Invertible Logits Transformation for Accuracy-Preserving Post-Hoc Uncertainty Calibration

Lening Zhao[*]
Department of Engineering and Applied Science
University of Pennsylvania
Philadelphia, PA
lnzhao@engineering.upenn.edu

Qipeng Zhan[*†]
School of Arts and Sciences
University of Pennsylvania
Philadelphia, PA
qipengz@sas.upenn.edu

Li Shen[†]
Perelman School of Medicine
University of Pennsylvania
Philadelphia, PA
li.shen@pennmedicine.upenn.edu

Abstract

Post-hoc calibration aligns a classifier's predicted confidences with its empirical accuracy without retraining. An ideal calibrator should correct nonlinear miscalibration, scale gracefully to large label spaces, and preserve the original predictions; existing methods typically violate at least one of these properties—temperature scaling lacks expressivity, more flexible parametric alternatives introduce parameters that grow with the number of classes C, and other expressive methods do not preserve the rank ordering of class scores and may alter the predicted class. We propose **Invertible Logits Transformation (InvLT)**, which applies a learned scalar MLP $f : \mathbb{R} \to \mathbb{R}$ element-wise to the pre-softmax logits. Sharing f across all logit dimensions makes the parameter count independent of C. Monotonicity of f—and hence preservation of the argmax prediction—is softly encouraged via a paired inverse network rather than enforced through the numerical integration required by prior monotone calibrators; this avoids their computational overhead while empirically preserving the original classification accuracy in every setting we evaluate. Across standard image classification benchmarks and a range of architectures, InvLT consistently outperforms a broad set of post-hoc baselines on standard calibration metrics. Our code is available at https://github.com/joannaLzhao/InvLT.

1 Introduction

Calibration is a fundamental requirement for trustworthy prediction: a classifier is calibrated if its predicted confidence matches its empirical accuracy [Naeini et al., 2015, Guo et al., 2017]. Calibration matters wherever downstream decisions depend on probabilistic predictions—medical diagnosis, autonomous driving, decision support, and selective prediction—yet modern deep networks are typically over-confident, even assigning high probability to incorrect predictions [Guo et al., 2017, Minderer et al., 2021]. *Post-hoc calibration* addresses this by learning a transformation of a fixed model's outputs on a held-out calibration set, leaving the trained weights untouched.

[*]Equal contribution.
[†]Corresponding authors.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

The most widely adopted post-hoc method is temperature scaling (TS) [Guo et al., 2017], which divides all logits by a single learned scalar $T > 0$. Because scalar division preserves the ordering of logit values, TS guarantees that the predicted class is unchanged—a property we call *accuracy preservation*. However, TS has only one degree of freedom: it applies the same linear transformation $f(x) = x/T$ to every logit of every sample, and therefore cannot correct miscalibration that varies across confidence levels.

Two complementary axes of expressivity. TS admits two natural accuracy-preserving extensions, distinguished by what the transformation depends on:

- **Sample-specific.** Make the temperature depend on the input. Parameterized Temperature Scaling (PTS) [Tomani et al., 2022] predicts a per-sample $T(\mathbf{z})$ via a network and applies

  $$\phi_{\mathrm{PTS}}(\mathbf{z})_c = z_c / T(\mathbf{z}).$$

  All logits of a given sample are still divided by the same positive scalar, so the within-sample ordering—and the argmax—is preserved.
- **Logit-specific.** Make the transformation depend on the value of each logit. Replace the per-sample temperature with a shared scalar function f applied logit-wise:

  $$\phi_{\mathrm{InvLT}}(\mathbf{z})_c = f(z_c).$$

  If f is strictly increasing, the ordering of logit values is preserved for every input, and so is the argmax.

These directions are complementary: PTS reshapes calibration across samples while keeping the per-sample logit map linear; the logit-specific approach reshapes the logit scale nonlinearly while sharing the same map across samples and classes.

The logit-specific direction has been explored through intra-order-preserving calibration functions [Rahimi et al., 2020], whose diagonal sub-family learns a strictly increasing scalar function using the UMNN architecture [Wehenkel and Louppe, 2019]. UMNN parameterizes f' as a positive network and recovers f by numerical quadrature at every forward pass—a procedure repeated during both training and inference, and one that constrains the architecture and tooling around f (e.g., choice of activations, integration solvers, gradient computation). Our central methodological contribution is to replace this hard architectural constraint with a soft regularizer rooted in a classical result: *a continuous injective function on an interval is strictly monotone.* We therefore train f jointly with an auxiliary inverse network g, minimizing the reconstruction error $\|g(f(u_k)) - u_k\|^2$ over reference points $\{u_k\}_{k=1}^{K}$ drawn from the operating range of the calibration logits. This approach—well-established in image translation but, to our knowledge, novel in the post-hoc calibration literature—drives f toward injectivity, and hence toward monotonicity, without imposing any architectural restriction. This reframing has three practical advantages. First, training avoids numerical integration: each step adds only K scalar evaluations of $g \circ f$ on a precomputed grid. Second, inference reduces to a small MLP applied element-wise, comparable in cost to TS. Third, f can be any standard MLP with any activation function, simplifying implementation and tuning. In general, our contributions are as follows:

1. We introduce InvLT, a post-hoc calibrator that applies a shared scalar MLP element-wise to logits. The number of learnable parameters is independent of the number of classes C.
2. We propose a reconstruction regularizer via an inverse network that softly enforces monotonicity, avoiding the numerical integration required by prior monotone calibration methods [Wehenkel and Louppe, 2019, Rahimi et al., 2020].
3. Across CIFAR-10, CIFAR-100, and ImageNet with seven architectures, InvLT consistently outperforms all baselines while being substantially faster to train than UMNN.

2 Related Work

Post-hoc calibration methods can be broadly organized along two axes: the space in which the transformation operates (logit space vs. probability space) and the expressiveness of the parameterization (single-parameter vs. multi-parameter or nonparametric).

Temperature and affine scaling. Temperature scaling [Guo et al., 2017] is the simplest and most widely used post-hoc method. It learns a single scalar $T > 0$ and replaces the logits $\mathbf{z}$ with $\mathbf{z}/T$. Despite having only one parameter, it is effective when the miscalibration is approximately uniform across confidence levels. Vector scaling extends this by learning a per-class scaling factor and bias, while matrix scaling further introduces cross-class interactions via a full weight matrix [Guo et al., 2017]. However, matrix scaling requires $O(C^2)$ parameters and is prone to overfitting when the validation set is small relative to the number of classes.

Ensemble and parameterized temperature scaling. Ensemble Temperature Scaling (ETS) [Zhang et al., 2020] learns a convex combination of temperature scaling, no calibration, and a uniform baseline, providing modest improvements with minimal additional complexity. Parameterized Temperature Scaling (PTS) [Tomani et al., 2022] conditions the temperature on the input logits via a small neural network, allowing instance-dependent recalibration while preserving the argmax prediction. PTS is conceptually related to InvLT in that both use neural networks for calibration, but PTS learns a per-instance temperature whereas InvLT learns a shared nonlinear transformation of the logit values themselves.

Uni-variate calibration methods. Histogram binning [Zadrozny and Elkan, 2001] partitions the confidence range into bins and assigns each bin the empirical accuracy of the samples falling within it. Isotonic regression [Zadrozny and Elkan, 2002] fits a monotone piecewise-constant function to map predicted confidences to calibrated probabilities. Platt scaling [Platt et al., 1999] fits a logistic regression on the classifier's outputs. These methods operate in probability space and are originally designed for binary calibration; extending them to the multiclass setting is typically done with a one-vs-rest scheme, after which a renormalization step is required to combine the per-class calibrated probabilities, which can re-shift the already-calibrated outputs.

Spline and Dirichlet calibration. Spline calibration [Gupta et al., 2020] models the calibration map using natural cubic splines in probability space, offering smooth and flexible recalibration. Dirichlet calibration [Kull et al., 2019] transforms the log-probabilities via a linear map followed by a softmax, parameterized as a Dirichlet distribution over the probability simplex. This approach captures cross-class interactions but introduces $O(C^2)$ parameters and can overfit or diverge on datasets with many classes and limited validation data, as evidenced by its poor performance on ImageNet in our experiments.

Order-preserving and monotone neural calibration. Rahimi et al. [Rahimi et al., 2020] introduced intra order-preserving functions for post-hoc calibration, a family of transformations that preserve the top-k predictions of a classifier. Their diagonal sub-family is particularly related to InvLT: it learns an increasing scalar function shared across logit dimensions, using the UMNN architecture of Wehenkel and Louppe [Wehenkel and Louppe, 2019] to enforce monotonicity. UMNN represents a monotone function by parameterizing its derivative as a positive neural network and computing the function value through numerical integration. While this gives strict monotonicity, the repeated numerical integration incurs substantial training and inference overhead. More recently, MCCT [Zhang et al., 2025] enforces instance-wise monotonicity by applying a constrained transformation to the *sorted* logits, giving an interpretable calibration map whose parameter count grows linearly with the number of classes C. In probability space, Berta et al. [Berta et al., 2023] generalize isotonic regression to the multiclass setting with a monotony criterion that preserves the classifier's K-dimensional ROC surface; by the authors' own account this construction becomes overly restrictive beyond a handful of classes, and in our experiments it does not scale to the label spaces we consider (Appendix A).

InvLT targets the same diagonal, order-preserving setting, but replaces hard architectural monotonicity with a soft inverse-reconstruction regularizer, allowing the calibrating function to be a standard scalar MLP. First, it shares a single scalar MLP across all logit dimensions, making its parameter count independent of C. Second, it replaces hard monotonicity like UMNN's numerical integration or MCCT's constrained optimization with a soft inverse-reconstruction regularizer that is discarded at inference. The inverse-reconstruction idea itself is well established in image-to-image translation and generative modeling [Zhu et al., 2017]; our contribution is not the regularizer but the observation that it serves as an effective, lightweight substitute for hard monotonicity constraints in the

post-hoc calibration setting. Third, it operates directly in logit space and scales to ImageNet-sized label spaces where ROC-regularized isotonic regression is inapplicable.

3 Method

We present *Invertible Logits Transformation* (InvLT), a post-hoc calibrator that learns a nonlinear transformation of the logit scale while preserving the classifier's predictions. The central idea is to recalibrate confidence by reshaping logits through a strictly increasing scalar function, obtained by training f to be invertible rather than constraining it to be monotone by construction.

3.1 Post-hoc Calibration and Accuracy Preservation

Let $\mathbf{z} \in \mathbb{R}^C$ denote the pre-softmax logits produced by a fixed classifier for an input $\mathbf{x}$, where C is the number of classes. The corresponding predictive distribution is $\hat{\mathbf{p}} = \mathrm{softmax}(\mathbf{z})$. Post-hoc calibration learns a transformation $\phi : \mathbb{R}^C \to \mathbb{R}^C$ on a held-out calibration set $\mathcal{D}_{\mathrm{cal}} = \{(\mathbf{z}_i, y_i)\}_{i=1}^N$ and returns calibrated probabilities $\hat{\mathbf{p}}^{\,\mathrm{cal}} = \mathrm{softmax}(\phi(\mathbf{z}))$. In many settings, the base classifier has already been optimized for accuracy, and calibration is intended only to correct its confidence estimates. We therefore require the calibration map to preserve the predicted class. We call a transformation ϕ accuracy-preserving if

$$\arg\max_c \phi(\mathbf{z})_c = \arg\max_c z_c \qquad \text{for all } \mathbf{z}.$$

This ensures that calibration changes only the probabilities assigned to classes, not the classifier's decision.

Logit-specific transformation restricts the calibration map to an element-wise transformation:

$$\phi_f(\mathbf{z}) = \big(f(z_1), f(z_2), \ldots, f(z_C)\big), \tag{1}$$

where $f : \mathbb{R} \to \mathbb{R}$ is invertible and shared across all classes and all samples. The calibrated probabilities are then

$$\hat{\mathbf{p}}^{\,\mathrm{cal}} = \mathrm{softmax}(\phi_f(\mathbf{z})).$$

The key property of this family is that strict increase of f is sufficient to preserve accuracy. The ordering of all logits is unchanged after applying f element-wise. Therefore the index of the largest logit is unchanged, and the predicted class is preserved.

3.2 Monotonicity via Invertibility

A standard neural network is flexible but does not by itself guarantee monotonicity. Prior work addresses this by enforcing strict monotonicity through architectural constraints: UMNN [Wehenkel and Louppe, 2019] and the Intra Order-Preserving method [Rahimi et al., 2020] parameterize the derivative f' as a positive network and recover f via numerical integration. This guarantees monotonicity by construction but incurs substantial computational overhead because numerical quadrature must be performed at every forward pass during both training and inference.

Rather than constraining the architecture, we ensure monotonicity through a reconstruction regularizer. The key theoretical insight is that a continuous scalar function that is invertible must be strictly monotone:

Proposition 1 (Continuous injective functions are strictly monotone)**.** *Let $I \subseteq \mathbb{R}$ be an interval and let $f : I \to \mathbb{R}$ be continuous. If f is injective, then f is strictly monotone on I.*

The proof, via the Intermediate Value Theorem, is given in Appendix C.

To establish that f is invertible, it suffice to exhibit a map g satisfying $g \circ f = \mathrm{id}$. This observation motivates a simple regularizer: we introduce an auxiliary inverse network $g_\psi : \mathbb{R} \to \mathbb{R}$ trained jointly with f_θ, and define a set of reference points $\{u_k\}_{k=1}^K$ covering the relevant logit range observed on the calibration set. The reconstruction loss encourages $g_\psi \circ f_\theta$ to approximate the identity:

$$\mathcal{L}_{\mathrm{rec}} = \frac{1}{K} \sum_{k=1}^{K} \big(g_\psi(f_\theta(u_k)) - u_k\big)^2. \tag{2}$$

Minimizing this loss drives f_θ toward injectivity and hence strict monotonicity, without constraining the network architecture. Although this enforcement is soft rather than exact, we find empirically that accuracy is preserved in every experiment we report (Section 4.3). This approach [Zhu et al., 2017] requires only K scalar evaluations per training step, adding negligible cost, while avoiding the numerical integration needed by UMNN. The auxiliary network g_ψ is the same architecture as f_θ used only during training and is discarded at inference.

Orientation. Proposition 1 guarantees strict monotonicity but not its direction; in principle f_θ could converge to a strictly *decreasing* function, which would reverse the argmax. In practice, no explicit orientation penalty is needed: a decreasing f_θ inverts class rankings and produces high training loss, so the calibration objective itself selects the increasing branch. We confirm this empirically: in every setting we tested, InvLT preserves both the increasing orientation and the original classification accuracy (Section 4.3).

3.3 Training and Inference

A post-hoc calibrator is usually fitted by optimizing a Negative Log-Likelihood (NLL) Loss

$$\mathcal{L}_{\text{NLL}} = -\frac{1}{N}\sum_{n=1}^{N} \log \operatorname{softmax}(\phi_{f_\theta}(\mathbf{z}_n))_{y_n}, \tag{3}$$

or a Brier loss

$$\mathcal{L}_{\text{Brier}} = \frac{1}{N}\sum_{n=1}^{N}\sum_{c=1}^{C} \left(\delta_{nc} - \operatorname{softmax}(\phi_{f_\theta}(\mathbf{z}_n))_c\right)^2, \tag{4}$$

where $\delta_{nc} = \mathbf{1}[y_n = c]$.

The complete objective for our InvLT is then:

$$\mathcal{L} = \mathcal{L}_{\text{Brier/NLL}} + \lambda \cdot \mathcal{L}_{\text{rec}}, \tag{5}$$

where $\lambda > 0$ controls the regularization strength. In practice we activate the regularization term after certain epochs. This warm-up technique allows the calibrator to first reduce calibration error before monotonicity is enforced. Both f_θ and g_ψ are optimized jointly with the Adam optimizer [Kingma and Ba, 2014]. Full hyperparameters are in Appendix D.

At inference, g_ψ is discarded. The calibrated probabilities are simply $\operatorname{softmax}(f_\theta(z_1), \ldots, f_\theta(z_C))$. When f_θ is strictly increasing, the predicted class is unchanged. The only inference cost is evaluating a small scalar network on each logit.

Remark. Proposition 1 establish that *exact* invertibility implies strict monotonicity. The reconstruction loss in Eq. (2), however, only enforces $g_\psi \circ f_\theta \approx \mathrm{id}$ at K finitely many reference points with a finite weight λ, and therefore does not provide a formal guarantee that f_θ is globally monotone or that the argmax is preserved on every input. We view the regularizer as a principled heuristic motivated by Proposition 1, and verify it empirically: the learned transformations visualized in Figure 1 are strictly increasing across all datasets and calibration set sizes, and the ablation in Section 4.3 confirms that the regularizer preserves the base classifier's accuracy.

4 Experiments

4.1 Setup

Datasets and models. We evaluate post-hoc calibration on three benchmark settings spanning different scales and class complexities. **CIFAR-10** and **CIFAR-100** [Krizhevsky et al., 2009] use ResNet-50 [He et al., 2016] as the primary backbone, evaluated across five random seeds. For architecture generalization, we additionally evaluate CIFAR-100 on VGG-16/19 [Simonyan and Zisserman, 2014], DenseNet-121 [Huang et al., 2017], Wide ResNet [Zagoruyko and Komodakis, 2016], and ViT-B/16 [Dosovitskiy et al., 2021]. **ImageNet** [Deng et al., 2009] uses pretrained ResNet-50 and ResNet-152 checkpoints. All logits are pre-extracted and fixed; calibration methods operate only on held-out validation logits. We use $N = 500$, $N = 5000$, $N = 25000$ held-out samples for calibrator fitting (denoted val 500, val 5 000, val 25 000) as our primary setting.

Baselines. We compare against No Calibration, Temperature Scaling (TS) [Guo et al., 2017], Vector/Matrix Scaling [Guo et al., 2017], Ensemble Temperature Scaling (ETS) [Zhang et al., 2020], Parameterized Temperature Scaling (PTS) [Tomani et al., 2022], Histogram Binning [Zadrozny and Elkan, 2001], Isotonic Regression [Zadrozny and Elkan, 2002], Spline Calibration [Gupta et al., 2020], Dirichlet Calibration [Kull et al., 2019], and UMNN Calibration [Wehenkel and Louppe, 2019, Rahimi et al., 2020]. We separate methods into those that may alter the predicted class and those that preserve the original classification decision by construction.

Metrics. We report Expected Calibration Error (ECE, 10 equal-width bins, %↓) [Naeini et al., 2015, Guo et al., 2017], Adaptive ECE (AdaECE, %↓) [Nixon et al., 2019], KDE-ECE (%↓) [Zhang et al., 2020], and Negative Log-Likelihood (NLL, ↓). Classification accuracy is unchanged for accuracy-preserving methods.

Hyperparameters. Activation function and hidden dimensions are selected per (dataset, calibration - set size) by grid search on the calibration set, using 5-seed-mean ECE as the selection criterion. To ensure a fair comparison, all tunable baselines (including PTS, UMNN, Histogram Binning, Spline, Matrix Scaling, and Dirichlet Calibration) are grid-searched under the same protocol. The selected configurations, the search grid, and the corresponding settings for all tunable methods are given in Appendix D.

4.2 Main Results

Table 1 reports all four calibration metrics at val 5 000. InvLT achieves the best ECE, AdaECE, KDE-ECE, and NLL on all three datasets, and its margin over the strongest accuracy-preserving baseline (UMNN) is widest on CIFAR-100.

On CIFAR-10, InvLT attains the best ECE (0.59%) and KDE-ECE (0.94%), ahead of UMNN(0.67% and 1.03%), and ties it for the lowest NLL(0.15). On CIFAR-100 the gap is largest: InvLT reaches ECE $= 1.75\%$, below both the best accuracy-preserving baseline (UMNN, 2.22%) and the best non-preserving one (Dirichlet, 2.09%), while matching the lowest NLL (0.79). On ImageNet (ResNet-152), InvLT again leads on every metric (ECE 0.70%, NLL 0.66), narrowly ahead of UMNN (0.72%, 0.67) and well below the best non-preserving baseline, Spline Calibration (1.10%).

Table 2 expands the ImageNet ResNet-152 evaluation across three calibration-set sizes. InvLT achieves the best ECE, AdaECE, KDE-ECE, and NLL at all three sizes. The difficulty of this setting is visible in the baselines at val 500: Vector Scaling reaches ECE $= 43.44\%$ and Isotonic Regression 33.45%, while Matrix Scaling collapses to the identity map (no calibration effect beyond No Cal.), as their $O(C^2)$ or $O(C)$ parameters cannot be estimated from 500 samples when $C = 1{,}000$. Among the accuracy-preserving methods only UMNN stays competitive, and InvLT leads it at every size (e.g. 0.40% vs. 0.49% ECE at val 25 000). Standard deviations over seeds are reported in Appendix E.

Computational efficiency. Table 16 compares wall-clock fitting and inference times across all methods. VS and TS are the fastest but achieve substantially worse calibration (Table 1); Spline Calibration is the slowest at inference (9.01 s on CIFAR-100, and 88.01 s on ImageNet) due to its non-parametric evaluation. The most informative comparison is against UMNN [Wehenkel and Louppe, 2019, Rahimi et al., 2020], the strongest baseline that—like InvLT—applies a monotone scalar function element-wise to logits. InvLT matches or outperforms UMNN on ECE and NLL in every setting while being $3.5\times$ faster to fit (22.7 s vs. 80.2 s on CIFAR-100) and $5\times$ faster at inference (74.8 ms vs. 372.7 ms). The gap stems from a fundamental design difference: UMNN enforces strict monotonicity by parameterizing f' as a positive network and computing f via numerical quadrature at every evaluation, whereas InvLT uses a standard network with a lightweight reconstruction penalty that is discarded at test time.

Robustness over seeds. Figure 1 visualizes the learned scalar function f_θ on three datasets, overlaying three independent random seeds (each fit on val 5 000 and re-centered so that $f_\theta(0) = 0$, removing the additive gauge freedom that cancels under softmax, and plotted over the 2nd–98th percentile of the logits the model actually produces). Across the range where the logits concentrate, the three per-seed curves are nearly indistinguishable: the mean pairwise relative L_1 distance between seeds is 8.8% (6.2% on CIFAR-100, 8.0% on CIFAR-10, 12.1% on ImageNet), and on

Table 1: ECE, AdaECE, KDE-ECE (%↓) and NLL (↓) on CIFAR-10/100 (ResNet-50) and ImageNet (ResNet-152), at val 5 000, means over 5 seeds. Methods above the double rule may change the predicted class; below are accuracy-preserving. **Bold**: best; underline: second best. ‡Matrix Scaling's grid optimum on ImageNet collapses to the identity map (no effect beyond No Cal.).

	CIFAR-10				CIFAR-100				ImageNet			
Method	ECE	Ada	KDE	NLL	ECE	Ada	KDE	NLL	ECE	Ada	KDE	NLL
Accuracy may change												
No Cal.	3.09	3.09	4.25	0.20	11.88	11.85	14.25	0.98	12.83	12.83	12.85	0.87
Hist. Bin.	0.99	2.35	1.09	0.40	5.92	8.84	6.64	2.40	5.61	6.82	5.87	3.49
Isotonic	1.03	0.90	1.47	0.20	6.14	6.04	6.97	1.30	6.84	6.84	7.62	2.97
Vector Sc.	0.91	1.18	1.41	0.16	4.96	4.77	5.35	0.89	10.95	10.78	11.45	2.40
Matrix Sc.	0.88	1.13	1.43	0.16	4.04	3.84	4.30	0.83	12.83‡	12.83	12.85	0.87
Dirichlet	0.89	1.14	1.43	0.16	2.09	2.08	2.25	0.99	2.93	3.06	3.12	1.53
Spline	1.54	1.70	2.12	0.27	7.74	7.66	8.55	1.17	1.10	1.19	1.34	0.74
Accuracy-preserving												
TS	1.69	1.54	1.86	0.17	4.67	4.64	5.17	0.91	4.66	4.72	4.74	0.77
ETS	1.64	1.48	1.81	0.17	2.72	2.78	2.81	0.90	2.79	2.81	2.79	0.75
PTS	0.76	0.69	1.07	0.16	2.55	2.98	2.93	0.89	3.02	3.63	3.35	0.74
UMNN	0.67	0.68	1.03	**0.15**	2.22	2.51	2.51	**0.79**	0.72	0.76	0.86	0.67
InvLT (ours)	**0.59**	**0.58**	**0.94**	**0.15**	**1.75**	**2.05**	**2.02**	**0.79**	**0.70**	**0.75**	**0.83**	**0.66**

Table 2: ECE, AdaECE, KDE-ECE (%↓) and NLL (↓) on ImageNet (ResNet-152) at three calibration-set sizes, means over 5 seeds (uncalibrated ECE $= 12.83\%$). Rule split and **bold**/underline as in Table 1. ‡Matrix Scaling collapses to the identity map at this val size. Isotonic Regression and Vector Scaling degrade catastrophically at val 500, where 500 samples cannot fit their per-class parameters.

	val 500				val 5 000				val 25 000			
Method	ECE	Ada	KDE	NLL	ECE	Ada	KDE	NLL	ECE	Ada	KDE	NLL
Accuracy may change												
No Cal.	12.83	12.83	12.85	0.87	12.83	12.83	12.85	0.87	12.83	12.83	12.85	0.87
Hist. Bin.	5.16	6.19	5.80	3.05	5.61	6.82	5.87	3.49	6.00	6.76	5.67	2.74
Isotonic	33.45	33.42	34.85	12.54	6.84	6.84	7.62	2.97	3.26	3.26	3.61	1.46
Vector Sc.	43.44	43.44	43.77	23.48	10.95	10.78	11.45	2.40	4.75	4.73	5.08	0.81
Matrix Sc.	12.83‡	12.83	12.85	0.87	12.83‡	12.83	12.85	0.87	10.54	10.54	11.45	0.86
Dirichlet	13.20	13.33	12.88	1.31	2.93	3.06	3.12	1.53	4.75	4.88	4.84	1.27
Spline	5.23	5.19	5.34	1.82	1.10	1.19	1.34	0.74	1.40	1.47	1.60	0.71
Accuracy-preserving												
TS	4.92	4.87	4.90	0.77	4.66	4.72	4.74	0.77	2.73	2.78	2.78	0.75
ETS	2.85	2.84	2.88	0.75	2.79	2.81	2.79	0.75	2.81	2.78	2.77	0.75
PTS	3.08	3.53	3.31	0.74	3.02	3.63	3.35	0.74	2.71	3.49	3.20	0.74
UMNN	1.02	1.08	1.17	0.68	0.72	0.76	0.86	0.67	0.49	0.56	0.68	0.67
InvLT (ours)	**0.98**	**1.00**	**1.16**	**0.67**	**0.70**	**0.75**	**0.83**	**0.66**	**0.40**	**0.52**	**0.63**	**0.66**

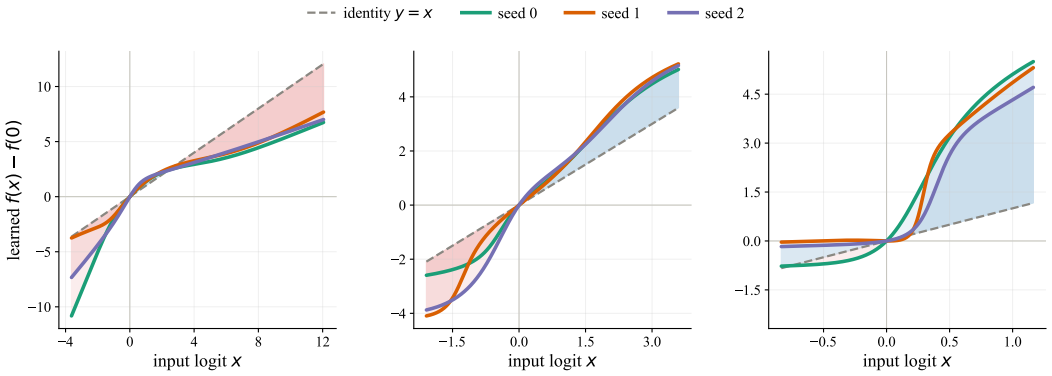
Table 3: ECE (%↓) of the accuracy-preserving methods on CIFAR-100 across network architectures (val 5 000). *Avg*: mean over the five non-ResNet-50 architectures. **Bold**: best per column. Underline: second best. The ResNet-50 column is a 5-seed mean from the same single evaluation pass as Tables 1 and 2, with PTS, UMNN and InvLT grid-searched (batch 1000, K=5). The other five columns are each a *single* run (seed 42) at default hyperparameters – no grid search or multi-seed evaluation has been performed for them, and they have not been re-run under the present protocol.

Method	ResNet-50	VGG-16	VGG-19	DN-121	WRN	ViT-B/16	*Avg*
TS	4.67	4.29	6.18	9.31	3.52	2.58	5.18
ETS	2.72	4.29	6.53	3.12	2.60	2.64	3.84
PTS	2.55	4.21	3.91	2.15	2.67	2.46	3.08
UMNN	2.22	5.15	5.19	1.94	2.66	2.54	3.50
InvLT (ours)	**1.75**	**2.89**	**3.17**	**1.87**	**2.31**	**1.81**	**2.41**

Table 4: Wall-clock efficiency across datasets. Fit: 1,000 iterations. Inference: 10,000 samples. All timings on a single CPU.

Dataset	Phase	TS	VS	PTS	IR	Spline	UMNN	**InvLT (ours)**
CIFAR-10	Fit	4.49 s	3.24 s	5.93 s	15.8 ms	57.7 ms	33.16 s	12.97 s
	Inference	1.4 ms	1.1 ms	3.2 ms	7.6 ms	935.4 ms	53.6 ms	6.9 ms
CIFAR-100	Fit	3.19 s	2.58 s	8.49 s	102.4 ms	687.6 ms	80.20 s	22.70 s
	Inference	0.6 ms	1.4 ms	12.1 ms	90.1 ms	9.01 s	372.7 ms	74.8 ms
ImageNet	Fit	3.54 s	3.33 s	17.14 s	799.9 ms	6.19 s	570.46 s	72.56 s
	Inference	2.9 ms	8.0 ms	120.7 ms	987.5 ms	88.01 s	3.94 s	516.3 ms

CIFAR-10 the seed-to-seed spread averages only 0.46 logits against a vertical extent of 18.5. InvLT therefore learns a stable transformation, robust to random initialization and to optimization noise. That transformation departs substantially from the identity in a dataset-dependent and clearly non-affine way: on CIFAR-10 f_θ flattens as x grows, compressing the upper logits, whereas on CIFAR-100 and ImageNet it runs steeper than the identity through the middle of the range. In every case the net effect is to pull confidence down towards accuracy, on CIFAR-100 the mean top-class probability falls from 0.910 to 0.80 against an accuracy of 0.794. This nonlinear reshaping, which temperature scaling cannot express, is what explains InvLT's calibration gains.



(a) CIFAR-10 (b) CIFAR-100 (c) ImageNet

Figure 1: Learned scalar transformation f_θ for three random seeds on (a) CIFAR-10 (ResNet-50), (b) CIFAR-100 (ResNet-50) and (c) ImageNet (ResNet-152), each fit on val 5 000, re-centered at $f_\theta(0) = 0$ and plotted over the 2nd–98th percentile of the model's logits. Dashed grey: identity $y = x$. Shading: per-seed deviation from identity (red where $f_\theta(x) < x$, blue where $f_\theta(x) > x$).

Reliability diagrams. Figure 2 shows reliability diagrams for eight post-hoc methods on CIFAR-100 (ResNet-50, val 5 000, averaged over 5 seeds, 10 equal-width bins). Blue bars give the empirical accuracy of each confidence bin and red segments the gap to the diagonal, which a perfectly calibrated model would follow exactly; the value above each bar is the mean fraction of test samples the bin contains. Those fractions are what make the panels readable, because ECE weights every bin by its occupancy, and on CIFAR-100 the mass is strongly right-skewed: each method places more than half of the test set in its most confident bin, so a wide red segment over a near-empty bin at the left edge costs far less than a thin one at the right. Histogram Binning and Isotonic Regression show the widest red segments, and they sit in the mid- and high-confidence bins that hold most of the data; Histogram Binning never emits a confidence above 0.9 at all, piling 81% of the test set into a single bin. Temperature Scaling, ETS and PTS track the diagonal through the middle of the range but leave a clearly visible gap in the two rightmost bins, where the bulk of their mass sits. Dirichlet Calibration, UMNN and InvLT stay close to the diagonal throughout, and UMNN and InvLT additionally leave the lowest bins empty, assigning no test sample a confidence below roughly 0.1. Among the eight, InvLT's bars lie closest to the diagonal across the occupied range, leaving only thin red slivers even in the densest bins, and it attains the lowest ECE at 1.75%.

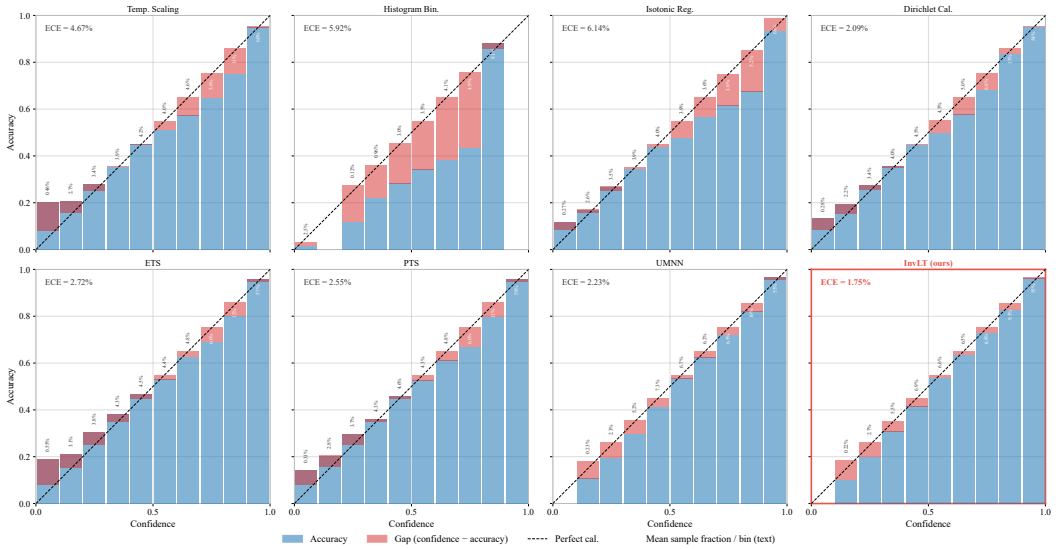


Figure 2: Reliability diagrams for eight post-hoc calibration methods on CIFAR-100(ResNet-50, val 5 000, 5 seeds, 10 equal-width bins). Blue bars: per-bin accuracy; red segments: gap to perfect calibration; text above each bar: mean fraction of samples in that bin. InvLT attains the lowest ECE.

4.3 Ablation Study

The reconstruction regularizer is the central design choice in InvLT: it softly enforces monotonicity of f_θ and thereby preserves the classifier's original predictions. We ablate its effect by comparing InvLT trained with and without the reconstruction loss on CIFAR-10 (ResNet-50, val 5 000).

Table 5 summarizes the results. Without the reconstruction loss, classification error increases ($4.52\% \rightarrow 4.69\%$), confirming that an unregularized f_θ can become non-monotone and alter the argmax. With the reconstruction loss enabled, InvLT achieves competitive ECE while classification error returns to the original model's accuracy (same error, 4.52%), verifying that the regularizer successfully enforces monotonicity. Importantly, the regularizer does not trade off calibration quality for accuracy preservation: both ECE and NLL also improve, suggesting that the invertibility constraint acts as a beneficial inductive bias that guides f_θ toward well-behaved calibration maps rather than merely preventing degenerate solutions.

We also ablate other design choices—the number of reconstruction samples K, warmup delay, network depth and width, and clipping range—in Appendix F. The main finding is that InvLT is robust to these choices once the reconstruction constraint is sufficiently strong and densely sampled.

Table 5: Effect of the reconstruction regularizer on CIFAR-10 (ResNet-50, val 5 000, seed 0). Without regularization, classification error increases; enabling it preserves the original accuracy while also improving calibration.

Setting	ECE (%↓)	AdaECE (%↓)	NLL (↓)	Error (%)
No cal.	3.05	3.04	0.21	4.52
Without reg.	0.94	0.93	0.16	4.69
With reg.	0.92	0.83	0.15	4.52

5 Discussion and Conclusion

The element-wise design of InvLT is both a strength and a limitation: sharing a single scalar function across all logit dimensions makes the parameter count independent of C and scales gracefully to ImageNet ($C = 1{,}000$), where class-dependent methods such as matrix scaling and Dirichlet calibration ($O(C^2)$ parameters) degrade severely (Table 1). However, this design cannot capture cross-class dependencies in the miscalibration structure. A natural extension would be to introduce lightweight cross-class interactions, though this risks overfitting on small calibration sets. Additionally, the soft monotonicity enforcement does not formally guarantee argmax preservation; in extremely safety-critical settings, a post-hoc argmax check may be advisable.

A central design choice is the use of soft rather than hard monotonicity. UMNN enforces strict monotonicity via numerical integration, which guarantees argmax preservation by construction but incurs substantial training and inference overhead. InvLT's reconstruction regularizer provides no such formal guarantee, yet accuracy is preserved in every experiment we report (Section 4.3). The practical advantage is twofold: it avoids the computational cost of quadrature, and it allows f to be any standard network with any activation function, simplifying implementation. At inference the auxiliary network is discarded, adding negligible cost over the original classifier.

In summary, InvLT offers a simple, efficient, and flexible approach to post-hoc calibration: a shared scalar network applied element-wise to logits, with monotonicity softly enforced via a paired inverse network that is discarded at test time. Across CIFAR-10, CIFAR-100, and ImageNet with seven architectures, InvLT consistently achieves the best ECE and NLL among all compared methods, and preserves accuracy while being substantially faster to train than UMNN. Future work includes exploring hard monotonicity constraints via non-negative weights, incorporating cross-class interactions, and extending evaluation to domains beyond image classification.

Acknowledgements

This work was supported in part by seed funding from the PSOM AI2D Center at Penn.

References

Eugene Berta, Francis R. Bach, and Michael I. Jordan. Classifier calibration with roc-regularized isotonic regression. *ArXiv*, abs/2311.12436, 2023. URL `https://api.semanticscholar.org/CorpusID:265308914`.

Jock A Blackard and Denis J Dean. Comparative accuracies of artificial neural networks and discriminant analysis in predicting forest cover types from cartographic variables. *Computers and electronics in agriculture*, 24(3):131–151, 1999.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *Proceedings of the 9th International Conference on Learning Representations (ICLR)*, 2021.

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

Kartik Gupta, Amir Rahimi, Thalaiyasingam Ajanthan, Thomas Mensink, Cristian Sminchisescu, and Richard Hartley. Calibration of neural networks using splines. *arXiv preprint arXiv:2006.12800*, 2020.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. *https://www. cs. toronto. edu/kriz/learning-features-2009-TR. pdf*, 2009.

Meelis Kull, Miquel Perello Nieto, Markus Kängsepp, Telmo Silva Filho, Hao Song, and Peter Flach. Beyond temperature scaling: Obtaining well-calibrated multi-class probabilities with dirichlet calibration. *Advances in neural information processing systems*, 32, 2019.

Matthias Minderer, Josip Djolonga, Rob Romijnders, Frances Hubis, Xiaohua Zhai, Neil Houlsby, Dustin Tran, and Mario Lucic. Revisiting the calibration of modern neural networks. *Advances in neural information processing systems*, 34:15682–15694, 2021.

Mahdi Pakdaman Naeini, Gregory Cooper, and Milos Hauskrecht. Obtaining well calibrated probabilities using bayesian binning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 29, 2015.

Jeremy Nixon, Michael W Dusenberry, Linchuan Zhang, Ghassen Jerfel, and Dustin Tran. Measuring calibration in deep learning. In *CVPR workshops*, 2019.

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

Amir Rahimi, Amirreza Shaban, Ching-An Cheng, Richard Hartley, and Byron Boots. Intra order-preserving functions for calibration of multi-class neural networks. *Advances in neural information processing systems*, 33:13456–13467, 2020.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

Christian Tomani, Daniel Cremers, and Florian Buettner. Parameterized temperature scaling for boosting the expressive power in post-hoc uncertainty calibration. In *European conference on computer vision*, pages 555–569. Springer, 2022.

Antoine Wehenkel and Gilles Louppe. Unconstrained monotonic neural networks. *Advances in neural information processing systems*, 32, 2019.

Bianca Zadrozny and Charles Elkan. Obtaining calibrated probability estimates from decision trees and naive bayesian classifiers. In *Icml*, volume 1, page 2001, 2001.

Bianca Zadrozny and Charles Elkan. Transforming classifier scores into accurate multiclass probability estimates. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 694–699, 2002.

Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016.

Jize Zhang, Bhavya Kailkhura, and T Yong-Jin Han. Mix-n-match: Ensemble and compositional methods for uncertainty calibration in deep learning. In *International conference on machine learning*, pages 11117–11128. PMLR, 2020.

Yunrui Zhang, Gustavo Batista, and Salil S. Kanhere. Instance-wise monotonic calibration by constrained transformation. In *Proceedings of the Forty-First Conference on Uncertainty in Artificial Intelligence*, UAI '25. JMLR.org, 2025.

Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232, 2017.

A Low-Class-Count Comparison

ROC-regularized isotonic regression (ROC-IR) [Berta et al., 2023] is designed for classifiers with few classes and, by its authors' account, becomes overly restrictive beyond a handful of classes; on our main benchmarks ($C \geq 10$) it does not yield usable calibration. We therefore compare against it in the regime it targets, using the Covertype dataset [Blackard and Dean, 1999] restricted to binary ($K = 2$) and ternary ($K = 3$) classification, following Berta et al. For ROC-IR we grid-search its regularization strength and `max_grid_points` and report the best configuration; InvLT uses its default protocol.

Table 6: Post-hoc calibration on Covertype in the low-class-count regime ROC-IR targets. ECE in %(↓); Error and NLL (↓). **Bold**: best ECE. InvLT attains the lowest ECE in both settings while preserving the base classifier's accuracy (Error equals No Cal.), whereas isotonic regression and ROC-IR both increase error.

Method	Error	ECE (%)	NLL
Binary ($K = 2$)			
No Calibration	0.2256	1.15	0.4781
Isotonic Regression	0.2263	1.05	0.5118
ROC-IR	0.2265	0.99	0.4804
InvLT (ours)	0.2256	**0.68**	0.4791
Ternary ($K = 3$)			
No Calibration	0.2254	1.12	0.4836
Isotonic Regression	0.2284	1.66	0.4946
ROC-IR	0.2293	0.74	0.4971
InvLT (ours)	0.2254	**0.54**	0.4833

ROC-IR performs as intended here: in the ternary setting it reduces ECE over standard isotonic regression from 1.66% to 0.74%, confirming its value in the low-K regime it was designed for. InvLT still achieves the lowest ECE in both settings (0.68% at $K = 2$, 0.54% at $K = 3$) while leaving the predicted class unchanged.

B Lambda sensitivity

C Proofs

Proof of Proposition 1. Suppose for contradiction that f is injective and continuous on I but not strictly monotone. Then there exist $a < b < c$ in I with (w.l.o.g.) $f(a) < f(b)$ and $f(b) > f(c)$. Pick $v \in (\max\{f(a), f(c)\},\ f(b))$. By the Intermediate Value Theorem applied to $[a, b]$, there exists $x_1 \in (a, b)$ with $f(x_1) = v$. Similarly, applied to $[b, c]$, there exists $x_2 \in (b, c)$ with $f(x_2) = v$. Since $x_1 \neq x_2$ but $f(x_1) = f(x_2) = v$, this contradicts injectivity. □

Table 7: Sensitivity to the reconstruction weight λ (CIFAR-10, ResNet-50). At $\lambda = 0.001$ accuracy preservation degrades (Error $>$ No Cal.'s 4.52%); for $\lambda \geq 0.01$ error returns to the base model's and calibration is stable. We use $\lambda = 0.01$ throughout and recommend it as a default without per-dataset search.

	val 500			val 1 000		
λ	ECE (%)	NLL	Error (%)	ECE (%)	NLL	Error (%)
0.001	1.02	0.1592	4.87	1.08	0.1565	4.60
0.01	1.17	0.1555	4.53	1.13	0.1555	4.54
0.1	1.18	0.1548	4.52	0.99	0.1541	4.52
1.0	1.13	0.1540	4.52	0.92	0.1536	4.52

D Hyperparameter Settings

Every tunable calibrator's configuration below is the winner of a grid search, selected by lowest 5-seed-mean ECE, at each (dataset, val size) reported in Tables 1–2. CIFAR-10/100 are grid-searched at val 500 and val 5 000; ImageNet at val 500, val 5 000, and val 25 000. No Cal., Temperature Scaling, ETS, Isotonic Regression, and Vector Scaling have no tunable hyperparameters (Vector Scaling fits its per-class scale/bias by NLL, matching the class default) and are omitted.

For the three neural calibrators (InvLT, PTS, UMNN) we search hidden dimensions $\in \{(8,8),(16,16),(24,24),(16,16,16)\}$, activation $\in \{\text{Tanh}, \text{SiLU}, \text{ReLU}\}$, and fitting loss $\in \{\text{NLL}, \text{Brier}\}$: $4 \times 3 \times 2 = 24$ configurations per (dataset, val size). UMNN constrains its activation internally, so for UMNN we search hidden dimensions $\times$ loss only ($4 \times 2 = 8$). Matrix Scaling and Dirichlet use the regularization grid of Table 12; Histogram Binning searches $\{10, 15, 20\}$ bins. In every case the selected configuration minimizes the 5-seed-mean *validation* ECE.

Table 8: InvLT (ours): grid-search-selected architecture, activation, and fitting loss. Fixed across all settings (not searched): learning rate $\eta = 10^{-3}$, max iterations 10,000, warmup $t_0 = 100$, reconstruction samples $K = 100$, $\lambda = 0.01$.

Dataset	val size	Hidden dims	Activation	Loss	Batch B
CIFAR-10	500	$(16,16)$	ReLU	NLL	1000†
CIFAR-10	5,000	$(8,8)$	Tanh	NLL	1000
CIFAR-100	500	$(8,8)$	Tanh	NLL	1000†
CIFAR-100	5,000	$(24,24)$	Tanh	NLL	1000
ImageNet	500	$(8,8)$	ReLU	NLL	1000†
ImageNet	5,000	$(24,24)$	SiLU	NLL	1000
ImageNet	25,000	$(16,16,16)$	ReLU	NLL	1000

†The fit draws a mini-batch of $\min(B, N)$ examples, so at val 500 the nominal batch size of 1000 is clamped to the full calibration set and the batch setting has no effect. Batch size is a fixed protocol constant, not a searched hyperparameter.

Table 9: PTS: grid-search-selected architecture, activation, and fitting loss.

Dataset	val size	Hidden dims	Activation	Loss
CIFAR-10	500	$(16,16)$	SiLU	Brier
CIFAR-10	5,000	$(16,16)$	ReLU	NLL
CIFAR-100	500	$(8,8)$	Tanh	Brier
CIFAR-100	5,000	$(16,16,16)$	Tanh	Brier
ImageNet	500	$(24,24)$	ReLU	Brier
ImageNet	5,000	$(24,24)$	SiLU	Brier
ImageNet	25,000	$(16,16,16)$	ReLU	Brier

Table 10: UMNN: grid-search-selected architecture and fitting loss. The number of Clenshaw–Curtis integration steps is $K = 5$ for all settings.

Dataset	val size	Hidden dims	Loss	K
CIFAR-10	500	$(8, 8)$	NLL	5
CIFAR-10	5,000	$(24, 24)$	NLL	5
CIFAR-100	500	$(8, 8)$	NLL	5
CIFAR-100	5,000	$(24, 24)$	NLL	5
ImageNet	500	$(16, 16)$	NLL	5
ImageNet	5,000	$(16, 16)$	NLL	5
ImageNet	25,000	$(8, 8)$	NLL	5

UMNN uses the same mini-batch size as InvLT (1000, clamped to N at val 500). Both $K = 5$ and $K = 10$ were run for every setting; $K = 5$ is reported. The choice is immaterial everywhere except ImageNet val 5 000, where $K = 10$ attains ECE 0.67% versus 0.72% at $K = 5$.

Table 11: Histogram Binning (number of bins) and Spline Calibration (number of knots): grid-search-selected configurations. Only CIFAR-10/100 at val 5 000 and ImageNet at all three val sizes are used in the main tables.

Dataset	val size	Hist. Bin. (bins)	Spline (knots)
CIFAR-10	5,000	10	9
CIFAR-100	5,000	10	9
ImageNet	500	10	9
ImageNet	5,000	10	9
ImageNet	25,000	10	3

Histogram Binning's ECE is bin-count-invariant here (10/15/20 bins give bit-identical ECE), so the reported "winner" is arbitrary among ties.

Table 12: Matrix Scaling and Dirichlet Calibration: grid-search-selected regularization strength ($\lambda = \mu$ on the diagonal) and fitting loss. ‡On ImageNet, both methods are searched only over the tied diagonal $\lambda = \mu$ (8 configs), not the full $\lambda \times \mu$ cross product used for CIFAR-10/100 (32 configs) – the full cross product does not fit the compute budget at $C = 1{,}000$ classes.

		Matrix Sc.		Dirichlet	
Dataset	val size	λ, μ	Loss	λ, μ	Loss
CIFAR-10	5,000	0.1, 100	NLL	0.1, 0.1	NLL
CIFAR-100	5,000	100, 0.1	NLL	10, 0.1	Brier
ImageNet‡	500	10, 10	Brier	100, 100	Brier
ImageNet‡	5,000	10, 10	Brier	1, 1	Brier
ImageNet‡	25,000	100, 100	NLL	1, 1	Brier

Matrix Scaling on ImageNet at val 500/val 5 000 selects Brier loss because cross-entropy catastrophically overfits the full $1{,}000 \times 1{,}000$ matrix at these val sizes (CE-loss ECE $> 20\%$ vs. Brier's $\approx 12.8\%$, see Table 2); the resulting Brier-loss fit is statistically indistinguishable from the identity map at those two val sizes.

E Results with Standard Deviations

Tables 1 and 3 report means over five random seeds. Table 13 gives the corresponding results with standard deviations for all accuracy-preserving methods on CIFAR-10 and CIFAR-100 (ResNet-50, val 5 000), and Table 14 does the same for ImageNet at all three calibration-set sizes.

Table 13: ECE, AdaECE, KDE-ECE (%↓) and NLL (↓) on CIFAR-10 and CIFAR-100 (ResNet-50, val 5 000, mean ± std over 5 seeds, accuracy-preserving methods). **Bold**: best mean.

	CIFAR-10				CIFAR-100			
Method	ECE	AdaECE	KDE	NLL	ECE	AdaECE	KDE	NLL
TS	$1.69 \pm .36$	$1.54 \pm .29$	$1.86 \pm .26$	$.169 \pm .003$	4.67 ± 3.65	4.64 ± 3.65	5.17 ± 4.58	$.912 \pm .038$
ETS	$1.64 \pm .27$	$1.48 \pm .24$	$1.81 \pm .21$	$.169 \pm .003$	$2.72 \pm .45$	$2.78 \pm .34$	$2.81 \pm .30$	$.895 \pm .011$
PTS	$0.76 \pm .17$	$0.69 \pm .23$	$1.07 \pm .28$	$.162 \pm .004$	$2.55 \pm .37$	$2.98 \pm .20$	$2.93 \pm .26$	$.889 \pm .010$
UMNN	$0.67 \pm .10$	$0.68 \pm .17$	$1.03 \pm .19$	$.153 \pm .003$	$2.22 \pm .49$	$2.51 \pm .56$	$2.51 \pm .43$	$.793 \pm .011$
InvLT	$\mathbf{0.59 \pm .10}$	$\mathbf{0.58 \pm .17}$	$\mathbf{0.94 \pm .17}$	$\mathbf{.152 \pm .002}$	$\mathbf{1.75 \pm .64}$	$\mathbf{2.05 \pm .75}$	$\mathbf{2.02 \pm .58}$	$\mathbf{.793 \pm .011}$

On CIFAR-100 the NLL of UMNN and InvLT agree to three decimals (0.7932 vs. 0.7929); InvLT is lower at the fourth.

F Additional Ablations

Table 15 ablates remaining design choices on CIFAR-10 (ResNet-50, val $= 500$). The number of reconstruction samples K controls a trade-off: smaller K achieves slightly better ECE but at the cost of accuracy preservation; $K \geq 25$ ensures accuracy preservation while maintaining competitive calibration. The warmup delay has negligible effect, indicating that the reconstruction loss does not interfere with early optimization. MLP architecture, residual connections, and clipping range are selected per dataset via grid search on the validation set; we find that performance is generally robust across reasonable settings.

G Efficiency on GPU

Table 14: ECE, AdaECE, KDE-ECE (%↓) and NLL (↓) on ImageNet with ResNet-152 under three calibration set sizes (mean±std over 5 seeds). The uncalibrated model has ECE $= 12.83\%$, making this a more challenging calibration setting. Methods above the double rule may change the predicted class; below are accuracy-preserving. ‡Matrix Scaling's optimum converges to the identity map at this val size, so all four metrics equal No Cal.'s exactly. **Bold**: best. Underline: second best.

(a) Calibration set size: val 500

Method	ECE	AdaECE	KDE-ECE	NLL
Accuracy may change				
No Cal.	$12.83 \pm .17$	$12.83 \pm .17$	$12.85 \pm .17$	$.867 \pm .003$
Hist. Bin.	$5.16 \pm .54$	$6.19 \pm .20$	$5.80 \pm .35$	$3.048 \pm .023$
Isotonic	33.45 ± 6.68	33.42 ± 6.69	34.85 ± 6.00	12.540 ± 1.535
Vector Sc.	43.44 ± 9.45	43.44 ± 9.45	43.77 ± 11.11	23.485 ± 2.249
Matrix Sc.	$12.83 \pm .17$‡	$12.83 \pm .17$	$12.85 \pm .17$	$.867 \pm .003$
Dirichlet	$13.20 \pm .91$	$13.33 \pm .76$	$12.88 \pm .67$	$1.307 \pm .057$
Spline	5.23 ± 1.32	5.19 ± 1.38	5.34 ± 1.39	$1.817 \pm .045$
Accuracy-preserving				
TS	4.92 ± 4.09	4.87 ± 4.11	4.90 ± 4.10	$.774 \pm .044$
ETS	$2.85 \pm .21$	$2.84 \pm .17$	$2.88 \pm .12$	$.751 \pm .003$
PTS	$3.08 \pm .35$	$3.53 \pm .54$	$3.31 \pm .45$	$.741 \pm .002$
UMNN	$1.02 \pm .26$	$1.08 \pm .23$	$1.17 \pm .22$	$.678 \pm .006$
InvLT (ours)	$\mathbf{0.98 \pm .37}$	$\mathbf{1.00 \pm .39}$	$\mathbf{1.16 \pm .43}$	$\mathbf{.672 \pm .005}$

(b) Calibration set size: val 5 000

Method	ECE	AdaECE	KDE-ECE	NLL
Accuracy may change				
No Cal.	$12.83 \pm .17$	$12.83 \pm .17$	$12.85 \pm .17$	$.867 \pm .003$
Hist. Bin.	$5.61 \pm .37$	$6.82 \pm .32$	$5.87 \pm .17$	$3.494 \pm .042$
Isotonic	$6.84 \pm .23$	$6.84 \pm .23$	$7.62 \pm .25$	$2.973 \pm .032$
Vector Sc.	$10.95 \pm .28$	$10.78 \pm .32$	$11.45 \pm .30$	$2.397 \pm .067$
Matrix Sc.	$12.83 \pm .17$‡	$12.83 \pm .17$	$12.85 \pm .17$	$.867 \pm .003$
Dirichlet	2.93 ± 1.17	3.06 ± 1.04	3.12 ± 1.09	$1.527 \pm .580$
Spline	$1.10 \pm .08$	$1.19 \pm .08$	$1.34 \pm .12$	$.740 \pm .003$
Accuracy-preserving				
TS	4.66 ± 4.00	4.72 ± 3.97	4.74 ± 3.97	$.775 \pm .045$
ETS	$2.79 \pm .20$	$2.81 \pm .16$	$2.79 \pm .13$	$.750 \pm .004$
PTS	$3.02 \pm .41$	$3.63 \pm .40$	$3.35 \pm .38$	$.739 \pm .004$
UMNN	$0.72 \pm .24$	$0.76 \pm .27$	$0.86 \pm .25$	$.666 \pm .004$
InvLT (ours)	$\mathbf{0.70 \pm .21}$	$\mathbf{0.75 \pm .23}$	$\mathbf{0.83 \pm .20}$	$\mathbf{.665 \pm .004}$

(c) Calibration set size: val 25 000

Method	ECE	AdaECE	KDE-ECE	NLL
Accuracy may change				
No Cal.	$12.83 \pm .17$	$12.83 \pm .17$	$12.85 \pm .17$	$.867 \pm .003$
Hist. Bin.	$6.00 \pm .07$	$6.76 \pm .18$	$5.67 \pm .08$	$2.740 \pm .048$
Isotonic	$3.26 \pm .10$	$3.26 \pm .10$	$3.61 \pm .12$	$1.456 \pm .044$
Vector Sc.	$4.75 \pm .11$	$4.73 \pm .12$	$5.08 \pm .13$	$.812 \pm .005$
Matrix Sc.	10.54 ± 2.01	10.54 ± 2.01	11.45 ± 1.27	$.856 \pm .007$
Dirichlet	4.75 ± 4.08	4.88 ± 4.01	4.84 ± 4.04	$1.268 \pm .216$
Spline	$1.40 \pm .11$	$1.47 \pm .07$	$1.60 \pm .13$	$.709 \pm .003$
Accuracy-preserving				
TS	$2.73 \pm .18$	$2.78 \pm .14$	$2.78 \pm .09$	$.752 \pm .003$
ETS	$2.81 \pm .14$	$2.78 \pm .14$	$2.77 \pm .10$	$.750 \pm .003$
PTS	$2.71 \pm .17$	$3.49 \pm .28$	$3.20 \pm .24$	$.740 \pm .003$
UMNN	$0.49 \pm .17$	$0.56 \pm .16$	$0.68 \pm .13$	$.666 \pm .003$
InvLT (ours)	$\mathbf{0.40 \pm .13}$	$\mathbf{0.52 \pm .04}$	$\mathbf{0.63 \pm .08}$	$\mathbf{.664 \pm .003}$

Table 15: Ablation on InvLT design choices (CIFAR-10, val $= 500$, ResNet-50). † marks the default used in all other experiments.

Component	Setting	ECE (%↓)	Error (%↓)	NLL (↓)
Rec. samples K	5	0.86	4.88	0.163
	10	0.92	4.72	0.157
	25	1.14	4.55	0.156
	$100^{\dagger}$	1.17	4.53	0.156
	500	1.17	4.53	0.156
Warmup delay τ	0	1.17	4.54	0.156
	$100^{\dagger}$	1.17	4.53	0.156
	1000	1.16	4.54	0.156

Table 16: Wall-clock efficiency on ImageNet (ResNet-152, val 25 000), single RTX 5070 GPU. Fit: grid-selected configuration to convergence. Inference: all 25 000 samples. All methods run under one aggressive budget (lr 0.1, 100 iterations where applicable). **Bold**: fastest expressive (non-TS/VS) method per column.

Method	Fit (s)	Inference (ms)
No Cal.	0.00	0.00
Hist. Bin.	0.70	464.80
Isotonic	3.13	763.10
Vector Sc.	0.53	1.29
Matrix Sc.	2.59	4.35
Dirichlet	2.92	4.77
Spline	143.27	246070.60
TS	0.14	0.65
ETS	8.83	834.20
PTS	0.79	4.65
UMNN	2.32	2788.53
InvLT (ours)	**1.98**	**69.57**

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: The abstract and Section 1 state three concrete contributions—(i) a shared scalar MLP applied element-wise to logits with parameter count independent of C, (ii) a reconstruction-based soft monotonicity regularizer, and (iii) consistent improvements over post-hoc baselines—each of which is substantiated in Sections 3 and 4 (Tables 1–4, Figure 1).

   Guidelines:

   - The answer [N/A] 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 [N/A] 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: Limitations are discussed at the end of Section 5: the element-wise design cannot capture cross-class miscalibration dependencies, and the soft monotonicity enforcement does not provide a formal argmax-preservation guarantee. We note that combining InvLT with a post-hoc argmax check may be advisable in safety-critical settings.

   Guidelines:

   - The answer [N/A] 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: The two theoretical statements (Theorem 1 on continuous injective functions being strictly monotone, and Corollary 2 on argmax preservation) are stated in Section 3.2 with full proofs given in Appendix A. Assumptions (continuity on an interval, existence of a left inverse) are explicit in the statements.

   Guidelines:

   - The answer [N/A] 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: Section 4.1 specifies datasets (CIFAR-10/100, ImageNet), backbones (ResNet-50/152, VGG-16/19, DenseNet-121, Wide ResNet, ViT-B/16), calibration set sizes, metrics, and baselines. All InvLT hyperparameters are provided in Appendix B (Table 6).

   Guidelines:

   - The answer [N/A] 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: Anonymized code to reproduce all experiments is provided in the supplementary material

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.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://neurips.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) necessary to understand the results?

   Answer: [Yes]

   Justification: Section 4.1 describes data splits (val 500 / 5 000 / 25 000), evaluation metrics (ECE, AdaECE, KDE-ECE, NLL), number of seeds (5), and baselines. Optimizer (Adam) and full hyperparameter grid are documented in Section 3.3 and Appendix B (Table 6).

   Guidelines:

   - The answer [N/A] 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: [Yes]

Justification: All main results are reported as means over 5 random seeds. Standard deviations across seeds are provided in Appendix C (Tables 7 and 8) for both CIFAR and ImageNet experiments. Variability captures the random seed used for calibrator fitting.

Guidelines:

- The answer [N/A] 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: [Yes]

   Justification: Wall-clock fitting and inference times are reported in Table 4 on a single CPU. Calibration is lightweight (fitting completes in seconds to tens of seconds),

   Guidelines:

   - The answer [N/A] 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: The research uses only public benchmarks (CIFAR-10/100, ImageNet) and does not involve human subjects, sensitive personal data, or deployments in high-risk settings. We are not aware of any deviation from the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] 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: [Yes]

    Justification: Better-calibrated classifiers support more trustworthy decision-making in downstream applications (medical diagnosis, autonomous systems, selective prediction), as motivated in Section 1. Potential negative impact is limited: post-hoc calibration does not change classifier predictions and could give a false sense of reliability if calibration sets are not representative of deployment data; users should validate calibration on representative data before relying on calibrated probabilities.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] 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., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper does not release pretrained generative models, scraped datasets, or other assets with elevated misuse risk. The proposed calibrator is a small MLP that operates on the logits of an already-trained classifier.

    Guidelines:

    - The answer [N/A] 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 datasets (CIFAR-10/100, ImageNet) and architectures (ResNet, VGG, DenseNet, Wide ResNet, ViT) are cited in Section 4.1. Baseline methods (TS, ETS, PTS, UMNN, Spline, Dirichlet, Histogram Binning, Isotonic, Platt, Vector/Matrix scaling) are cited at first use. License terms of these public assets are respected.

    Guidelines:

    - The answer [N/A] 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: [Yes]

    Justification: we will provided a documented code base along with this paper

    Guidelines:

    - The answer [N/A] 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: [N/A]

    Justification: The work involves no crowdsourcing or human-subject experiments.

    Guidelines:

    - The answer [N/A] 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: [N/A]

    Justification: The work involves no human-subject research and therefore does not require IRB review.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

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

    Answer: [N/A]

    Justification: LLMs were not used as a component of the core methodology. Any use of LLMs was limited to non-substantive editing of writing, which does not require declaration per the NeurIPS policy.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.