NEIGHBORHOOD SMOOTHING FOR CALIBRATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Modern neural networks are often miscalibrated, with a tendency to overconfidence. Existing train-time calibration methods largely modify task losses or calibration penalties, leaving neighborhood structure in learned representations underexploited. We introduce graph smoothing as a general principle for train-time calibration, which encourages similar predictive distributions across neighboring samples in representation space. We analyze the effects of graph smoothing, deriving bounds that connect predictive divergence between neighboring samples to local confidence variation and to the propagation of pointwise calibration error, and characterize the conditions under which smoothing can or cannot improve calibration. In light of this analysis, we propose NeSCal, a graph-based train-time regularizer that penalizes the Jensen–Shannon divergence between predictive distributions of neighboring samples. We present a thorough empirical analysis, showing that across standard calibration benchmarks, NeSCal improves predictive quality, and the improvement is complementary to post-hoc calibration: after temperature scaling, NeSCal attains the lowest NLL of all evaluated train-time methods in seven of the eight image and tabular settings. These findings demonstrate the value of graph smoothing over learned representations for neural network calibration.

1 INTRODUCTION

Confidence calibration in classification refers to the alignment between the empirical correctness of a model's predictions and its confidence in these predictions. A well-calibrated confidence serves as a reliable indication of prediction correctness and an essential component of downstream decisions. Modern deep neural networks tend to be over-confident (Guo et al., 2017). Approaches to addressing miscalibration are typically either post-hoc, learning a function on a withheld data set to reshape confidence values (Guo et al., 2017; Kull et al., 2019; Platt, 2000; Zadrozny & Elkan, 2002), or train-time, modifying the learning objective to encourage the model to be better calibrated, either by some explicit, calibration-aware penalty term (Hebbalaguppe et al., 2022; Karandikar et al., 2021; Krishnan & Tickoo, 2020; Kumar et al., 2018), or implicitly, through a loss shown to improve calibration (Mukhoti et al., 2020; Müller et al., 2019).

Existing train-time calibration methods largely modify the learning objective based primarily on model outputs and labels, without explicitly exploiting the geometry of the representations learned by the model. Yet these representations encode information about which samples the model regards as similar. Nearby samples in representation space should, when they correspond to similar underlying inputs, receive similar predictions, providing a natural structure over which predictive behavior can be regularized.

In this work, we investigate model regularization to encourage adjacent samples in a neighborhood graph over its representation space to receive similar predictive distributions, a principle we term *graph smoothing*. Penalizing the variation of the learned function in such a graph is the basis of manifold regularization (Belkin et al., 2006), where it has been used to improve classification accuracy under limited supervision. Applied to calibration, suppressing predictive disagreement constrains local confidence variation: a well-calibrated prediction can constrain neighboring calibration errors when the connected samples have similar true probabilities of correctness. By exploiting this additional structure during training, graph smoothing can also improve the learned predictor in ways that remain beneficial when combined with subsequent post-hoc recalibration.

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

We instantiate graph smoothing as NeSCal (**Ne**ighborhood **S**moothing for **Cal**ibration), a train-time regularizer that penalizes Jensen–Shannon (JS) divergence between neighboring predictive distributions, added to a task loss through a single weighting term at no inference-time cost.

The paper's contributions are quintuple. (i) We introduce *graph smoothing* as a general principle for train-time calibration, in which a model is regularized to encourage similar predictive distributions for samples adjacent in a neighborhood graph over representation space (Section 3). (ii) We demonstrate the relationship between graph-based predictive smoothness and calibration, showing that the variation of the predictive distribution translates into a bound on the propagation of pointwise calibration error along paths in the graph (Proposition 1; Theorem 1). (iii) We characterize when graph smoothing improves calibration and when it cannot, identifying the variation of true correctness probability across an edge as the quantity governing both (Propositions 2 and 3). (iv) We instantiate the principle as a train-time regularizer NeSCal, penalizing the Jensen–Shannon divergence between the predictive distributions of neighbors in a k-nearest-neighbor graph 4. (v) We show empirically that NeSCal improves negative log-likelihood (NLL) in all eleven main benchmarks and ranks first or second among the evaluated methods in seven of eight image and tabular settings. After temperature scaling, it attains the lowest NLL in seven of those eight settings (Section 6).

2 Background

Consider a generic K-class classification problem, with a model f that receives inputs $x \in X$ and produces softmax probability vectors $p(x) := f(x) \in \Delta^{K-1}$, the $(K-1)$-dimensional probability simplex. Write $p_i := p(x_i)$ and let $p_{i,k}$ denote the ith sample's class-k probability. Each input x has an associated true label $y \in \{1, \ldots, K\}$, with Y the corresponding label random variable. The model's predicted label and its associated confidence for x are $\hat{y}(x) = \arg\max_k p_k(x)$ and $c(x) = \max_k p_k(x)$.

We write the true conditional class-probability vector (the Bayes posterior) $\eta(x) := \big(P(Y = 1 \mid X = x), \ldots, P(Y = K \mid X = x)\big) \in \Delta^{K-1}$. For the predicted class, the conditional probability of correctness is $a(x) := P(Y = \hat{y}(x) \mid X = x) = \eta_{\hat{y}(x)}(x)$, a single coordinate of $\eta(x)$. Thus, $c(x)$ is the model's predicted probability of being correct, while $a(x)$ is the corresponding true conditional probability of correctness. We define $\mathcal{E}_{\text{cal}}(x) := |a(x) - c(x)|$ as the pointwise calibration error at x, writing $\mathcal{E}_{\text{cal}_i} := \mathcal{E}_{\text{cal}}(x_i)$ for samples. Perfect pointwise calibration is then achieved if $c(x) = a(x)$ for all $x \in \mathcal{X}$. The posterior $\eta(x)$ is a property of the data distribution and $a(x)$ also depends on the model's predicted label; neither is directly observed from an single labeled example, so empirical calibration metrics assess aggregated notions of calibration. The most widely used metric is Expected Calibration Error (ECE) Naeini et al. (2015), which measures *confidence calibration* (Guo et al., 2017) by binning predictions according to confidence and comparing confidence means with empirical accuracy within each bin. ECE is a binning-based estimator, it is thus sensitive to the binning scheme and to finite-sample bias (Nixon et al., 2019; Roelofs et al., 2022; Zhang et al., 2020). Moreover, since ECE aggregates across predictions, pointwise over- and under-confidence can cancel. A low ECE therefore does not imply that predictions are calibrated at a specific point. A discussion of the alternative metrics can be found in Appendix D.6.

3 Graph Smoothing for Confidence Calibration

3.1 Problem Setup

We adopt the setting of Section 2, with a dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$ of i.i.d. draws from $P(X, Y)$ with $X \subseteq \mathbb{R}^d$, and a probabilistic classifier $f_\theta : X \to \Delta^{K-1}$ with softmax output layer, trained to minimize $\mathbb{E}_{(X,Y)\sim P}[\mathcal{L}(f_\theta(X), Y)]$ for a task loss $\mathcal{L}$. Write $p_i := f_\theta(x_i)$, $a_i := a(x_i)$ and $c_i := c(x_i)$, so that $\mathcal{E}_{\text{cal}_i} = |a_i - c_i|$. Our analysis targets pointwise calibration: the top-label guarantee is required at each point rather than on average, without necessitating the full predictive vector to match the posterior.

Let $h : \mathcal{X} \to \mathcal{Z} \subseteq \mathbb{R}^l$ produce representations $z_i = h(x_i)$. A neighborhood graph $\mathcal{G} = (V, E)$ has nodes $V = \{z_i\}_{i=1}^n \subset \mathcal{Z}$ and connects nearby representations under a chosen distance. Its

adjacency matrix W assigns nonnegative weights W_{ij} to edges $(i, j) \in E$. Let $\mathrm{D}(p_i \| p_j)$ be a nonnegative divergence between the predictive probability distributions at observations i and j.

We define $R_n(p) := \sum_{i,j} W_{ij}\, \mathrm{D}(p_i \| p_j)$ as the *neighborhood consistency regularizer* and consider the penalized formulation

$$\min_{f_\theta}\ \mathcal{L}(\theta) = \mathcal{L}_{\mathrm{task}}(\theta) + \lambda\, R_n(p), \tag{1}$$

where $\mathcal{L}_{\mathrm{task}}$ is the task loss, and $\lambda > 0$ controls the regularization strength.

3.2 Formal Guarantees

We relate predictive disagreement to confidence variation and pointwise calibration-error propagation on the training graph, at a fixed predictor. Held-out calibration is assessed empirically in Section 6. Proofs and the local quadratic interpretation of the regularizer appear in Appendix A. We make the standing requirement that $\mathrm{D}(p\|q) \geq C_{\mathrm{D}} \|p - q\|_1^2$ for all $p, q \in \Delta^{K-1}$, with $C_{\mathrm{D}} > 0$. This is a Pinsker-type condition that implies $|c_i - c_j| \leq \frac{1}{2}\|p_i - p_j\|_1 \leq \frac{1}{2\sqrt{C_{\mathrm{D}}}} \sqrt{\mathrm{D}(p_i \| p_j)}$, which bounds the propagation of pointwise calibration error:

Proposition 1 (Calibration Propagation) *Under the standing domination condition, for any edge $(i, j) \in E$,*

$$\mathcal{E}_{\mathrm{cal}_j} \ \leq\ \mathcal{E}_{\mathrm{cal}_i} + \frac{1}{2\sqrt{C_{\mathrm{D}}}} \sqrt{\mathrm{D}(p_i \| p_j)} + |a_i - a_j| \tag{2}$$

The bound decomposes propagation into three terms: the calibration error already present at i, a smoothness term controlled directly by the regularizer through $\mathrm{D}(p_i \| p_j)$, and the local variation $|a_i - a_j|$ of the true correctness probability. Only the second is directly penalized. With a fixed graph, the correctness gap depends on θ only through the predicted labels. A calibrated prediction therefore constrains neighboring errors when connected points have similar correctness probabilities. The bound, however, does not detail how a calibrated prediction arises in the first place. This is left to the task loss.

Theorem 1 extends Proposition 1 from a single edge to paths through the grap:

Theorem 1 (Pathwise Calibration Propagation) *Under the standing domination condition, let $P = (v_0, v_1, \ldots, v_r)$ be a simple directed path in $\mathcal{G}$ from $v_0 = s$ to $v_r = v$. Write $D_\ell := \mathrm{D}(p_{v_{\ell-1}} \| p_{v_\ell})$. Then*

$$\mathcal{E}_{\mathrm{cal}_v} \leq \mathcal{E}_{\mathrm{cal}_s} + \sqrt{\frac{r}{4C_{\mathrm{D}}} \sum_{\ell=1}^{r} D_\ell} + |a_v - a_s| \tag{3}$$

A well-calibrated prediction transfers to points connected by short paths along which both the predictive divergence and the variation of the true correctness probability remain small. The bound grows with path length and thus the guarantee is most informative locally. Additional disagreement bounds appear in Appendix A (Propositions 5 and 6, Corollary 1).

3.3 When Smoothing Improves Calibration

Section 3.2 bounds the propagation of calibration error, given that it is small somewhere. We next identify an obstruction to calibrating both endpoints of an edge and a condition under which idealized pairwise averaging reduces their summed expected squared calibration error.

Proposition 2 (Smoothing Obstruction) *Under the standing domination condition, for every edge $(i, j) \in E$,*

$$\mathcal{E}_{\mathrm{cal}_i} + \mathcal{E}_{\mathrm{cal}_j} \geq |a_i - a_j| - \frac{1}{2\sqrt{C_{\mathrm{D}}}} \sqrt{\mathrm{D}(p_i \| p_j)} \tag{4}$$

For fixed predicted labels, reducing predictive disagreement raises the lower bound toward the correctness gap $|a_i - a_j|$. A rising floor, however, does not establish that the realized error rose, as the error may already have exceeded it. Proposition 2 therefore identifies the regime in which smoothing

is provably obstructed from calibrating both endpoints. The next result characterizes when averaging reduces expected squared error.

To analyze the error removed by averaging, we fix an edge and randomize over training. Let $\mathcal{T}$ denote randomness from initialization, data order, and stochastic optimization. We analyze idealized pairwise averaging on a single edge, in which both endpoints adopt the mixture $m = \frac{1}{2}(p_i + p_j)$. Both samples must predict the same fixed label almost surely under $\mathcal{T}$, so that a_i and a_j are fixed targets, and then averaging the predictive distributions is equivalent to averaging the confidences, $\max_k m_k = \frac{1}{2}(c_i + c_j)$. The regularizer has access to neither a nor the labels; it only draws predictions toward one another. The question is when doing so reduces the summed expected squared calibration error.

Proposition 3 (Smoothing threshold) *Let (i, j) be an edge with $\hat{y}_i = \hat{y}_j = k^*$ almost surely under $\mathcal{T}$, and write $e_i := c_i - a_i$, $e_j := c_j - a_j$ for the pointwise errors. Replacing both confidences by their average strictly reduces the summed expected squared calibration error of the edge iff*

$$(a_i - a_j)^2 \;<\; \mathbb{E}_{\mathcal{T}}\big[(e_i - e_j)^2\big]. \tag{5}$$

A bias–variance decomposition and correlation-dependent specialization appear in Appendix A.9. Only the difference $e_i - e_j$ enters equation 5. Averaging can reduce error that differs between neighbors, whether random or systematic, but cannot touch the component they share, and systematic overconfidence is precisely the error deep classifiers are known to exhibit (Guo et al., 2017). This motivates testing whether gains from neighborhood smoothing remain useful after a global confidence adjustment. Section 6.2 evaluates this empirically by applying the same post-hoc calibrators to every train-time method.

4 NeSCal

4.1 The Regularizer

NeSCal instantiates the graph-smoothing objective of Equation 1 with the Jensen–Shannon divergence and a k-nearest-neighbor graph over learned representations:

$$\min_{f_\theta} \; \mathcal{L}(\theta) = \mathcal{L}_{\mathrm{CE}}(\theta) + \lambda \sum_{i,j} W_{ij}\, \mathrm{JS}(p_i \| p_j) \tag{6}$$

where $\mathcal{L}_{\mathrm{CE}}$ is the cross-entropy (CE) loss and $\mathrm{JS}(p_i \| p_j)$ is the Jensen–Shannon divergence (Endres & Schindelin, 2003; Fuglede & Topsoe, 2004), $\mathrm{JS}(p_i \| p_j) = \frac{1}{2} D_{\mathrm{KL}}(p_i \| m) + \frac{1}{2} D_{\mathrm{KL}}(p_j \| m)$, with $m = \frac{1}{2}(p_i + p_j)$ and $D_{\mathrm{KL}}(u \| v) = \sum_k u_k \log(u_k / v_k)$. We use base-2 logarithms in the JS divergence, so that $\mathrm{JS}(p_i \| p_j) \in [0, 1]$. Cross-entropy and reported NLL use natural logarithms.

The adjacency matrix $W \in \{0, 1\}^{n \times n}$ encodes a k-nearest-neighbor graph (kNN) in the representation space $\mathcal{Z}$, with $W_{ij} = 1 \iff j \in \mathrm{kNN}(i)$. For optimization, we average the penalty over the directed edge set $E = \{(i, j) : W_{ij} = 1\}$, using $\bar{R}_n(p) = R_n(p)/|E|$.

JS divergence satisfies the total-variation domination condition with $C_{\mathrm{JS}} = 1/(8 \ln 2)$, allowing the general confidence and propagation bounds to apply. Its symmetry and bounded range make it a sufficient and practical (rather than uniquely optimal) choice for comparing neighboring predictive distributions, while comparison of the full vectors captures disagreement across all classes. The induced local quadratic geometry is described in Appendix B.

4.2 Design Choices

Neighborhood Graph. Neighbors are identified by cosine distance in $\mathcal{Z}$. The graph is constructed once at activation and held fixed, limiting the construction cost and keeping the neighborhoods unchanged during regularized training. We use $k = 10$ by default. Section 6.3 examines sensitivity to this choice.

Activation and Schedule. The activation epoch is a hyperparameter, swept jointly with λ on validation NLL. Proposition 2 motivates delaying graph construction until the representation matures and neighborhoods become semantically meaningful. If the across-run dispersion of neighboring errors shrinks as training converges, the tolerable correctness gap narrows (Proposition 3), motivating a reduction in regularization strength. NeSCal accordingly activates the regularizer after an initial warm-up and decays λ over the remainder of training on a cosine schedule. Further discussion appears in Appendix C.4.

Task Loss. NeSCal uses cross-entropy as its task loss. Cross-entropy is a strictly proper scoring rule: its expected loss is uniquely minimized at the true conditional distribution. In contrast, the regularizer R_n is minimized whenever neighboring predictions agree, regardless of their correctness. Train-time calibration methods are frequently combined with label smoothing Szegedy et al. (2016) or focal loss Lin et al. (2017), neither of which is a proper scoring rule since population minimizers generally differ from the true conditional distribution. Replacing cross-entropy with either would therefore remove the strictly proper scoring rule that anchors the supervised term to that distribution. The smoothing penalty, which does not itself identify the correct posterior, would then be susceptible to propagating a shared bias. We therefore retain a strictly proper task loss.

4.3 Implementation

Training uses minibatch updates with detached neighbor targets: current predictions for neighbors in the minibatch and cached predictions otherwise. Inference is unchanged. Algorithm 1 and the full implementation and complexity analysis appear in Appendix C, while measured costs are reported in Appendix E.10.

5 Experimental Setup

We compare NeSCal with train-time calibration methods across image, text, and tabular classification, evaluating each trained model both directly and after post-hoc recalibration. This section states the protocol; full implementation details are deferred to Appendix D.

5.1 Datasets and Architectures

Image. CIFAR-10 and CIFAR-100 (Krizhevsky & Hinton, 2009) with ResNet-50 (He et al., 2016) and WideResNet-28-10 (Zagoruyko & Komodakis, 2016), and Tiny-ImageNet (Le & Yang, 2015) with ResNet-50.

Text. 20 Newsgroups (Lang, 1995) (20NG) with the global-pooling CNN (GPCNN) (Lin et al., 2013), and Amazon Reviews (Hou et al., 2024) (binary and five-class) with a multilayer perceptron (MLP) head on fine-tuned and then frozen RoBERTa-base (Liu et al., 2019) representations. In the frozen-encoder setting, NeSCal regularizes only the head with the encoder representation remaining fixed during training.

Tabular. Helena, Jannis (Guyon et al., 2019) and a subsample of Covertype (Blackard, 1998), with an MLP classifier.

5.2 Compared Methods

We compare with CE and six train-time calibration methods: focal loss (FL) (Mukhoti et al., 2020; Lin et al., 2017), label smoothing (LS) (Müller et al., 2019; Szegedy et al., 2016), and margin-based label smoothing (MbLS) (Liu et al., 2022), which modify the training objective; and MMCE (Kumar et al., 2018), MDCA (Hebbalaguppe et al., 2022), and s-AvUC (Karandikar et al., 2021), which introduce explicit calibration-related penalties. Every method is tuned under the protocol below. The main comparison uses NeSCal with cosine decay; the constant schedule is examined separately.

To assess composition, we apply five post-hoc calibrators to every train-time method. Temperature scaling (TS) (Guo et al., 2017) and Dirichlet calibration (Kull et al., 2019) provide established output-based adjustments, ranging from a single temperature to a multiclass calibration map. LoCal Nets (LoCal) (Barbera et al., 2026), LoRe (Luo et al., 2022), and DAC (Tomani et al., 2023)

use local or density information in representation space. Together, they test whether the gains from training persist under both global output recalibration and recalibration informed by representations.

5.3 Model Selection and Evaluation

All methods are trained for a fixed epoch budget, except the frozen-encoder heads, which use early stopping. Checkpoints and hyperparameters are selected by validation NLL, a strictly proper scoring rule for the full predictive distribution (Gneiting & Raftery, 2007). ECE measures agreement between confidence and empirical correctness after aggregation and can therefore be low when predictions are uninformative. This distinction matters when selecting regularization strength: strong smoothing can reduce ECE while degrading accuracy and NLL. Selection by a proper score remains sensitive to this loss of predictive information, although it can select a configuration with higher ECE. The shared criterion gives every method a common selection target, although this may differ from its original protocol. Appendix E.9 examines the resulting trade-offs.

The primary evaluation metric is also NLL, for the reasons stated above. We additionally report the multiclass Brier score (Brier, 1950), accuracy and ECE, alongside secondary calibration metrics and the calibration–refinement decomposition (Kull & Flach, 2015) in Appendix D.6. We assess consistency across settings rather than significance within them (Appendix E.2).

6 Experimental Results

We address three questions: whether NeSCal improves calibration and predictive quality relative to CE and the competing train-time methods (Section 6.1); whether these gains persist after post-hoc recalibration (Section 6.2); and whether they depend on the neighborhood structure the theory relies on, and in which settings they arise (Sections 6.3 and 6.4). Tables report test-set means, with standard deviations where shown. Unless a caption states otherwise, bold and italic identify the best and second-best distinct means, with ties at displayed precision sharing emphasis. NeSCal denotes the cosine schedule on real-data benchmarks; synthetic results use the constant schedule. R50 and WRN denote ResNet-50 and WideResNet-28-10, respectively.

6.1 Predictive quality and calibration

NeSCal improves NLL over CE in all eleven main benchmark settings and Brier score in ten (Table 1; Appendix Table 6). These gains generally accompany improved accuracy, which increases in nine settings, including seven of the eight image and tabular settings .

Against the other train-time methods, NeSCal ranks first or second (including ties) on NLL in seven of the eight image and tabular settings. Because NLL rewards both calibration and accuracy, we assess calibration improvement by asking whether any method is better on both at once. A method is Pareto-optimal if no other method is at least as accurate and has no lower ECE, with strict improvement in at least one. NeSCal is Pareto-optimal in five of the eight image and tabular settings both before and after TS. Base-model ECE is where NeSCal is least competitive; Section 6.2 shows that this ranking changes once confidence scale is corrected. On the text benchmarks, NeSCal's improvements over CE are smaller and do not yield the lowest NLL against other methods. Section 6.4 characterizes this variation across settings.

6.2 Composition Under Post-hoc Recalibration

An important assessment of a train-time calibration method is whether its gains survive the cheapest post-hoc correction. After the same validation-fitted temperature scaling is applied to every trained model, NeSCal retains lower NLL than CE in all eleven settings and has the lowest NLL among the evaluated methods in seven of the eight image and tabular settings (Table 1). On CIFAR-10 and CIFAR-100 with ResNet-50, its NLL is respectively 6.3% and 6.0% below the strongest evaluated baseline after TS. TS compresses the spread of base-model ECE across methods on CIFAR from 2.2–6.3 to 0.5–1.1 percentage points, after which NeSCal ranks first or second in five of the eight image and tabular settings (Table 7). Much of the base-model ECE advantage of methods that lower confidence directly is therefore recoverable by a single temperature, whereas NeSCal retains its NLL advantage after recalibration.

Table 1: Test NLL before and after validation-fitted temperature scaling; lower is better. Values are means ± SD over seeds, with seed counts specified in Appendix D.

Setting	CE	FL	LS	MMCE	MDCA	s-AvUC	MbLS	NeSCal
Base model								
CIFAR-10 / R50	0.2065 ± 0.0091	**0.1821 ± 0.0045**	0.2285 ± 0.0064	0.2031 ± 0.0043	0.2089 ± 0.0074	0.2065 ± 0.0052	0.2227 ± 0.0052	*0.1843 ± 0.0058*
CIFAR-10 / WRN	0.1972 ± 0.0077	**0.1572 ± 0.0033**	0.1838 ± 0.0034	0.1791 ± 0.0083	0.1951 ± 0.0071	0.2090 ± 0.0102	0.1791 ± 0.0025	*0.1752 ± 0.0028*
CIFAR-100 / R50	0.8782 ± 0.0249	*0.8463 ± 0.0044*	0.8966 ± 0.0186	0.8935 ± 0.0393	0.8889 ± 0.0139	0.8763 ± 0.0170	0.8614 ± 0.0174	**0.8026 ± 0.0155**
CIFAR-100 / WRN	0.8785 ± 0.0049	**0.7681 ± 0.0058**	0.8388 ± 0.0086	0.8838 ± 0.0219	0.8786 ± 0.0139	0.8691 ± 0.0043	0.8206 ± 0.0117	*0.7760 ± 0.0173*
Tiny-ImageNet	1.4704 ± 0.0254	1.4960 ± 0.0101	1.4816 ± 0.0254	1.4797 ± 0.0344	1.4406 ± 0.0225	1.4403 ± 0.0317	*1.4287 ± 0.0251*	**1.4126 ± 0.0254**
Helena	2.5651 ± 0.0034	2.5909 ± 0.0032	*2.5504 ± 0.0075*	2.5648 ± 0.0030	2.5679 ± 0.0036	2.5653 ± 0.0017	**2.5500 ± 0.0030**	2.5534 ± 0.0042
Jannis	0.7125 ± 0.0001	0.7955 ± 0.0055	0.7168 ± 0.0038	0.7127 ± 0.0029	*0.7121 ± 0.0019*	0.7141 ± 0.0033	0.7125 ± 0.0026	**0.7066 ± 0.0014**
Covertype	0.2516 ± 0.0033	0.3647 ± 0.0052	0.2771 ± 0.0035	0.2505 ± 0.0016	0.2527 ± 0.0021	0.2507 ± 0.0043	**0.2489 ± 0.0019**	*0.2499 ± 0.0015*
20NG	0.8127 ± 0.0496	0.7854 ± 0.0257	*0.6405 ± 0.0214*	0.7648 ± 0.0251	0.7799 ± 0.0188	0.7789 ± 0.0195	**0.6313 ± 0.0157**	0.8023 ± 0.0369
Amazon (5-class)	0.7809 ± 0.0029	0.7948 ± 0.0006	*0.7779 ± 0.0027*	**0.7725 ± 0.0028**	0.7799 ± 0.0033	0.7808 ± 0.0021	0.7782 ± 0.0010	0.7781 ± 0.0032
Amazon (binary)	0.2254 ± 0.0012	0.3023 ± 0.0091	0.2234 ± 0.0014	**0.2025 ± 0.0008**	0.2255 ± 0.0053	0.2241 ± 0.0020	*0.2233 ± 0.0015*	0.2240 ± 0.0020
After temperature scaling								
CIFAR-10 / R50	0.1863 ± 0.0084	*0.1813 ± 0.0044*	0.2191 ± 0.0074	0.2027 ± 0.0045	0.1858 ± 0.0144	0.1857 ± 0.0099	0.2206 ± 0.0059	**0.1698 ± 0.0046**
CIFAR-10 / WRN	0.1575 ± 0.0083	*0.1568 ± 0.0033*	0.1832 ± 0.0031	0.1776 ± 0.0077	0.1678 ± 0.0104	0.1640 ± 0.0105	0.1710 ± 0.0075	**0.1566 ± 0.0052**
CIFAR-100 / R50	0.8489 ± 0.0267	*0.8452 ± 0.0038*	0.8951 ± 0.0188	0.8631 ± 0.0402	0.8577 ± 0.0137	0.8478 ± 0.0137	0.8558 ± 0.0168	**0.7944 ± 0.0191**
CIFAR-100 / WRN	0.8070 ± 0.0061	*0.7681 ± 0.0057*	0.8374 ± 0.0076	0.8309 ± 0.0151	0.8141 ± 0.0155	0.8212 ± 0.0099	0.8173 ± 0.0103	**0.7613 ± 0.0178**
Tiny-ImageNet	1.4415 ± 0.0246	1.4950 ± 0.0099	1.4563 ± 0.0255	1.4614 ± 0.0319	*1.4096 ± 0.0228*	**1.4066 ± 0.0307**	1.4208 ± 0.0246	1.4098 ± 0.0255
Helena	2.5618 ± 0.0015	2.5907 ± 0.0035	2.5503 ± 0.0072	2.5615 ± 0.0036	2.5653 ± 0.0025	2.5624 ± 0.0027	*2.5498 ± 0.0033*	**2.5497 ± 0.0049**
Jannis	0.7115 ± 0.0008	0.7582 ± 0.0097	0.7160 ± 0.0039	0.7111 ± 0.0025	*0.7104 ± 0.0011*	0.7106 ± 0.0014	0.7121 ± 0.0027	**0.7042 ± 0.0011**
Covertype	0.2478 ± 0.0028	0.2930 ± 0.0026	0.2589 ± 0.0037	0.2463 ± 0.0011	0.2490 ± 0.0015	*0.2453 ± 0.0033*	0.2485 ± 0.0021	**0.2449 ± 0.0012**
20NG	0.7311 ± 0.0274	0.7073 ± 0.0292	*0.6361 ± 0.0235*	0.6971 ± 0.0200	0.7339 ± 0.0374	0.7209 ± 0.0281	**0.6320 ± 0.0158**	0.7290 ± 0.0231
Amazon (5-class)	0.7676 ± 0.0044	0.7694 ± 0.0012	0.7779 ± 0.0020	**0.7579 ± 0.0042**	*0.7593 ± 0.0037*	0.7678 ± 0.0066	0.7652 ± 0.0037	0.7625 ± 0.0024
Amazon (binary)	0.2207 ± 0.0061	0.2113 ± 0.0071	0.2205 ± 0.0017	**0.2024 ± 0.0008**	0.2235 ± 0.0038	0.2237 ± 0.0017	0.2214 ± 0.0029	*0.2102 ± 0.0019*

This pattern is consistent with the distinction motivated by Proposition 3: neighborhood smoothing can reduce variation in neighboring prediction errors, while temperature scaling adjusts confidence globally. On the two CIFAR settings with ResNet-50, NeSCal reduces the reported proxy for neighboring-error dispersion and the across-seed spread of individual confidences (Appendix E.11). Failure-prediction AUROC and AURC also improve over cross-entropy in all five image settings (Appendix E.5). These observations support a change in predictive quality beyond confidence scale, without requiring the two interventions to act on strictly independent error components.

The broader composition analysis supports the same conclusion: NeSCal attains the lowest best-observed post-hoc NLL in six of eight tabular and image settings when each training method is evaluated with all five post-hoc calibrators (Appendix E.4).

6.3 Design choices, cost and robustness

If the NeSCal's gains came from indiscriminate smoothing, a random graph of the same degree would reproduce them. It does not: kNN graphs yield lower NLL than random graphs in all three evaluated controls (Appendix E.7, Table 14). This is consistent with the role of the correctness gap $|a_i - a_j|$ in Section 3: random edges connect points whose correctness probabilities differ, where enforcing agreement conflicts with calibration.

The validation-selected cosine configurations improve NLL, Brier score, and accuracy over constant regularization in all four CIFAR settings (Table 15). Yet constant regularization yields lower final edge JS on both evaluated ResNet-50 graphs (Table 19): stronger final agreement does not necessarily imply better predictive quality. Sensitivity to neighborhood size is smaller than the NLL gain over CE in six of eight ablated settings.

No divergence uniformly dominates in the stable, scale-matched comparisons and activation rules do not transfer consistently across settings, so activation epoch and regularization strength remain validation-selected hyperparameters (Appendices E.7 and E.8).

The regularizer is inexpensive per run: on CIFAR, the reported training-time increase is below 1%, with one-time graph construction taking 17–27 seconds per run. Inference time is unchanged (Appendix E.10).

Finally, we test whether the gains survive shift and perturbation. Under input corruption, NeSCal improves NLL and ECE over CE in four of five image settings, with higher accuracy in all five. On those same 4, it leads all other methods in accuracy. Under label noise, ECE improves in all four

evaluated conditions, while NLL improves in only one. The gains therefore depend on the evaluation conditions and metric (Appendices E.6 and E.12).

6.4 When does NeSCal help?

Proposition 3 motivates two requirements for useful smoothing: neighboring errors must differ enough for averaging to help, and the connected points must have sufficiently similar correctness probabilities. Table 2 describes these conditions through mean edge JS and edge agreement ($|E|^{-1}\sum_{(i,j)\in E}\mathbf{1}\{\hat{y}_i=\hat{y}_j\}$), measured at activation before regularized training. These diagnostics characterize the available disagreement and neighborhood coherence, although neither directly measures the true correctness gap. The small gains on Covertype, Jannis, and the two Amazon settings are consistent with limited disagreement to remove: their mean edge JS is only 0.003–0.013, and their relative NLL improvements are 0.4–0.8%. Tiny-ImageNet presents a different limitation. Its high disagreement occurs on the least coherent graph, increasing the risk that smoothing suppresses genuine differences between neighbors as well as estimation error. Its relative NLL gain is 3.9%, compared with 8.6–11.7% across the four CIFAR settings.

Helena and 20NG show why graph diagnostics alone are insufficient. Although their numbers overlap those of the CIFAR settings, their gains are much smaller. A plausible additional factor is how the model responds to the consistency constraint. The tabular MLPs learn representations, but through much shallower networks than the image ResNets. The 20NG model learns convolutional features over frozen word embeddings, while the Amazon models retain a frozen sentence encoder. These architectures offer different scopes for adapting features under the regularizer, including learning rate schedules. If neighboring predictions share errors rooted in the available features, making them agree will not correct those errors; the constraint would be satisfied by changing output probabilities. The observed gains are consistent with this limitation: they are 3.9–11.7% in the image settings and 0.4–1.3% in the tabular and text settings. Local consistency is therefore most useful when disagreement is both removable and supported by an informative, learnable representation.

Table 2: Mean edge JS and Edge Agreement at activation and improvement in test NLL over CE, sorted by edge JS. Positive ΔNLL indicates improvement; the relative column is as a fraction of CE's NLL.

Setting	Edge JS	Edge agreement	ΔNLL	Relative
Covertype	0.003	0.958	0.0017	0.7%
Amazon (binary)	0.004	0.987	0.0014	0.6%
Jannis	0.009	0.935	0.0059	0.8%
Amazon (5-class)	0.013	0.909	0.0028	0.4%
CIFAR-10 / WRN	0.026	0.901	0.0220	11.2%
20NG	0.027	0.748	0.0104	1.3%
CIFAR-10 / R50	0.029	0.775	0.0222	10.8%
Helena	0.078	0.678	0.0117	0.5%
CIFAR-100 / R50	0.188	0.677	0.0756	8.6%
CIFAR-100 / WRN	0.202	0.673	0.1025	11.7%
Tiny-ImageNet	0.254	0.535	0.0578	3.9%

7 Related Work

Train-time calibration. Train-time approaches encourage calibration either explicitly or implicitly, through some modification to training. Explicit objectives include accuracy-uncertainty alignment (Krishnan & Tickoo, 2020), matching average class probabilities to empirical frequencies (Hebbalaguppe et al., 2022), and penalizing confidence-accuracy discrepancies through soft binning (Karandikar et al., 2021) or a Huber loss (Patra et al., 2023). Other approaches construct differentiable calibration objectives using kernel methods (Kumar et al., 2018; Popordanoska et al., 2022). Distribution matching (Marx et al., 2023) provides a flexible calibration objective whose conditioning kernels can explicitly target local calibration. Implicit approaches improve calibration without directly optimizing a calibration discrepancy, including FL (Lin et al., 2017; Mukhoti et al., 2020), LS (Szegedy et al., 2016; Müller et al., 2019), and MbLS (Liu et al., 2022). SVLS (Islam &

Glocker, 2021) constructs soft targets from neighboring ground-truth pixel labels and NACL (Murugesan et al., 2025) penalizes logit deviations from a neighborhood-derived prior; both utilize local pixel structure specifically in image segmentation tasks. Mixup (Zhang et al., 2018) is also known to improve calibration (Thulasidasan et al., 2019), but it is an augmentation protocol rather than an objective: it synthesizes inputs and interpolates labels between randomly paired samples. In comparison, NeSCal uses representation-space neighborhoods to regularize agreement between full predictive distributions, with no label access required by the smoothing penalty.

Local post-hoc calibration. Several post-hoc methods exploit locality to recalibrate pretrained models. LoRe (Luo et al., 2022) estimates correctness through kernel-weighted local accuracy, while KCal (Lin et al., 2023) constructs class probabilities using kernel density estimates in a learned metric space and DAC (Tomani et al., 2023) uses density proxies from hidden representations to determine sample-dependent temperatures. LoCal (Barbera et al., 2026) learns a representation projection and new logits, aligning predictions with kernel-estimated local class frequencies using Jensen–Shannon distance. LoCal is the post-hoc method most closely related to NeSCal , but the objectives differ: LoCal matches predictions to local label-frequency estimates in a separate calibration stage, directly targeting multiclass local calibration, whereas NeSCal encourages agreement between neighboring predictive distributions during classifier training, connecting predictive smoothness to confidence variation and the propagation of pointwise confidence calibration.

Graph regularization. Our approach builds on graph-based manifold regularization, which encourages functions to vary smoothly over the data geometry (Belkin et al., 2006). Similarly, NCR (Iscen et al., 2022) regularizes using a KL penalty between the similarity-weighted average nearest neighbor predictions. This differs from NeSCal in both purpose and construction. We study representation-space graph smoothing specifically as a principle for calibration, deriving bounds connecting neighboring predictive divergence to confidence variation and the propagation of pointwise confidence error. NeSCal instantiates this principle through a train-time penalty between full predictive distributions on a representation-space neighborhood graph.

Our approach builds on graph-based manifold regularization, which encourages functions to vary smoothly over the data geometry (Belkin et al., 2006). Similarly, NCR (Iscen et al., 2022) it penalizes the KL divergence between each example's prediction and the similarity-weighted average of its neighbors' predictions, so that neighbors can override noisy labels. A mixture target can be met while individual neighbors disagree sharply; NeSCal instead penalizes every edge, as its bounds require. NCR also draws neighbors from the current minibatch, whereas NeSCal fixes a kNN graph over the full training set, and NCR targets accuracy under label noise rather than calibration.

8 Conclusion

We introduced graph smoothing for train-time calibration and instantiated it as NeSCal, a JS neighborhood regularizer with no inference-time cost. Our analysis bounds pointwise calibration-error propagation and identifies both an obstruction and a condition for improvement under idealized averaging. Across eleven benchmarks, NeSCal improves NLL over CE; after TS, it attains the lowest NLL among the evaluated train-time methods in seven of eight image and tabular settings. Three findings characterize the method's behavior: representation-based neighborhoods yield lower NLL than random neighborhoods; stronger final neighborhood agreement does not necessarily imply better predictive quality; and gains persist under a common temperature-scaling procedure. Together, these findings support neighborhood smoothing as a framework from train-time calibration and predictive improvement.

Future work should consider further development of the graph smoothing framework, and may focus on encouraging smoothing in the regimes where it helps, either through the penalty or through hyperparameters, and to suppressing it on harmful edges. It would also be worthwhile to investigate alternative graph-based penalties or underlying graph structures, and explore graph smoothing on benchmarks where the graph is inherent rather than induced.

REPRODUCIBILITY STATEMENT

We provide all materials necessary to reproduce and verify our results throughout the paper and appendices. Appendix D specifies data processing, splits, architectures, optimization, search spaces, selection, metrics, and hardware. Appendix C and Algorithm 1 specify NeSCal; Appendices A and B provide assumptions and proofs. Seed counts and split conventions, including exceptions, are documented in Appendix D. Run manifests identify selected configurations and reported aggregates. Reproduction code is available at `https://anonymous.4open.science/r/anon_nescal_paper_code/`.

DISCLOSURE OF GENERATIVE AI USE

In this work, generative AI tools were used for the following tasks:

- *Formulate mathematical claims*. Several claims were formalized, refined or partially developed using the assistance of AI.
- *Provide critical ingredients for proving mathematical claims* and *assist in the writing of proofs*. A number of proofs were sharpened, refined or partially developed using AI. All proofs were passed to AI for error checking, and AI feedback, corrections and suggested improvements of the proofs were considered and in some cases, adopted.
- *Propose or refine hypotheses*. A number of secondary hypotheses in the paper were proposed or refined by AI.
- *Design or provide feedback on research methodology or experiments* and *interpret results*. Sanity checks and direct feedback and critiques of methodology, experiments and results for the paper were provided by AI.
- *Implement methods*. One specific post-hoc calibration method was implemented by AI since the authors' official implementation is not publicly available.

Generative AI tools were not used to *help develop theoretical models or conceptual frameworks* or *generate synthetic datasets*, and *support qualitative and thematic data analysis*, *assist with translation* and *clean and reformat datasets* are not applicable to this work. Additionally, Generative AI tools were used to *create or edit software code*, *draft parts of a research paper*, *summarize or analyze existing literature*, *discover research topics or identify gaps*, *source information*, *brainstorm*, *edit a research paper to improve readability*, *identify relevant literature*, *propose a title or keywords for a research paper*. AI was used for the above tasks for sanity and fact checking and to improve the validity and interpretability of this work. AI was used to write code for the orchestration of multi-step experimental pipelines, find bugs and verify code validity, but not to implement the core methodology, training and evaluation protocols, barring the specific post-hoc method implemented.

The authors have thoroughly manually reviewed all AI-assisted outputs. The authors take full responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

S. Amari and H. Nagaoka. *Methods of Information Geometry*. Fields Institute Communications. American Mathematical Society, 2000. ISBN 9780821805312. URL `https://books.google.co.za/books?id=qGzCwAEACAAJ`.

Imanol Arrieta-Ibarra, Paman Gujral, Jonathan Tannen, Mark Tygert, and Cherie Xu. Metrics of calibration for probabilistic predictions. *J. Mach. Learn. Res.*, 23(1), January 2022. ISSN 1532-4435.

Cesare Barbera, Lorenzo Perini, Giovanni De Toni, Andrea Passerini, and Andrea Pugnana. Multiclass local calibration with the jensen-shannon distance. In *The 29th International Conference on Artificial Intelligence and Statistics*, 2026. URL `https://openreview.net/forum?id=atUXOpyCox`.

Mikhail Belkin, Partha Niyogi, and Vikas Sindhwani. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *Journal of Machine Learning Research*, 7(85):2399–2434, 2006. URL http://jmlr.org/papers/v7/belkin06a.html.

Jock Blackard. Covertype. UCI Machine Learning Repository, 1998. DOI: https://doi.org/10.24432/C50K5N.

Glenn W. Brier. Verification of forecasts expressed in terms of probability. *Monthly Weather Review*, 78(1):1 – 3, 1950. doi: 10.1175/1520-0493(1950)078⟨0001:VOFEIT⟩2.0.CO;2. URL https://journals.ametsoc.org/view/journals/mwre/78/1/1520-0493_1950_078_0001_vofeit_2_0_co_2.xml.

Jarosław Błasiok, Parikshit Gopalan, Lunjia Hu, and Preetum Nakkiran. A unifying theory of distance from calibration. In *Proceedings of the 55th Annual ACM Symposium on Theory of Computing*, STOC 2023, pp. 1727–1740, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 9781450399135. doi: 10.1145/3564246.3585182. URL https://doi.org/10.1145/3564246.3585182.

D.M. Endres and J.E. Schindelin. A new metric for probability distributions. *IEEE Transactions on Information Theory*, 49(7):1858–1860, 2003. doi: 10.1109/TIT.2003.813506.

B. Fuglede and F. Topsoe. Jensen-shannon divergence and hilbert space embedding. In *International Symposium onInformation Theory, 2004. ISIT 2004. Proceedings.*, pp. 31–, 2004. doi: 10.1109/ISIT.2004.1365067.

Yonatan Geifman, Guy Uziel, and Ran El-Yaniv. Bias-reduced uncertainty estimation for deep neural classifiers. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=SJfb5jCqKm.

Tilmann Gneiting and Adrian E Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 102(477):359–378, 2007. doi: 10.1198/016214506000001437. URL https://doi.org/10.1198/016214506000001437.

Yury Gorishniy, Ivan Rubachev, Valentin Khrulkov, and Artem Babenko. Revisiting deep learning models for tabular data. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 18932–18943. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/9d86d83f925f2149e9edb0ac3b49229c-Paper.pdf.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, pp. 1321–1330. JMLR.org, 2017.

Isabelle Guyon, Lisheng Sun-Hosoya, Marc Boullé, Hugo Jair Escalante, Sergio Escalera, Zhengying Liu, Damir Jajetic, Bisakha Ray, Mehreen Saeed, Michèle Sebag, Alexander Statnikov, Wei-Wei Tu, and Evelyne Viegas. Analysis of the automl challenge series 2015–2018. In Frank Hutter, Lars Kotthoff, and Joaquin Vanschoren (eds.), *Automated Machine Learning: Methods, Systems, Challenges*, pp. 177–219. Springer International Publishing, Cham, 2019. ISBN 978-3-030-05318-5. doi: 10.1007/978-3-030-05318-5_10. URL https://doi.org/10.1007/978-3-030-05318-5_10.

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 (CVPR)*, June 2016.

Ramya Hebbalaguppe, Jatin Prakash, Neelabh Madan, and Chetan Arora. A stitch in time saves nine: A train-time regularizing loss for improved neural network calibration. In *CVPR*, pp. 16060–16069, 2022. URL https://doi.org/10.1109/CVPR52688.2022.01561.

Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=HJz6tiCqYm.

Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=Hkg4TI9xl.

Yupeng Hou, Jiacheng Li, Zhankui He, An Yan, Xiusi Chen, and Julian McAuley. Bridging language and items for retrieval and recommendation. *arXiv preprint arXiv:2403.03952*, 2024.

Ahmet Iscen, Jack Valmadre, Anurag Arnab, and Cordelia Schmid. Learning with Neighbor Consistency for Noisy Labels . In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4662–4671, Los Alamitos, CA, USA, June 2022. IEEE Computer Society. doi: 10.1109/CVPR52688.2022.00463. URL https://doi.ieeecomputersociety.org/10.1109/CVPR52688.2022.00463.

Mobarakol Islam and Ben Glocker. Spatially varying label smoothing: Capturing uncertainty from expert annotations. In *Information Processing in Medical Imaging: 27th International Conference, IPMI 2021, Virtual Event, June 28–June 30, 2021, Proceedings*, pp. 677–688, Berlin, Heidelberg, 2021. Springer-Verlag. ISBN 978-3-030-78190-3. doi: 10.1007/978-3-030-78191-0_52. URL https://doi.org/10.1007/978-3-030-78191-0_52.

Archit Karandikar, Nicholas Cain, Dustin Tran, Balaji Lakshminarayanan, Jonathon Shlens, Michael C Mozer, and Becca Roelofs. Soft calibration objectives for neural networks. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 29768–29779. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/f8905bd3df64ace64a68e154ba72f24c-Paper.pdf.

Ranganath Krishnan and Omesh Tickoo. Improving model calibration with accuracy versus uncertainty optimization. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 18237–18248. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/d3d9446802a44259755d38e6d163e820-Paper.pdf.

Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical report, University of Toronto, Toronto, Ontario, 2009. URL https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf.

Meelis Kull and Peter Flach. Novel decompositions of proper scoring rules for classification: Score adjustment as precursor to calibration. In Annalisa Appice, Pedro Pereira Rodrigues, Vítor Santos Costa, Carlos Soares, João Gama, and Alípio Jorge (eds.), *Machine Learning and Knowledge Discovery in Databases*, pp. 68–85, Cham, 2015. Springer International Publishing. ISBN 978-3-319-23528-8.

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. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/8ca01ea920679a0fe3728441494041b9-Paper.pdf.

Aviral Kumar, Sunita Sarawagi, and Ujjwal Jain. Trainable calibration measures for neural networks from kernel mean embeddings. In Jennifer Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 2805–2814. PMLR, 10–15 Jul 2018. URL https://proceedings.mlr.press/v80/kumar18a.html.

Ken Lang. Newsweeder: Learning to filter netnews. In *Proceedings of the 12th International Conference on Machine Learning (ICML)*, pp. 331–339, 1995. URL http://qwone.com/~jason/20Newsgroups/. Retrived from source, which documents that the cited paper does not explicitly acknowledge collecting this dataset.

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

Min Lin, Qiang Chen, and Shuicheng Yan. Network in network. *arXiv preprint arXiv:1312.4400*, 2013.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollar. Focal loss for dense object detection. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

Zhen Lin, Shubhendu Trivedi, and Jimeng Sun. Taking a step back with KCal: Multi-class kernel-based calibration for deep neural networks. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=p_jIy5QFB7.

Bingyuan Liu, Ismail Ben Ayed, Adrian Galdran, and Jose Dolz. The devil is in the margin: Margin-based label smoothing for network calibration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 80–88, June 2022.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach, 2019. URL https://arxiv.org/abs/1907.11692.

Rachel Luo, Aadyot Bhatnagar, Yu Bai, Shengjia Zhao, Huan Wang, Caiming Xiong, Silvio Savarese, Stefano Ermon, Edward Schmerling, and Marco Pavone. Local calibration: metrics and recalibration. In James Cussens and Kun Zhang (eds.), *Proceedings of the Thirty-Eighth Conference on Uncertainty in Artificial Intelligence*, volume 180 of *Proceedings of Machine Learning Research*, pp. 1286–1295. PMLR, 01–05 Aug 2022. URL https://proceedings.mlr.press/v180/luo22a.html.

Charlie Marx, Sofian Zalouk, and Stefano Ermon. Calibration by distribution matching: Trainable kernel calibration metrics. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 25910–25928. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/52493d82db00e73abb2858a5a5f28717-Paper-Conference.pdf.

Jishnu Mukhoti, Viveka Kulharia, Amartya Sanyal, Stuart Golodetz, Philip Torr, and Puneet Dokania. Calibrating deep neural networks using focal loss. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 15288–15299. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/aeb7b30ef1d024a76f21a1d40e30c302-Paper.pdf.

Rafael Müller, Simon Kornblith, and Geoffrey E Hinton. When does label smoothing help? In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/f1748d6b0fd9d439f71450117eba2725-Paper.pdf.

Balamurali Murugesan, Sukesh Adiga Vasudeva, Bingyuan Liu, Herve Lombaert, Ismail Ben Ayed, and Jose Dolz. Neighbor-aware calibration of segmentation networks with penalty-based constraints. *Medical Image Analysis*, 101:103501, 2025. ISSN 1361-8415. doi: https://doi.org/10.1016/j.media.2025.103501. URL https://www.sciencedirect.com/science/article/pii/S1361841525000490.

Mahdi Pakdaman Naeini, Gregory F. Cooper, and Milos Hauskrecht. Obtaining well calibrated probabilities using bayesian binning. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, AAAI'15, pp. 2901–2907. AAAI Press, 2015. ISBN 0262511290.

Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Ng. Reading digits in natural images with unsupervised feature learning. *NIPS*, 01 2011.

Jeremy Nixon, Michael W. Dusenberry, Linchuan Zhang, Ghassen Jerfel, and Dustin Tran. Measuring calibration in deep learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2019.

Rishabh Patra, Ramya Hebbalaguppe, Tirtharaj Dash, Gautam Shroff, and Lovekesh Vig. Calibrating deep neural networks using explicit regularisation and dynamic data pruning. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 1541–1549, January 2023.

Jeffrey Pennington, Richard Socher, and Christopher Manning. GloVe: Global vectors for word representation. In Alessandro Moschitti, Bo Pang, and Walter Daelemans (eds.), *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1532–1543, Doha, Qatar, October 2014. Association for Computational Linguistics. doi: 10.3115/v1/D14-1162. URL https://aclanthology.org/D14-1162/.

John Platt. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Adv. Large Margin Classif.*, 10, 06 2000.

Teodora Popordanoska, Raphael Sayer, and Matthew Blaschko. A consistent and differentiable lp canonical calibration error estimator. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 7933–7946. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/33d6e648ee4fb24acec3a4bbcd4f001e-Paper-Conference.pdf.

Rebecca Roelofs, Nicholas Cain, Jonathon Shlens, and Michael C. Mozer. Mitigating bias in calibration error estimation. In Gustau Camps-Valls, Francisco J. R. Ruiz, and Isabel Valera (eds.), *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics*, volume 151 of *Proceedings of Machine Learning Research*, pp. 4036–4054. PMLR, 28–30 Mar 2022. URL https://proceedings.mlr.press/v151/roelofs22a.html.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

Sunil Thulasidasan, Gopinath Chennupati, Jeff A Bilmes, Tanmoy Bhattacharya, and Sarah Michalak. On mixup training: Improved calibration and predictive uncertainty for deep neural networks. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/36ad8b5f42db492827016448975cc22d-Paper.pdf.

Christian Tomani, Futa Waseda, Yuesong Shen, and Daniel Cremers. Beyond in-domain scenarios: robust density-aware calibration. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org, 2023.

Juozas Vaicenavicius, David Widmann, Carl Andersson, Fredrik Lindsten, Jacob Roll, and Thomas Schön. Evaluating model calibration in classification. In Kamalika Chaudhuri and Masashi Sugiyama (eds.), *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*, volume 89 of *Proceedings of Machine Learning Research*, pp. 3459–3467. PMLR, 16–18 Apr 2019. URL https://proceedings.mlr.press/v89/vaicenavicius19a.html.

David Widmann, Fredrik Lindsten, and Dave Zachariah. Calibration tests in multi-class classification: A unifying framework. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/1c336b8080f82bcc2cd2499b4c57261d-Paper.pdf.

Bianca Zadrozny and Charles Elkan. Transforming classifier scores into accurate multiclass probability estimates. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 08 2002. doi: 10.1145/775047.775151.

Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In Edwin R. Hancock Richard C. Wilson and William A. P. Smith (eds.), *Proceedings of the British Machine Vision Conference (BMVC)*, pp. 87.1–87.12. BMVA Press, September 2016. ISBN 1-901725-59-6. doi: 10.5244/C.30.87. URL https://dx.doi.org/10.5244/C.30.87.

Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=r1Ddp1-Rb.

Jize Zhang, Bhavya Kailkhura, and T. Yong-Jin Han. Mix-n-match : Ensemble and compositional methods for uncertainty calibration in deep learning. In Hal Daumé III and Aarti Singh (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 11117–11128. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/zhang20k.html.

A Further Theoretical Results and Proofs

A.1 Graph Laplacian Approximation

Definition 1 *Given an unweighted, symmetric adjacency matrix $W \in \{0,1\}^{n\times n}$, the graph Laplacian is $L := \mathrm{diag}(d_1,\dots,d_n) - W$, where $d_i := \sum_j W_{ij}$. For any function $u : V \to \mathbb{R}$ on the nodes, the Laplacian quadratic form*

$$u^\top L u = \frac{1}{2}\sum_{i,j} W_{ij}\,(u_i - u_j)^2 \tag{7}$$

measures the smoothness of u over the graph, growing as u varies more sharply between neighboring nodes.

Proposition 4 (Graph Laplacian Approximation) *Let $\mathrm{D} : \Delta^{K-1}\times\Delta^{K-1} \to \mathbb{R}_{\geq 0}$ satisfy $\mathrm{D}(p\|p) = 0$ and, for some $\varepsilon > 0$, let $\mathrm{D}(p\|\cdot)$ be three times continuously differentiable on a neighborhood of $\Delta_\varepsilon := \{u \in \Delta^{K-1} : \min_k u_k \geq \varepsilon\}$ in $\mathbb{R}^K$, with third derivatives jointly continuous in (p,q) on $\Delta_\varepsilon\times\Delta_\varepsilon$; write $M_3(\varepsilon) := \sup_{p,q\in\Delta_\varepsilon}\sup_{\|u\|_2=1}\left|\nabla_q^3\mathrm{D}(p\|q)[u,u,u]\right|$, which is finite by compactness. Suppose the predictive distributions obey the probability floor $\min_{i,k} p_{i,k} \geq \varepsilon$. Define the induced local form: $G_{\mathrm{D}}(p) := \frac{1}{2}\nabla_q^2\mathrm{D}(p\|q)\big|_{q=p}$. Then for any edge $(i,j) \in E$,*

$$\mathrm{D}(p_i\|p_j) = (p_i - p_j)^{\mathsf{T}} G_{\mathrm{D}}(p_i)(p_i - p_j) + \rho_{ij} \tag{8}$$

where $|\rho_{ij}| \leq \frac{1}{6}M_3(\varepsilon)\,\|p_i - p_j\|_2^3$. Consequently,

$$R_n(p) = \sum_{i,j} W_{ij}\,(p_i - p_j)^{\mathsf{T}} G_{\mathrm{D}}(p_i)\,(p_i - p_j) + \rho, \qquad |\rho| \leq \tfrac{1}{6}M_3(\varepsilon)\sum_{i,j} W_{ij}\,\|p_i - p_j\|_2^3. \tag{9}$$

Expanding D on each edge about the source prediction shows that $R_n(p)$ is, up to an explicitly bounded remainder, a weighted graph-Laplacian quadratic form, giving it a geometric interpretation as a smoothness penalty over some representation manifold $\mathcal{M} \subseteq \mathcal{Z}$. The result holds for any sufficiently regular divergence; the specific weighting is determined by the divergence's local curvature. The expansion is anchored at the source prediction p_i, weighting edge (i,j) by $G_{\mathrm{D}}(p_i)$. Since $p_i = f_\theta(x_i)$, minimizing R_n acts as a smoothness penalty on f_θ. Its magnitude depends on both the predictive distributions and the edge weights.

Proposition 4 expresses R_n as a weighted graph-Laplacian-like smoothness penalty and determines the weighting $G_{\mathrm{D}}(p_i)$, but no result below depends on it; the guarantees follow instead from Proposition 5. The quadratic form is an accurate approximation only where $|p_{i,k} - p_{j,k}| \ll p_{i,k}$; it is an interpretive device, not a bound we optimize.

Proof. Fix an edge (i,j) and write $p := p_i$, $q := p_j$, and $\delta := q - p$. Since $p, q \in \Delta^{K-1}$ are probability vectors, they satisfy $\sum_{k=1}^K p_k = \sum_{k=1}^K q_k = 1$, so $\mathbf{1}^\top\delta = \sum_k \delta_k = 0$. The expansion below is Taylor's theorem with Lagrange remainder applied to $t \mapsto \mathrm{D}(p\|p + t\delta)$; it is exact for any δ.

We take the multivariate Taylor expansion of $D(p\|q)$ about $q = p$ in its second argument:

$$\mathrm{D}(p\|q) = \mathrm{D}(p\|p) + \nabla_q\mathrm{D}(p\|q)\big|_{q=p}^\top\delta + \tfrac{1}{2}\delta^\top\nabla_q^2\mathrm{D}(p\|q)\big|_{q=p}\,\delta + \rho_{ij} \tag{10}$$

Zeroth order. By hypothesis $\mathrm{D}(p\|p) = 0$.

First order. For fixed p, $\mathrm{D}(p\|q)$ is nonnegative in its second argument and equals 0 at $q = p$, so $q = p$ minimizes $\mathrm{D}(p\|\cdot)$ over the simplex. Since p lies in the interior, the only binding constraint at this minimum is that probabilities sum to one, $\mathbf{1}^\top q = 1$. Stationarity then forces the gradient to be constant across coordinates, $\nabla_q\mathrm{D}(p\|q)\big|_{q=p} = \mu\mathbf{1}$ for some $\mu \in \mathbb{R}$. Because p_i and p_j are both probability vectors, their difference satisfies $\mathbf{1}^\top\delta = \sum_k \delta_k = 0$, and hence the first-order term vanishes:

$$\nabla_q\mathrm{D}(p\|q)\big|_{q=p}^\top\delta = \mu\mathbf{1}^\top\delta = 0.$$

Second order. By the definition of G_{D},

$$\frac{1}{2}\delta^\top \nabla_q^2 \mathrm{D}(p\|q)\big|_{q=p}\delta = \delta^\top G_{\mathrm{D}}(p)\delta = (p_i - p_j)^\top G_{\mathrm{D}}(p_i)(p_i - p_j), \tag{11}$$

with the last equality holding because the quadratic form is even in δ. Since $q = p$ minimizes $\mathrm{D}(p\|\cdot)$ over the simplex, the second-order necessary condition gives $\delta^\top G_{\mathrm{D}}(p)\delta \geq 0$ for every admissible difference δ (those with $\sum_k \delta_k = 0$); that is, $G_{\mathrm{D}}(p)$ is positive semidefinite on the tangent space of the simplex, though not necessarily positive definite as a matrix.

Remainder. By Taylor's theorem with Lagrange remainder, $\rho_{ij} = \frac{1}{6}\nabla_q^3 \mathrm{D}(p\|q)\big|_{q=\xi}[\delta,\delta,\delta]$ for some ξ on the segment between p_i and p_j. Both endpoints lie in Δ_ε, which is convex, so $\xi \in \Delta_\varepsilon$ and $|\rho_{ij}| \leq \frac{1}{6}M_3(\varepsilon)\|\delta\|_2^3$. For log-based divergences the third derivatives grow as a coordinate approaches zero, which is why the floor is required. Summing over edges with $W_{ij} \geq 0$ gives the bound on ρ. □

A.2 Predictive Smoothness

Definition 2 (Quadratic total-variation domination) *For $p, q \in \Delta^{K-1}$, let $\mathrm{TV}(p,q) := \frac{1}{2}\|p - q\|_1$ denote the total variation distance. A divergence* D *on* Δ^{K-1} quadratically dominates total variation *with constant $C_{\mathrm{D}} > 0$ if*

$$\mathrm{D}(p\|q) \;\geq\; 4\,C_{\mathrm{D}}\,\mathrm{TV}(p,q)^2 \qquad \forall\, p, q \in \Delta^{K-1}, \tag{12}$$

equivalently $\mathrm{D}(p\|q) \geq C_{\mathrm{D}}\|p - q\|_1^2$.

This is a Pinsker-type condition requiring D to grow at least quadratically in total variation, which rules out divergences that vanish faster than $\|p - q\|_1^2$ near the diagonal.

Lemma 1 (Predictive-distribution smoothness) *Let* D *dominate squared total variation with constant C_{D} according to Definition 2. Then, for every edge $(i,j) \in E$,*

$$\|p_i - p_j\|_1 \leq \frac{1}{\sqrt{C_{\mathrm{D}}}}\sqrt{\mathrm{D}(p_i\|p_j)} \tag{13}$$

Proof. By Definition 2, $\mathrm{D}(p_i\|p_j) \geq C_{\mathrm{D}}\|p_i - p_j\|_1^2$. Since both sides are non-negative, taking square roots and rearranging gives

$$\|p_i - p_j\|_1 \leq \frac{1}{\sqrt{C_{\mathrm{D}}}}\sqrt{\mathrm{D}(p_i\|p_j)} \tag{14}$$

□

A.3 Confidence Smoothness

Proposition 5 (Confidence Smoothness) *Under the conditions of Lemma 1, for every edge $(i,j) \in E$,*

$$|c_i - c_j| \leq \tfrac{1}{2}\|p_i - p_j\|_1 \leq \frac{1}{2\sqrt{C_{\mathrm{D}}}}\sqrt{\mathrm{D}(p_i\|p_j)}. \tag{15}$$

Proof. The maximum operator is 1-Lipschitz with respect to the supremum norm. Let $k_i \in \arg\max_k p_{i,k}$. Then

$$c_i - c_j = p_{i,k_i} - \max_k p_{j,k} \leq p_{i,k_i} - p_{j,k_i} \leq \|p_i - p_j\|_\infty. \tag{16}$$

Exchanging the roles of i and j yields $c_j - c_i \leq \|p_j - p_i\|_\infty$, and hence

$$|c_i - c_j| \leq \|p_i - p_j\|_\infty \tag{17}$$

Since $p_i, p_j \in \Delta^{K-1}$, we have

$$\sum_{k=1}^K (p_{i,k} - p_{j,k}) = \sum_{k=1}^K p_{i,k} - \sum_{k=1}^K p_{j,k} = 0. \tag{18}$$

Let $k^\star \in \arg\max_k |p_{i,k} - p_{j,k}|$. Then $\sum_{k \neq k^\star}(p_{i,k} - p_{j,k}) = -(p_{i,k^\star} - p_{j,k^\star})$ and we have:

$$\sum_{k \neq k^\star} |p_{i,k} - p_{j,k}| \geq \left| \sum_{k \neq k^\star} (p_{i,k} - p_{j,k}) \right| = |p_{i,k^\star} - p_{j,k^\star}|$$

$$\implies \|p_i - p_j\|_1 = |p_{i,k^\star} - p_{j,k^\star}| + \sum_{k \neq k^\star} |p_{i,k} - p_{j,k}|$$

$$\geq 2|p_{i,k^\star} - p_{j,k^\star}| = 2\|p_i - p_j\|_\infty.$$

and thus

$$\|p_i - p_j\|_\infty \leq \frac{1}{2}\|p_i - p_j\|_1 \tag{19}$$

Combining these inequalities with Lemma 1 gives

$$|c_i - c_j| \leq \|p_i - p_j\|_\infty \leq \frac{1}{2}\|p_i - p_j\|_1 \leq \frac{1}{2\sqrt{C_{\mathrm{D}}}}\sqrt{\mathrm{D}(p_i\|p_j)}, \tag{20}$$

□

A.4 Local Disagreement Bound

Proposition 6 (Local Disagreement Bound) *Let $\mathcal{L}(\theta) = \mathcal{L}_{\text{task}}(\theta) + \lambda R_n(p)$ with $\mathcal{L}_{\text{task}} \geq 0$, and sample an edge (i, j) with probability $W_{ij}/|E|$, where $|E| = \sum_{i,j} W_{ij}$. Then for any $t > 0$,*

$$\Pr\left(\mathrm{D}(p_i\|p_j) \geq t\right) \leq \frac{R_n(p)}{|E|\, t} \leq \frac{\mathcal{L}(\theta)}{\lambda\, |E|\, t} \tag{21}$$

Proof. Since $\mathcal{L}_{\text{task}}(\theta) \geq 0$, rearranging $\mathcal{L}(\theta) = \mathcal{L}_{\text{task}}(\theta) + \lambda R_n(p)$ gives

$$R_n(p) = \frac{\mathcal{L}(\theta) - \mathcal{L}_{\text{task}}(\theta)}{\lambda} \;\leq\; \frac{\mathcal{L}(\theta)}{\lambda}. \tag{22}$$

Let $X := \mathrm{D}(p_i\|p_j)$ for an edge sampled with probability $W_{ij}/|E|$. Since $\mathrm{D} \geq 0$, X is a nonnegative random variable with

$$\mathbb{E}[X] = \frac{1}{|E|}\sum_{i,j} W_{ij}\, \mathrm{D}(p_i\|p_j) = \frac{R_n(p)}{|E|} \;\leq\; \frac{\mathcal{L}(\theta)}{\lambda\, |E|}. \tag{23}$$

Markov's inequality then gives, for any $t > 0$,

$$\Pr(X \geq t) \leq \frac{\mathbb{E}[X]}{t} = \frac{R_n(p)}{|E|t} \leq \frac{\mathcal{L}(\theta)}{\lambda|E|t} \tag{24}$$

□

A.5 Probabilistic Confidence Smoothness Bound

Corollary 1 (Probabilistic Confidence Smoothness) *Under the conditions of Proposition 5 and Proposition 6, for any $t > 0$, with probability at least $1 - \frac{\mathcal{L}(\theta)}{\lambda|E|t}$ over a sampled edge (i, j),*

$$|c_i - c_j| \;\leq\; \frac{1}{2}\sqrt{\frac{t}{C_{\mathrm{D}}}} \tag{25}$$

Proof. By Proposition 6, the event $\{\mathrm{D}(p_i\|p_j) < t\}$ holds with probability at least $1 - \mathcal{L}(\theta)/(\lambda|E|t)$. On this event, Proposition 5 gives

$$|c_i - c_j| \leq \frac{1}{2\sqrt{C_{\mathrm{D}}}}\sqrt{\mathrm{D}(p_i\|p_j)} < \frac{1}{2}\sqrt{t/C_{\mathrm{D}}} \tag{26}$$

□

A.6 Proof of Proposition 1

By the triangle inequality,

$$|a_j - c_j| \;\le\; |a_j - a_i| + |a_i - c_i| + |c_i - c_j|. \tag{27}$$

The second term is $\mathcal{E}_{\mathrm{cal}_i}$ and the third is bounded by Proposition 5. □

A.7 Proof of Theorem 1

By the triangle inequality applied at the endpoints of P,

$$\mathcal{E}_{\mathrm{cal}_v} = |a_v - c_v| \le |a_v - a_s| + |a_s - c_s| + |c_s - c_v| = \mathcal{E}_{\mathrm{cal}_s} + |c_s - c_v| + |a_v - a_s| \tag{28}$$

Telescoping the confidence term along P and applying Proposition 5 to each edge,

$$|c_s - c_v| \;\le\; \sum_{\ell=1}^{r} \big|c_{v_{\ell-1}} - c_{v_\ell}\big| \;\le\; \frac{1}{2\sqrt{C_{\mathrm{D}}}} \sum_{\ell=1}^{r} \sqrt{D_\ell}. \tag{29}$$

By Cauchy–Schwarz, $\sum_{\ell=1}^{r} \sqrt{D_\ell} \;\le\; \sqrt{r \sum_{\ell=1}^{r} D_\ell}$, so $|c_s - c_v| \;\le\; \sqrt{\frac{r}{4C_{\mathrm{D}}} \sum_\ell D_\ell}$. Substituting yields the stated bound. □

A.8 Proof of Proposition 2

By the triangle inequality,

$$|a_i - a_j| \;\le\; |a_i - c_i| + |c_i - c_j| + |c_j - a_j| = \mathcal{E}_{\mathrm{cal}_i} + |c_i - c_j| + \mathcal{E}_{\mathrm{cal}_j} \tag{30}$$

Rearranging gives

$$\mathcal{E}_{\mathrm{cal}_i} + \mathcal{E}_{\mathrm{cal}_j} \ge |a_i - a_j| - |c_i - c_j|. \tag{31}$$

Substituting $|c_i - c_j| \le \frac{1}{2\sqrt{C_{\mathrm{D}}}} \sqrt{\mathrm{D}(p_i \| p_j)}$ from Proposition 5 yields equation 4. □

Perfect calibration at both endpoints forces $|c_i - c_j| = |a_i - a_j|$ and hence, by Proposition 5, $\mathrm{D}(p_i \| p_j) \ge 4C_{\mathrm{D}}(a_i - a_j)^2$: no predictor with divergence below this level can be perfectly calibrated at both endpoints unless $a_i = a_j$.

A.9 Proposition 3: Bias–Variance Decomposition and Proof

Under the fixed-class assumption of Proposition 3, a_i and a_j are fixed under $\mathcal{T}$. Decompose

$$c_i = a_i + b_i + \varepsilon_i, \qquad b_i := \mathbb{E}_{\mathcal{T}}[c_i] - a_i, \qquad \mathbb{E}_{\mathcal{T}}[\varepsilon_i] = 0, \tag{32}$$

and analogously at j, so that the expected squared pointwise calibration error decomposes as $\mathbb{E}_{\mathcal{T}}[(c_i - a_i)^2] = b_i^2 + \mathrm{Var}(\varepsilon_i)$ into a systematic and a random part. Hence the right-hand side of equation 5 is

$$\mathbb{E}_{\mathcal{T}}\big[(e_i - e_j)^2\big] = (b_i - b_j)^2 + \mathrm{Var}(\varepsilon_i - \varepsilon_j). \tag{33}$$

In particular, if $b_i = b_j$, $\mathrm{Var}(\varepsilon_i) = \mathrm{Var}(\varepsilon_j) = \sigma^2$ and $\mathrm{Corr}(\varepsilon_i, \varepsilon_j) = \rho$, the condition is $|a_i - a_j| < \sigma\sqrt{2(1-\rho)}$.

Proof. Write $\Delta := a_i - a_j$ and $\bar{c} := \frac{1}{2}(c_i + c_j)$, with $c_i = p_{i,k^*}$ and $c_j = p_{j,k^*}$. For the mixture $m = \frac{1}{2}(p_i + p_j)$ we have $m_{k^*} = \bar{c}$, while for every coordinate k, $m_k = \frac{1}{2}(p_{i,k} + p_{j,k}) \le \bar{c}$. Hence $\max_k m_k = \bar{c}$, the predicted class is preserved, and a_i, a_j are unchanged. Before averaging, the summed expected squared calibration error is $\mathbb{E}[e_i^2] + \mathbb{E}[e_j^2]$. After, both endpoints receive $\bar{c}$, with errors $\bar{c} - a_i = \frac{1}{2}(e_i + e_j) - \frac{1}{2}\Delta$ and $\bar{c} - a_j = \frac{1}{2}(e_i + e_j) + \frac{1}{2}\Delta$, whose squares sum to $\frac{1}{2}(e_i + e_j)^2 + \frac{1}{2}\Delta^2$. Taking expectations and subtracting, the change is

$$\tfrac{1}{2}\Delta^2 + \tfrac{1}{2}\mathbb{E}[(e_i + e_j)^2] - \mathbb{E}[e_i^2] - \mathbb{E}[e_j^2] = \tfrac{1}{2}\Delta^2 - \tfrac{1}{2}\mathbb{E}[(e_i - e_j)^2], \tag{34}$$

strictly negative if and only if $\Delta^2 < \mathbb{E}[(e_i - e_j)^2]$. Since $\mathbb{E}[\varepsilon_i] = \mathbb{E}[\varepsilon_j] = 0$, $\mathbb{E}[(e_i - e_j)^2] = (b_i - b_j)^2 + \mathrm{Var}(\varepsilon_i - \varepsilon_j)$, which equals $2\sigma^2(1-\rho)$ under the symmetric assumptions. □

Interpretation. For fixed predicted labels, Proposition 2 limits how small the summed edge calibration error can be at a given predictive divergence. Under Proposition 3's assumptions, pairwise averaging removes $\frac{1}{2}\mathbb{E}[(e_i - e_j)^2]$ from summed expected squared error while adding $(a_i - a_j)^2/2$ due to the correctness gap. Improvement occurs exactly when the former exceeds the latter.

Under the equal-bias, equal-variance specialization, the tolerable correctness gap scales with $\sqrt{1-\rho}$ and vanishes at $\rho = 1$, so predictions that err together cannot correct one another. Since neighboring predictions from a single network are plausibly positively correlated, $\sqrt{2}\sigma$ is a ceiling rather than an estimate of the operative threshold, and whatever benefit smoothing confers is a reduction of within-neighborhood error dispersion, not a correction of the neighborhood's mean error.

The threshold depends on the training stage through $\mathbb{E}_{\mathcal{T}}[(e_i - e_j)^2]$, which includes systematic differences between neighboring errors as well as random variation. If this quantity decreases, the tolerable correctness gap narrows, motivating weaker smoothing later in training.

B NeSCal-Specific Guarantees

Here we instantiate the results of Section 3.2. For base-2 JS divergence, the domination constant is $C_{\mathrm{JS}} = 1/(8\ln 2)$. Substituting this constant into the general bounds yields the following guarantees.

Corollary 2 (Confidence smoothness and propagation under NeSCal) *For every edge $(i,j) \in E$,*

$$|c_i - c_j| \le \sqrt{2\ln(2)\mathrm{JS}(p_i\|p_j)} \tag{35}$$

and consequently

$$\mathcal{E}_{\mathrm{cal}_j} \le \mathcal{E}_{\mathrm{cal}_i} + \sqrt{2\ln(2)\,\mathrm{JS}(p_i\|p_j)} + |a_i - a_j|. \tag{36}$$

Along any simple directed path $P = (v_0, \dots, v_r)$ in $\mathcal{G}$,

$$\mathcal{E}_{\mathrm{cal}_{v_r}} \le \mathcal{E}_{\mathrm{cal}_{v_0}} + \sqrt{2\ln(2)\, r \sum_{\ell=1}^{r} \mathrm{JS}(p_{v_{\ell-1}}\|p_{v_\ell})} + \left|a_{v_r} - a_{v_0}\right|. \tag{37}$$

Proof. Write $m = (p+q)/2$, so that $\mathrm{JS}_{\ln}(p\|q) = \frac{1}{2}D_{\mathrm{KL}}(p\|m) + \frac{1}{2}D_{\mathrm{KL}}(q\|m)$. Since $p - m = \frac{1}{2}(p-q)$ and $q - m = -\frac{1}{2}(p-q)$, we have $\|p-m\|_1 = \|q-m\|_1 = \frac{1}{2}\|p-q\|_1$. Pinsker's inequality $D_{\mathrm{KL}}(u\|v) \ge \frac{1}{2}\|u-v\|_1^2$ (in nats) then gives

$$D_{\mathrm{KL}}(p\|m),\ D_{\mathrm{KL}}(q\|m) \ge \tfrac{1}{2}\cdot\tfrac{1}{4}\|p-q\|_1^2 = \tfrac{1}{8}\|p-q\|_1^2 \tag{38}$$

thus $\mathrm{JS}_{\ln}(p\|q) \ge \frac{1}{8}\|p-q\|_1^2$. Converting to base 2 via $\mathrm{JS} = \mathrm{JS}_{\ln}/\ln 2$ yields $\mathrm{JS}(p\|q) \ge \frac{1}{8\ln 2}\|p-q\|_1^2$, i.e. $C_{\mathrm{JS}} = 1/(8\ln 2)$. Substituting into Proposition 5 gives $\frac{1}{2\sqrt{C_{\mathrm{JS}}}} = \frac{1}{2}\sqrt{8\ln 2} = \sqrt{2\ln 2}$, which is the stated confidence bound.

The remaining claims follow by the same substitution. In Proposition 1, $\frac{1}{2}C_{\mathrm{D}}^{-1/2}\sqrt{\mathrm{D}(p_i\|p_j)}$ becomes $\sqrt{2\ln(2)\,\mathrm{JS}(p_i\|p_j)}$; in Theorem 1, $\frac{r}{4C_{\mathrm{D}}}$ becomes $2\ln(2)r$ inside the first radical; and in Proposition 2 the same replacement gives the NeSCal-specific obstruction. □

Proposition 6 bounds large disagreement at a fixed predictor: for an edge sampled with probability $W_{ij}/|E|$ and $t > 0$, $\Pr(\mathrm{JS}(p_i\|p_j) \ge t) \le \mathcal{L}(\theta)/(\lambda|E|t)$. Instantiating Proposition 2 gives

$$\mathcal{E}_{\mathrm{cal}_i} + \mathcal{E}_{\mathrm{cal}_j} \ge |a_i - a_j| - \sqrt{2\ln(2)\,\mathrm{JS}(p_i\|p_j)} \tag{39}$$

Taken together, these delimit what NeSCal can and cannot achieve. The penalty $\mathrm{JS}(p_i\|p_j)$ appears in every bound as the quantity the regularizer directly controls, and λ governs how strongly it is suppressed. The residual $|a_i - a_j|$ appears in each with the opposite sign; it is not penalized, and is determined by which points the frozen graph connects and by the predicted labels. NeSCal is therefore expected to improve calibration to the extent that its graph connects points which genuinely share a true probability of correctness - a condition the design choices of Section 4.2 are intended to promote and which Section 6 examines empirically.

B.1 JS-Specific Laplacian Approximation Lemma

Corollary 3 (JS-induced weighting) *For* $\mathrm{D} = \mathrm{JS}$ *(base* 2*), under the probability floor* $\min_{i,k} p_{i,k} \geq \varepsilon$ *of Proposition 4, the induced local form is diagonal,*

$$G_{\mathrm{JS}}(p_i) = H(p_i) = \operatorname{diag}\left(\frac{1}{8\ln(2)\, p_{i,1}}, \ldots, \frac{1}{8\ln(2)\, p_{i,K}}\right) \tag{40}$$

which is positive definite since $p_{i,k} > 0$ *for softmax outputs. Consequently,*

$$R_n(p) = \sum_{i,j} W_{ij}\,(p_i - p_j)^{\mathsf{T}} H(p_i)(p_i - p_j) + \rho, \qquad |\rho| \leq \frac{1}{12\ln 2\,\varepsilon^2} \sum_{i,j} W_{ij} \|p_i - p_j\|_2^3. \tag{41}$$

Proof. The JS divergence satisfies $\mathrm{JS}(p\|p) = 0$, and its formula is smooth on the positive orthant with third derivatives jointly continuous in (p, q), so under the probability floor of Proposition 4 its conditions hold. By Proposition 4, the zeroth- and first-order terms of the expansion of $\mathrm{JS}(p_i\|p_j)$ about $p_i = p_j$ vanish, and the weighting is determined by $G_{\mathrm{JS}}(p) = \frac{1}{2}\nabla_q^2 \mathrm{JS}(p\|q)|_{q=p}$. It therefore suffices to extract the second-order coefficient. Write $\delta_k = q_k - p_k$ and $m_k = p_k + \frac{\delta_k}{2}$, and expand JS in natural-log units,

$$\begin{aligned}
\mathrm{JS}_{\ln}(p\|q) &= \sum_k \frac{1}{2}\Big[p_k \ln p_k + q_k \ln q_k - (p_k + q_k)\ln m_k\Big] \\
&= \sum_k \frac{1}{2}\Big[p_k \ln p_k + (p_k + \delta_k)\ln(p_k + \delta_k) - (2p_k + \delta_k)\ln(p_k + \frac{\delta_k}{2})\Big]
\end{aligned}$$

Using the second-order Taylor expansions

$$\begin{aligned}
\ln(p_k + \delta_k) &= \ln p_k + \tfrac{\delta_k}{p_k} - \tfrac{\delta_k^2}{2p_k^2} + O(\delta^3), \\
\ln\left(p_k + \tfrac{\delta_k}{2}\right) &= \ln p_k + \tfrac{\delta_k}{2p_k} - \tfrac{\delta_k^2}{8p_k^2} + O(\delta^3),
\end{aligned}$$

and retaining only the second-order terms (the lower orders cancel by Proposition 4) gives

$$\sum_k \tfrac{1}{2}\left[\tfrac{\delta_k^2}{p_k} - \tfrac{\delta_k^2}{2p_k} + \tfrac{\delta_k^2}{4p_k} - \tfrac{\delta_k^2}{2p_k}\right] = \sum_k \frac{\delta_k^2}{8p_k}. \tag{42}$$

Hence $G_{\mathrm{JS},\ln}(p) = \operatorname{diag}(1/(8p_k))$. For the remainder, $\mathrm{JS}_{\ln}(p\|q) = \sum_k \varphi_k(q_k)$ is separable, with $\varphi_k'''(q) = \frac{1}{2}\big(-q^{-2} + \frac{1}{4}m^{-2}\big)$, $m = (p_k + q)/2$, so $|\varphi_k'''| \leq 1/(2\varepsilon^2)$ on the floor; since $\sum_k |u_k|^3 \leq \|u\|_2^3$, $M_3(\varepsilon) \leq 1/(2\varepsilon^2)$ in natural-log units, and Proposition 4 gives $|\rho_{ij}| \leq \|p_i - p_j\|_2^3/(12\varepsilon^2)$, which becomes $\|p_i - p_j\|_2^3/(12\ln 2\,\varepsilon^2)$ in base 2. Converting to base 2 via the exact identity $\mathrm{JS} = \mathrm{JS}_{\ln}/\ln 2$ (each D_{KL} term rescales identically under change of base) gives $H(p) = \operatorname{diag}(1/(8\ln(2)\, p_k))$, and substituting $\delta_k = p_{i,k} - p_{j,k}$ yields the matrix form. □

Up to the constant $1/(8\ln 2)$, $H(p_i)$ is the Fisher information metric of the categorical distribution (Amari & Nagaoka, 2000), which every f-divergence induces up to $\frac{1}{2}f''(1)$. Penalizing JS therefore measures disagreement in the intrinsic geometry of the simplex rather than the Euclidean geometry of the probability vector. The weighting is a property of the geometry rather than a performance guarantee, and it holds only where the expansion is accurate, $|p_{i,k} - p_{j,k}| \ll p_{i,k}$.

C NeSCal Implementation and Computational Cost

C.1 Objective Normalization

The objective in Section 4.1 uses the summed penalty $R_n(p)$; optimization uses the directed-edge mean $\bar{R}_n(p) = R_n(p)/|E|$. This rescales the coefficient: $\lambda_{\mathrm{mean}} = |E|\lambda_{\mathrm{sum}}$. Algorithm 1 and all reported regularization strengths use the edge-mean convention.

Algorithm 1 NeSCal

Require: Training set $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$; model f_θ with representation map h; number of neighbors k; regularization strength λ_0; activation epoch T_a; total epochs T

Initialize model parameters θ
for $t = 1$ **to** T **do**
  if $t = T_a$ **then**
    Compute $z_i \leftarrow h(x_i)$ for all i ▷ no gradient
    $W \leftarrow \textsc{kNNGraph}(\{z_i\}_{i=1}^n, k, \text{cosine})$ ▷ fixed thereafter
    $\widehat{p}_i \leftarrow \text{stopgrad}(f_\theta(x_i))$ for all i ▷ initialize cache
  end if
  if $t \geq T_a$ **then**
    $\lambda_t \leftarrow \dfrac{\lambda_0}{2}\left[1 + \cos\left(\pi \dfrac{t - T_a}{T - T_a}\right)\right]$
  else
    $\lambda_t \leftarrow 0$
  end if
  for all minibatches $B \subseteq \mathcal{D}$ **do**
    $p_i \leftarrow f_\theta(x_i)$ for all $i \in B$
    $\mathcal{L}_{\text{CE}} \leftarrow \dfrac{1}{|B|} \sum_{i \in B} \text{CE}(p_i, y_i)$
    if $t \geq T_a$ **then**
      $\widehat{p}_i \leftarrow \text{stopgrad}(p_i)$ for all $i \in B$ ▷ current targets for neighbors in this batch
      $E_B \leftarrow \{(i, j) : i \in B,\ W_{ij} = 1\}$
      $R \leftarrow \dfrac{1}{|E_B|} \sum_{(i,j) \in E_B} \text{JS}(p_i \,\|\, \widehat{p}_j)$ ▷ $\widehat{p}_j$ detached
    else
      $R \leftarrow 0$
    end if
    $\theta \leftarrow \theta - \alpha \nabla_\theta \left(\mathcal{L}_{\text{CE}} + \lambda_t R\right)$
  end for
end for

C.2 Training Procedure and Cost

Algorithm 1 summarizes the training procedure. Training proceeds by standard cross-entropy minimization until the activation epoch T_a, at which point a single inference pass supplies the embeddings from which the k-nearest-neighbor graph is constructed, and a second seeds a cache of predictive distributions for every training sample. From T_a onward each minibatch incurs the neighborhood-consistency penalty in addition to the task loss, with λ annealed over the remaining epochs and the cache refreshed with the predictions computed in that step. The graph itself is never rebuilt.

Computing live predictions for every neighbor would require additional network evaluations. Neighbor targets are therefore detached current-step predictions when the neighbor belongs to the minibatch, and cached predictions otherwise. Each cached prediction is refreshed when its sample is visited, once per epoch. The predictive bounds are evaluated using current predictions, as described in Appendix E.11.

The graph construction is a one-off cost. The k-nearest-neighbor search is exact, computed as a chunked cosine similarity, and is therefore $O(n^2 l)$ in the number of samples n and the embedding dimension l. Because the graph is built once and held fixed (Section 4.2), this cost is incurred a single time and amortized over the remaining $T - T_a$ epochs.

The recurring cost is the penalty itself. It requires one divergence evaluation per edge, each $O(K)$ in the number of classes, giving $O(nkK)$ per epoch against the $O(nC)$ cost of the forward and backward passes, where C is the per-sample cost of the network. Since $C \gg kK$ for most of the architectures considered in Section 5, the relative overhead is small. Memory overhead is $O(nk)$ for the edge list, held on host memory, and $O(nK)$ for the cached predictions. Measured graph-

construction times, peak memory, and training wall-clock relative to the baselines are reported in Appendix E.10.

C.3 Directed Neighborhoods

Because the k-nearest-neighbor relation is not symmetric (j may be among the k nearest neighbors of i without the opposite being true), $\mathcal{G}$ is a directed graph, and W is not in general symmetric. Nothing in Section 3 requires symmetry. Larger neighborhoods can reduce averaging variance while admitting more heterogeneous points.

C.4 Activation and Schedule Rationale

Activation epoch and λ are selected jointly by validation NLL; the theory motivates the schedule without determining it. Under Proposition 3's assumptions, a decrease in $\mathbb{E}[(e_i - e_j)^2]$ narrows the correctness gap that averaging can tolerate, motivating decay. Proposition 2 also motivates delaying graph construction until neighborhoods are informative: smoothing across a large correctness gap can preclude calibration at both endpoints. Delaying may stabilize predicted labels, which fixes the corresponding correctness probabilities. Activation balances these considerations.

NeSCal activates after a warm-up and then decays λ on a cosine schedule. Appendix E.7 compares constant and cosine schedules; Appendix E.8 evaluates diagnostic activation rules and their transfer across settings.

D Training and Implementation Details

Unless stated otherwise, every method in a given setting shares the architecture, optimizer, schedule, epoch budget, batch size, data splits and seeds; only the training objective differs.

D.1 Datasets, Splits and Preprocessing

Table 3: Datasets. Splits are train/val/test

Dataset	Modality	Classes	Size	Split
CIFAR-10 / CIFAR-100	image	10 / 100	60,000	45k / 5k / 10k
Tiny-ImageNet	image	200	110,000	90k / 10k / 10k
20 Newsgroups	text	20	18,846	13,570 / 1,507 / 3,769
Amazon (binary) (frozen RoBERTa)	text	2	11,349	7,264 / 1,816 / 2,269
Amazon (5-class) (frozen RoBERTa)	text	5	22,815	14,602 / 3,650 / 4,563
Helena	tabular	100	65,196	41,724 / 10,432 / 13,040
Jannis	tabular	4	83,733	53,589 / 13,397 / 16,747
Covertype (subsampled)	tabular	7	100,000	64,000 / 16,000 / 20,000

Image. Standard augmentation (random crop with 4-pixel padding, random horizontal flip) and per-channel normalization; evaluation uses normalization only. The validation split is a seeded 10% partition of the training set, fixed across seeds.

Tabular. Following Gorishniy et al. (2021), features are quantile-transformed to a normal output distribution, except for Helena, which is standardized. The transform is fitted on the training partition only. The test and validation splits are fixed across seeds.

Text. The 20 Newsgroups pipeline follows the MMCE reference implementation: vocabulary of $20{,}000$, sequence length $1{,}000$, GloVe-100d initialization (Pennington et al., 2014). Frozen-encoder embeddings are mean-pooled over tokens with the attention mask and extracted once, so the head sees a fixed vector dataset. The test split is fixed across seeds in every setting; the validation split is redrawn per seed on the 20NG setting, and for the frozen-encoder Amazon settings it is held fixed so that the same encoder embeddings are reused.

Table 4: Training configuration per setting.

Setting	Optimizer	LR	LR schedule	Epochs	Batch
CIFAR-10/100, ResNet-50 & WRN-28-10	SGD (0.9, wd 5×10^{-4})	0.1	$\times0.1$ at $50\%, 75\%$	200	128
Tiny-ImageNet, ResNet-50	SGD (0.9, wd 5×10^{-4})	0.1	$\times0.1$ at $40\%, 60\%$	100	64
20 Newsgroups, GPCNN	Adam	10^{-3}	constant	50	128
Frozen encoder + MLP head	SGD (0.9, wd 10^{-4})	10^{-3}	constant	50 (early stopping)	256
Tabular, MLP 3×256 (dropout 0.1)	AdamW (wd 10^{-5})	10^{-3}	constant	100	512

D.2 Architectures and Optimization

Image runs use mixed-precision training; all others use full precision. Tabular and text settings use a constant learning rate by design, which removes any interaction between the λ schedule and LR decay - a useful control for the schedule comparison.

D.3 NeSCal Configuration and Selection

Unless ablated, $k = 10$, cosine distance, JS divergence, a directed unweighted kNN graph over penultimate features, frozen at the activation epoch. The penalty uses the directed-edge mean $\bar{R}_n = R_n/|E|$, removing its direct scaling with the number of edges. Cached neighbor predictions are refreshed when their samples are visited (Appendix C). All reported regularization strengths use the edge-mean convention (Appendix C.1).

Activation epoch and λ are selected jointly. A single cross-entropy warm-up run per seed is checkpointed at every candidate onset; each grid point resumes from the corresponding checkpoint, so all branches share a bit-identical prefix and differ only in the intervention under study. Resumption restores optimizer state and the scheduler's absolute epoch. This checkpoint-resumption setup enables a wider grid search space at lower cost. Within each schedule, activation epoch and λ are selected by mean validation NLL over seeds 18 and 42. Evaluation of the selected configuration is on seeds $18, 42, 613$ and 2026. The same selection-evaluation protocol applies to all methods, with the following exceptions: for 20NG, CE and NeSCal additionally include evaluation on seeds 7 and 1234; s-AvUC for Tiny-ImageNet uses three evaluation seeds, excluding 2026. Table 5 gives the search grids. Divergent configurations are excluded from selection; Helena omits $\lambda = 0.5$.

Table 5: NeSCal search grids. Onsets are epochs of the total budget.

Setting	Candidate onsets	λ grid (cosine / constant)
CIFAR-10, ResNet-50	$\{50, 65, 80, 95, 110, 125\}$	$\{1, 3, 5, 10, 20\}$ / $\{1, 3, 5\}$
CIFAR-10, WRN-28-10	$\{50, 65, 80, 95, 110, 125\}$	$\{5, 10, 20, 30, 50\}$ / $\{3, 5, 10\}$
CIFAR-100, ResNet-50	$\{50, 65, 80, 95, 100, 110, 125\}$	$\{5, 10, 20, 30, 50\}$ / $\{3, 5, 8, 10\}$
CIFAR-100, WRN-28-10	$\{50, 65, 80, 95, 100, 110, 125\}$	$\{5, 10, 20, 30, 40\}$ / $\{3, 5, 8, 10\}$
Tiny-ImageNet	$\{20, 30, 38, 40, 50, 58\}$	$\{5, 10, 20, 30, 40\}$ / $\{5, 10, 20\}$
20NG	$\{3, 5, 8, 12, 20\}$	$\{0.5, 1, 2, 5\}$ / $\{0.5, 1, 2, 5\}$
Tabular	$\{5, 10, 20, 30, 50\}$	$\{0.5, 1, 2, 5, 10\}$ / $\{1, 2, 5\}$
Frozen encoder	$\{1, 3, 5, 10, 20, 30\}$	$\{0.01, 0.05, 0.1, 0.3, 0.5, 0.7, 1.0, 2.0\}$ / constant identical

D.4 Competitor Search Spaces

Each competitor is tuned over setting-dependent candidate ranges informed by the search ranges or parameter-sensitivity analyses reported in its original paper. Configurations are selected by seed-averaged validation NLL, using the same selection criterion as NeSCal. For compactness, the values below report the widest range of candidate values considered for each method across settings; individual settings may use subsets of these values, dependent on author recommendations under parallel settings. We use each method's own notation. FL is searched over fixed $\gamma \in \{2, 3\}$ and the adaptive, sample-dependent variant with $\gamma_{\text{high}} \in \{2, 3\}$, selecting once across both families. LS uses $\alpha \in \{0.05, 0.1\}$; MMCE $\lambda \in \{1, 3, 7, 10, 15, 20\}$ at the Laplacian kernel width of Kumar et al. (2018), owing to their reported insensitivity to λ in that range; MDCA $\beta \in \{1, 5, 10, 15, 20, 25\}$;

MbLS $\lambda \in \{0, 0.05, 0.1, 0.2, 0.3\}$ with margin $m \in \{4, 6, 8, 10, 12\}$. s-AvUC is searched sequentially in the order $\kappa \in \{0.1, 0.5, 0.9\}$, temperature $T \in \{0.5, 1, 2\}$, penalty weight $\beta \in \{0.1, 1, 3\}$, matching the authors' own recommendation, as a full grid is prohibitive. Note that λ denotes the penalty weight for MMCE, MbLS and NeSCal, but the ℓ_2 coefficient in the notation of Karandikar et al. (2021).

D.5 Post-hoc Calibrators

Post-hoc calibrators are fitted on the validation split used for base-model selection and evaluated on the test split, following the protocol of Guo et al. (2017). TS uses one-dimensional numerical minimization of validation NLL. Dirichlet calibration uses the full matrix parameterization with off-diagonal and intercept regularization (ODIR), with regularization strength selected on validation. LoCal uses its local calibration objective and early stopping. Feature preprocessing and other departures from the reference methods are described below.

D.6 Metrics

The limitations of ECE have motivated debiased and binning-free alternatives (Arrieta-Ibarra et al., 2022; Błasiok et al., 2023). Other notions retain different information: classwise calibration requires $\mathbb{P}(Y = k \mid p_k(X)) = p_k(X)$ for each class, while canonical calibration requires $\mathbb{P}(Y = k \mid p(X)) = p_k(X)$ (Kull et al., 2019; Vaicenavicius et al., 2019; Widmann et al., 2019). Local calibration assesses reliability within neighborhoods in representation space (Luo et al., 2022).

We report accuracy, ECE (Naeini et al., 2015), an equal-mass-binned ECE variant (Nixon et al., 2019), classwise ECE (Kull et al., 2019), the local calibration error (LCE), using the classwise extension of Barbera et al. (2026) to the confidence-based measure of Luo et al. (2022), and the binned calibration–refinement decomposition of the top-label Brier score (Brier, 1950; Kull & Flach, 2015).

ECE and MCE use 15 equal-width bins of top-label confidence, the last bin closed; the equal-mass variant uses 15 equal-count bins. Classwise ECE applies the same binning to each class probability and averages over classes. LCE uses a Gaussian kernel ($\gamma = 10$) in a 50-dimensional PCA projection of the penultimate features, fitted on the validation split, excluding bins with at most 20 samples. The calibration–refinement decomposition bins top-label confidence into the same 15 bins; because it ignores within-bin variation, its two terms need not sum exactly to the top-label Brier score.

D.7 Additional Experiments and Ablations

We ablate the components of Section 4.2: neighborhood size, random versus nearest-neighbor graphs, graphs built on PCA- and UMAP-reduced representations, and alternative divergences at matched initial penalty magnitude. We also compare regularization schedules and diagnostic activation rules. Further studies assess input corruption, out-of-distribution detection, class imbalance, and label noise. Results appear in Appendix E.

D.8 Theoretical Diagnostics

The quantities of Sections 3.2 and Appendix B are computed on the training nodes of the frozen graph, at the selected weights, from a full inference pass with current predictions rather than from the cached distributions used during training. We report whether bounds hold on every edge, the median per-edge tightness of Corollary 2 and the fraction of edges satisfying the small-disagreement condition under which the Laplacian expansion of Proposition 4 is valid. We do not report an aggregate ratio of the Laplacian quadratic form to R_n: it is dominated by a small number of edges carrying near-zero class probabilities, where the third-derivative remainder diverges, and is therefore uninformative about the graph as a whole. This is consistent with the lemma being an interpretive device rather than a bound we optimize.

The regime diagnostics of Table 2 are logged by the training runs at the activation epoch and averaged over seeds, on the same frozen graph.

D.9 Robustness and Perturbation Studies

Corruption robustness is evaluated on CIFAR-10-C, CIFAR-100-C, and Tiny-ImageNet-C using the 15 standard corruption types (14 on Tiny-ImageNet-C) at all five severity levels (Hendrycks & Dietterich, 2019); out-of-distribution (OOD) separation uses SVHN (Netzer et al., 2011) as the OOD dataset for models trained on CIFAR-10, CIFAR-100, and Tiny-ImageNet. SVHN images are resized where necessary to match the model input resolution and normalized using the corresponding in-distribution statistics. Detection is based on maximum softmax probability (Hendrycks & Gimpel, 2017) and evaluated using AUROC. We additionally train under long-tailed class imbalance (imbalance factors 10, 50, 100) and symmetric label noise (20%, 40%), reusing the clean-selected configuration.

D.10 Cost Measurement

We report wall-clock training time excluding evaluation, one-off graph construction time, and peak memory, measured on the same hardware for all methods within a setting. The reported NeSCal wall-clock includes the warm-up prefix, so the comparison against baselines is like-for-like. All settings ran on either a single NVIDIA RTX 5000 Ada Gen. or NVIDIA RTX A6000 GPU, with the GPU model kept consistent for every run in that setting.

D.11 Deviations from Reference Implementations

We note the following deviations from the original implementations and protocols.

Protocol. All methods share a validation split carved from the training set (45k/5k on CIFAR), are evaluated at the epoch of best validation NLL rather than at the final epoch, and are tuned on validation NLL; several of the compared methods tune on a calibration metric in their own papers. CIFAR budgets (Table 4) are shorter than the 350-epoch schedule of Mukhoti et al. (2020).

Architectures. ResNet-50 uses the standard small-input stem (3×3 convolution, no max-pooling) on both CIFAR and Tiny-ImageNet. The 20 Newsgroups model follows the canonical global-pooling CNN of Kumar et al. (2018), with the GloVe embedding matrix held frozen rather than fine-tuned, and a 50-epoch budget. Tabular models use a fixed 3×256 MLP with a fixed epoch budget, rather than the per-dataset tuned architectures and early stopping of Gorishniy et al. (2021); accuracies are correspondingly 0.5–1.5 points below their tuned reference values. Covertype is subsampled to 100k rows.

Baselines. MDCA is evaluated as cross-entropy plus MDCA, without the focal or label-smoothing compositions reported by its authors. Two further departures are shared across methods rather than specific to one. All CIFAR runs use batch size 128, following Mukhoti et al. (2020) and Kumar et al. (2018), rather than the 512 of Karandikar et al. (2021). MDCA is trained end-to-end on the standard architecture rather than on the frozen-backbone head with additional convolutional and fully connected layers used by Hebbalaguppe et al. (2022). MbLS on 20 Newsgroups uses the 50-epoch constant-rate schedule shared by all methods in that setting rather than the 100-epoch decayed schedule of Liu et al. (2022). Calibration metrics use 15 bins throughout, following three of the four reference papers; Kumar et al. (2018) use 20.

Post-hoc. LoRe is applied in the same standardized PCA feature space and Gaussian kernel as our locality metric rather than at its published Laplacian bandwidth, which was tuned on different features. LoCal is run at its authors' published configuration in every setting, without per-setting tuning.

E Additional Results

E.1 Full per-setting metrics

Tables 6–8 report the metrics accompanying the NLL comparison in the main text. NeSCal improves Brier score over CE in ten of eleven settings and accuracy in nine.

NeSCal attains the lowest or second-lowest LCE in five of eight image and tabular settings, counting ties at displayed precision. Improvements are not uniform across settings or calibration measures. Classwise ECE values are small in absolute terms on the many-class datasets and should be interpreted alongside the other metrics.

Table 6: Brier score (top, lower is better) and accuracy in percent (bottom, higher is better). Conventions as in Table 1.

Setting	CE	FL	LS	MMCE	MDCA	s-AvUC	MbLS	NeSCal
Brier score								
CIFAR-10 / R50	0.0953 ± 0.0045	**0.0866 ± 0.0029**	*0.0872 ± 0.0035*	0.0988 ± 0.0021	0.0943 ± 0.0087	0.0949 ± 0.0054	0.0948 ± 0.0057	*0.0872 ± 0.0015*
CIFAR-10 / WRN	0.0782 ± 0.0051	**0.0733 ± 0.0016**	*0.0750 ± 0.0019*	0.0854 ± 0.0030	0.0852 ± 0.0066	0.0831 ± 0.0052	0.0807 ± 0.0052	0.0785 ± 0.0018
CIFAR-100 / R50	0.3447 ± 0.0081	*0.3296 ± 0.0019*	0.3430 ± 0.0083	0.3481 ± 0.0128	0.3482 ± 0.0041	0.3443 ± 0.0057	0.3372 ± 0.0045	**0.3178 ± 0.0050**
CIFAR-100 / WRN	0.3321 ± 0.0033	**0.3024 ± 0.0022**	0.3162 ± 0.0015	0.3396 ± 0.0066	0.3353 ± 0.0058	0.3337 ± 0.0025	0.3161 ± 0.0060	*0.3057 ± 0.0061*
Tiny-ImageNet	0.4778 ± 0.0057	0.4991 ± 0.0030	0.4757 ± 0.0063	0.4819 ± 0.0079	0.4696 ± 0.0071	*0.4691 ± 0.0096*	**0.4651 ± 0.0073**	0.4693 ± 0.0064
Helena	0.7597 ± 0.0004	0.7707 ± 0.0011	*0.7563 ± 0.0029*	0.7590 ± 0.0010	0.7589 ± 0.0013	0.7593 ± 0.0007	**0.7559 ± 0.0004**	0.7573 ± 0.0016
Jannis	0.4012 ± 0.0005	0.4520 ± 0.0033	*0.3992 ± 0.0023*	0.4012 ± 0.0015	0.3998 ± 0.0008	0.4013 ± 0.0010	0.4012 ± 0.0014	**0.3973 ± 0.0006**
Covertype	0.1401 ± 0.0021	0.2085 ± 0.0030	0.1401 ± 0.0019	0.1396 ± 0.0006	0.1417 ± 0.0011	**0.1387 ± 0.0017**	*0.1388 ± 0.0014*	0.1389 ± 0.0015
20NG	0.3247 ± 0.0116	0.2944 ± 0.0095	**0.2456 ± 0.0084**	0.2842 ± 0.0160	0.3263 ± 0.0126	0.3128 ± 0.0134	*0.2491 ± 0.0113*	0.3200 ± 0.0061
Amazon (5-class)	0.4303 ± 0.0013	0.4418 ± 0.0007	**0.4241 ± 0.0009**	*0.4264 ± 0.0013*	0.4303 ± 0.0017	0.4296 ± 0.0018	0.4291 ± 0.0009	0.4288 ± 0.0017
Amazon (binary)	0.1272 ± 0.0010	0.1597 ± 0.0059	*0.1239 ± 0.0009*	**0.1211 ± 0.0005**	0.1275 ± 0.0033	0.1275 ± 0.0012	0.1261 ± 0.0014	0.1262 ± 0.0007
Accuracy (%)								
CIFAR-10 / R50	93.70 ± 0.47	94.24 ± 0.23	**94.56 ± 0.31**	93.22 ± 0.22	93.77 ± 0.78	93.76 ± 0.50	93.94 ± 0.64	*94.28 ± 0.15*
CIFAR-10 / WRN	95.11 ± 0.51	*95.19 ± 0.18*	**95.30 ± 0.14**	94.26 ± 0.12	94.49 ± 0.59	94.86 ± 0.45	94.76 ± 0.59	94.94 ± 0.11
CIFAR-100 / R50	75.29 ± 0.82	*76.41 ± 0.22*	75.18 ± 0.82	75.04 ± 1.07	75.02 ± 0.33	75.36 ± 0.37	75.66 ± 0.41	**77.19 ± 0.52**
CIFAR-100 / WRN	76.84 ± 0.26	**78.47 ± 0.19**	77.46 ± 0.28	75.94 ± 0.40	76.60 ± 0.45	76.32 ± 0.31	77.40 ± 0.52	*78.10 ± 0.65*
Tiny-ImageNet	64.29 ± 0.52	62.37 ± 0.37	64.75 ± 0.53	63.83 ± 0.74	*64.92 ± 0.61*	64.89 ± 0.67	**65.28 ± 0.56**	64.51 ± 0.34
Helena	37.55 ± 0.14	37.30 ± 0.15	*37.83 ± 0.40*	37.59 ± 0.16	37.57 ± 0.21	37.46 ± 0.12	**37.91 ± 0.07**	37.76 ± 0.18
Jannis	70.46 ± 0.11	69.62 ± 0.46	*70.84 ± 0.25*	70.49 ± 0.23	70.67 ± 0.07	70.54 ± 0.14	70.47 ± 0.20	**70.94 ± 0.10**
Covertype	90.43 ± 0.20	89.24 ± 0.25	90.56 ± 0.09	90.58 ± 0.05	90.37 ± 0.09	*90.59 ± 0.13*	90.55 ± 0.15	**90.62 ± 0.17**
20NG	76.94 ± 1.55	79.53 ± 0.69	**83.24 ± 0.53**	80.13 ± 1.48	76.46 ± 1.62	77.71 ± 1.40	*82.67 ± 1.05*	77.20 ± 0.81
Amazon (5-class)	68.58 ± 0.06	68.47 ± 0.33	*68.67 ± 0.28*	68.60 ± 0.08	68.55 ± 0.18	68.53 ± 0.17	**68.70 ± 0.10**	68.46 ± 0.15
Amazon (binary)	91.74 ± 0.10	91.64 ± 0.11	*91.79 ± 0.07*	**91.82 ± 0.10**	91.70 ± 0.10	91.70 ± 0.15	91.77 ± 0.13	**91.82 ± 0.05**

Table 7: ECE in percent for the base model (top) and after temperature scaling (bottom); lower is better.

Setting	CE	FL	LS	MMCE	MDCA	s-AvUC	MbLS	NeSCal
Base model								
CIFAR-10 / R50	2.46 ± 0.49	*1.14 ± 0.04*	3.98 ± 0.21	**0.83 ± 0.10**	2.56 ± 0.40	2.55 ± 0.53	2.08 ± 0.49	2.29 ± 0.11
CIFAR-10 / WRN	2.75 ± 0.30	**0.78 ± 0.07**	1.64 ± 0.31	*1.13 ± 0.25*	2.66 ± 0.14	3.00 ± 0.20	2.10 ± 0.51	2.39 ± 0.18
CIFAR-100 / R50	5.90 ± 0.58	*2.73 ± 0.50*	**1.39 ± 0.18**	6.02 ± 0.49	6.08 ± 0.24	5.83 ± 0.94	3.84 ± 0.51	3.95 ± 0.67
CIFAR-100 / WRN	8.19 ± 0.25	**1.87 ± 0.12**	*3.41 ± 0.57*	7.45 ± 0.38	7.91 ± 0.18	7.03 ± 0.61	3.62 ± 0.29	4.60 ± 0.62
Tiny-ImageNet	5.75 ± 0.29	3.12 ± 0.23	4.63 ± 0.29	4.56 ± 0.35	6.00 ± 0.25	6.27 ± 0.47	*2.69 ± 0.63*	**2.02 ± 0.36**
Helena	1.36 ± 0.24	5.66 ± 0.42	1.32 ± 0.25	1.53 ± 0.33	**1.19 ± 0.08**	1.36 ± 0.14	*1.23 ± 0.17*	1.25 ± 0.07
Jannis	1.51 ± 0.49	14.36 ± 0.93	**1.37 ± 0.26**	1.96 ± 0.32	1.77 ± 0.39	2.55 ± 0.56	*1.46 ± 0.38*	2.09 ± 0.39
Covertype	1.14 ± 0.15	14.25 ± 0.58	3.23 ± 0.21	*1.11 ± 0.29*	1.15 ± 0.25	1.39 ± 0.16	**0.48 ± 0.10**	1.23 ± 0.27
20NG	5.72 ± 2.40	4.60 ± 1.65	*2.32 ± 0.83*	5.30 ± 0.65	4.29 ± 2.32	5.18 ± 1.20	**2.11 ± 0.73**	4.80 ± 1.18
Amazon (5-class)	8.14 ± 0.62	10.63 ± 0.14	**5.50 ± 0.88**	*7.49 ± 0.67*	8.82 ± 0.27	7.80 ± 1.33	8.01 ± 0.59	8.37 ± 0.75
Amazon (binary)	3.07 ± 0.63	13.09 ± 0.93	4.17 ± 0.18	**0.92 ± 0.18**	2.93 ± 0.72	*2.58 ± 0.62*	2.60 ± 0.42	3.75 ± 0.04
After temperature scaling								
CIFAR-10 / R50	*0.64 ± 0.12*	0.91 ± 0.15	1.41 ± 0.23	0.81 ± 0.16	0.67 ± 0.16	0.66 ± 0.18	1.52 ± 0.09	**0.60 ± 0.14**
CIFAR-10 / WRN	**0.70 ± 0.15**	0.80 ± 0.07	1.19 ± 0.13	0.82 ± 0.20	0.73 ± 0.11	0.94 ± 0.14	1.20 ± 0.17	*0.71 ± 0.20*
CIFAR-100 / R50	**1.61 ± 0.25**	2.08 ± 0.16	2.52 ± 0.22	1.81 ± 0.31	1.68 ± 0.28	1.88 ± 0.25	1.94 ± 0.28	*1.63 ± 0.24*
CIFAR-100 / WRN	1.86 ± 0.23	1.93 ± 0.20	2.65 ± 0.34	1.92 ± 0.25	1.94 ± 0.12	*1.85 ± 0.33*	2.42 ± 0.26	**1.54 ± 0.23**
Tiny-ImageNet	1.98 ± 0.25	2.26 ± 0.17	2.46 ± 0.37	1.80 ± 0.32	1.59 ± 0.12	**1.39 ± 0.38**	*1.45 ± 0.29*	1.74 ± 0.23
Helena	2.19 ± 0.10	5.84 ± 0.28	**1.33 ± 0.28**	2.18 ± 0.24	2.24 ± 0.42	2.15 ± 0.21	*1.59 ± 0.31*	2.35 ± 0.32
Jannis	*0.93 ± 0.15*	5.58 ± 0.97	2.12 ± 0.38	1.01 ± 0.07	1.08 ± 0.19	1.08 ± 0.10	1.21 ± 0.30	**0.74 ± 0.22**
Covertype	1.27 ± 0.21	2.90 ± 0.44	1.19 ± 0.23	1.10 ± 0.15	*0.99 ± 0.23*	1.09 ± 0.08	**0.58 ± 0.14**	1.02 ± 0.10
20NG	**2.08 ± 0.53**	4.53 ± 1.30	2.67 ± 0.84	3.81 ± 1.28	*2.39 ± 0.60*	2.66 ± 0.35	2.56 ± 0.47	3.05 ± 0.35
Amazon (5-class)	4.84 ± 0.15	5.20 ± 0.60	5.56 ± 0.30	**4.11 ± 0.20**	*4.49 ± 0.30*	4.77 ± 0.22	4.87 ± 0.17	4.84 ± 0.08
Amazon (binary)	2.68 ± 0.78	*1.85 ± 0.24*	3.31 ± 0.18	**0.95 ± 0.19**	2.26 ± 0.40	2.69 ± 0.45	2.67 ± 0.42	1.95 ± 0.22

Table 8: Classwise ECE, LCE, and equal-mass ECE, all in percent; lower is better.

Setting	CE	FL	LS	MMCE	MDCA	s-AvUC	MbLS	NeSCal
Classwise ECE								
CIFAR-10 / R50	0.58 ± 0.08	*0.43 ± 0.02*	0.83 ± 0.02	**0.40 ± 0.05**	0.58 ± 0.08	0.58 ± 0.07	0.57 ± 0.04	0.54 ± 0.02
CIFAR-10 / WRN	0.61 ± 0.06	*0.45 ± 0.04*	0.53 ± 0.04	**0.42 ± 0.02**	0.61 ± 0.03	0.68 ± 0.04	0.50 ± 0.07	0.59 ± 0.03
CIFAR-100 / R50	0.21 ± 0.00	**0.18 ± 0.00**	*0.19 ± 0.00*	0.21 ± 0.01	0.21 ± 0.01	0.21 ± 0.01	*0.19 ± 0.01*	**0.18 ± 0.01**
CIFAR-100 / WRN	0.24 ± 0.00	**0.18 ± 0.00**	0.20 ± 0.01	0.23 ± 0.01	0.24 ± 0.01	0.22 ± 0.01	0.20 ± 0.00	*0.19 ± 0.00*
Tiny-ImageNet	*0.14 ± 0.00*	*0.14 ± 0.00*	*0.14 ± 0.00*	*0.14 ± 0.00*	*0.14 ± 0.00*	0.15 ± 0.00	*0.14 ± 0.00*	**0.13 ± 0.00**
Helena	*0.16 ± 0.01*	0.20 ± 0.01	*0.16 ± 0.00*	*0.16 ± 0.01*	*0.16 ± 0.01*	*0.16 ± 0.01*	*0.16 ± 0.00*	**0.15 ± 0.00**
Jannis	1.34 ± 0.18	6.71 ± 0.23	1.74 ± 0.27	1.39 ± 0.28	1.36 ± 0.23	1.69 ± 0.21	**1.23 ± 0.23**	*1.25 ± 0.11*
Covertype	0.53 ± 0.05	4.05 ± 0.18	1.16 ± 0.06	*0.49 ± 0.09*	0.50 ± 0.05	0.54 ± 0.03	**0.40 ± 0.04**	0.54 ± 0.05
20NG	1.01 ± 0.15	0.98 ± 0.07	*0.80 ± 0.06*	0.85 ± 0.06	0.90 ± 0.12	0.91 ± 0.09	**0.73 ± 0.03**	0.95 ± 0.05
Amazon (5-class)	3.59 ± 0.17	4.45 ± 0.10	**3.04 ± 0.24**	*3.41 ± 0.23*	3.77 ± 0.05	3.44 ± 0.39	3.54 ± 0.16	3.57 ± 0.25
Amazon (binary)	3.33 ± 0.62	13.22 ± 0.81	4.34 ± 0.22	**2.01 ± 0.22**	3.70 ± 0.96	*2.78 ± 0.57*	3.14 ± 0.52	3.79 ± 0.07
LCE								
CIFAR-10 / R50	0.70 ± 0.13	0.90 ± 0.05	1.07 ± 0.05	0.93 ± 0.06	*0.69 ± 0.17*	0.71 ± 0.17	0.95 ± 0.29	**0.61 ± 0.04**
CIFAR-10 / WRN	0.45 ± 0.04	0.46 ± 0.03	0.45 ± 0.04	0.45 ± 0.03	0.49 ± 0.04	0.50 ± 0.03	**0.42 ± 0.03**	*0.43 ± 0.04*
CIFAR-100 / R50	**0.24 ± 0.01**	0.28 ± 0.00	0.32 ± 0.01	*0.25 ± 0.01*	**0.24 ± 0.01**	**0.24 ± 0.01**	0.28 ± 0.01	0.26 ± 0.01
CIFAR-100 / WRN	0.13 ± 0.00	*0.10 ± 0.00*	0.11 ± 0.00	0.13 ± 0.01	0.13 ± 0.01	0.12 ± 0.00	*0.10 ± 0.00*	**0.09 ± 0.01**
Tiny-ImageNet	*0.18 ± 0.00*	0.25 ± 0.00	0.24 ± 0.00	0.19 ± 0.00	*0.18 ± 0.00*	**0.17 ± 0.00**	0.23 ± 0.00	0.20 ± 0.00
Helena	*0.14 ± 0.01*	0.19 ± 0.01	0.15 ± 0.01	0.15 ± 0.01	*0.14 ± 0.01*	0.15 ± 0.01	*0.14 ± 0.00*	**0.13 ± 0.00**
Jannis	*1.42 ± 0.14*	6.81 ± 0.22	1.89 ± 0.27	1.47 ± 0.19	*1.42 ± 0.23*	1.77 ± 0.18	**1.34 ± 0.15**	**1.34 ± 0.10**
Covertype	0.52 ± 0.05	4.07 ± 0.16	1.22 ± 0.06	*0.49 ± 0.08*	0.50 ± 0.07	0.53 ± 0.03	**0.42 ± 0.04**	0.53 ± 0.05
20NG	0.71 ± 0.13	0.67 ± 0.05	*0.54 ± 0.06*	0.55 ± 0.05	0.66 ± 0.08	0.61 ± 0.09	**0.45 ± 0.03**	0.68 ± 0.04
Amazon (5-class)	4.10 ± 0.15	4.97 ± 0.09	**3.71 ± 0.16**	*3.84 ± 0.19*	4.22 ± 0.11	4.03 ± 0.27	4.07 ± 0.10	4.07 ± 0.22
Amazon (binary)	3.63 ± 0.47	13.47 ± 0.79	4.73 ± 0.15	**2.77 ± 0.14**	4.52 ± 0.91	3.56 ± 0.50	3.73 ± 0.62	*3.22 ± 0.14*
Equal-mass ECE								
CIFAR-10 / R50	2.45 ± 0.50	*0.93 ± 0.12*	3.93 ± 0.29	**0.77 ± 0.17**	2.53 ± 0.39	2.53 ± 0.53	2.62 ± 0.31	2.23 ± 0.13
CIFAR-10 / WRN	2.73 ± 0.31	**0.71 ± 0.14**	2.17 ± 0.15	*0.94 ± 0.16*	2.62 ± 0.11	2.99 ± 0.20	2.14 ± 0.49	2.33 ± 0.15
CIFAR-100 / R50	5.85 ± 0.59	*2.70 ± 0.48*	**1.41 ± 0.18**	6.00 ± 0.48	6.07 ± 0.23	5.81 ± 0.95	3.82 ± 0.49	3.87 ± 0.71
CIFAR-100 / WRN	8.18 ± 0.25	**1.81 ± 0.22**	*3.27 ± 0.51*	7.42 ± 0.40	7.87 ± 0.21	7.02 ± 0.61	3.55 ± 0.30	4.52 ± 0.70
Tiny-ImageNet	5.75 ± 0.29	3.19 ± 0.27	4.73 ± 0.21	4.55 ± 0.33	5.97 ± 0.29	6.27 ± 0.47	*2.69 ± 0.59*	**1.92 ± 0.37**
Helena	1.47 ± 0.38	5.74 ± 0.47	1.39 ± 0.28	1.49 ± 0.28	1.37 ± 0.18	1.46 ± 0.07	*1.35 ± 0.17*	**1.33 ± 0.10**
Jannis	*1.64 ± 0.32*	14.35 ± 0.92	**1.45 ± 0.26**	1.98 ± 0.23	1.84 ± 0.55	2.64 ± 0.46	1.65 ± 0.22	2.03 ± 0.38
Covertype	1.13 ± 0.15	14.24 ± 0.58	3.21 ± 0.22	*1.04 ± 0.31*	1.12 ± 0.17	1.33 ± 0.21	**0.55 ± 0.03**	1.21 ± 0.28
20NG	5.62 ± 2.52	4.77 ± 1.44	*2.45 ± 0.69*	5.14 ± 0.65	4.19 ± 2.37	5.13 ± 1.22	**2.22 ± 0.52**	4.76 ± 1.15
Amazon (5-class)	8.14 ± 0.57	10.64 ± 0.13	**5.82 ± 0.81**	*7.52 ± 0.65*	8.89 ± 0.23	7.84 ± 1.32	7.98 ± 0.57	8.43 ± 0.68
Amazon (binary)	3.89 ± 0.39	13.15 ± 1.08	4.00 ± 0.29	**2.26 ± 0.16**	4.38 ± 0.86	3.73 ± 0.20	3.54 ± 0.38	*3.31 ± 0.09*

E.2 Across-setting comparisons

We summarize pairwise wins across setting means and report exploratory Wilcoxon signed-rank comparisons in Table 9, which counts, for each baseline, the settings in which NeSCal attains the lower mean, and applies an exact two-sided Wilcoxon signed-rank test to the per-setting mean differences. p-values for are uncorrected; with eight settings the smallest attainable is 0.008. Several settings share datasets, thus the uncorrected p-values are therefore interpreted as exploratory summaries. For NLL after temperature scaling NeSCal wins in all eight image and tabular settings against FL, LS, MMCE and MbLS, and in seven of eight against MDCA and s-AvUC.

Table 9: NeSCal against each baseline across settings. "All" covers the eleven completed settings, "I+T" the eight image and tabular settings.

	NLL (all)		NLL (I+T)		NLL+TS (all)		NLL+TS (I+T)		Brier (all)	
vs.	wins	p	wins	p	wins	p	wins	p	wins	p
CE	11/11	0.001	8/8	0.008	11/11	0.001	8/8	0.008	10/11	0.002
FL	7/11	0.067	5/8	0.109	10/11	0.014	8/8	0.008	7/11	0.083
LS	7/11	0.175	7/8	0.016	10/11	0.042	8/8	0.008	5/11	0.878
MMCE	8/11	0.206	8/8	0.008	8/11	0.083	8/8	0.008	8/11	0.240
MDCA	10/11	0.019	8/8	0.008	9/11	0.005	7/8	0.016	11/11	0.001
s-AvUC	10/11	0.019	8/8	0.008	9/11	0.024	7/8	0.023	8/11	0.054
MbLS	7/11	0.278	6/8	0.039	10/11	0.054	8/8	0.008	6/11	0.520

E.3 Calibration–refinement decomposition

Table 10 reports the binned refinement component of the base-model top-label Brier decomposition. This component measures within-bin variation in correctness and is lower for NeSCal than for CE in ten of eleven real-data settings. It provides a complementary diagnostic under the specified binning scheme; recalibration can change bin membership and therefore the estimate.

Table 10: Reported binned refinement component of the base-model top-label Brier decomposition; lower is better.

Setting	CE	FL	LS	MMCE	MDCA	s-AvUC	MbLS	NeSCal
CIFAR-10 / R50	0.0424 ± 0.0020	0.0470 ± 0.0046	**0.0381 ± 0.0012**	0.0439 ± 0.0012	0.0420 ± 0.0036	0.0424 ± 0.0026	0.0424 ± 0.0027	*0.0388 ± 0.0006*
CIFAR-10 / WRN	*0.0346 ± 0.0024*	0.0377 ± 0.0038	**0.0326 ± 0.0008**	0.0382 ± 0.0011	0.0377 ± 0.0030	0.0365 ± 0.0023	0.0359 ± 0.0023	0.0352 ± 0.0010
CIFAR-100 / R50	0.1242 ± 0.0014	*0.1229 ± 0.0015*	0.1249 ± 0.0007	0.1239 ± 0.0032	0.1248 ± 0.0013	0.1237 ± 0.0019	0.1231 ± 0.0016	**0.1160 ± 0.0019**
CIFAR-100 / WRN	0.1185 ± 0.0018	0.1151 ± 0.0010	**0.1069 ± 0.0005**	0.1203 ± 0.0016	0.1197 ± 0.0018	0.1189 ± 0.0009	0.1161 ± 0.0026	*0.1118 ± 0.0014*
Tiny-ImageNet	0.1463 ± 0.0014	0.1558 ± 0.0011	0.1497 ± 0.0016	0.1484 ± 0.0022	*0.1448 ± 0.0016*	**0.1434 ± 0.0029**	0.1468 ± 0.0019	0.1453 ± 0.0030
Helena	0.1777 ± 0.0008	0.1789 ± 0.0013	0.1772 ± 0.0006	0.1770 ± 0.0010	*0.1767 ± 0.0009*	**0.1762 ± 0.0013**	0.1772 ± 0.0006	0.1773 ± 0.0009
Jannis	0.1783 ± 0.0005	0.1860 ± 0.0007	0.1784 ± 0.0001	0.1783 ± 0.0004	*0.1773 ± 0.0003*	0.1781 ± 0.0003	0.1783 ± 0.0004	**0.1770 ± 0.0002**
Covertype	0.0690 ± 0.0011	0.0787 ± 0.0019	**0.0673 ± 0.0010**	0.0690 ± 0.0003	0.0698 ± 0.0005	*0.0683 ± 0.0007*	0.0685 ± 0.0007	0.0685 ± 0.0007
20NG	0.1227 ± 0.0039	0.1397 ± 0.0045	**0.0930 ± 0.0025**	0.1058 ± 0.0048	0.1231 ± 0.0043	0.1178 ± 0.0055	*0.0953 ± 0.0042*	0.1220 ± 0.0024
Amazon (5-class)	0.1808 ± 0.0008	0.1796 ± 0.0008	0.1798 ± 0.0010	*0.1793 ± 0.0008*	**0.1790 ± 0.0004**	0.1805 ± 0.0007	0.1801 ± 0.0002	*0.1793 ± 0.0003*
Amazon (binary)	0.0618 ± 0.0007	0.0608 ± 0.0006	**0.0603 ± 0.0012**	*0.0605 ± 0.0003*	0.0621 ± 0.0013	0.0617 ± 0.0011	0.0615 ± 0.0011	0.0608 ± 0.0005

E.4 Post-hoc composition

Table 11 reports each train-time method's best observed NLL and ECE across the five calibrators, selected separately and retrospectively on test means. NeSCal attains the lowest NLL in six of eleven real-data settings, all of them image or tabular.

Composition does not improve every metric: on Helena, TS slightly lowers NeSCal's NLL while increasing ECE from 1.25% to 2.35%.

E.5 Failure prediction and selective classification

We assess how well confidence separates correct from incorrect predictions using AUROC (Hendrycks & Gimpel, 2017) and area under the risk–coverage curve (AURC) (Geifman et al., 2019). Temperature scaling preserves predicted labels but can change confidence rankings across multiclass examples, so we report both base-model and recalibrated results.

Table 11: Best observed post-hoc results for each train-time method. NLL and ECE are minimized separately over TS (T), LoCal (L), Dirichlet (D), DAC (A), and LoRe (R); ECE is in percent. Best method selection is retrospective on test means. Calibrator ties at reported precision use the order T, L, D, A, R.

Setting	CE	FL	LS	MMCE	MDCA	s-AvUC	MbLS	NeSCal
				NLL ↓				
CIFAR-10 / R50	0.1863 ± 0.0084^{T}	$\mathit{0.1790 \pm 0.0052}^{L}$	0.2108 ± 0.0046^{D}	0.1993 ± 0.0061^{L}	0.1858 ± 0.0144^{T}	0.1855 ± 0.0121^{L}	0.2156 ± 0.0064^{A}	$\mathbf{0.1698 \pm 0.0046}^{T}$
CIFAR-10 / WRN	$\mathit{0.1495 \pm 0.0129}^{L}$	0.1508 ± 0.0043^{L}	0.1637 ± 0.0028^{L}	0.1726 ± 0.0063^{L}	0.1618 ± 0.0115^{L}	0.1540 ± 0.0117^{L}	0.1599 ± 0.0095^{L}	$\mathbf{0.1444 \pm 0.0041}^{L}$
CIFAR-100 / R50	0.8489 ± 0.0267^{T}	$\mathit{0.8452 \pm 0.0038}^{T}$	0.8944 ± 0.0190^{A}	0.8631 ± 0.0402^{T}	0.8577 ± 0.0137^{T}	0.8478 ± 0.0137^{T}	0.8558 ± 0.0168^{T}	$\mathbf{0.7944 \pm 0.0191}^{T}$
CIFAR-100 / WRN	0.8070 ± 0.0061^{T}	$\mathit{0.7681 \pm 0.0057}^{T}$	0.8334 ± 0.0049^{D}	0.8309 ± 0.0151^{T}	0.8141 ± 0.0155^{T}	0.8212 ± 0.0099^{T}	0.8173 ± 0.0103^{T}	$\mathbf{0.7613 \pm 0.0178}^{T}$
Tiny-ImageNet	1.4415 ± 0.0246^{T}	1.4950 ± 0.0099^{T}	1.4563 ± 0.0255^{T}	1.4614 ± 0.0319^{T}	$\mathit{1.4096 \pm 0.0228}^{T}$	$\mathbf{1.4066 \pm 0.0307}^{T}$	1.4208 ± 0.0246^{T}	1.4098 ± 0.0255^{T}
Helena	2.5618 ± 0.0015^{T}	2.5760 ± 0.0058^{D}	$\mathbf{2.5461 \pm 0.0099}^{D}$	2.5615 ± 0.0036^{T}	2.5653 ± 0.0025^{T}	2.5624 ± 0.0027^{T}	$\mathit{2.5463 \pm 0.0039}^{D}$	2.5497 ± 0.0049^{T}
Jannis	0.7085 ± 0.0014^{D}	0.7267 ± 0.0040^{D}	$\mathit{0.7068 \pm 0.0019}^{D}$	0.7090 ± 0.0025^{D}	0.7085 ± 0.0015^{D}	0.7084 ± 0.0011^{D}	0.7094 ± 0.0021^{D}	$\mathbf{0.7038 \pm 0.0010}^{D}$
Covertype	0.2461 ± 0.0023^{D}	0.2728 ± 0.0023^{R}	0.2483 ± 0.0028^{D}	0.2455 ± 0.0011^{A}	0.2485 ± 0.0013^{A}	$\mathit{0.2444 \pm 0.0033}^{A}$	0.2454 ± 0.0016^{D}	$\mathbf{0.2443 \pm 0.0014}^{A}$
20NG	0.7305 ± 0.0267^{A}	0.7008 ± 0.0318^{L}	$\mathit{0.6301 \pm 0.0161}^{D}$	0.6969 ± 0.0201^{A}	0.7337 ± 0.0377^{A}	0.7209 ± 0.0281^{T}	$\mathbf{0.6075 \pm 0.0171}^{L}$	0.7285 ± 0.0219^{A}
Amazon (5-class)	0.7534 ± 0.0024^{L}	0.7534 ± 0.0025^{R}	$\mathbf{0.7486 \pm 0.0027}^{L}$	$\mathit{0.7499 \pm 0.0016}^{L}$	0.7513 ± 0.0021^{L}	0.7526 ± 0.0015^{L}	0.7513 ± 0.0007^{L}	0.7521 ± 0.0016^{L}
Amazon (binary)	0.2111 ± 0.0021^{R}	0.2097 ± 0.0052^{A}	$\mathit{0.2045 \pm 0.0012}^{L}$	$\mathbf{0.2012 \pm 0.0010}^{D}$	0.2107 ± 0.0047^{R}	0.2098 ± 0.0032^{R}	0.2086 ± 0.0041^{R}	0.2052 ± 0.0014^{R}
Synthetic-noise	1.8432 ± 0.0224^{T}	1.8465 ± 0.0243^{A}	1.8475 ± 0.0222^{T}	1.8434 ± 0.0229^{T}	1.8448 ± 0.0215^{T}	1.8450 ± 0.0228^{T}	$\mathit{1.8431 \pm 0.0223}^{T}$	$\mathbf{1.8294 \pm 0.0241}^{T}$
Synthetic-geometry	0.9275 ± 0.0205^{D}	0.9262 ± 0.0221^{D}	$\mathbf{0.9243 \pm 0.0219}^{D}$	0.9271 ± 0.0193^{D}	0.9417 ± 0.0162^{D}	0.9286 ± 0.0201^{D}	$\mathit{0.9250 \pm 0.0187}^{D}$	0.9291 ± 0.0183^{D}
				ECE (%) ↓				
CIFAR-10 / R50	$\mathit{0.64 \pm 0.12}^{T}$	0.85 ± 0.16^{A}	1.23 ± 0.15^{A}	0.81 ± 0.16^{T}	0.67 ± 0.16^{T}	0.66 ± 0.18^{T}	1.22 ± 0.17^{A}	$\mathbf{0.59 \pm 0.12}^{A}$
CIFAR-10 / WRN	0.67 ± 0.15^{A}	0.76 ± 0.17^{L}	$\mathbf{0.56 \pm 0.05}^{R}$	0.63 ± 0.07^{R}	0.63 ± 0.15^{L}	0.88 ± 0.11^{A}	0.65 ± 0.15^{R}	$\mathit{0.59 \pm 0.04}^{L}$
CIFAR-100 / R50	1.61 ± 0.25^{T}	1.63 ± 0.25^{L}	$\mathit{1.42 \pm 0.36}^{L}$	1.80 ± 0.32^{A}	1.68 ± 0.28^{T}	1.88 ± 0.25^{T}	1.60 ± 0.27^{L}	$\mathbf{1.11 \pm 0.39}^{L}$
CIFAR-100 / WRN	1.86 ± 0.23^{T}	1.70 ± 0.19^{R}	1.88 ± 0.25^{R}	1.92 ± 0.25^{T}	1.94 ± 0.12^{T}	1.85 ± 0.33^{T}	$\mathit{1.61 \pm 0.36}^{R}$	$\mathbf{1.54 \pm 0.23}^{T}$
Tiny-ImageNet	1.98 ± 0.25^{T}	2.20 ± 0.20^{A}	2.46 ± 0.37^{T}	1.80 ± 0.32^{T}	1.58 ± 0.11^{A}	$\mathbf{1.38 \pm 0.33}^{A}$	$\mathit{1.45 \pm 0.29}^{T}$	1.74 ± 0.23^{T}
Helena	1.39 ± 0.24^{D}	1.48 ± 0.17^{R}	$\mathbf{1.33 \pm 0.28}^{T}$	1.38 ± 0.14^{D}	1.42 ± 0.10^{D}	1.43 ± 0.30^{D}	$\mathit{1.34 \pm 0.30}^{D}$	1.36 ± 0.14^{D}
Jannis	$\mathit{0.87 \pm 0.19}^{D}$	1.18 ± 0.21^{D}	1.07 ± 0.36^{D}	1.01 ± 0.07^{T}	1.08 ± 0.19^{T}	0.92 ± 0.17^{R}	1.03 ± 0.29^{D}	$\mathbf{0.74 \pm 0.22}^{T}$
Covertype	0.61 ± 0.16^{R}	0.50 ± 0.10^{R}	$\mathit{0.42 \pm 0.17}^{R}$	0.53 ± 0.18^{R}	0.47 ± 0.08^{R}	0.59 ± 0.13^{R}	$\mathbf{0.39 \pm 0.05}^{D}$	0.67 ± 0.17^{R}
20NG	$\mathbf{1.44 \pm 0.25}^{D}$	2.06 ± 0.80^{D}	2.12 ± 0.45^{R}	2.14 ± 0.50^{R}	$\mathit{1.68 \pm 0.31}^{D}$	2.34 ± 0.16^{D}	1.96 ± 0.37^{D}	2.04 ± 0.28^{D}
Amazon (5-class)	2.99 ± 0.12^{L}	2.80 ± 0.34^{R}	3.00 ± 0.24^{L}	$\mathbf{2.42 \pm 0.24}^{L}$	$\mathit{2.63 \pm 0.23}^{L}$	3.29 ± 0.39^{L}	3.16 ± 0.26^{L}	2.73 ± 0.17^{L}
Amazon (binary)	1.38 ± 0.20^{R}	1.27 ± 0.36^{R}	1.39 ± 0.10^{R}	$\mathbf{0.93 \pm 0.18}^{D}$	1.49 ± 0.21^{R}	1.87 ± 0.33^{R}	1.50 ± 0.42^{R}	$\mathit{1.22 \pm 0.28}^{R}$
Synthetic-noise	$\mathbf{1.96 \pm 0.88}^{R}$	2.84 ± 0.63^{R}	2.60 ± 0.68^{R}	2.59 ± 1.00^{R}	3.04 ± 1.55^{R}	2.96 ± 0.91^{R}	2.67 ± 1.01^{R}	$\mathit{2.12 \pm 0.96}^{R}$
Synthetic-geometry	1.97 ± 0.57^{R}	1.71 ± 0.14^{R}	1.93 ± 0.59^{R}	$\mathbf{1.47 \pm 0.31}^{R}$	2.81 ± 1.36^{R}	2.43 ± 0.29^{R}	$\mathit{1.57 \pm 0.51}^{R}$	1.98 ± 0.08^{R}

NeSCal attains the best or second-best AUROC (higher is better) and AURC (lower is better) on all CIFAR settings, winning outright in three, both on the base model and after TS. Relative to CE, base-model AURC decreases by approximately 15.5% on CIFAR-10/ResNet-50 and 13.5% on CIFAR-100/ResNet-50. (Table 12).

Table 12: Failure-prediction AUROC (higher is better) and AURC ($\times 10^3$, lower is better), before and after temperature scaling. Means over seeds; SDs omitted.

Setting	Metric	CE	FL	LS	MMCE	MDCA	s-AvUC	MbLS	NeSCal
CIFAR-10 / R50	AUROC	0.9357	0.9278	0.8537	*0.9359*	0.9351	0.9352	0.8995	**0.9392**
	AUROC+TS	0.9352	0.9270	0.8563	*0.9361*	0.9349	0.9349	0.9001	**0.9395**
	AURC	6.60	8.57	19.08	7.25	*6.53*	*6.53*	11.21	**5.58**
	AURC+TS	6.64	8.66	18.69	7.24	*6.55*	*6.55*	11.11	**5.57**
CIFAR-10 / WRN	AUROC	0.9343	0.9290	0.8951	**0.9393**	0.9379	0.9350	0.9340	*0.9384*
	AUROC+TS	0.9316	0.9295	0.8957	**0.9391**	0.9362	0.9322	0.9331	*0.9370*
	AURC	*5.18*	6.58	8.89	5.64	5.51	5.46	5.69	**4.92**
	AURC+TS	*5.41*	6.53	8.83	5.64	5.67	5.73	5.77	**5.04**
CIFAR-100 / R50	AUROC	0.8695	0.8633	0.8594	*0.8709*	0.8693	0.8697	0.8677	**0.8764**
	AUROC+TS	0.8699	0.8637	0.8601	*0.8713*	0.8694	0.8700	0.8674	**0.8763**
	AURC	69.84	*69.26*	73.31	70.60	71.05	69.78	69.74	**60.41**
	AURC+TS	69.75	*69.14*	72.95	70.54	71.03	69.71	69.91	**60.44**
CIFAR-100 / WRN	AUROC	0.8771	0.8688	0.8744	0.8761	0.8755	*0.8777*	0.8747	**0.8826**
	AUROC+TS	0.8773	0.8690	0.8742	0.8763	0.8759	*0.8779*	0.8741	**0.8817**
	AURC	61.29	57.50	**52.95**	64.97	62.47	62.93	60.19	*54.83*
	AURC+TS	61.17	57.44	**53.05**	64.92	62.39	62.91	60.40	*55.07*
Tiny-ImageNet	AUROC	0.8594	0.8441	0.8506	0.8567	*0.8614*	**0.8644**	0.8551	0.8611
	AUROC+TS	0.8587	0.8443	0.8537	0.8557	0.8606	**0.8637**	0.8564	*0.8608*
	AURC	128.52	147.03	130.41	132.39	*123.87*	**122.88**	125.62	126.24
	AURC+TS	128.86	146.95	128.97	132.82	*124.25*	**123.23**	124.98	126.36

E.6 Robustness

Table 13 reports corruption robustness and SVHN detection. NeSCal improves corrupted NLL and ECE over CE in four of five image settings and accuracy in all five, attaining the highest corrupted accuracy among the methods in Table 13 in four settings. OOD detection results are mixed: AUROC improves over CE on both CIFAR-100 settings and Tiny-ImageNet, but decreases on both CIFAR-10 settings. On Tiny-ImageNet, NeSCal attains the highest reported SVHN AUROC, 0.9221.

Table 13: Robustness: corruption metrics averaged over five severity levels and 15 corruption types for CIFAR-10/100-C or 14 for Tiny-ImageNet-C; SVHN detection AUROC uses maximum softmax probability.

Setting	CE	FL	LS	MMCE	MDCA	s-AvUC	MbLS	NeSCal
Corrupted NLL								
CIFAR-10 / R50	1.2363	0.9956	1.0565	**0.9828**	1.2799	1.2882	1.0554	1.2483
CIFAR-10 / WRN	1.3416	1.0069	1.0651	**0.9919**	1.3808	1.4140	1.1209	1.2342
CIFAR-100 / R50	2.6843	**2.4157**	2.4532	2.6950	2.7057	2.6799	2.5766	2.4443
CIFAR-100 / WRN	3.1114	**2.5448**	2.6913	2.9448	3.1199	2.9370	2.6783	2.5604
Tiny-ImageNet	4.1629	4.0258	3.8927	4.1066	4.1212	4.1864	**3.8153**	3.9413
Corrupted ECE (%)								
CIFAR-10 / R50	15.89	8.89	13.83	**8.80**	16.25	16.45	14.50	15.99
CIFAR-10 / WRN	17.43	10.68	14.35	**9.48**	17.68	18.11	15.51	16.07
CIFAR-100 / R50	21.36	**12.83**	16.48	21.44	21.53	21.18	19.79	18.58
CIFAR-100 / WRN	25.46	**16.13**	17.77	24.04	25.77	23.79	20.45	18.98
Tiny-ImageNet	22.21	**13.88**	15.24	20.41	22.13	22.83	14.08	17.17
Corrupted accuracy (%)								
CIFAR-10 / R50	71.66	**72.79**	72.58	70.94	71.46	71.29	72.35	72.43
CIFAR-10 / WRN	73.85	73.70	73.05	72.24	72.14	74.20	73.11	**74.55**
CIFAR-100 / R50	47.60	47.76	47.44	47.26	47.26	47.47	47.47	**49.26**
CIFAR-100 / WRN	47.45	47.65	47.45	47.31	47.39	48.05	47.90	**49.27**
Tiny-ImageNet	24.73	22.47	24.55	24.07	24.88	24.71	24.99	**25.45**
SVHN OOD AUROC								
CIFAR-10 / R50	0.9355	0.8996	0.8592	**0.9462**	0.9267	0.9338	0.9064	0.9282
CIFAR-10 / WRN	0.8996	0.8715	0.9208	**0.9761**	0.9150	0.9190	0.9397	0.8765
CIFAR-100 / R50	0.7463	**0.8078**	0.7621	0.7259	0.7749	0.7357	0.7384	0.7692
CIFAR-100 / WRN	0.7340	0.7315	0.7388	0.7385	0.7096	0.7263	**0.7524**	0.7448
Tiny-ImageNet	0.9018	0.9085	0.8463	0.8875	0.8987	0.8967	0.8416	**0.9221**

E.7 Ablations

Component ablations use seeds 18 and 42, the cosine vs constant schedule comparison matches the headline in seed numbers, with activation epoch and regularization strength selected separately for each schedule. The Fisher-weighted arm uses $D_{\mathrm{F}}(p_i, p_j) = \frac{1}{8 \ln 2} \sum_k \frac{(p_{i,k} - p_{j,k})^2}{\max(p_{i,k}, 10^{-2})}$. The floor applies only to the source probabilities. This is a stabilized version of the local quadratic weighting in Proposition 4.

Graph construction. Projection-based graphs retain gains over CE in some of the evaluated settings, without a clear, consistent winner emerging. The choice of feature transformation is therefore not uniformly neutral. Table 14 displays the random-graph results discussed in Section 6.3.

Table 14: Test NLL for neighborhood controls; lower is better. Random uses headline λ; Random (scaled) matches penalty magnitude.

Setting	CE	kNN	Random	Random (scaled)
CIFAR-10 / R50	0.2065	**0.1843**	0.2156	*0.2058*
CIFAR-100 / R50	*0.8782*	**0.8026**	1.0386	0.8785
Helena	2.5651	**2.5534**	2.5890	*2.5639*

Constant versus cosine regularization. With each schedule selected by validation NLL, cosine decay improves NLL, Brier score, and accuracy in all four CIFAR settings; ECE improves in three (Table 15). On the two ResNet-50 settings with graph diagnostics, constant regularization produces

Table 15: Constant versus cosine regularization on CIFAR.

Setting	Schedule	NLL ↓	Brier ↓	Accuracy ↑	ECE ↓
CIFAR-10 / R50	Constant	0.1917 ± 0.0068	0.0941 ± 0.0033	0.9358 ± 0.0026	**0.0155 ± 0.0012**
	Cosine	**0.1843 ± 0.0058**	**0.0872 ± 0.0015**	**0.9428 ± 0.0015**	0.0229 ± 0.0011
CIFAR-10 / WRN	Constant	0.1966 ± 0.0092	0.0897 ± 0.0037	0.9413 ± 0.0028	0.0252 ± 0.0021
	Cosine	**0.1752 ± 0.0028**	**0.0785 ± 0.0018**	**0.9494 ± 0.0011**	**0.0239 ± 0.0018**
CIFAR-100 / R50	Constant	0.8728 ± 0.0238	0.3423 ± 0.0062	0.7541 ± 0.0052	0.0539 ± 0.0067
	Cosine	**0.8026 ± 0.0155**	**0.3178 ± 0.0050**	**0.7719 ± 0.0052**	**0.0395 ± 0.0067**
CIFAR-100 / WRN	Constant	0.8319 ± 0.0109	0.3230 ± 0.0068	0.7713 ± 0.0076	0.0636 ± 0.0057
	Cosine	**0.7760 ± 0.0173**	**0.3057 ± 0.0061**	**0.7810 ± 0.0065**	**0.0460 ± 0.0062**

lower final edge JS, whereas cosine decay produces lower across-seed error dispersion (Table 19). These observations support relaxing the consistency constraint during training: stronger agreement at the selected weights does not necessarily imply better predictive quality. They motivate cosine decay as the default without implying uniform improvement across calibration metrics.

Neighborhood structure. At matched initial penalty magnitude, random graphs provide little improvement over CE in the three evaluated settings. Using the headline λ produces worse NLL than CE in all three (Table 16), supporting the importance of which samples the graph connects.

Divergence sensitivity. Divergence arms are at matched initial penalty magnitude, with no divergence uniformly dominating among the stable, scale-matched runs. The common penalty target over-regularizes some settings, so cells marked † and divergent runs are excluded from the comparison of divergence quality.

Table 16: Ablations: test NLL, lower is better. – : not run. †: the common penalty target over-regularizes this setting; div.: diverged.

Arm	C10/R50	C10/WRN	C100/R50	C100/WRN	Helena	Jannis	Covertype	20NG
Headline (k=10, JS, raw features)	0.1843	0.1752	0.8026	0.7760	2.5534	0.7066	0.2499	0.8023
Cross-entropy (no regularizer)	0.2065	0.1972	0.8782	0.8785	2.5651	0.7125	0.2516	0.8127
Random graph, same λ	0.2156	–	1.0386	–	2.5890	–	–	–
Random graph, matched penalty	0.2058	–	0.8785	–	2.5639	–	–	–
$k = 2$	0.1830	0.1736	0.8368	0.7936	2.5554	0.7069	0.2487	0.7466
$k = 5$	0.1789	0.1702	0.8148	0.7985	2.5553	0.7072	0.2531	0.7650
$k = 20$	0.1777	0.1765	0.8106	0.7993	2.5561	0.7072	0.2514	0.7726
$k = 50$	0.1781	0.1754	0.8204	0.7974	2.5573	0.7079	0.2551	0.7795
Graph on PCA-50	0.1777	0.1774	0.8355	0.8440	2.5549	0.7053	0.2505	0.7608
Graph on PCA-16	0.1822	0.1702	0.8581	0.8354	2.5553	0.7051	0.2516	0.7580
Graph on UMAP-30	0.1801	0.1748	0.8291	0.7758	2.5535	0.7050	0.2498	0.7586
Divergence: JS	1.1048†	–	0.8167	0.7642	2.5604	0.7128	0.3903†	0.8382
Divergence: squared L_2	0.2548†	–	0.8084	0.7498	2.6115	0.7182	0.3905†	0.8263
Divergence: KL	0.2749†	–	0.8383	0.8154	2.5806	0.7145	0.4239†	0.8869
Divergence: Fisher-weighted	div.	div.	0.8754	0.9818	2.6091	0.7057	0.3933†	0.7666

E.8 Activation rules

If the activation epoch could be placed from a quantity observable during the warm-up, NeSCal would lose one swept hyperparameter. We tested three such rules on the cross-entropy trajectory:

- *entropy drop*: mean predictive entropy falls below a fraction of its initial value;
- *CWCC*: class-weighted cross-class connectivity falls below a fraction of its initial value;
- *probe*: kNN-probe accuracy reaches a fraction of its maximum over the warm-up.

For these rule experiments, diagnostics are logged every 10 epochs and activation gates are rounded down to a separate 20-epoch checkpoint grid. Table 17 reports each rule's regret, the test-NLL difference against the grid-selected onset at the same λ.

No rule transfers consistently across settings. Probe-rule regret ranges from -0.039 to $+0.072$ nats. Because this rule uses the maximum probe accuracy over the warm-up trajectory, it is a retrospective diagnostic rather than an online activation rule.

Table 17: Regret of activation rules: test NLL relative to the grid-selected onset at the same λ; positive is worse. –: rule not evaluated.

Setting	Entropy drop	CWCC	Probe (hindsight)
CIFAR-10 / R50	$+0.0222$	-0.0018	$+0.0178$
CIFAR-10 / WRN	-0.0012	$+0.0298$	$+0.0170$
CIFAR-100 / R50	$+0.0940$	$+0.2987$	$+0.0123$
CIFAR-100 / WRN	$+0.0439$	-0.0427	-0.0388
Tiny-ImageNet	$+0.0686$	–	$+0.0718$
Helena	–	–	$+0.0075$
Jannis	$+0.4471$	–	$+0.0005$
Covertype	–	–	$+0.0000$
20NG	$+0.0393$	$+0.0393$	-0.0059

E.9 Selection criterion

Every train-time method is selected by validation NLL across checkpoints and hyperparameters. Selecting by ECE can favor less informative predictions. Across the eleven settings, ECE-based selection improves NeSCal's test ECE by 0.85 percentage points on average but worsens NLL by 0.017 nats. On CIFAR-100/R50, for example, ECE improves from 3.95% to 2.47% while NLL worsens from 0.8026 to 0.871. Similar trade-offs occur among the baselines. These results support the shared proper-score criterion while confirming that it does not optimize each method's ECE.

E.10 Training cost

Table 18 reports training-loop time, one-off graph-construction time, and peak GPU memory. Peak GPU memory is the maximum allocated at any point in the run, including the one-off graph construction. Where NeSCal 's peak clearly exceeds CE's (Tiny-ImageNet, CIFAR-100/WRN and the tabular settings), the maximum occurs during the chunked similarity computation of graph construction; differences of a few hundredths of a GB are within run-to-run allocator variation. Relative overhead is small on the image models and larger on the smaller tabular and text models, where the absolute increase in the reported training time is at most 27.4 seconds. Peak memory increases vary across settings. Inference is unchanged.

Table 18: Training cost: wall-clock training time (s), relative overhead, one-off graph construction (s), and peak GPU memory (GB). Means follow headline seeds except CIFAR-10/ResNet-50 NeSCal (one seed).

Setting	CE time	NeSCal time	Overhead	Graph build	CE memory	NeSCal memory
CIFAR-10 / R50	3430	3456	$+0.7\%$	20.4	2.05	2.05
CIFAR-10 / WRN	5028	5053	$+0.5\%$	26.9	2.29	2.26
CIFAR-100 / R50	2702	2725	$+0.9\%$	17.3	2.11	2.15
CIFAR-100 / WRN	5068	5094	$+0.5\%$	27.1	2.59	3.05
Tiny-ImageNet	11036	11085	$+0.4\%$	146.2	3.89	6.89
Helena	36.5	50.1	$+37\%$	1.49	0.04	1.44
Jannis	148.9	168.7	$+13\%$	2.72	0.03	0.24
Covertype	71.6	99.0	$+38\%$	2.42	0.03	0.29
20NG	61.0	66.9	$+10\%$	1.65	0.61	0.62
Amazon (5-class)	3.37	4.63	$+38\%$	0.56	0.02	0.14
Amazon (binary)	1.38	2.73	$+97\%$	0.39	0.02	0.08

E.11 Theory diagnostics

We evaluate bound tightness and related diagnostics on the frozen training graphs for CIFAR-10 and CIFAR-100 with ResNet-50, using CE and both NeSCal schedules over four seeds (Table 19).

All evaluated bounds hold on every edge in all evaluated configurations. As expected of worst-case inequalities, they are conservative: median confidence-bound tightness is 0.02–0.04 on CIFAR-10

and 0.10–0.13 on CIFAR-100, where neighboring disagreement is larger. We therefore use them to identify which quantities govern calibration propagation rather than to predict its magnitude. The Markov bound of Proposition 6 is informative when the penalty is small relative to the threshold: on CIFAR-10 it bounds the fraction of edges with JS ≥ 0.25 by 0.52, against an observed 0.15; on CIFAR-100, where mean disagreement is higher, it is not informative at this threshold.

The diagnostics bearing on Proposition 3 are consistent with its mechanism. Cosine-scheduled NeSCal reduces the across-seed dispersion of neighboring errors and of individual confidences on both settings, and both schedules reduce the NLL gap to a test-fitted temperature. Final edge agreement behaves differently across schedules: the constant schedule lowers mean edge JS relative to CE on both settings, whereas under cosine decay λ_t approaches zero by the end of training, and edge JS at the selected checkpoint is close to CE's. Better test NLL therefore need not coincide with the smallest final penalty, consistent with the regularizer acting on the dispersion of neighboring errors rather than on end-state agreement alone. Finally, the quadratic expansion of Corollary 3 describes the geometry near agreement; at the trained weights, neighboring predictions lie outside this regime on nearly all edges, so we treat the Fisher-type weighting as an interpretation of the penalty's local geometry rather than a description of training, in line with the ablations, where Fisher weighting shows no consistent advantage.

Table 19: Theory diagnostics at the selected weights on the frozen training graph. Tightness concerns the observable predictive and confidence bounds. The dispersion proxy corresponds to the equal-bias, equal-variance specialization of Proposition 3, not the full expected squared error difference.

	CIFAR-10 / R50			CIFAR-100 / R50		
Quantity	CE	NeSCal (const)	NeSCal (cos)	CE	NeSCal (const)	NeSCal (cos)
Bound violations (L1, confidence)	0	0	0	0	0	0
Confidence-bound tightness, median	0.019	0.038	0.019	0.131	0.117	0.101
Edges within the expansion's regime	0%	0%	0%	0.005%	0.005%	0.005%
Across-seed dispersion proxy $\sigma\sqrt{2(1-\rho)}$	0.055	0.061	0.038	0.102	0.102	0.088
Across-seed confidence spread σ_c	0.059	0.059	0.035	0.103	0.112	0.076
NLL gap to the test-fitted oracle temperature	0.020	0.005	0.015	0.030	0.025	0.008
Mean edge JS at the selected weights	0.134	0.102	0.129	0.317	0.307	0.323
Reciprocal edges	–	0.435	0.435	–	0.439	0.439

Synthetic setup. Each scenario draws equiprobable Gaussian classes in $\mathbb{R}^{20}$ with identity covariance, whose centers differ only in the first two coordinates. In the label-noise scenario, the centers lie on a circle of radius 8. In the geometric scenario, half lie on that circle and half on a circle of radius 2.8 centered at $(12.8, 0)$, so the second group's classes are more closely spaced. In both scenarios, the labels of half the classes are redrawn uniformly at random with probability $H \in \{0, 0.2, 0.4, 0.6\}$. Data are split $60/20/20$ into training, validation and test sets, and the model is an MLP with two hidden layers of width 64 and a 16-dimensional representation. NeSCal uses $k = 10$ and constant regularization, with activation epoch and strength selected by validation NLL and held fixed across the heterogeneity sweep.

Evaluation compares predictions with pointwise reference distributions q_i and reports the top-label reference error $\frac{1}{n}\sum_i \big(c_i - q_{i,\hat{y}_i}\big)^2$, the mean squared pointwise calibration error of Section 2 with $a_i = q_{i,\hat{y}_i}$. In the label-noise scenario, q_i is the analytical noisy-label posterior. In the geometric scenario, q_i mixes the clean posterior with the uniform distribution at weight H when sample i's generating class is noisy, and equals the clean posterior otherwise. This coincides with the noisy-label posterior wherever the class posterior is concentrated within either the noisy or the clean group, and differs from it only where classes from the two groups overlap. The comparator labeled "oracle" chooses, on the test set, the temperature minimizing this error among 47 candidates (six equally spaced in $[0.5, 1)$ and 41 in $[1, 5]$); it is a test-informed diagnostic rather than a deployable calibrator.

The synthetic regimes. Figure 1 compares NeSCal with CE and the test-informed temperature search. In the label-noise scenario, the regional confidence biases have opposite signs, which no single temperature can correct together, and NeSCal attains lower top-label reference error than the searched temperatures at every evaluated $H > 0$. In the geometric scenario, the biases share a sign: the temperature search is best, and NeSCal has higher reference error than CE. Both outcomes are

consistent with the idealized analysis of Proposition 3. Averaging an edge changes its summed expected squared error by $\frac{1}{2}(a_i - a_j)^2 - \frac{1}{2}\mathbb{E}[(e_i - e_j)^2]$ (Appendix A.9), so smoothing helps only when neighboring errors differ more than neighboring correctness probabilities do. When neighbors share their error, as under a common same-sign bias, the second term is small, and any correctness gap, which the more closely spaced classes of the geometric scenario plausibly create, makes smoothing increase error. A global temperature acts on the shared component, is therefore the appropriate correction. The geometric scenario therefore exhibits the failure mode the analysis predicts, and the label-noise scenario the regime in which smoothing and rescaling are not interchangeable.

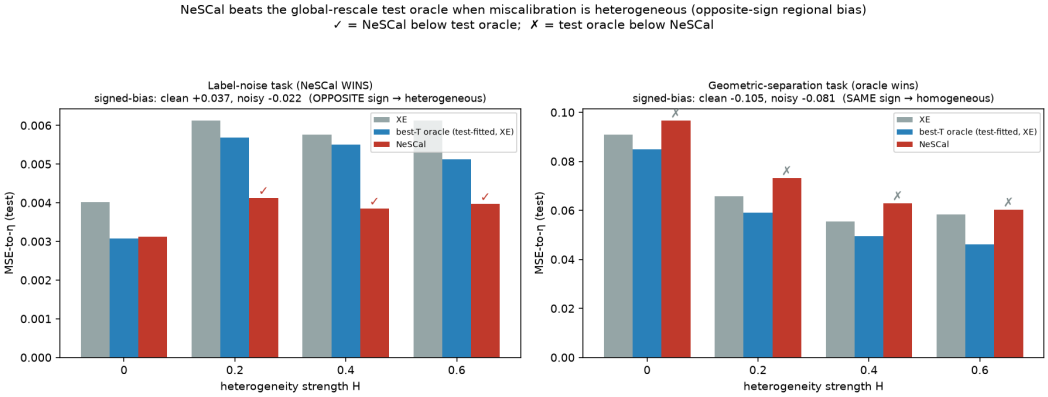

Figure 1: Mean squared top-label reference error across heterogeneity strengths H, per scenario, on CE, CE with the test-informed temperature (*oracle*), and NeSCal.

E.12 Perturbation studies

These studies test neighborhood smoothing under class imbalance, which can make neighborhoods less representative for rare classes and label noise, which weakens the supervised anchor. Both studies use CIFAR-100 with the clean-selected configurations of R50 and WRN (Appendix D.9).

Label noise. NeSCal lowers ECE relative to CE in all four architecture–noise conditions, but improves NLL and accuracy in only one (Table 20). These results show that improved aggregate confidence calibration under label noise need not accompany improved predictive quality.

Table 20: Symmetric label noise on CIFAR-100, without re-tuning λ.

		NLL ↓		ECE (%) ↓		Accuracy (%) ↑	
Architecture	Noise	CE	NeSCal	CE	NeSCal	CE	NeSCal
ResNet-50	20%	**1.2521**	1.3123	6.42	**4.32**	**68.20**	67.64
	40%	**1.7125**	1.9218	15.96	**9.66**	**61.01**	56.95
WRN-28-10	20%	1.1743	**1.1528**	2.92	**1.70**	70.19	**70.44**
	40%	**1.5154**	1.6212	9.58	**4.85**	**64.35**	61.74

Class imbalance. Under long-tailed training, the JS penalty improves NLL over CE in three of six settings. Squared L_2 attains the lowest NLL in all six. Because these arms use equal λ rather than matched penalty magnitude, the effect of divergence weighting (Table 21) is not isolated.

Table 21: Long-tailed CIFAR-100: test NLL for CE and three divergences at equal λ.

Architecture	Imbalance	Cross-entropy	JS	Fisher-weighted	Uniform (L_2)
ResNet-50	10	1.8104	1.8018	1.8403	**1.7629**
	50	2.9898	3.0154	3.0717	**2.8144**
	100	3.4013	3.4364	3.5764	**3.2381**
WRN-28-10	10	1.7965	1.6838	1.6325	**1.5680**
	50	2.5238	2.4804	2.6973	**2.3800**
	100	2.9397	2.9967	3.2001	**2.8844**