	ResNet50	0.908	0.962	0.919	0.984	0.884	0.977	0.922	0.982	0.921	0.984	0.837	0.972	0.951	0.985	0.988	0.991	0.988	0.995	0.985	0.996	0.979	0.994	**0.995**	**0.997**
	WRN-50-2	0.924	0.970	0.971	0.984	0.968	0.981	0.955	0.977	0.978	0.993	0.865	0.987	0.980	0.991	0.990	0.991	0.995	0.994	0.987	0.997	0.962	0.988	**0.996**	**0.998**
	Average	0.898	0.956	0.936	0.987	0.915	0.983	0.927	0.982	0.946	0.988	0.864	0.976	0.962	0.988	0.985	0.987	0.986	0.994	0.988	0.996	0.977	0.993	**0.996**	**0.997**
TinyImageNet	ResNet18	0.786	0.946	0.670	0.869	0.592	0.842	0.561	0.853	0.751	0.945	0.826	0.970	0.966	0.986	0.970	0.981	0.941	0.993	**0.985**	0.994	0.983	0.994	0.981	**0.996**
	ResNet50	0.786	0.947	0.670	0.869	0.651	0.892	0.560	0.853	0.751	0.945	0.826	0.971	0.977	0.986	0.979	0.987	0.941	0.993	**0.980**	0.994	0.965	0.994	**0.980**	**0.996**
	WRN-50-2	0.878	0.967	0.757	0.951	0.704	0.935	0.654	0.904	0.635	0.897	0.884	0.984	0.968	0.986	0.965	0.983	0.961	0.996	**0.985**	0.997	0.962	0.988	0.979	**0.997**
	Average	0.805	0.959	0.727	0.920	0.650	0.890	0.599	0.878	0.693	0.921	0.847	0.976	0.970	0.987	0.972	0.984	0.950	0.995	**0.984**	0.995	0.968	0.993	0.980	**0.996**
ImageNet	ResNet18	-	-	0.979	0.991	0.963	0.991	-	-	0.974	0.983	0.802	0.974	0.940	0.971	0.975	0.993	0.924	0.994	**0.996**	**0.998**	0.992	0.997	0.992	0.997
	ResNet50	-	-	0.980	0.994	0.967	0.992	-	-	0.970	0.983	0.855	0.974	0.938	0.968	0.986	0.993	0.886	0.994	**0.993**	0.996	0.985	0.997	0.991	**0.998**
	WRN-50-2	-	-	0.983	0.991	0.963	0.991	-	-	0.983	0.993	0.909	0.988	0.939	0.976	0.978	0.993	0.880	0.997	0.989	0.994	0.987	0.998	**0.996**	**0.998**
	Average	-	-	0.981	0.993	0.969	0.992	-	-	0.976	0.987	0.855	0.979	0.939	0.972	0.980	0.993	0.897	0.995	0.993	0.996	0.988	0.998	**0.993**	**0.998**

5 Experiments

In this section, we conduct experiments with MANO that uses `softrun` to properly normalize logits.

5.1 Experimental Setup

Pre-training datasets. For pre-training the neural network, we use a diverse set of datasets including CIFAR-10, CIFAR-100 (Krizhevsky and Hinton, 2009), TinyImageNet (Le and Yang, 2015), ImageNet (Deng et al., 2009), PACS (Li et al., 2017), Office-Home (Venkateswara et al., 2017), DomainNet (Peng et al., 2019) and RR1-WILDS (Taylor et al., 2019), and BREEDS (Santurkar et al., 2020) which leverages class hierarchy of ImageNet (Deng et al., 2009) to create 4 datasets including Living-17, Nonliving-26, Entity-13 and Entity-30.

Test datasets. In our comprehensive evaluation, we consider 12 datasets with 3 types of distribution shifts: the synthetic, the natural, and the subpopulation shifts. To verify the effectiveness of our method under the synthetic shift, we use CIFAR-10C, CIFAR-100C, and ImageNet-C (Hendrycks and Dietterich, 2019) that span 19 types of corruption across 5 severity levels, as well as TinyImageNet-C (Hendrycks and Dietterich, 2019) with 15 types of corruption and 5 severity levels. For the natural shift, we use the domains excluded from training from PACS, Office-Home, DomainNet, and RR1-WILDS as the OOD datasets. For the novel subpopulation shift, we consider the BREEDS benchmark with Living-17, Nonliving-26, Entity-13, and Entity-30 which were constructed from ImageNet-C.

Training details. To show the versatility of our method across different architectures, we perform experiments with ResNet18, ResNet50 (He et al., 2016), and WRN-50-2 (Zagoruyko and Komodakis, 2016) models. We train them for 20 epochs for CIFAR-10 (Krizhevsky and Hinton, 2009) and 50 epochs for the other datasets. In all cases, we use SGD with a learning rate of 10^{-3}, cosine learning rate decay (Loshchilov and Hutter, 2016), a momentum of 0.9, and a batch size of 128.

Evaluation metrics. We use the coefficient of determination $R^2 \in [0, 1]$ (Nagelkerke et al., 1991) and the Spearman's rank correlation coefficient $\rho \in [-1, 1]$ (Kendall, 1948) to evaluate performance. The former measures the linearity and goodness of fit and 1 indicates a perfect fit. ρ measures monotonicity and values close to $\{-1, 1\}$ indicate strong correlation while 0 indicates no correlation.

Baselines. We consider 11 baselines commonly evaluated in the unsupervised accuracy estimation studies, including *Rotation Prediction* (Rotation) (Deng et al., 2021), *Averaged Confidence* (ConfScore) (Hendrycks and Gimpel, 2016), *Entropy* (Guillory et al., 2021), *Agreement Score* (AgreeScore) (Jiang et al., 2021), *Averaged Threshold Confidence* (ATC) (Garg et al., 2022), *AutoEval* (Fréchet) (Deng and Zheng, 2021), *ProjNorm* (Yu et al., 2022), *Dispersion Score* (Dispersion) (Xie et al., 2023), MDE (Peng et al., 2024), COT (Lu et al., 2024), and *Nuclear Norm* (Nuclear) (Deng et al., 2023).

Table 2: Method comparison on four benchmarks using ResNet18, ResNet50, and WRN-50-2 under **subpopulation shift** with R^2 and ρ metrics (the higher the better). The best results for each metric are in **bold**. Overall, MANO surpasses all its competitors.

Dataset	Model	Subpopulation Shift																					
		Rotation		ConfScore		Entropy		AgreeScore		ATC		Fréchet		Dispersion		ProjNorm		MDE		COT		Nuclear		MANO	
		R^2	ρ	R^2	ρ	R^2	ρ	R^2	ρ	R^2	ρ	R^2	ρ	R^2	ρ	R^2	ρ	R^2	ρ	R^2	ρ	R^2	ρ	R^2	ρ
Entity-13	ResNet18	0.927	0.961	0.795	0.940	0.794	0.935	0.543	0.919	0.823	0.945	0.950	0.981	0.937	0.968	0.952	0.981	0.927	0.995	0.960	0.985	0.978	0.991	**0.992**	**0.996**
	ResNet50	0.932	0.976	0.728	0.941	0.698	0.928	0.901	0.964	0.783	0.950	0.903	0.959	0.764	0.892	0.944	0.974	0.912	0.993	0.935	0.971	0.989	0.996	**0.993**	**0.998**
	WRN-50-2	0.939	0.983	0.930	0.977	0.919	0.973	0.871	0.935	0.936	0.980	0.906	0.958	0.815	0.905	0.950	0.977	0.925	0.995	0.944	0.979	0.989	0.995	**0.992**	**0.996**
	Average	0.933	0.973	0.817	0.953	0.804	0.945	0.772	0.939	0.847	0.958	0.920	0.966	0.948	0.977	0.839	0.922	0.921	0.995	0.947	0.979	0.985	0.994	**0.993**	**0.996**
Entity-30	ResNet18	0.964	0.979	0.570	0.836	0.553	0.832	0.542	0.935	0.611	0.845	0.849	0.978	0.929	0.968	0.952	0.987	0.931	0.994	0.971	0.993	0.980	0.993	**0.991**	**0.996**
	ResNet50	0.961	0.980	0.878	0.969	0.838	0.956	0.914	0.975	0.924	0.973	0.835	0.956	0.783	0.914	0.937	0.986	0.918	0.995	0.958	0.982	0.978	0.994	**0.988**	**0.997**
	WRN-50-2	0.940	0.978	0.897	0.974	0.878	0.970	0.826	0.955	0.936	0.984	0.927	0.973	0.927	0.973	0.959	0.986	0.925	0.995	0.944	0.979	0.985	0.996	**0.988**	**0.997**
	Average	0.955	0.978	0.781	0.926	0.756	0.919	0.728	0.956	0.823	0.934	0.871	0.969	0.880	0.952	0.949	0.987	0.925	0.995	0.970	0.988	0.981	0.994	**0.989**	**0.996**
Living-17	ResNet18	0.876	0.973	0.913	0.973	0.898	0.970	0.586	0.736	0.940	0.973	0.768	0.950	0.900	0.958	0.923	0.970	0.927	0.985	0.972	0.984	0.975	0.987	**0.980**	**0.991**
	ResNet50	0.906	0.956	0.880	0.967	0.853	0.961	0.633	0.802	0.938	0.976	0.771	0.926	0.851	0.929	0.903	0.924	0.914	0.985	0.953	0.973	0.967	0.976	**0.975**	**0.997**
	WRN-50-2	0.909	0.957	0.928	0.980	0.921	0.977	0.652	0.793	0.966	0.984	0.931	0.967	0.931	0.966	0.915	0.970	0.914	0.983	0.965	0.990	0.951	0.978	**0.961**	**0.996**
	Average	0.933	0.974	0.907	0.973	0.814	0.969	0.623	0.777	**0.948**	**0.978**	0.817	0.949	0.894	0.951	0.913	0.969	0.918	0.984	0.963	0.982	0.964	0.980	**0.972**	**0.995**
Nonliving-26	ResNet18	0.906	0.955	0.781	0.925	0.739	0.909	0.543	0.810	0.854	0.939	0.914	0.980	0.958	0.981	0.939	0.978	0.929	0.989	**0.982**	**0.992**	0.970	0.989	0.978	0.991
	ResNet50	0.916	0.970	0.832	0.942	0.776	0.918	0.638	0.837	0.893	0.960	0.848	0.950	0.805	0.907	0.873	0.972	0.907	0.993	0.962	0.984	0.956	0.985	**0.975**	**0.995**
	WRN-50-2	0.917	0.977	0.932	0.971	0.912	0.959	0.676	0.861	0.945	0.969	0.885	0.942	0.893	0.939	0.924	0.973	0.909	0.991	0.962	0.979	0.960	0.988	**0.978**	**0.992**
	Average	0.913	0.967	0.849	0.946	0.809	0.929	0.618	0.836	0.897	0.956	0.882	0.957	0.913	0.974	0.886	0.943	0.915	0.991	0.969	0.985	0.962	0.987	**0.977**	**0.992**

5.2 Main Takeaways

MANO improves over state-of-the-art. Tables 1 and 2 present the numerical results of unsupervised accuracy estimation across 8 datasets using 3 different network architectures, evaluated under synthetic and subpopulation shifts. These shifts are quantified by R^2 and ρ. Empirical results demonstrate that these distribution shifts do not significantly impact calibration (*i.e.*, $\Phi(\mathcal{D}_{\text{test}}) > \eta$). We observe that MANO consistently outperforms other baselines, achieving state-of-the-art performance. For instance, MANO achieves $R^2 > 0.960$ and $\rho > 0.990$ under subpopulation shift, whereas the performance of other baselines does not reach such consistently high levels.

MANO significantly boosts performance under the natural shift. Table 3 illustrates the results of accuracy estimation under the natural shift on four datasets. Under the challenging and natural shift that is more complex than the other distribution shifts, we empirically observe $\Phi(\mathcal{D}_{\text{test}}) \leq \eta$. From Table 3, we observe a significant improvement compared with the other baselines. In particular, most methods have an R^2 and a ρ under 0.9 while MANO reaches higher values. In addition, our method achieves the best performance on average on all four datasets. To visualize the estimation performance, we provide the scatter plots for *Dispersion Score*, *ProjNorm* and MANO in Figure 3 on Entity-18 with ResNet18. We find that MANO scores present a robust linear relationship with ground-truth OOD errors, while the other state-of-the-art baselines tend to exhibit a biased estimation of high test errors.

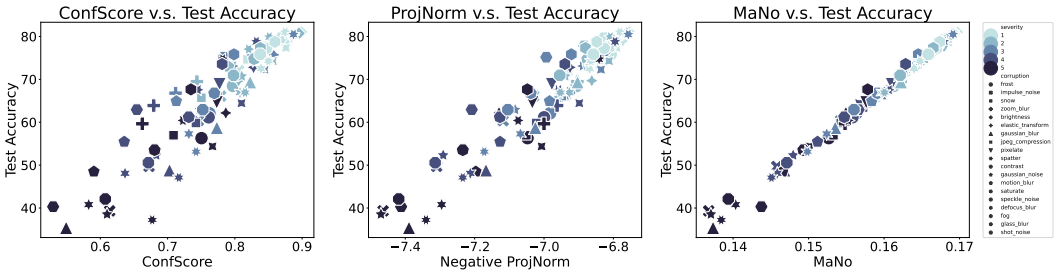


Figure 3: OOD error prediction versus ground-truth error on Entity-13 with ResNet18. This scatter plot compares MANO with Dispersion Score and ProjNorm. Each point represents one dataset under a specific type and severity of corruption. Different shapes indicate different types of corruption, while darker colors indicate higher severity levels. This indicates the qualitative superiority of MANO.

Improved robustness. Figure 4 presents a box plot showing the estimation robustness across different distribution shifts on all datasets except ImageNet, using ResNet18. Results for ImageNet are excluded due to the lack of *Rotation* and *AgreeScore* data for this dataset, as these two methods require retraining the networks. We observe that the estimation performance of MANO is more stable than other baselines across three types of distribution shifts. Additionally, MANO achieves the highest median estimation performance.

Table 3: Method comparison on four benchmarks using ResNet18, ResNet50 and WRN-50-2 under **natural shift** with R^2 and ρ metrics (the higher the better). The best results for each metric are in **bold**. Overall, MANO surpasses all the other baselines.

		Natural Shift																							
Dataset	Model	Rotation		ConfScore		Entropy		AgreeScore		ATC		Fréchet		Dispersion		ProjNorm		MDE		COT		Nuclear		MANO	
		R^2	ρ	R^2	ρ	R^2	ρ	R^2	ρ	R^2	ρ	R^2	ρ	R^2	ρ	R^2	ρ	R^2	ρ	R^2	ρ	R^2	ρ	R^2	ρ
PACS	ResNet18	0.822	0.895	0.594	0.755	0.624	0.755	0.613	0.832	0.514	0.650	0.624	0.804	0.832	0.825	0.161	0.419	0.003	0.153	0.790	0.783	0.609	0.874	**0.827**	**0.909**
	ResNet50	0.860	0.923	0.070	0.069	0.061	0.055	0.463	0.622	0.192	0.265	0.463	0.622	0.073	0.167	0.244	0.587	0.059	0.104	0.891	0.790	0.611	0.888	**0.923**	**0.958**
	WRN-50-2	0.865	0.902	0.646	0.678	0.629	0.671	0.377	0.858	0.752	0.832	0.558	0.832	0.111	0.167	0.474	0.650	0.072	0.244	0.890	0.888	0.607	0.867	**0.924**	**0.972**
	Average	0.849	0.906	0.437	0.501	0.438	0.494	0.488	0.770	0.486	0.582	0.548	0.337	0.338	0.275	0.293	0.552	0.045	0.065	0.857	0.820	0.609	0.876	**0.891**	**0.946**
Office-Home	ResNet18	0.822	0.930	0.795	0.909	0.761	0.881	0.054	0.146	0.571	0.615	0.605	0.755	0.453	0.664	0.064	0.202	0.331	0.650	0.863	0.874	0.692	0.783	**0.926**	**0.930**
	ResNet50	**0.851**	**0.944**	0.769	0.895	0.742	0.853	0.026	0.216	0.487	0.734	0.607	0.685	0.383	0.727	0.169	0.475	0.342	0.622	0.762	0.846	0.731	0.895	0.838	0.916
	WRN-50-2	0.823	**0.958**	0.741	0.874	0.696	0.846	0.132	0.405	0.383	0.643	0.589	0.706	0.456	0.713	0.172	0.531	0.342	0.650	**0.863**	0.874	0.766	0.874	0.800	0.895
	Average	0.832	**0.944**	0.768	0.892	0.733	0.860	0.071	0.256	0.480	0.664	0.601	0.715	0.431	0.702	0.135	0.403	0.339	0.650	0.781	0.855	0.730	0.850	**0.854**	0.913
DomainNet	ResNet18	0.568	0.692	0.670	0.736	0.423	0.609	0.326	0.668	0.429	0.597	0.704	0.903	0.202	0.516	0.219	0.443	0.358	0.445	0.897	0.910	0.758	0.789	**0.902**	**0.937**
	ResNet50	0.588	0.703	0.570	0.706	0.344	0.573	0.455	0.697	0.245	0.404	0.746	0.872	0.002	0.041	0.220	0.430	0.379	0.527	0.903	0.927	0.809	0.879	**0.910**	**0.950**
	WRN-50-2	0.609	0.712	0.774	0.874	0.711	0.845	0.437	0.698	0.846	0.918	0.585	0.831	0.003	0.034	0.363	0.466	0.520	0.713	0.885	0.935	0.850	0.911	**0.893**	**0.978**
	Average	0.588	0.702	0.671	0.722	0.493	0.676	0.406	0.688	0.507	0.639	0.678	0.869	0.069	0.197	0.234	0.446	0.419	0.562	0.894	0.919	0.805	0.895	**0.899**	**0.949**
RR1-WILDS	ResNet18	0.821	1.000	0.951	1.000	0.836	1.000	0.929	1.000	0.342	0.500	0.936	1.000	0.843	1.000	0.859	1.000	0.927	1.000	0.969	1.000	0.885	1.000	**0.983**	**1.000**
	ResNet50	0.740	1.000	0.918	1.000	0.819	1.000	0.938	1.000	**0.986**	**1.000**	0.935	1.000	0.737	1.000	0.867	1.000	0.938	1.000	0.960	1.000	0.906	1.000	0.978	**1.000**
	WRN-50-2	0.031	0.500	0.941	1.000	0.846	1.000	0.946	1.000	**0.988**	**1.000**	0.922	1.000	0.824	1.000	0.878	1.000	0.954	1.000	0.934	1.000	0.840	1.000	0.969	**1.000**
	Average	0.530	0.833	0.937	1.000	0.833	1.000	0.938	1.000	0.779	0.833	0.931	1.000	0.801	0.833	0.868	1.000	0.940	1.000	0.953	1.000	0.877	1.000	**0.977**	**1.000**

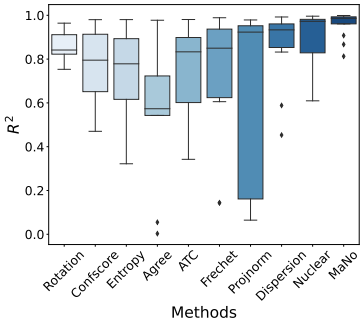


Figure 4: R^2 distribution with ResNet18 on all distribution shifts. Overall, MANO leads to the best and most robust estimations.

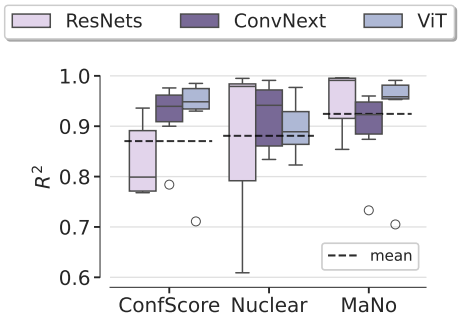


Figure 5: R^2 distribution using ResNets (average), ConvNext, or ViT on all distribution shifts. Again, MANO is the best method.

5.3 Additional Experiments

In this section, we discuss the results obtained with additional architectures, the ablation studies we conducted to validate our implementation choices, and the generalization capabilities of MANO.

Beyond ResNets. To demonstrate the efficiency and versatility of MANO, we conduct experiments on recent models such as Vision Transformers (Dosovitskiy, 2020, ViT) and ConvNeXt (Liu et al., 2022). We compare MANO to its best competitors on 6 datasets for 3 distribution shifts in Figure 5. The full results are gathered in Table 5 of Appendix F. We note that *ConfScore* is particularly strong with ConvNexts while MANO works the best with ResNets and ViT. Again, we observe that MANO is the best method overall.

Ablation study. To motivate our choices of implementation, we provide in Appendix G ablation studies on the L_p norm and the Taylor order n as well as a sensitive analysis on the calibration threshold η.

Generalization capabilities of MANO. To verify the generalization capabilities of MANO, we utilize designed scores calculated from ImageNet-C and their corresponding accuracy to fit a linear regression model. This model is then used to predict the test accuracy on ImageNet-V2-$\bar{C}$, which is generated using the 10 new corruptions provided by (Mintun et al., 2021) on ImageNet-V2 (Recht et al., 2019). These new corruptions are perceptually dissimilar from those in ImageNet-C, including warps, blurs, color distortions, noise additions, and obscuring effects. Figure 6 shows that ConfScore and Dispersion give two distinct trends, while Nuclear exhibits some deviations for ImageNet-V2-$\bar{C}$. In comparison, our MANO exhibits a consistent prediction pattern for both ImageNet-C and ImageNet-V2-$\bar{C}$, aligning well with the linear regression model trained on ImageNet-C. Additionally,

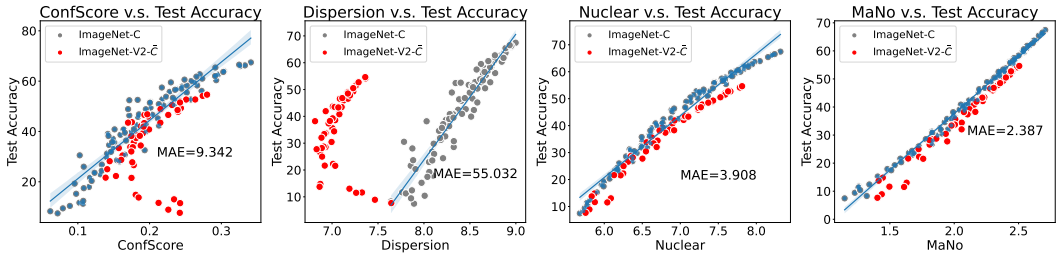


Figure 6: Comparison of generalization capability across four methods. Each subplot displays a linear regression model fitted on ImageNet-C, which is used to predict the accuracy on ImageNet-V2-$\bar{C}$. The mean absolute error (MAE) is reported. All experiments are conducted using ResNet18.

experimental results on ImageNet-C and ImageNet-$\bar{C}$, generated from the validation set of ImageNet, are provided in Appendix G.5, further demonstrating the superiority of MANO.

5.4 Discussion

In this section, we discuss how MANO can be applied in practice, the benefit of combining `softrun` with other estimation baselines, and the limitations of our approach.

Real-world applications. In Appendix H, we discuss how MANO can be used in real-world applications. In particular, our additional results with the Mean Absolute Error (MAE) metric confirm the superiority of MANO.

Can `softrun` enhance other logit-based methods? To study this, we conducted an ablation study by equipping `softrun` with *Nuclear* (Deng et al., 2023), *ConfScore* (Hendrycks and Gimpel, 2016), and our MANO. In Table 4, we observe that `softrun` significantly enhances the estimation performance R^2 of Nuclear. For example, Nuclear is improved from 0.692 to 0.826 on poorly-calibrated Office-Home.

Table 4: Impact of `softrun` on other logit-based methods. `softrun` significantly boosts the performance of Nuclear. The metric used is R^2.

Dataset	ConfScore		MANO		Nuclear	
	w/o	w/	w/o	w/	w/o	w/
PACS	**0.594**	0.574	0.541	**0.827**	0.609	**0.851**
Office-Home	0.795	**0.829**	**0.929**	0.926	0.692	**0.826**

Limitations. Despite its soundness and strong empirical performance, we acknowledge that our method has potential areas for improvement. One of these is the dependence on η of the selection criterion in Eq. (6). We elaborate on this discussion in Appendix E.3. In future work, we will explore a smoother way to automatically select the optimal normalization function without requiring hyperparameters. Additionally, if multiple validation sets are provided, as in (Deng et al., 2021; Deng and Zheng, 2021), we could select η based on those sets.

6 Conclusion

In this paper, we introduce MANO, a simple yet effective training-free method to estimate test accuracy in an unsupervised manner using the Matrix Norm of neural network predictions on test data. Our approach is inspired by the LDS assumption that optimal decision boundaries should lie in low-density regions. To mitigate the negative impact of different distribution shifts on estimation performance, we first demonstrate the failure of `softmax` normalization under poor calibration, due to the accumulation of overconfident errors. We then propose a normalization strategy based on Taylor polynomial approximation, balancing logits information and error accumulation. Extensive experiments show that MANO consistently outperforms previous methods across various distribution shifts. This work highlights that logits imply the feature-to-boundary distance and considers the impact of calibration on estimation performance. We hope our insights inspire future research to explore the relationship between model outputs and generalization.

Acknowledgements

Ambroise Odonnat would like to thank Alexandre Ramé and Youssef Attia El Hili for the fruitful discussions that led to this work. The authors thank the anonymous reviewers and meta-reviewers for their time and constructive feedback. This work was enabled thanks to open-source software such as Python (Van Rossum and Drake Jr, 1995), PyTorch (Paszke et al., 2019) and Matplotlib (Hunter, 2007). This research is supported by the National Research Foundation Singapore and DSO National Laboratories under the AI Singapore Programme (AISGAward No: AISG2-GC-2023-009).

References

Amini, M.-R., Feofanov, V., Pauletto, L., Hadjadj, L., Devijver, E., and Maximov, Y. (2022). Self-training: A survey. *arXiv preprint arXiv:2202.12040*.

Banerjee, A. (2006). On bayesian bounds. In *Proceedings of the 23rd International Conference on Machine Learning*, ICML '06, page 81–88, New York, NY, USA. Association for Computing Machinery.

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*, page 738. Springer.

Blondel, M., Martins, A., and Niculae, V. (2019). Learning classifiers with Fenchel-Young losses: Generalized entropies, margins, and algorithms. In Chaudhuri, K. and Sugiyama, M., editors, *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*, pages 606–615.

Blondel, M., Martins, A. F. T., and Niculae, V. (2020). Learning with Fenchel-Young losses. *J. Mach. Learn. Res.*, 21(1).

Chapelle, O. and Zien, A. (2005). Semi-supervised classification by low density separation. In *Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics*, pages 57–64.

Chaudhari, P., Choromanska, A., Soatto, S., LeCun, Y., Baldassi, C., Borgs, C., Chayes, J., Sagun, L., and Zecchina, R. (2019). Entropy-SGD: Biasing gradient descent into wide valleys. *Journal of Statistical Mechanics: Theory and Experiment*, 2019(12):124018.

Chen, B., Deng, W., and Du, J. (2017). Noisy softmax: Improving the generalization ability of dcnn via postponing the early softmax saturation. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4021–4030.

Chen, D., Wang, D., Darrell, T., and Ebrahimi, S. (2022). Contrastive test-time adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 295–305.

Chen, J., Liu, F., Avci, B., Wu, X., Liang, Y., and Jha, S. (2021). Detecting errors and estimating accuracy on unlabeled data with self-training ensembles. *Advances in Neural Information Processing Systems*, 34:14980–14992.

Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. (2020). A simple framework for contrastive learning of visual representations. In *International Conference on Machine Learning*, pages 1597–1607.

Choromanska, A., Henaff, M., Mathieu, M., Arous, G. B., and LeCun, Y. (2015). The loss surfaces of multilayer networks. In *Artificial Intelligence and Statistics*, pages 192–204.

Chuang, C.-Y., Torralba, A., and Jegelka, S. (2020). Estimating generalization under distribution shifts via domain-invariant representations. *arXiv preprint arXiv:2007.03511*.

Croce, F. and Hein, M. (2020). Minimally distorted adversarial examples with a fast adaptive boundary attack. In *International Conference on Machine Learning*, pages 2196–2205.

Dauphin, Y. N., Pascanu, R., Gulcehre, C., Cho, K., Ganguli, S., and Bengio, Y. (2014). Identifying and attacking the saddle point problem in high-dimensional non-convex optimization. *Advances in Neural Information Processing Systems*, 27.

de Brébisson, A. and Vincent, P. (2016). An exploration of softmax alternatives belonging to the spherical loss family. In *International Conference on Learning Representations, (ICLR)*.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). ImageNet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 248–255. Ieee.

Deng, J., Guo, J., Xue, N., and Zafeiriou, S. (2019). ArcFace: Additive angular margin loss for deep face recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4690–4699.

Deng, W., Gould, S., and Zheng, L. (2021). What does rotation prediction tell us about classifier accuracy under varying testing environments? In *International Conference on Machine Learning (ICML)*, pages 2579–2589.

Deng, W., Suh, Y., Gould, S., and Zheng, L. (2023). Confidence and dispersity speak: Characterising prediction matrix for unsupervised accuracy estimation. *arXiv preprint arXiv:2302.01094*.

Deng, W. and Zheng, L. (2021). Are labels always necessary for classifier accuracy evaluation? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15069–15078.

Dinh, L., Pascanu, R., Bengio, S., and Bengio, Y. (2017). Sharp minima can generalize for deep nets. In *International Conference on Machine Learning*, pages 1019–1028.

Dohmatob, E. (2020). Distance from a point to a hyperplane. https://math.stackexchange.com/questions/1210545/distance-from-a-point-to-a-hyperplane.

Donmez, P., Lebanon, G., and Balasubramanian, K. (2010). Unsupervised supervised learning i: Estimating classification and regression errors without labels. *Journal of Machine Learning Research*, 11(4).

Dosovitskiy, A. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

Fan, X., Wang, Q., Ke, J., Yang, F., Gong, B., and Zhou, M. (2021). Adversarially adaptive normalization for single domain generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8208–8217.

Fawzi, A., Moosavi-Dezfooli, S.-M., Frossard, P., and Soatto, S. (2018). Empirical study of the topology and geometry of deep networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3762–3770.

Feofanov, V., Devijver, E., and Amini, M.-R. (2019). Transductive bounds for the multi-class majority vote classifier. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 3566–3573.

Feofanov, V., Devijver, E., and Amini, M.-R. (2024). Multi-class probabilistic bounds for majority vote classifiers with partially labeled data. *Journal of Machine Learning Research*, 25(104):1–47.

Feofanov, V., Tiomoko, M., and Virmaux, A. (2023). Random matrix analysis to balance between supervised and unsupervised learning under the low density separation assumption. In *Proceedings of the 40th International Conference on Machine Learning*, pages 10008–10033.

Freeman, C. D. and Bruna, J. (2016). Topology and geometry of half-rectified network optimization. *arXiv preprint arXiv:1611.01540*.

Garg, S., Balakrishnan, S., Lipton, Z. C., Neyshabur, B., and Sedghi, H. (2022). Leveraging unlabeled data to predict out-of-distribution performance. *arXiv preprint arXiv:2201.04234*.

Geirhos, R., Rubisch, P., Michaelis, C., Bethge, M., Wichmann, F. A., and Brendel, W. (2018). ImageNet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness. *arXiv preprint arXiv:1811.12231*.

Gell-Mann, M. and Tsallis, C. (2004). *Nonextensive Entropy: Interdisciplinary Applications*. Oxford University Press.

Gini, C. (1912). *Variabilità e mutabilità: contributo allo studio delle distribuzioni e delle relazioni statistiche. [Fasc. I.]*. Studi economico-giuridici pubblicati per cura della facoltà di Giurisprudenza della R. Università di Cagliari. Tipogr. di P. Cuppini.

Guillory, D., Shankar, V., Ebrahimi, S., Darrell, T., and Schmidt, L. (2021). Predicting with confidence on unseen distributions. In *Proceedings of the IEEE/CVF international Conference on Computer Vision (ICCV)*, pages 1134–1144.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778.

He, W., Li, B., and Song, D. (2018). Decision boundary analysis of adversarial examples. In *International Conference on Learning Representations*.

Hendrycks, D. and Dietterich, T. (2019). Benchmarking neural network robustness to common corruptions and perturbations. *arXiv preprint arXiv:1903.12261*.

Hendrycks, D. and Gimpel, K. (2016). A baseline for detecting misclassified and out-of-distribution examples in neural networks. *arXiv preprint arXiv:1610.02136*.

Hendrycks, D. and Mazeika, M. (2022). X-risk analysis for ai research. *arXiv preprint arXiv:2206.05862*.

Heo, B., Lee, M., Yun, S., and Choi, J. Y. (2019). Knowledge distillation with adversarial samples supporting decision boundary. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 3771–3778.

Hunter, J. D. (2007). Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9(3):90–95.

Jiang, Y., Nagarajan, V., Baek, C., and Kolter, J. Z. (2021). Assessing generalization of SGD via disagreement. *arXiv preprint arXiv:2106.13799*.

Karimi, H., Derr, T., and Tang, J. (2019). Characterizing the decision boundary of deep neural networks. *arXiv preprint arXiv:1912.11460*.

Kendall, M. G. (1948). Rank correlation methods. *Michigan University*.

Koh, P. W., Sagawa, S., Marklund, H., Xie, S. M., Zhang, M., Balsubramani, A., Hu, W., Yasunaga, M., Phillips, R. L., Gao, I., et al. (2021). Wilds: A benchmark of in-the-wild distribution shifts. In *International Conference on Machine Learning*, pages 5637–5664.

Krizhevsky, A. and Hinton, G. (2009). Learning multiple layers of features from tiny images. *Technical Report*.

Le, Y. and Yang, X. (2015). Tiny ImageNet visual recognition challenge. *CS 231N*, 7(7):3.

Lee, D., Yu, S., and Yu, H. (2020). Multi-class data description for out-of-distribution detection. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1362–1370.

Lee, D.-H. (2013). Pseudo-Label : The Simple and Efficient Semi-Supervised Learning Method for Deep Neural Networks. *ICML 2013 Workshop : Challenges in Representation Learning (WREPL)*.

Li, D., Yang, Y., Song, Y.-Z., and Hospedales, T. M. (2017). Deeper, broader and artier domain generalization. In *Proceedings of the IEEE international Conference on Computer Vision*, pages 5542–5550.

Li, J., Shen, C., Kong, L., Wang, D., Xia, M., and Zhu, Z. (2022). A new adversarial domain generalization network based on class boundary feature detection for bearing fault diagnosis. *IEEE Transactions on Instrumentation and Measurement*, 71:1–9.

Li, Y., Ding, L., and Gao, X. (2018). On the decision boundary of deep neural networks. *arXiv preprint arXiv:1808.05385*.

Liu, W., Wen, Y., Yu, Z., Li, M., Raj, B., and Song, L. (2017). Sphereface: Deep hypersphere embedding for face recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 212–220.

Liu, Z., Mao, H., Wu, C.-Y., Feichtenhofer, C., Darrell, T., and Xie, S. (2022). A convnet for the 2020s. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11976–11986.

Loshchilov, I. and Hutter, F. (2016). SGDR: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*.

Lu, Y., Qin, Y., Zhai, R., Shen, A., Chen, K., Wang, Z., Kolouri, S., Stepputtis, S., Campbell, J., and Sycara, K. (2023). Characterizing out-of-distribution error via optimal transport. *arXiv preprint arXiv:2305.15640*.

Lu, Y., Qin, Y., Zhai, R., Shen, A., Chen, K., Wang, Z., Kolouri, S., Stepputtis, S., Campbell, J., and Sycara, K. (2024). Characterizing out-of-distribution error via optimal transport. *Advances in Neural Information Processing Systems*, 36.

Madani, O., Pennock, D., and Flake, G. (2004). Co-validation: Using model disagreement on unlabeled data to validate classification algorithms. *Advances in Neural Information Processing Systems (NeurIPS)*, 17.

Mensch, A., Blondel, M., and Peyré, G. (2019). Geometric losses for distributional learning. In *Proceedings of the 36th International Conference on Machine Learning*, pages 4516–4525.

Mickisch, D., Assion, F., Greßner, F., Günther, W., and Motta, M. (2020). Understanding the decision boundary of deep neural networks: An empirical study. *arXiv preprint arXiv:2002.01810*.

Mintun, E., Kirillov, A., and Xie, S. (2021). On interaction between augmentations and corruptions in natural corruption robustness. *Advances in Neural Information Processing Systems*, 34:3571–3583.

Montufar, G. F., Pascanu, R., Cho, K., and Bengio, Y. (2014). On the number of linear regions of deep neural networks. *Advances in Neural Information Processing Systems*, 27.

Muzellec, B., Nock, R., Patrini, G., and Nielsen, F. (2017). Tsallis regularized optimal transport and ecological inference. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, AAAI'17, page 2387–2393.

Nagelkerke, N. J. et al. (1991). A note on a general definition of the coefficient of determination. *Biometrika*, 78(3):691–692.

Negrinho, R. and Martins, A. (2014). Orbit regularization. In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N., and Weinberger, K., editors, *Advances in Neural Information Processing Systems*, volume 27.

Odonnat, A., Feofanov, V., and Redko, I. (2024). Leveraging ensemble diversity for robust self-training in the presence of sample selection bias. In Dasgupta, S., Mandt, S., and Li, Y., editors, *Proceedings of The 27th International Conference on Artificial Intelligence and Statistics*, volume 238 of *Proceedings of Machine Learning Research*, pages 595–603. PMLR.

Oord, A. v. d., Li, Y., and Vinyals, O. (2018). Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. (2019). Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.

Peng, R., Zou, H., Wang, H., Zeng, Y., Huang, Z., and Zhao, J. (2024). Energy-based automated model evaluation. *arXiv preprint arXiv:2401.12689*.

Peng, X., Bai, Q., Xia, X., Huang, Z., Saenko, K., and Wang, B. (2019). Moment matching for multi-source domain adaptation. In *Proceedings of the IEEE/CVF international Conference on Computer Vision*, pages 1406–1415.

Peters, B., Niculae, V., and Martins, A. F. T. (2019). Sparse sequence-to-sequence models. In Korhonen, A., Traum, D., and Màrquez, L., editors, *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1504–1519.

Platanios, E., Poon, H., Mitchell, T. M., and Horvitz, E. J. (2017). Estimating accuracy from unlabeled data: A probabilistic logic approach. *Advances in Neural Information Processing Systems (NeurIPS)*, 30.

Platanios, E. A., Dubey, A., and Mitchell, T. (2016). Estimating accuracy from unlabeled data: A bayesian approach. In *International Conference on Machine Learning (ICML)*, pages 1416–1425.

Poole, B., Lahiri, S., Raghu, M., Sohl-Dickstein, J., and Ganguli, S. (2016). Exponential expressivity in deep neural networks through transient chaos. *Advances in Neural Information Processing Systems*, 29.

Quionero-Candela, J., Sugiyama, M., Schwaighofer, A., and Lawrence, N. D. (2009). *Dataset Shift in Machine Learning*. The MIT Press.

Ranjan, R., Castillo, C. D., and Chellappa, R. (2017). L2-constrained softmax loss for discriminative face verification. *arXiv preprint arXiv:1703.09507*.

Recht, B., Roelofs, R., Schmidt, L., and Shankar, V. (2019). Do imagenet classifiers generalize to imagenet? In *International Conference on Machine Learning*, pages 5389–5400.

Rusak, E., Schneider, S., Pachitariu, G., Eck, L., Gehler, P. V., Bringmann, O., Brendel, W., and Bethge, M. (2022). If your data distribution shifts, use self-learning. *Transactions on Machine Learning Research*. Expert Certification.

Santurkar, S., Tsipras, D., and Madry, A. (2020). Breeds: Benchmarks for subpopulation shift. *arXiv preprint arXiv:2008.04859*.

Seldin, Y. and Tishby, N. (2010). Pac-bayesian analysis of co-clustering and beyond. *Journal of Machine Learning Research*, 11(117):3595–3646.

Seo, S., Suh, Y., Kim, D., Kim, G., Han, J., and Han, B. (2020). Learning to optimize domain specific normalization for domain generalization. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXII 16*, pages 68–83. Springer.

Sneddon, R. (2007). The Tsallis entropy of natural information. *Physica A: Statistical Mechanics and its Applications*, 386(1):101–118.

Sohn, K. (2016). Improved deep metric learning with multi-class n-pair loss objective. *Advances in Neural Information Processing Systems*, 29.

Sohn, K., Berthelot, D., Li, C.-L., Zhang, Z., Carlini, N., Cubuk, E. D., Kurakin, A., Zhang, H., and Raffel, C. (2020). Fixmatch: simplifying semi-supervised learning with consistency and confidence. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*.

Taylor, J., Earnshaw, B., Mabey, B., Victors, M., and Yosinski, J. (2019). Rxrx1: An image set for cellular morphological variation across many experimental batches. In *International Conference on Learning Representations (ICLR)*.

Teimoori, Z., Rezazadeh, K., and Rostami, A. (2024). Inflation based on the Tsallis entropy. *The European Physical Journal C*, 84(1):80.

Tian, J., Hsu, Y.-C., Shen, Y., Jin, H., and Kira, Z. (2021). Exploring covariate and concept shift for out-of-distribution detection. In *NeurIPS 2021 Workshop on Distribution Shifts: Connecting Methods and Applications*.

Tsallis, C. (1988). Possible generalization of Boltzmann-Gibbs statistics. *Journal of Statistical Physics*, 52(1):479–487.

Tu, W., Deng, W., Gedeon, T., and Zheng, L. (2023). A bag-of-prototypes representation for dataset-level applications. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2881–2892.

Van Rossum, G. and Drake Jr, F. L. (1995). *Python reference manual*. Centrum voor Wiskunde en Informatica Amsterdam.

Vapnik, V. N. (1998). *Statistical Learning Theory*. Wiley-Interscience.

Veličković, P., Perivolaropoulos, C., Barbero, F., and Pascanu, R. (2024). softmax is not enough (for sharp out-of-distribution).

Venkateswara, H., Eusebio, J., Chakraborty, S., and Panchanathan, S. (2017). Deep hashing network for unsupervised domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5018–5027.

Wang, D., Shelhamer, E., Liu, S., Olshausen, B., and Darrell, T. (2021). Tent: Fully test-time adaptation by entropy minimization. In *International Conference on Learning Representations*.

Wang, F., Xiang, X., Cheng, J., and Yuille, A. L. (2017). Normface: L2 hypersphere embedding for face verification. In *Proceedings of the 25th ACM international conference on Multimedia*, pages 1041–1049.

Wei, H., Xie, R., Cheng, H., Feng, L., An, B., and Li, Y. (2022a). Mitigating neural network overconfidence with logit normalization. *arXiv preprint arXiv:2205.09310*.

Wei, H., Xie, R., Cheng, H., Feng, L., An, B., and Li, Y. (2022b). Mitigating neural network overconfidence with logit normalization. In *Proceedings of the 39th International Conference on Machine Learning*, pages 23631–23644.

Wu, Z., Xiong, Y., Yu, S. X., and Lin, D. (2018). Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3733–3742.

Xie, R., Odonnat, A., Feofanov, V., Redko, I., Zhang, J., and An, B. (2024). Characterising gradients for unsupervised accuracy estimation under distribution shift. *arXiv preprint arXiv:2401.08909*.

Xie, R., Wei, H., Cao, Y., Feng, L., and An, B. (2023). On the importance of feature separability in predicting out-of-distribution error. *arXiv preprint arXiv:2303.15488*.

Yousefzadeh, R. (2021). Deep learning generalization and the convex hull of training sets. *arXiv preprint arXiv:2101.09849*.

Yu, Y., Yang, Z., Wei, A., Ma, Y., and Steinhardt, J. (2022). Predicting out-of-distribution error with the projection norm. *arXiv preprint arXiv:2202.05834*.

Zagoruyko, S. and Komodakis, N. (2016). Wide residual networks. In *British Machine Vision Conference (BMVC)*.

Zekri, O., Odonnat, A., Benechehab, A., Bleistein, L., Boullé, N., and Redko, I. (2024). Large language models as markov chains.

Zhang, X., Zhao, R., Qiao, Y., Wang, X., and Li, H. (2019). Adacos: Adaptively scaling cosine logits for effectively learning deep face representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10823–10832.

Appendix

Roadmap. We provide the pseudo-code of MANO in Appendix A. We discuss related work in Appendix B and provide some background on Tsallis entropies in Appendix C. Appendix D contains detailed proofs of our theoretical results. Additional discussion and theoretical insights into Section 4 are given in Appendix E. In Appendix F, we conduct experiments with ViT and ConvNext architectures and provide a thorough ablation study and sensitivity analysis in Appendix G. Finally, we explain how MANO can be used in practice in Appendix H. We display the corresponding table of contents below.

Table of Contents

A Pseudo-Code of MANO

Algorithm 1 summarizes MANO introduced in Section 3.2, which is a lightweight, training-free method for unsupervised accuracy estimation using the neural network's outputs. We open-sourced the code of MANO at https://github.com/Renchunzi-Xie/MaNo.

Algorithm 1: Our proposed algorithm, MANO, for unsupervised accuracy estimation.

Input: Model f pre-trained on $\mathcal{D}_{\text{train}}$, test dataset $\mathcal{D}_{\text{test}} = \{\mathbf{x}_i\}_{i=1}^{N}$.
Parameters: Hyperparameter $p > 1$.
Initialization: Empty prediction matrix $\mathbf{Q} \in \mathbb{R}^{N \times K}$.
Criterion: compute criterion $\Phi(\mathcal{D}_{\text{test}})$ and select σ following Eq. (5) and Eq. (6).
for $i \in [\![1, N]\!]$ **do**
  Inference: recover logits $\mathbf{q}_i = f(\mathbf{x}_i) \in \mathbb{R}^K$.
  Normalization: obtain normalized logits $\sigma(\mathbf{q}_i) \in \Delta_K$.
  Update: fill the prediction matrix $\mathbf{Q}_i \leftarrow \sigma(\mathbf{q}_i)$ following Eq. (1).
end
Output Estimation score $\mathcal{S}(f, \mathcal{D}_{\text{test}}) = \frac{1}{\sqrt[p]{NK}} \|\mathbf{Q}\|_p$ following Eq. (2).

B Extended Related Work

Unsupervised accuracy estimation. This task aims to estimate model generalization performance on unlabeled test sets. To achieve this, several directions have been proposed. *(1) Utilizing model outputs:* One popular research direction is to use the model outputs on distribution-shifted data to construct a linear relationship with the test accuracy (Deng et al., 2023; Garg et al., 2022; Guillory et al., 2021; Hendrycks and Gimpel, 2016; Xie et al., 2024). Some of these approaches are limited by requiring access to the training set (Chen et al., 2021; Chuang et al., 2020). The most recent work (Deng et al., 2023) uses the nuclear norm of the softmax probability matrix as a training-free accuracy estimator. However, it significantly suffers from the overconfidence issues (Wei et al., 2022a), leading to fluctuating estimation performance across natural distribution shifts. Our work focuses on addressing this issue by balancing logit-information completeness and overconfidence-information accumulation. *(2) Considering distribution discrepancy:* another direction examines the negative relation between test accuracy and the distribution discrepancy between the training and test datasets (Deng and Zheng, 2021; Lu et al., 2023; Tu et al., 2023; Yu et al., 2022). However, commonly-used distribution distances do not guarantee stable accuracy estimation under different distribution shifts (Guillory et al., 2021; Xie et al., 2023), and some of these methods are time-consuming on large-scale datasets due to the requirement of training data (Deng and Zheng, 2021). *(3) Constructing unsupervised losses:* methods such as data augmentation and multiple-classifier agreement have also been introduced (Jiang et al., 2021; Madani et al., 2004; Platanios et al., 2017, 2016). However, they usually require special model architectures, undermining their practical applicability.

Distance to decision boundaries. The idea of treating distance to the decision boundary as an indicator of confidence originates from classical support vector machines (Vapnik, 1998). Decision boundaries of deep neural networks have been studied in various contexts. For example, some works explore the geometric properties of deep neural networks either in the input space (Fawzi et al., 2018; Karimi et al., 2019; Montufar et al., 2014; Poole et al., 2016) or in the weight space (Chaudhari et al., 2019; Choromanska et al., 2015; Dauphin et al., 2014; Dinh et al., 2017; Freeman and Bruna, 2016). Some works apply the properties of decision boundaries to address practical questions, such as adversarial defense (Croce and Hein, 2020; He et al., 2018; Heo et al., 2019), OOD detection (Lee et al., 2020), and domain generalization (Li et al., 2022; Yousefzadeh, 2021). As we discuss in Section 3.1, those approaches that use the distance to the decision boundary as an *unsupervised* indicator of confidence, rely on the low-density separation assumption (Chapelle and Zien, 2005), which states that the classifier must mistake mostly in the low margin zone (Feofanov et al., 2019, 2024). In our work, under this assumption, similar to Li et al. (2018), we use the distance between the learned intermediate feature to each decision boundary in the last hidden space.

Normalization in deep learning. Normalization is a crucial technique extensively utilized across various fields in deep learning, including domain generalization (Fan et al., 2021; Seo et al., 2020; Wang et al., 2021), metric learning (Oord et al., 2018; Sohn, 2016; Wu et al., 2018), face recognition (Deng et al., 2019; Liu et al., 2017; Ranjan et al., 2017; Wang et al., 2017; Zhang et al., 2019) and self-supervised learning (Chen et al., 2020). For example, TENT (Wang et al., 2021) normalizes features of test data using the mean value and standard deviation estimated from the target data. L_2-constrained softmax (Ranjan et al., 2017) introduces L_2 normalization on features. These normalization techniques are primarily employed to adapt new samples to familiar domains, calculate similarity, and speed up convergence. However, our proposed normalization focuses on reducing the negative implications of poorly calibrated scenarios.

C Background on Tsallis Entropies

The definition of Tsallis α-entropies (Tsallis, 1988) is given below.

Definition C.1 (Tsallis α-entropies)**.** *Let $\mathbf{p} \in \Delta_K$ be a probability distribution. Let $\alpha > 1$ and $k \geq 0$. The Tsallis α-entropy is defined as:*

$$\mathbf{H}_\alpha^{\mathrm{T}}(\mathbf{p}) = k(\alpha - 1)^{-1}(1 - \|\mathbf{p}\|_\alpha^\alpha).$$

It is common to take $k = 1$ or $k = \frac{1}{\alpha}$ following Blondel et al. (2019). The Tsallis α-entropy generalizes the Boltzmann-Gibbs theory of statistic mechanics to nonextensive systems. It has been used as a measure of disorder and uncertainty in many applications (Gell-Mann and Tsallis, 2004; Negrinho and Martins, 2014; Sneddon, 2007; Teimoori et al., 2024), including in Machine Learning (Blondel et al., 2019, 2020; Muzellec et al., 2017). Moreover, they generalize two widely-known measures of uncertainty. Indeed, the limit case $\alpha \to 1$ leads to the Shannon entropy $\mathbf{H}_S$ (see Peters et al., 2019, Appendix A.1), *i.e.*, $\lim_{\alpha \to 1} \mathbf{H}_\alpha^{\mathrm{T}}(\mathbf{p}) = \mathbf{H}_S(\mathbf{p}) = -\sum_{j=1}^K p_j \ln(p_j)$, while taking $\alpha = 2$ leads to the Gini index $\mathbf{G}$, a popular impurity measure fo decision trees (Gini, 1912), *i.e.*, $\mathbf{H}_2^{\mathrm{T}}(\mathbf{p}) = \frac{1}{2}(1 - \|p\|_2^2) = \mathbf{G}(\mathbf{p})$. Tsallis entropies measure the uncertainty: the higher the entropy the greater the uncertainty. From a probabilistic perspective, the entropy will take high values for *uncertain* probability distributions, *i.e.*, close to the uniform distribution. We visualize the evolution of the Tsallis entropy for varying parameters α in Figure 7, where the case $\alpha = 1$ corresponds to the Shannon entropy.

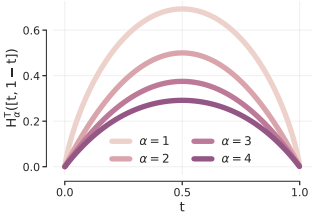


Figure 7: Tsallis α-entropies of $[t, 1-t]$ for $t \in [0, 1]$.

D Proofs

In this section, we detail the proofs of our theoretical results.

Notations. Scalar values are denoted by regular letters (e.g., parameter λ), vectors are represented in bold lowercase letters (e.g., vector $\mathbf{x}$) and matrices are represented by bold capital letters (e.g., matrix $\mathbf{A}$). The i-th row of the matrix $\mathbf{A}$ is denoted by $\mathbf{A}_i$, its j-th column is denoted by $\mathbf{A}_{\cdot,j}$ and its elements are denoted by $\mathbf{A}_{ij}$. The trace of a matrix $\mathbf{A}$ is denoted by $\mathrm{Tr}(\mathbf{A})$ and its transpose by $\mathbf{A}^\top$. The L_p norm of a vector $\mathbf{x}$ is denoted by $\|\mathbf{x}\|_p$, and by abuse of notation we denote it by $\|\mathbf{A}\|_p$ for a matrix $\mathbf{A}$ with $\|\mathbf{A}\|_p^p = \sum_i \|\mathbf{A}_i\|_p^p = \sum_{ij} |\mathbf{A}_{ij}|^p$. Let $\Delta_K \coloneqq \{\mathbf{p} \in [0, 1]^K \,|\, \sum_{i=1}^K \mathbf{p}_i = 1\}$ be the K-dimensional probability simplex.

D.1 Impact of Prediction Errors

Let $\mathbf{x} \in \mathcal{D}_{\text{test}}$ be a test sample with ground-truth label $y \in \{1, \ldots, K\}$. In multi-class classification, the `softmax` operator is used to approximate the posterior probability $p(y|\mathbf{x})$ (see Bishop, 2006, chap.4, p.198). Reusing the notations of Section 4, because of the distribution shifts between source and target, logits are subject to a prediction bias $\boldsymbol{\varepsilon} = (\varepsilon_k)_k$ and write $f(\mathbf{x}) = \mathbf{q}^* + \boldsymbol{\varepsilon}$ where $\mathbf{q}^*$ are ground-truth logits. In this section, we study the impact of such bias on the approximation of the posterior $p(y|\mathbf{x})$.

Impact on the posterior approximation. Proposition 3.1 shows the impact of the prediction bias on the KL divergence between the true class posterior probabilities, assumed modeled as $\mathbf{p} = \mathrm{softmax}(\mathbf{q}^*) \in \Delta_K$, and the estimated ones $\mathbf{s} = \mathrm{softmax}(f(\mathbf{x})) \in \Delta_K$. In particular, it states that

$$0 \leq \mathrm{KL}(\mathbf{p}||\mathbf{s}) \leq \boldsymbol{\varepsilon}_+^T \mathbf{p}, \tag{7}$$

where $\boldsymbol{\varepsilon}_+ = (\max_l\{\varepsilon_l\} - \varepsilon_k)_k \in \mathbb{R}_+^K$. The proof is given below.

Proof. We denote $\mathbf{q} = f(\mathbf{x}) \in \mathbb{R}^K$ the neural network's outputs on a given test sample $\mathbf{x}$. We first remark that

$$\begin{aligned}
\mathrm{KL}(\mathbf{p}||\mathbf{s}) &= \sum_k \mathbf{p}_k \ln\left(\frac{\mathbf{p}_k}{\mathbf{s}_k}\right) \\
&= \sum_k \mathbf{p}_k \ln\left(\frac{\exp(\mathbf{q}_k^*)}{\sum_{j=1}^K \exp(\mathbf{q}_j^*)} \cdot \frac{\sum_{j=1}^K \exp(\mathbf{q}_j^* + \varepsilon_j)}{\exp(\mathbf{q}_k^* + \varepsilon_k)}\right) \\
&= \sum_k \mathbf{p}_k \ln\left(\exp(-\varepsilon_k) \cdot \frac{\sum_{j=1}^K \exp(\mathbf{q}_j^* + \varepsilon_j)}{\sum_{j=1}^K \exp(\mathbf{q}_j^*)}\right).
\end{aligned} \tag{8}$$

To obtain the upper-bound, we notice that

$$\exp(\mathbf{q}_j^* + \varepsilon_j) = \exp(\mathbf{q}_j^*)\exp(\varepsilon_j) \leq \exp(\mathbf{q}_j^*) \cdot \max_l\{\exp(\varepsilon_l)\}.$$

This leads to

$$\frac{\sum_{j=1}^K \exp(\mathbf{q}_j^* + \varepsilon_j)}{\sum_{j=1}^K \exp(\mathbf{q}_j^*)} \leq \frac{\max_l\{\exp(\varepsilon_l)\}\sum_{j=1}^K \exp(\mathbf{q}_j^*)}{\sum_{j=1}^K \exp(\mathbf{q}_j^*)} = \max_l\{\exp(\varepsilon_l)\}. \tag{9}$$

Using the fact that all the terms are positive and that $\ln$ and $\exp$ are increasing functions, we obtain from Eq. (8) that

$$\begin{aligned}
\sum_k \mathbf{p}_k \ln\left(\exp(-\varepsilon_k) \cdot \frac{\sum_{j=1}^K \exp(\mathbf{q}_j^* + \varepsilon_j)}{\sum_{j=1}^K \exp(\mathbf{q}_j^*)}\right) &\leq \sum_k \mathbf{p}_k \ln(\exp(-\varepsilon_k) \cdot \max_l\{\exp(\varepsilon_l)\}) \\
&\leq \sum_k \mathbf{p}_k [\ln(\exp(\max_l\{\varepsilon_l\})) - \varepsilon_k] \\
&\leq \sum_k \mathbf{p}_k [\max_l\{\varepsilon_l\} - \varepsilon_k] \\
&= \boldsymbol{\varepsilon}_+^T \mathbf{p}.
\end{aligned} \tag{10}$$

Combining Eq. (8) and Eq. (10) gives the upper bound. □

The quantities $\boldsymbol{\varepsilon}_+$ is a linear transformation of the prediction bias $\boldsymbol{\varepsilon} \in \mathbb{R}^K$ and has nonnegative entries, which means each class is overestimated, representing an *overconfident* model. Proposition 3.1 shows that the discrepancy between the error approximation of the posterior probabilities is controlled by the alignment between the posterior and this extreme prediction bias. In addition, by a simple application of Cauchy-Schwartz in Eq. (7) and using the fact that $|\mathbf{p}\| = \sum_{k=1} \mathbf{p}_k^2 \leq \sum_{k=1} \mathbf{p}_k = 1$, we have $\mathrm{KL}(\mathbf{p}||\mathbf{s}) \leq \|\boldsymbol{\varepsilon}_+\|_2$. In particular, in the perfect situation where $\boldsymbol{\varepsilon} = \mathbf{0}$, $\boldsymbol{\varepsilon}_+$ is equal to 0 and the `softmax` probabilities perfectly approximate the posterior. In summary, Proposition 3.1 indicates that not only the norm of the prediction bias but also its alignment to the posterior is responsible for the approximation error of the posterior. In our setting, it means that logits-methods need a low prediction bias on classes on which the model is confident such that `softmax` probabilities can be reliably used to estimate accuracy. This follows our analysis and empirical verification from Section 4.

A real-world example. Although we usually tend to think that a high prediction bias shifts the predicted posterior towards the uniform distribution, in the general case, other situations may happen that hinder the quality of the accuracy estimation. For example, one may think of a letter recognition task with a neural network pre-trained on the Latin alphabet and tested on the Cyrillic one. In this case, some prediction probabilities will be adversarial as the neural network will not be aware of the semantic differences between the Latin "B" and the Cyrillic "B", therefore predicting a wrong class with high probability.

D.2 Distance to the Hyperplane

Lemma D.1 (Dohmatob (2020)). *Let $\boldsymbol{\omega} \in \mathbb{R}^n$ be non zero and $b \in \mathbb{R}$. The distance between any point $\mathbf{z} \in \mathbb{R}^n$ and the hyperplane $\{\mathbf{x} | \boldsymbol{\omega}^\top \mathbf{x} + b = 0\}$ writes $\mathrm{d}(\boldsymbol{\omega}, \mathbf{z}) = |\boldsymbol{\omega}^\top \mathbf{x} + b| / \|\boldsymbol{\omega}\|$.*

Proof. The proof follows the geometric intuition from Dohmatob (2020). We recall it here for the sake of self-consistency. The distance between $\mathbf{z}$ and the hyperplane $\mathcal{H} = \{\mathbf{x} \in \mathbb{R}^n | \boldsymbol{\omega}^\top \mathbf{x} + b = 0\}$ is equal to the distance between $\mathbf{z}$ and its orthogonal projection on $\mathcal{H}$. We consider the line $L = \{\mathbf{z} + t\boldsymbol{\omega} | t \in \mathbb{R}\}$ that is orthogonal to $\mathcal{H}$ and passes through $\mathbf{z}$. The desired orthogonal projection is simply the point $\mathbf{z} + t^*\boldsymbol{\omega}$ such that L and $\mathcal{H}$ intersects, *i.e.*, such that

$$\begin{aligned} \boldsymbol{\omega}^\top(\mathbf{z} + t^*\boldsymbol{\omega}) + b = 0 &\Leftrightarrow \boldsymbol{\omega}^\top \mathbf{z} + b = -t^* \|\boldsymbol{\omega}\|^2 \\ &\Leftrightarrow t^* = -\frac{\boldsymbol{\omega}^\top \mathbf{z} + b}{\|\boldsymbol{\omega}\|^2}. \qquad (\|\boldsymbol{\omega}\| \neq 0) \end{aligned}$$

It follows that the distance between $\mathbf{z}$ and $\mathcal{H}$ writes

$$\mathrm{d}(\boldsymbol{\omega}, \mathbf{z}) = \|\mathbf{z} - \mathbf{z} + t^*\boldsymbol{\omega}\| = \left\| -\frac{\boldsymbol{\omega}^\top \mathbf{z} + b}{\|\boldsymbol{\omega}\|^2} \times \boldsymbol{\omega} \right\| = \frac{|\boldsymbol{\omega}^\top \mathbf{z} + b|}{\|\boldsymbol{\omega}\|}.$$

□

D.3 Proof of Theorem 3.3

Proof. Reusing the notations introduced in Section 3.2 and Algorithm 1, we have that

$$\begin{aligned} \mathcal{S}(f, \mathcal{D}_{\text{test}})^p = \frac{1}{NK}\|\mathbf{Q}\|_p^p &= \frac{1}{NK}\sum_{i=1}^N \|\mathbf{Q}_i\|_p^p && \text{(Definition of } \|\mathbf{Q}\|_p\text{)} \\ &= \frac{1}{NK}\sum_{i=1}^N \|\sigma(\mathbf{q}_i)\|_p^p && \text{(Definition of } \mathbf{Q}_i \text{ in Algorithm 1)} \\ &= \frac{1}{NK}\sum_{i=1}^N 1 - (1 - \|\sigma(\mathbf{q}_i)\|_p^p) \\ &= \frac{1}{K} - \frac{p(p-1)}{NK}\sum_{i=1}^N \frac{1}{p(p-1)}(1 - \|\sigma(\mathbf{q}_i)\|_p^p) \\ &= \frac{1}{K} - \frac{p(p-1)}{NK}\sum_{i=1}^N \mathbf{H}_p^{\mathrm{T}}(\sigma(\mathbf{q}_i)) && \text{(Definition of } \mathbf{H}_p^{\mathrm{T}}\text{)} \\ &= b - a\left(\frac{1}{N}\sum_{i=1}^N \mathbf{H}_p^{\mathrm{T}}(\sigma(\mathbf{q}_i))\right), \end{aligned}$$

where $a = \frac{p(p-1)}{K} > 0$ and $b = \frac{1}{K}$. Rearranging the terms concludes the proof. □

D.4 Properties of ϕ

The `softmax` can be decomposed as $\mathrm{softmax}(\mathbf{q}) = \exp(\mathbf{q}) / \sum_{k=1}^K \exp(\mathbf{q})_k = (\phi \circ \exp)(\mathbf{q})$, where $\phi \colon \mathbb{R}_+^K \to \Delta_K$ writes $\phi(\mathbf{u}) = \mathbf{u} / \sum_{k=1}^K \mathbf{u}_k = \mathbf{u} / \|\mathbf{u}\|_1$. We extend the domain of ϕ to $\mathbb{R}_+^K$ by setting $\phi(\mathbf{0}) = \frac{1}{K}\mathbb{1}_K$. The following proposition states the properties of ϕ.

Proposition D.2 (Properties of ϕ).

1. ***Generalized injectivity.*** $\forall \mathbf{u}, \mathbf{v} \in \mathbb{R}_+^K \setminus \{\mathbf{0}\}, \quad \phi(\mathbf{u}) = \phi(\mathbf{v}) \iff \exists \alpha \in \mathbb{R}^*, \text{ s.t. } \mathbf{u} = \alpha\mathbf{v}$

2. ***Evaluation on constant inputs.*** *Let* $\mathbf{u} = \alpha \mathbb{1}_K$ *with* $\alpha \geq 0$*. Then, we have* $\phi(\mathbf{u}) = \frac{1}{K}\mathbb{1}_K$*.*

Proof. We start by proving the first part of Proposition D.2. Let $\mathbf{u}, \mathbf{v} \in \mathbb{R}_+^K \setminus \mathbf{0}$. We have

$$\phi(\mathbf{u}) = \phi(\mathbf{v}) \iff \frac{\mathbf{u}}{\|\mathbf{u}\|_1} = \frac{\mathbf{v}}{\|\mathbf{v}\|_1} \iff \mathbf{u} = \underbrace{\frac{\|\mathbf{u}\|_1}{\|\mathbf{v}\|_1}}_{\alpha > 0} \times \mathbf{v}.$$

Then, we prove the second part of the proposition. Let $\alpha \geq 0$ and consider $\mathbf{u} = \alpha \mathbb{1}_K \in \mathbb{R}_+^K$. If $\alpha = 0$, then $\mathbf{u} = \mathbf{0}$ and by definition, $\phi(\mathbf{u}) = \phi(\mathbf{0}) = \frac{1}{K}\mathbb{1}_K$. Assuming $\alpha > 0$, we have

$$\phi(\mathbf{u}) = \frac{\mathbf{u}}{\|\mathbf{u}\|_1} = \frac{\mathbf{u}}{\sum_{k=1}^K \mathbf{u}_k} = \frac{\alpha}{\sum_{k=1}^K \alpha} \times \mathbb{1}_K = \frac{1}{K}\mathbb{1}_K.$$

□

The first part of the proposition is dubbed "generalized injectivity" as the injectivity can be retrieved by fixing $\alpha = 1$ in Proposition D.2. It ensures that ϕ only has *equal* outputs if the inputs are *similar*. To illustrate that, consider the logits $\mathbf{q}, \boldsymbol{\delta} \in \mathbb{R}^K$. From Proposition D.2, having $\mathrm{softmax}(\mathbf{q}) = \mathrm{softmax}(\boldsymbol{\delta})$ is equivalent to having $\exp(\mathbf{q}) = \alpha \exp(\boldsymbol{\delta})$ for some $\alpha \neq 0$. By positivity of both sides, it implies $\alpha > 0$, and taking the logarithm leads to $\mathbf{q} = \boldsymbol{\delta} + \ln \alpha$. It means that $\mathbf{q}$ equals $\boldsymbol{\delta}$ up to a fixed constant. From a learning perspective, those logits will thus have the same predicted label and normalized logits. Proposition D.2 shows that using ϕ preserves the information from the neural network. In addition, if the neural network's output is not informative,*i.e.*, all entries are equal, then the link function gives equal probability to all classes.

E Theoretical Insights into Criterion $\Phi(\mathcal{D}_{\text{test}})$

In Section 4, we explained the main drawbacks of using the softmax normalization in the presence of the prediction bias proposing a new alternative normalization, `softrun`, that we recall is

$$v(\mathbf{q}_i) = \begin{cases} 1 + \mathbf{q}_i + \frac{\mathbf{q}_i^2}{2}, & \text{if } \Phi(\mathcal{D}_{\text{test}}) \leq \eta \quad \text{(Taylor)} \\ \exp(\mathbf{q}_i), & \text{if } \Phi(\mathcal{D}_{\text{test}}) > \eta \quad \text{(softmax)} \end{cases},$$

where $\Phi(\mathcal{D}_{\text{test}})$ is the selection criterion defined as

$$\Phi(\mathcal{D}_{\text{test}}) = -\frac{1}{NK} \sum_{j=1}^N \sum_{k=1}^K \log\left(\frac{\exp(\mathbf{q}_j)_k}{\sum_{j=1}^K \exp(\mathbf{q}_i)_j}\right).$$

In this section, we first would like to give more insights on the choice of $\Phi(\mathcal{D}_{\text{test}})$ and the selection rule. Then, we motivate our choice of the hyperparameter η. Finally, we discuss the potential limitations of our approach.

E.1 Choice of Criterion $\Phi(\mathcal{D}_{\text{test}})$

Intuition. We first provide some high-level intuition behind the selection criterion of Section 4. As we discussed before, the main idea of the logit-based approach is to rely on the model's confidence whose reliability depends on the prediction bias induced by possible distribution shift. Depending on exact values of confidence and bias, we can roughly distinguish the five following cases illustrated in Figure 8 and described as follows:

1. *High confidence, high bias.* The model is self-confident but practically makes a lot of mistakes. This corresponds to the case when assumptions are not met (Section 3.1), so logits are generally uninformative, and no normalization technique can really fix it. Thus, in practice, we have to make sure that the low-density separation (LDS) assumption holds, which is generally the case for well-calibrated models trained on diverse training sets. However, a user has to be careful when applying test-time adaptation methods (Chen et al., 2022; Rusak et al., 2022; Wang et al., 2021) to an original pre-trained model, since these approaches perform unsupervised confidence maximization making LDS not guaranteed anymore.

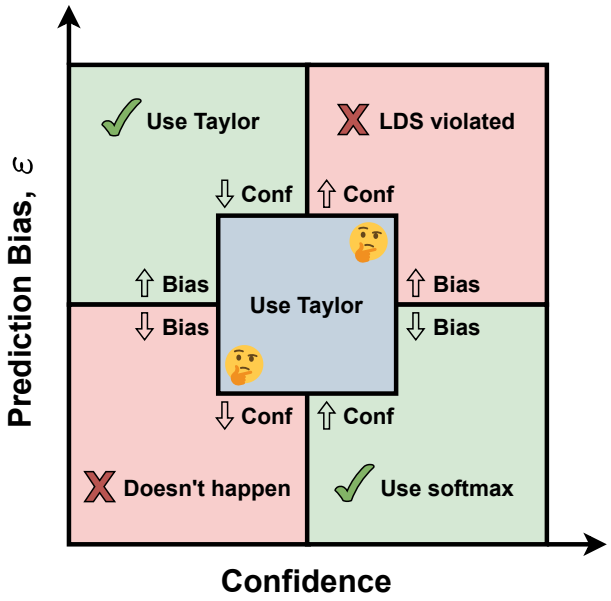


Figure 8: A schematic illustration of possible cases that may or may not happen and what normalization technique we use depending on the model's confidence and the value of the prediction bias. The two *unadmissible* scenarios correspond to the red sub-squares.

2. *Low confidence, low bias.* The model tends to output confidence that is low but overall precise. This situation is unlikely to happen mainly for the following two reasons. Firstly, the classification problem is poorly posed as the considered case implies that the true posterior probabilities are close to uniform ones. Secondly, it is known that deep neural networks tend to be overconfident in their predictions (Wei et al., 2022a), which we also observed in our experiments. Thus, we do not consider this case and leave it as a subject of future work.
3. *High confidence, low bias.* The model tends to be self-confident being overall precise. This is a favorable case as logits are very reliable, so we can use softmax normalization without being afraid to be too optimistic.
4. *Low confidence, high bias.* The model is not confident in their predictions and it indeed makes a lot of mistakes due to the high prediction bias. This is also a favorable scenario as low confidence correlates with low performance. In this case, we want posterior probabilities to be close to uniform, and we use the Taylor normalization due to its smoother behavior.
5. *Grey zone.* It corresponds to a mixed scenario when different examples may refer to different cases. As it is generally difficult to know what normalization technique would be the most relevant, we would opt for a more conservative solution in this situation. This is where the Taylor normalizer becomes useful as it does not exacerbate prediction bias to the same degree as the softmax (see Section 4.1 for more details).

Formulation of $\Phi(\mathcal{D}_{\text{test}})$. As we discussed in the main body of the paper, the criterion $\Phi(\mathcal{D}_{\text{test}})$ is equal, up to a constant, to the average KL divergence between the uniform distribution and the predicted `softmax` probabilities (Tian et al., 2021). This implies that the criterion reflects the model's confidence being high when predicted probabilities are far away from the uniform ones. As we rely on the LDS assumption, high values of $\Phi(\mathcal{D}_{\text{test}})$ correspond to the 3^{rd} case (*high confidence, low bias*), and the softmax normalization is selected. Conversely, when the $\Phi(\mathcal{D}_{\text{test}})$ is lower than the threshold η, we apply Taylor, which corresponds to either the 4^{th} case (*low confidence, high bias*) or the 5^{th} case (*grey zone*).

Connection between criterion and misclassification error. The next proposition provides further insights into our selection process. Let $\mathbf{u} = \frac{1}{K}\mathbb{1}_K \in \Delta_K$ be the uniform probability. The test dataset writes $\mathcal{D}_{\text{test}} = \{\mathbf{x}_i\}_{i=1}^N$ with corresponding logits $\mathbf{q}_i = f(\mathbf{x}_i)$ and ground-truth labels $\mathcal{Y}_{\text{test}} =$

$\{y_i\}_{i=1}^N$ (unavailable in practice). We denote the `softmax` probabilities by $\mathbf{s}^i = \mathrm{softmax}(\mathbf{q}_i) = \exp(\mathbf{q}_i)/\sum_{k=1}^K \exp(\mathbf{q}_i)_k \in \Delta_K$. We introduce the entropy of a probability vector as $\mathrm{H}(\mathbf{p}) = -\frac{1}{K}\sum_{k=1}^K \mathbf{p}_k \ln(\mathbf{p}_k)$. In particular, it is a measure of uncertainty and takes a high value when the model is uncertain, *i.e.*, outputs probabilities close to the uniform. We establish in the following proposition the connection between the criterion $\Phi(\mathcal{D}_{\text{test}})$, the miscalibration error, the model's confidence, and its entropy.

Proposition E.1 ($\Phi(\mathcal{D}_{\text{test}})$, misclassification error, confidence and entropy)**.** *We have*

$$\underbrace{\xi(\mathcal{D}_{\text{test}}, \mathcal{Y}_{\text{test}})}_{\textit{misclassification}} + \underbrace{\mathcal{U}(\mathcal{D}_{\text{test}})}_{\textit{confidence}} + \underbrace{\mathcal{H}(\mathcal{D}_{\text{test}})}_{\textit{entropy}} \leq \underbrace{\Phi(\mathcal{D}_{\text{test}})}_{\textit{criterion}} + \ln(e + \frac{1}{K}),$$

where $\xi(\mathcal{D}_{\text{test}}, \mathcal{Y}_{\text{test}}) = \frac{1}{N}\sum_{i=1}^N (1 - \mathbf{s}^i_{y_i})$, $\mathcal{U}(\mathcal{D}_{\text{test}}) = \frac{1}{N}\sum_{i=1}^N \mathrm{KL}(\mathbf{u} \parallel \mathbf{s}^i)$, *and* $\mathcal{H}(\mathcal{D}_{\text{test}}) = \frac{1}{N}\sum_{i=1}^N \mathrm{H}(\mathbf{s}^i)$.

Proof. We first present the following lemma that introduces the change of measure inequality (Banerjee, 2006; Seldin and Tishby, 2010).

Lemma E.2 (Change of measure inequality (Seldin and Tishby, 2010))**.** *Let Z be a random variable on $\{1, \ldots, K\}$ and $\boldsymbol{\mu} = (\mu_k)_k \in \Delta_K$ and $\boldsymbol{\nu} = (\nu_k)_k \in \Delta_K$ be two probability distributions. For any measurable function $\psi \colon \mathcal{Z} \to \mathbb{R}$, the following inequality holds:*

$$\sum_{k=1}^K \mu_k \psi(k) \leq \mathrm{KL}(\boldsymbol{\mu} \parallel \boldsymbol{\nu}) + \ln\left(\sum_{k=1}^K \nu_k \exp(\psi(k))\right).$$

Proof. We have

$$\begin{aligned}
\sum_{k=1}^K \mu_k \psi(k) &= \sum_{k=1}^K \mu_k \ln\left(\frac{\mu_k}{\nu_k}\exp(\psi(k))\frac{\nu_k}{\mu_k}\right) \\
&= \sum_{k=1}^K \mu_k \ln\left(\frac{\mu_k}{\nu_k}\right) + \sum_{k=1}^K \mu_k \ln\left(\exp(\psi(k))\frac{\nu_k}{\mu_k}\right) \\
&= \mathrm{KL}(\mu \parallel \nu) + \sum_{k=1}^K \mu_k \ln\left(\exp(\psi(k))\frac{\nu_k}{\mu_k}\right) && \text{(Definition of } \mathrm{KL}(\cdot \parallel \cdot)) \\
&\leq \mathrm{KL}(\mu \parallel \nu) + \ln\left(\sum_{k=1}^K \mu_k \exp(\psi(k))\frac{\nu_k}{\mu_k}\right) && \text{(Jensen inequality)} \\
&= \mathrm{KL}(\mu \parallel \nu) + \ln\left(\sum_{k=1}^K \nu_k \exp(\psi(k))\right).
\end{aligned}$$

□

We now proceed to the proof of Proposition (E.1). For a given test sample $\mathbf{x}_i \in \mathcal{D}_{\text{test}}$, we first notice that

$$\mathrm{KL}(\mathbf{u} \parallel \mathbf{s}^i) = \sum_{k=1}^K \mathbf{u}_k \ln\left(\frac{\mathbf{u}_k}{\mathbf{s}^i_k}\right) = \frac{1}{K}\sum_{k=1}^K \ln(\frac{1}{K}) - \ln(\mathbf{s}^i_k) = -\ln(K) - \frac{1}{K}\sum_{k=1}^K \ln(\mathbf{s}^i_k).$$

Similarly, we obtain $\mathrm{KL}(\mathbf{s}^i \parallel \mathbf{u}) = \sum_{k=1}^K \mathbf{s}_k^i \ln(\mathbf{s}_k^i) + \ln(K)$. Combining those results leads to

$$\mathrm{KL}(\mathbf{u} \parallel \mathbf{s}^i) + \mathrm{KL}(\mathbf{s}^i \parallel \mathbf{u}) = -\frac{1}{K}\sum_{k=1}^K \ln(\mathbf{s}_k^i) + \sum_{k=1}^K \mathbf{s}_k^i \ln(\mathbf{s}_k^i)$$

$$\iff \mathrm{KL}(\mathbf{s}^i \parallel \mathbf{u}) = -\mathrm{KL}(\mathbf{u} \parallel \mathbf{s}^i) - \frac{1}{K}\sum_{k=1}^K \ln(\mathbf{s}_k^i) + \sum_{k=1}^K \mathbf{s}_k^i \ln(\mathbf{s}_k^i).$$

Consider the function $\psi(k) = \mathbb{I}(y_i \neq k)$ that takes the value 1 when $y_i \neq k$ and 0 otherwise. Using Lemma E.2 with the measures $\boldsymbol{\mu} = \mathbf{s}^i, \boldsymbol{\nu} = \mathbf{u}$ and ψ, and the previous equation, we obtain

$$\sum_{k=1}^K \mathbf{s}_k^i \mathbb{I}(y_i \neq k) \leq \mathrm{KL}(\mathbf{s}^i \parallel \mathbf{u}) + \ln\left(\sum_{k=1}^K \mathbf{u}_k \exp(\mathbb{I}(y_i \neq k))\right)$$

$$\iff \sum_{k=1}^K \mathbf{s}_k^i \mathbb{I}(y_i \neq k) \leq -\mathrm{KL}(\mathbf{u} \parallel \mathbf{s}^i) - \frac{1}{K}\sum_{k=1}^K \ln(\mathbf{s}_k^i) + \sum_{k=1}^K \mathbf{s}_k^i \ln(\mathbf{s}_k^i) + \ln\left(\sum_{k=1}^K \mathbf{u}_k \exp(\mathbb{I}(y_i \neq k))\right)$$

$$\iff 1 - \mathbf{s}_{y_i}^i \leq -\mathrm{KL}(\mathbf{u} \parallel \mathbf{s}^i) - \frac{1}{K}\sum_{k=1}^K \ln(\mathbf{s}_k^i) - \mathrm{H}(\mathbf{s}^i) + \ln\left(\frac{1}{K}(Ke+1)\right) \qquad (\textstyle\sum_{k=1}^K \mathbf{s}_k^i = 1)$$

$$\iff 1 - \mathbf{s}_{y_i}^i + \mathrm{KL}(\mathbf{u} \parallel \mathbf{s}^i) + \mathrm{H}(\mathbf{s}^i) \leq -\frac{1}{K}\sum_{k=1}^K \ln(\mathbf{s}_k^i) + \ln\left(e + \frac{1}{K}\right).$$

Summing over all the test samples and dividing by N leads to

$$\frac{1}{N}\sum_{i=1}^N (1 - \mathbf{s}_{y_i}^i) + \frac{1}{N}\sum_{i=1}^N \mathrm{KL}(\mathbf{u} \parallel \mathbf{s}^i) + \frac{1}{N}\sum_{i=1}^N \mathrm{H}(\mathbf{s}^i) \leq \underbrace{-\frac{1}{NK}\sum_{i=1}^N\sum_{k=1}^K \ln(\mathbf{s}_k^i)}_{=\Phi(\mathcal{D}_{\text{test}})} + \ln\left(e + \frac{1}{K}\right),$$

which concludes the proof by using the notations introduced in Proposition E.1. □

Interpretation. The term $\xi(\mathcal{D}_{\text{test}}, \mathcal{Y}_{\text{test}})$, dubbed misclassification error, is the average error on the test set between the optimal probability on the true label (*i.e.*, 1) and the predicted probability $\mathbf{s}_y^i$. It takes high values when the model makes a lot of mistakes, assigning low confidence to the true class labels, and low values otherwise. $\mathcal{U}(\mathcal{D}_{\text{test}})$ is the average KL divergence on the test set between the predicted probabilities and the uniform distribution and it measures the model's confidence (Tian et al., 2021). It takes high values when the predicted probabilities are far from the uniform (confidence) and low values when they are close to the uniform (uncertain). $\mathcal{H}(\mathcal{D}_{\text{test}})$ is the average entropy on the test set of the predicted probabilities. It takes high values when predicted probabilities are close to the uniform and low values otherwise. Proposition E.1 implies that when the model makes few mistakes ($\xi(\mathcal{D}_{\text{test}}, \mathcal{Y}_{\text{test}})$ is low) and is confident ($\mathcal{U}(\mathcal{D}_{\text{test}})$ is high and $\mathcal{H}(\mathcal{D}_{\text{test}})$ is low), then the criterion $\Phi(\mathcal{D}_{\text{test}})$ takes high values. This matches the empirical evidence from from Tian et al. (2021). Proposition E.1 is harder to analyze in other scenarios, *i.e*, when the misclassification error, the confidence, or the entropy behaves differently, mostly because of the interplay between $\mathcal{U}(\mathcal{D}_{\text{test}})$ and $\mathcal{H}(\mathcal{D}_{\text{test}})$. However, we experimentally show the benefits of $\Phi(\mathcal{D}_{\text{test}})$ and `softrun` in Section 5 where MANO achieves superior performance against 11 commonly used baselines for various architectures and types of shifts on 12 datasets.

E.2 Choice of hyperparameter η

In all our experiments, we take $\eta = 5$ for the selection criterion in Eq. (6). We motivate this choice in what follows. In our setting, we consider test samples $\mathbf{x}_i \in \mathcal{D}_{\text{test}}$ drawn i.i.d. from the test distribution p_T. As the model f pre-trained on $\mathcal{D}_{\text{train}}$ is a deterministic function, the logits $\mathbf{q}_i$ are i.i.d. random variables and the decision threshold $\Phi(\mathcal{D}_{\text{test}}) = -\frac{1}{NK}\sum_{i=1}^N\sum_{k=1}^K \ln\Big(\exp(\mathbf{q}_i)_k / \sum_{j=1}^K \exp(\mathbf{q}_i)_j\Big)$ is a random variable with mean μ and variance ν. Applying the Chebyshev's inequality leads to

$$\mathbb{P}(|\Phi(\mathcal{D}_{\text{test}}) - \mu| > \nu\eta) \leq \frac{1}{\eta^2}. \tag{11}$$

The threshold $\Phi(\mathcal{D}_{\text{test}})$ is used to determine how calibrated the model is on a given test dataset $\mathcal{D}_{\text{test}}$. Figure 2(b) shows that our proposed normalization is optimal in poorly calibrated datasets and performs slightly below the `softmax` in calibrated situations. Hence, we can afford to be conservative and we want to consider the model calibrated only for *extreme* values of $\Phi(\mathcal{D}_{\text{test}})$. From Eq. (11), taking $\eta = 5$ ensures that the probability that $\Phi(\mathcal{D}_{\text{test}})$ deviates from its mean by several standard deviations with probability smaller than 5% ($\frac{1}{25} < 0.05$). It should be noted that we do not claim the optimality of this choice nor the optimality of our automatic selection in Eq. (6). However, it is particularly difficult to define decision rules in unsupervised and semi-supervised settings (Amini et al., 2022). Moreover, using Eq. (6), MANO remains suitable even when test labels are not available which is often the case in real-world applications, and we demonstrate state-of-the-art performance for various architecture and types of shifts in Section 5. For the sake of self-consistency, we also provide a sensitivity analysis on the values of η in Appendix G.3.

E.3 Potential Limitations

It should be noted that the selection criterion of Eq. (6) remains somehow heuristic and might depend on the model, the data, or the threshold η. As stated above, the chosen value of η is motivated by a probabilistic argument and by our experiments. However, as it can be seen in Eq. (11), the mean and standard deviation of $\Phi(\mathcal{D}_{\text{test}})$ can impact the validity of η. In particular, this could be the case when applying MANO on other data modalities than images or in other learning settings (*e.g.*, classification with a huge number of classes, regression tasks, auto-regressive settings). We believe this is the subject of future work to improve the robustness and versatility of our method.

Impact of the number of classes. As a first research direction, we provide a motivating example with synthetic data on the impact of the number of classes K on the values of $\Phi(\mathcal{D}_{\text{test}})$. We uniformly draw random vectors of $\mathbb{R}^K$ in $[-5, 5]$ to mimic the logits obtained from 100000 independent models. We compute the corresponding $\Phi(\mathcal{D}_{\text{test}})$ for each model and recover the 0.5^{th} and 99.5^{th} percentile to obtain a 99% confidence interval. We repeat this experiment for $K \in [\![2, 100]\!]$. The evolution of the confidence interval is displayed in Figure 9. We observe that as soon as $K > 3$, the upper bound of the confidence interval has a very slow increase. However, the lower bound increases quickly in the beginning until $K \sim 25$ and then adopts the same increase pace as the upper bound. In summary, the range of values of $\Phi(\mathcal{D}_{\text{test}})$ becomes thinner and more concentrated on high values for $K > 25$. In particular, we observe that 99% of the models have an associated $\Phi(\mathcal{D}_{\text{test}})$, higher than $\eta = 5$, which means that in this situation, the `softmax` would always be selected as a normalization σ. While the conclusions from this experiment are not directly applicable to our real experimental setting (in particular, Taylor and `softmax` cases of Eq. (6) occur both for datasets with $K > 25$ and $K \leq 25$), we believe it motivates further work to make η more robust to the values of $\Phi(\mathcal{D}_{\text{test}})$. In particular, one could propose to compute η based on statistics of the data or as a function of the number of classes.

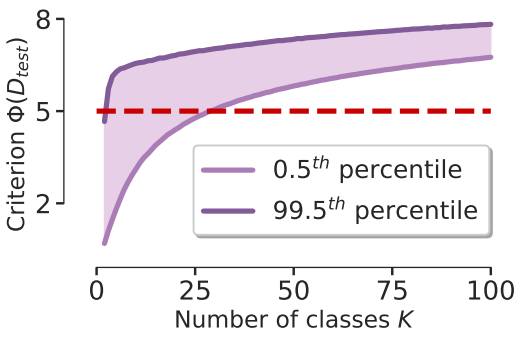


Figure 9: Evolution of the 99% confidence interval on $\Phi(\mathcal{D}_{\text{test}})$ with the number of classes K. As K increases, $\Phi(\mathcal{D}_{\text{test}})$ will likely be higher than $\eta = 5$.

Dispersion of the softmax. In the previous paragraph, we showed that the number of classes K can have an impact on the values taken by criterion $\Phi(\mathcal{D}_{\text{test}})$. As the criterion relies on the softmax probabilities of the model on test data, it is natural to investigate the impact of K on the `softmax` function. The lemma below shows that the `softmax` must disperse on all entries as the number of classes K increases. It means that the total weights on the `softmax` entries (equal to 1) cannot be concentrated on a few entries as the number of classes K increases.

Lemma E.3. *Let $\boldsymbol{\theta} \in \mathbb{R}^K$ be logits of a neural network f such that $\|\boldsymbol{\theta}\|_1 \leq c$ for some $c > 0$. Then, as the number of classes grows, i.e., $K \to \infty$, we have*

$$\operatorname{softmax}(\boldsymbol{\theta}) = \mathcal{O}\left(\frac{1}{K}\right),$$

where the equality holds at the component level.

Proof. Following the proof of Zekri et al. (2024, Lemma D.7), we can show for all $i \in [K]$ that

$$\frac{\exp(-c)}{\sum_{j=1}^{K} \exp(c)} \leq \frac{\exp(\boldsymbol{\theta}_i)}{\sum_{j=1}^{m} \exp(\boldsymbol{\theta}_j)} \leq \frac{\exp(c)}{\sum_{j=1}^{K} \exp(-c)} \iff \frac{a}{K} \leq \operatorname{softmax}(\boldsymbol{\theta})_i \leq \frac{b}{K},$$

where $a = \exp(-2c), b = \exp(2c)$ are constant. This concludes the proof. □

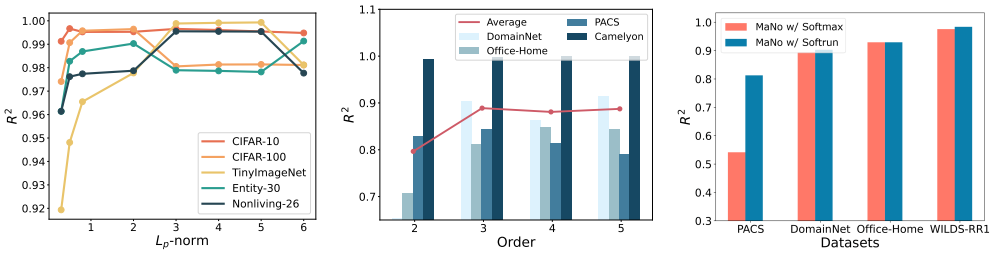
Lemma E.3 implies that, as the number of classes grows, the highest value an individual entry can have decreases. Hence, the number of classes impacts the distribution of the weights among the `softmax` entries (recalling that it must sum at 1 as it is a probability vector). As by definition, $\Phi(\mathcal{D}_{\text{test}})$ depends on the `softmax` probability distributions, this will impact its value. We believe that empirically and theoretically studying this phenomenon could be insightful in deriving more robust selection criteria and threshold values. We note that Lemma E.3 is similar but more general than Veličković et al. (2024, Lemma 2.1) as our global bounding condition on $\boldsymbol{\theta}$ encompasses their entry-wise condition.

F Beyond ResNets: Experiments with Vision Transformers and ConvNeXts

To evaluate the efficiency of MANO across diverse model architectures, we conducted additional experiments with the Vision Transformer (Dosovitskiy, 2020, ViT) and the ConvNeXt (Liu et al., 2022) architectures. The numerical results are gathered in Table 5. Two methods stand out from the rest of the baselines: *ConfScore* and MANO. In particular, *ConfScore* is particularly strong with the ConvNeXt architecture while MANO is better with the Vision Transformer. Overall, MANO leads to a better accuracy estimation on average.

Table 5: Method comparison using Vision Transformer (ViT) and ConvNext under **synthetic, subpopulation and natural shifts** with R^2 and ρ metrics (the higher the better). The best results for each metric are in **bold**. Overall, MANO surpasses its competitors while *ConfScore* appears to be stronger with ViT and ConvNext than with ResNets.

Dataset	Network	ConfScore		Entropy		ATC		MDE		COT		Nuclear		MANO	
		R^2	ρ	R^2	ρ	R^2	ρ	R^2	ρ	R^2	ρ	R^2	ρ	R^2	ρ
CIFAR 10	ViT	0.985	0.996	0.980	0.996	0.991	0.997	0.871	0.873	0.950	**0.997**	0.937	0.988	**0.991**	0.996
	ConvNeXt	0.936	0.996	0.924	0.995	0.911	0.994	0.002	0.534	0.978	0.994	**0.991**	0.994	0.916	**0.995**
	Average	0.961	0.996	0.952	0.996	0.951	0.996	0.435	0.704	**0.964**	0.996	0.964	0.991	0.954	**0.996**
CIFAR 100	ViT	0.983	**0.997**	0.981	0.995	0.987	0.995	0.974	0.983	**0.993**	0.996	0.977	0.995	0.989	0.996
	ConvNeXt	0.976	**0.995**	0.957	0.992	0.976	0.993	0.617	0.399	0.981	0.996	**0.982**	0.994	0.954	0.994
	Average	0.961	**0.996**	0.952	**0.996**	0.982	0.994	0.795	0.691	**0.987**	**0.996**	0.977	0.995	0.971	0.995
PACS	ViT	0.711	0.783	0.631	0.727	0.426	0.503	0.180	0.209	0.742	0.797	**0.823**	**0.860**	0.705	0.755
	ConvNeXt	**0.900**	**0.895**	0.872	0.853	0.727	0.580	0.004	0.062	0.814	0.748	0.834	0.790	0.874	0.755
	Average	0.806	**0.839**	0.752	0.790	0.577	0.541	0.092	0.073	0.778	0.772	**0.829**	0.825	0.789	0.755
Office-Home	ViT	0.947	0.958	0.928	0.979	0.896	0.902	0.217	0.755	0.642	0.856	0.861	0.958	**0.953**	**0.979**
	ConvNeXt	**0.784**	0.860	0.649	0.825	0.769	**0.923**	0.040	0.475	0.642	0.856	0.514	0.514	0.733	0.818
	Average	**0.865**	0.909	0.788	0.902	0.832	**0.912**	0.128	0.615	0.642	0.856	0.687	0.736	0.843	0.898
Entity-13	ViT	0.930	0.950	0.925	0.950	0.950	0.971	0.816	0.884	0.923	0.958	0.873	0.882	**0.958**	**0.971**
	ConvNeXt	**0.943**	**0.970**	0.931	0.960	0.901	0.902	0.868	0.805	0.937	0.938	0.941	0.942	0.930	0.963
	Average	0.937	0.960	0.928	0.955	0.926	0.937	0.842	0.844	0.930	0.948	0.907	0.912	**0.944**	**0.967**
Entity-30	ViT	0.950	0.972	0.937	0.968	0.948	0.970	0.819	0.908	0.939	0.971	0.905	0.927	**0.959**	**0.975**
	ConvNeXt	**0.968**	0.988	0.955	0.981	0.916	0.936	0.959	0.961	0.942	0.976	0.942	0.959	0.960	**0.990**
	Average	0.959	0.980	0.946	0.975	0.932	0.953	0.889	0.934	0.941	0.973	0.924	0.943	**0.960**	**0.982**



(a) Impact of the L_p norm (b) Approximation order n (c) Benefits of `softrun`

Figure 10: **Sensitivity analysis with Resnet18. (a)** Effect of the L_p norm types. **(b)** Impact of the Taylor approximation order, *i.e.*, the number of terms in Eq 4. For instance, an order of 3 means that 3 terms are taken, which corresponds to the default setting in Eq. (6) and is used in all our experiments. **(c)** Type of normalization function.

G Sensitivity Analysis and Ablation Study

G.1 Choice of L_p Norm

To reflect the impact of different L_p norms on estimation performance, we conduct a sensitivity study on 5 datasets with ResNet18, whose results are shown in Figure 10(a). The performance for $p = 1$ is ignored as in this case, $\|\mathbf{Q}\|_1 = 1$ because $\mathbf{Q}$ is right-stochastic. We can see that when we choose $p \in [2, 5]$, the results fluctuate within a satisfying range. This can be explained by the fact that within this range, we emphasize adequately the large positive feature-to-boundary distances without ignoring the other comparatively small distances.

G.2 Choice of Taylor Approximation Order

In Figure 10(b), we verify the impact of Taylor formula approximation on final accuracy estimation performance. It should be noted that for orders higher than in the default setting in Eq. (6), the positivity is lost. To alleviate this issue, we consistently remove for all orders the minimum value of the obtained vector to each of its entries to ensure having an output in $\mathbb{R}_+^K$. This extends de Brébisson and Vincent (2016) to orders higher than 2. From this figure, we can see that when we reserve the first three terms in the Taylor formula, the average estimation performance is optimal. For well-calibrated datasets such as Office-Home and WILDS, there exists an increased trend of estimation performance when we reserve more Taylor formula terms. As for suboptimal-calibrated datasets such as PACS and Office-31, their performance rises when fewer terms are reserved. It empirically certifies that the normalization technique is a trade-off tool between the ground-truth logits' information and error accumulation. In addition, the optimal choice is to keep 3 terms in Eq. (4) which motivates our default setting in Eq (6).

G.3 Choice of Calibration Threshold η

In Table 6, we display the performance comparison for varying values of threshold η on three datasets with ResNet18. It should be noted that taking $\eta = 0$ corresponds to the case where the `softmax` is always taken, i.e., the common choice in the literature. This matches our theoretical insights in Appendix E.2 and confirms that taking $\eta = 5$ is a robust and effective choice for `softrun`.

G.4 Superiority of `softrun`

To verify the effectiveness of our proposed normalization technique, `softrun`, we conduct an ablation study by replacing our normalization with the softmax function under the natural shift. In Figure 10(c), we observe that our proposed normalization significantly enhances the estimation performance of datasets from the natural shift. Especially, R^2 for poorly-calibrated datasets such as PACS is improved from 0.541 to 0.812.

Table 6: Performance comparison for varying $\eta \in \{0, 1, 3, 5, 7, 9\}$ on CIFAR-10, Office-Home, and PACS with ResNet18. The metric used is R^2 (the higher the better). The best results are in **bold**. The results motivate our choice of $\eta = 5$.

Dataset	$\eta = 0$	$\eta = 1$	$\eta = 3$	$\eta = 5$	$\eta = 7$	$\eta = 9$
Cifar-10	**0.995**	**0.995**	**0.995**	**0.995**	**0.995**	**0.995**
Office-Home	**0.926**	**0.926**	**0.926**	**0.926**	0.777	0.777
PACS	0.541	0.541	0.541	**0.827**	**0.827**	**0.827**
Average	0.820	0.820	0.820	**0.916**	0.866	0.866

G.5 Generalization Capabilities of MANO on ImageNet-$\bar{C}$

To further demonstrate the generalization capability of MANO, we provide a similar experiment with that in Section 5.4 on ImageNet-C and ImageNet-$\bar{C}$ (Mintun et al., 2021) in Figure 11. In particular, we fit a linear regression function on ImageNet-C and use the linear function to predict the accuracy of ImageNet-$\bar{C}$. This figure shows that MANO has better estimation performance than the other baselines when meeting different corruption types.

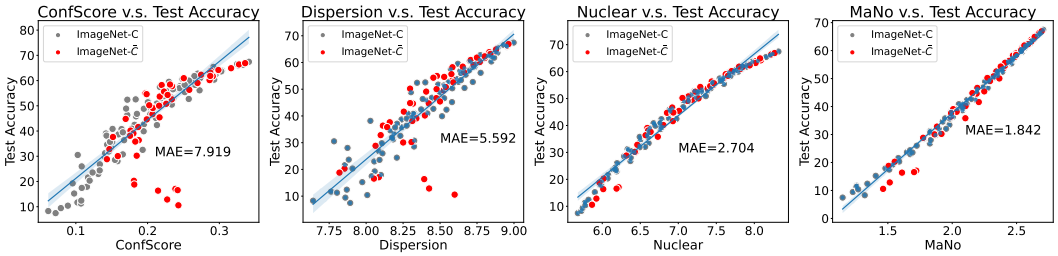


Figure 11: Comparison of generalization capability across four methods. Each subplot shows a linear regression model fitted on ImageNet-C to predict accuracy on ImageNet-$\bar{C}$. Mean absolute error (MAE) is calculated on ImageNet-$\bar{C}$ (Mintun et al., 2021). All experiments use ResNet18.

H How to Use MANO in Real-World Applications?

This work demonstrates the strong correlation between ground-truth OOD accuracy and the designed score, which can be particularly useful for model deployment applications. In this section, we provide two examples.

- **Find difficult (under-performed) test set.** In cases such as retraining on under-performed datasets or annotating hard datasets, we only need to know the rank of datasets by accuracy. Therefore, we can calculate the proposed score for each dataset directly and fulfill the task based on this score's ranking.
- **Deployement risk estimation.** When deploying the model into production, it is important to estimate its safety. If the cost of getting test labels is prohibitive, our method can help to estimate the model's accuracy on the product's test data. A practitioner can additionally look at the variability of the score on multiple test sets. When multiple datasets are not available, we can alternatively construct adequate synthetic datasets via various visual transformations.

In Table 7, we provide an example, using 90% of datasets to train a linear regression model and estimating the test accuracy of the rest 10% of datasets via the trained linear regression model. The results are measured by Mean Absolute Error (MAE). From this table, we observe the superiority of MANO for application in the real world.

Table 7: Method comparison on four benchmarks using ResNet18, ResNet50, and WRN-50-2 under **natural shift** with the MAE metric (the lower the better). The best results are highlighted in **bold**. MANO provides the best accuracy estimation overall.

Dataset	Network	ConfScore	Entropy	ATC	Fréchet	Dispersion	MDE	COT	Nuclear	MANO
CIFAR-10	ResNet18	5.131	5.265	3.968	1.964	3.842	1.763	0.952	1.357	**0.394**
	ResNet50	1.945	1.891	1.706	2.477	3.842	0.928	0.670	1.024	**0.450**
	WRN-50-2	2.956	3.160	3.086	3.547	2.671	0.846	**0.355**	0.843	0.406
	Average	3.344	3.439	2.920	2.662	2.539	1.179	0.659	1.075	**0.417**
CIFAR-100	ResNet18	2.944	5.265	3.968	1.964	3.846	1.763	0.952	1.357	**0.394**
	ResNet50	2.128	1.891	1.706	2.477	2.671	0.928	0.670	1.024	**0.450**
	WRN-50-2	1.323	3.160	3.086	3.547	1.102	0.846	**0.355**	0.843	0.406
	Average	3.344	3.439	2.920	2.662	2.539	1.179	0.659	1.075	**0.417**
TinyImageNet	ResNet18	3.822	3.559	3.752	5.998	2.822	1.926	1.297	1.165	**0.612**
	ResNet50	3.376	3.696	3.435	5.616	1.667	2.687	1.653	1.226	**1.005**
	WRN-50-2	2.712	2.854	5.011	4.862	1.054	1.703	1.286	**1.053**	1.145
	Average	3.303	3.370	4.066	5.492	1.848	2.106	1.412	1.148	**0.921**
ImageNet	ResNet18	3.616	3.812	2.503	3.750	2.602	4.818	0.705	2.679	**1.057**
	ResNet50	3.325	3.357	2.911	3.242	7.318	6.346	1.796	3.203	**1.388**
	WRN-50-2	2.990	4.132	3.388	5.210	10.050	6.755	2.564	4.091	**0.695**
	Average	3.310	3.767	2.934	4.067	6.656	5.973	1.688	3.325	**1.047**
Office-Home	ResNet18	1.987	2.878	10.014	1.404	7.976	8.228	**0.602**	0.885	0.880
	ResNet50	0.855	1.874	7.509	1.254	7.285	9.374	**0.730**	2.176	3.995
	WRN-50-2	2.607	3.933	11.272	0.924	7.591	7.974	**2.382**	2.764	4.330
	Average	1.816	2.895	9.598	1.947	7.617	8.525	**1.238**	1.941	3.068
DomainNet	ResNet18	**3.375**	7.067	10.875	7.416	8.200	7.066	5.954	7.313	3.830
	ResNet50	4.778	7.742	10.039	3.533	8.949	9.917	5.005	7.407	**4.230**
	WRN-50-2	**5.513**	6.310	3.513	10.001	8.695	9.230	4.605	6.953	5.827
	Average	**4.555**	7.039	8.142	6.983	8.615	8.738	5.188	7.224	4.629
Entity-13	ResNet18	7.448	7.416	7.391	5.544	2.343	3.798	2.205	2.182	**0.790**
	ResNet50	5.183	5.367	6.155	2.895	6.696	4.140	4.132	2.272	**0.969**
	WRN-50-2	2.748	2.893	3.128	1.817	5.431	3.159	2.276	1.297	**0.674**
	Average	5.126	5.225	5.558	3.419	4.824	3.699	2.871	1.917	**0.811**
Entity-30	ResNet18	5.060	5.544	5.731	4.098	3.768	3.665	1.867	2.269	**0.830**
	ResNet50	4.415	5.14	4.630	4.499	7.185	4.326	4.767	2.158	**1.455**
	WRN-50-2	3.477	4.200	3.363	2.490	4.17	4.080	2.103	1.797	**0.918**
	Average	4.317	4.961	4.575	3.695	5.042	4.024	2.912	2.075	**1.068**
living-17	ResNet18	4.095	4.098	3.699	3.767	4.262	3.737	1.373	2.569	**1.975**
	ResNet50	2.802	2.757	1.574	9.687	4.361	3.535	3.260	1.860	**1.573**
	WRN-50-2	4.059	4.250	2.535	3.889	4.925	3.833	**2.922**	2.974	3.281
	Average	3.652	3.701	2.603	5.781	4.516	3.641	2.519	2.467	**2.277**
Nonliving-26	ResNet18	2.907	3.767	3.386	1.891	2.773	3.453	1.881	2.168	**1.010**
	ResNet50	4.004	4.663	4.685	3.547	6.461	4.236	3.955	**2.087**	2.189
	WRN-50-2	1.903	2.444	2.593	2.259	4.220	3.781	2.785	2.094	**1.403**
	Average	2.938	3.625	3.555	2.565	4.485	3.823	2.874	2.116	**1.534**

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification:

   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: [Yes]

   Justification:

   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: [Yes]

Justification:

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: [Yes]

   Justification:

   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: [Yes]

Justification: All datasets used are open-source and all the implementation details are given to reproduce the experimental results. The pseudo-code and all the implementation details are given and an extensive ablation study was conducted. Only the code is not provided.

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: [Yes]

   Justification:

   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: [NA]

   Justification: We use coefficients of determination and Spearman coefficient on test data, hence no random seed is needed, and error bars are not applicable.

   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: [No]

   Justification: All details to reproduce the experiments are given and our proposed method is training-free once a pre-trained model is available.

   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: [Yes]

   Justification:

   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: [NA]

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

    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: [NA]

    Justification:

    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: [Yes]

    Justification: All creators and owners are properly credited in the paper and code.

    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: [NA]

    Justification:

    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: [NA]

    Justification:

    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: [NA]

    Justification:

    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.