Semantic Geometric Calibration in Randomized Neural Feature Space

Itay Abuhazera[1], **Liron Cohen**[1] and **Gil Einziger**[1]
[1]Stein Faculty of Computer and Information Science, Ben-Gurion University of the Negev, Israel
itayab@post.bgu.ac.il, cliron@bgu.ac.il, gilein@bgu.ac.il

Abstract

Modern neural networks can be accurate yet poorly calibrated. We present Random Geometric Calibration (RGC), a post hoc method that augments confidence estimation with a geometric signal derived from distances to training data in an aggregated semantic representation space. Prior geometric calibration methods in feature space often rely on architecture-specific layer choices, which can be unstable across models and datasets. To avoid this dependence, RGC samples a fixed number of intermediate blocks uniformly at random and reuses this set at both calibration and test time.

We instantiate RGC with $\mathrm{RGC_L}$, which applies spatial pyramid pooling to each sampled activation, projects features to a fixed dimension, and computes a nearest-neighbor separation score that is mapped to probabilities via isotonic regression. We also present $\mathrm{RGC_C}$, which replaces pooling and projection with uniform coordinate sampling for improved computational efficiency.

We evaluate $\mathrm{RGC_L}$ and $\mathrm{RGC_C}$ across multiple datasets, including CIFAR-10, CIFAR-100, and Tiny-ImageNet, and across a diverse set of architectures such as ResNet, DenseNet, and DINOv2. Our results show that $\mathrm{RGC_L}$ achieves state-of-the-art or near state-of-the-art calibration performance, reducing expected calibration error by approximately 78% on average and by up to 97% for some models.

Finally, we provide representation analysis that explains why $\mathrm{RGC_L}$ and $\mathrm{RGC_C}$ succeed without layer selection and characterize their computational overheads. We observe that layers exhibiting favorable geometric structure tend to emerge in later network blocks, and that randomized aggregation implicitly emphasizes these layers, leading to robust and reliable calibration performance.

1 Introduction

Neural network classifiers underpin critical applications in computer vision, natural language processing, medical diagnosis, and autonomous systems [Caruana and Niculescu-Mizil, 2006; Zhang and Haghani, 2015]. While modern architectures achieve remarkable accuracy, accuracy alone does not capture predictive reliability. A model's *accuracy* summarizes the probability of a correct prediction over a test set, but does not quantify the reliability of individual predictions.

Although modern classifiers expose native confidence estimates, these are often biased. Confidence calibration aligns reported confidence with empirical correctness, transforming raw scores into interpretable uncertainty estimates. Calibration methods are commonly divided into post-hoc approaches, which fit a transformation on top of a trained model [Kull *et al.*, 2019; Guo *et al.*, 2017; Gupta and Ramdas, 2021], and ad-hoc approaches, which modify the training algorithm to produce calibrated confidence estimations by construction [Thulasidasan *et al.*, 2019; Hendrycks *et al.*, 2019]. Post-hoc methods are attractive when pretrained models are reused, but they can only correct systematic bias rather than introduce new information. In contrast, ad-hoc methods can encode richer uncertainty signals but require retraining or confidence-aware fine-tuning, which is often impractical.

These limitations motivate approaches that preserve the base model while augmenting confidence estimation with an additional signal. A natural candidate is geometry. Neural networks implicitly project data into high-dimensional representation spaces and separate classes within them. Low-confidence predictions tend to arise near decision boundaries, where inputs are close to samples of different labels, or for atypical inputs that are far from nearby training examples [Mandelbaum and Weinshall, 2017; Jiang *et al.*, 2018; Papernot and McDaniel, 2018; Lee *et al.*, 2018; Tomani *et al.*, 2023; Xiong *et al.*, 2023; Yüksekgönül *et al.*, 2023]. Since we focus on high-accuracy models, which necessarily induce meaningful class separation, geometric structure provides a promising signal for confidence estimation.

Existing representation-space geometric methods depend critically on layer selection: practitioners must identify which intermediate layers yield representations with sufficient class separation. For example, DAC [Tomani *et al.*, 2023] and TULIP [Benkert *et al.*, 2024] require architecture-specific configurations, with DAC using prescribed layer sets and TULIP using manually determined placement rules that lack guidance beyond ResNets. Optimal selection of layers varies across architectures, datasets, and even training runs.

In this work, we ask whether such careful layer selection is

truly necessary. Specifically, we ask: *Is careful layer selection actually required for effective geometric calibration?*

1.1 Contributions

1. We introduce **Random Geometric Calibration** (RGC)[code], a post-hoc geometric calibration framework that consistently outperforms prior work, reducing calibration error by approximately 78% on average relative to uncalibrated predictions. Instead of performing layer selection, we instantiate RGC with layer-based randomized sampling ($\mathrm{RGC_L}$) and with coordinate-based randomized sampling ($\mathrm{RGC_C}$).
2. We show that $\mathrm{RGC_L}$ and $\mathrm{RGC_C}$ **eliminate the need for explicit layer selection** by replacing it with randomized sampling, achieving calibration performance comparable to carefully tuned layer-selection strategies.
3. In $\mathrm{RGC_L}$, representations from the sampled layers are efficiently aggregated using spatial pyramid pooling. We further introduce $\mathrm{RGC_C}$, a more computationally efficient variant that attains performance comparable to $\mathrm{RGC_L}$. The efficiency of $\mathrm{RGC_C}$ stems from the fact that it requires no feature compression and instead directly samples a fixed number of coordinates in the feature space. We provide an equivalence analysis showing that $\mathrm{RGC_L}$ and $\mathrm{RGC_C}$ achieve statistically indistinguishable performance at the 95% confidence level in 9 out of 13 model–dataset combinations.
4. Our evaluation demonstrates an average calibration error reduction of approximately 12% compared to several recent geometric calibrators, and up to 19% on individual configurations. We additionally analyze the computational overheads and show that $\mathrm{RGC_C}$ is considerably faster than $\mathrm{RGC_L}$, while both methods require a longer one-time preprocessing step than prior work. Despite this overhead, both methods achieve several hundred confidence estimates per second at inference time.

2 Background and Related Work

This section defines confidence calibration and situates RGC within prior work on post-hoc and geometric calibration.

2.1 Confidence Calibration

Consider a classifier $f\colon \mathcal{X} \to \Delta^{C-1}$ that outputs a probability vector over C classes. Let $\hat{y}$ denote the predicted class and let $\hat{p}$ be the associated confidence (e.g., the maximum predicted probability). The model is said to be *perfectly calibrated* if

$$\mathbb{P}(\hat{y} = y \mid \hat{p} = p) = p, \quad \forall p \in [0, 1].$$

In practice, calibration is assessed using binning-based estimates such as the *Expected Calibration Error (ECE)*. We partition predictions into M bins $\{B_m\}_{m=1}^{M}$ according to $\hat{p}$. The ECE is defined as

$$\mathrm{ECE} = \sum_{m=1}^{M} \frac{|B_m|}{N} \left|\mathrm{acc}(B_m) - \mathrm{conf}(B_m)\right|, \qquad (1)$$

where N is the total number of samples, $\mathrm{acc}(B_m)$ denotes the empirical accuracy in bin B_m, and $\mathrm{conf}(B_m)$ denotes the average predicted confidence in that bin.

2.2 Post Hoc Calibration

Logit-based methods. Temperature Scaling (TS) [Guo *et al.*, 2017] fits a single scalar T on a validation set and rescales logits as z/T before applying softmax. TS can substantially reduce ECE on in-distribution data, but as a monotone transformation it cannot correct failures in the underlying ranking of confidence (e.g., confident mistakes that remain confident). More expressive alternatives such as isotonic regression [Zadrozny and Elkan, 2002] or Dirichlet calibration [Kull *et al.*, 2019] fit richer mappings, but still rely exclusively on the model's logits or predicted probabilities. When these signals degrade under shift, purely logit-based calibration has no independent information to exploit.

Ensemble and Bayesian methods. Deep ensembles [Lakshminarayanan *et al.*, 2017] and MC Dropout [Gal and Ghahramani, 2016] can produce strong uncertainty estimates by aggregating multiple predictors or stochastic forward passes. However, these approaches typically require multiple models, architectural modifications, or increased inference cost. Our focus is on *single-model, post-hoc* methods. RGC is complementary to ensemble-based approaches and can be applied on top of ensemble predictions.

2.3 Geometric and Representation-Space Methods

Geometric confidence signals. Geometric approaches derive confidence from distances to training data or local neighborhoods [Mandelbaum and Weinshall, 2017; Jiang *et al.*, 2018; Papernot and McDaniel, 2018]. A representative example is the *separation score* used in pixel-space methods [Chouraqui *et al.*, 2022], which compares the distance from a test input to its nearest same-class neighbor versus its nearest different-class neighbor:

$$s(x) = \frac{D(x, n_{\mathrm{non}}(x)) - D(x, n_{\mathrm{friend}}(x))}{2}. \qquad (2)$$

While effective in settings where pixel geometry aligns with semantic class structure, such measures are less robust for normalized inputs or modalities where pixel-level distances are poorly aligned with class identity.

Latent-space methods and layer selection. To bridge this semantic gap, recent methods compute geometric or density signals in internal representation spaces [Lee *et al.*, 2018; Tomani *et al.*, 2023; Benkert *et al.*, 2024], typically requiring architecture-specific layer configurations (Section 1). Our RGC instead replaces layer selection with uniform sampling, reducing tuning and mitigating overfitting risks.

3 Randomized Geometric Calibration

This section introduces *Randomized Geometric Calibration* (RGC), a post-hoc framework that augments confidence estimation with a geometric signal computed in an aggregated representation space. The central departure from prior representation-space methods is that RGC removes layer selection as a tuned component. Instead of prescribing architecture-specific taps (e.g., DAC, TULIP) [Tomani *et al.*, 2023; Benkert *et al.*, 2024], RGC samples intermediate indices uniformly at random and uses the resulting representation to derive a separation-based confidence score.

Algorithm 1 Randomized Geometric Calibration (RGC)

Input: Classifier f, training set $\mathcal{D}_{\text{tr}}$, validation set $\mathcal{D}_{\text{val}}$, number of layers L, dimension d (RGC_L) or K (RGC_C)
Output: Calibrator parameters $(\mathcal{L}, \text{feat-params}, \mathcal{S}_{\text{val}}, g)$

Offline (fit)
Sample random indices
for $x \in \mathcal{D}_{\text{tr}} \cup \mathcal{D}_{\text{val}}$ **do**
    Compute normalized embedding $z(x)$
end for
Build an NN index over $\{z(x_i) : (x_i, y_i) \in \mathcal{D}_{\text{tr}}\}$.
for $(x_i, y_i) \in \mathcal{D}_{\text{val}}$ **do**
    Compute separation score $s(x_i)$
end for
Store sorted scores $\mathcal{S}_{\text{val}}$ and compute ranks $\tilde{s}(x_i)$ using Eq. 12
Fit isotonic regression g on $(\tilde{s}(x_i), \mathbf{1}[\hat{y}(x_i) = y_i])$.

Online (predict)
Compute normalized embedding $z(x)$
Compute $s(x)$ via NN queries
Normalize $\tilde{s}(x)$ using ranks in $\mathcal{S}_{\text{val}}$
return calibrated confidence $\hat{p}(x) = g(\tilde{s}(x))$.

Algorithm 1 summarizes RGC. Concretely, RGC consists of four steps:

1. **Random Index Sampling (RIS):** sample a fixed-size subset of representation components uniformly at random (e.g., intermediate layers or individual coordinates).
2. **Feature Construction:** map the sampled components to a compact normalized embedding $z(x)$.
3. **Geometric Scoring:** compute a separation score $s(x)$ from nearest-neighbor distances to training representations.
4. **Calibration Mapping:** map scores to calibrated probabilities via isotonic regression.

Unless stated otherwise, we use fixed hyperparameters $L = 6$ and $d = K = 256$ across all architectures and datasets.

The RGC framework defines a single randomized geometric calibration pipeline for which we study two instantiations of the sampling step. RGC_L samples intermediate layers and aggregates them using multi-scale pooling and random projection, while RGC_C samples activation coordinates to obtain a compact embedding with lower computational overhead. Both variants share the same geometric scoring and calibration mapping.

3.1 Random Index Sampling

Let f be a neural network with extractable intermediate representations indexed by $\mathcal{L}_{\text{all}} = \{1, \dots, L_{\text{total}}\}$, obtained via the following architecture-agnostic discovery-and-filtering procedure. We enumerate every `torch.nn.Module` via `model.named_modules()` and retain feature-bearing modules - convolutional, normalization, linear, and pooling layers, and named composite blocks (e.g., ResNet bottleneck blocks, DenseNet dense blocks, transformer encoder blocks) - while excluding non-feature operations (activations, dropout, identity, flatten). Throughout, we use the term layer to denote one such extractable representation point.

The RGC framework fixes a random index set once and reuses it for both calibration and inference. We consider two instantiations: RGC_L samples layer indices, and RGC_C samples coordinate indices.

Layer index sampling in RGC_L. RGC_L samples a subset $\mathcal{L} \subset \mathcal{L}_{\text{all}}$ of size $|\mathcal{L}| = L$ uniformly at random:

$$\mathcal{L} \sim \text{Uniform}\Big(\big\{\mathcal{S} \subseteq \mathcal{L}_{\text{all}} : |\mathcal{S}| = L\big\}\Big). \tag{3}$$

The sampled set $\mathcal{L}$ is used to extract activations $\{\phi_\ell(x)\}_{\ell \in \mathcal{L}}$. By avoiding performance-based index selection, this procedure reduces validation-set adaptivity beyond the final calibration map and removes architecture-specific heuristics for selecting layers or blocks.

Coordinate index sampling in RGC_C. Unlike RGC_L, RGC_C does not sample a subset of layers. Instead, it defines a *global coordinate space* over all extractable layer outputs $\{\phi_\ell(x)\}_{\ell \in \mathcal{L}_{\text{all}}}$ and samples coordinates directly from this space.

For each layer $\ell \in \mathcal{L}_{\text{all}}$, let $\phi_\ell(x) \in \mathbb{R}^{C_\ell \times H_\ell \times W_\ell}$ be its activation tensor and $u_\ell(x) = \text{vec}(\phi_\ell(x)) \in \mathbb{R}^{D_\ell}$ its vectorization, where $D_\ell = C_\ell H_\ell W_\ell$. We define the global activation vector by concatenation:

$$\phi(x) = \big[u_\ell(x)\big]_{\ell \in \mathcal{L}_{\text{all}}} \in \mathbb{R}^D, \qquad D = \sum_{\ell \in \mathcal{L}_{\text{all}}} D_\ell. \tag{4}$$

Let $[D] = \{1, \dots, D\}$. We then sample K coordinates uniformly without replacement:

$$\mathcal{I} \sim \text{Uniform}\left(\binom{[D]}{K}\right). \tag{5}$$

In practice, we precompute the mapping from each global index $i \in [D]$ to a tuple (ℓ, c, h, w) and group sampled indices by layer to form an efficient per-layer *sampling plan*. This is an implementation detail; conceptually, sampling is performed over the global coordinate space without any layer selection.

3.2 Feature Construction

Given a fixed random index set from Section 3.1, RGC maps each input x to a compact embedding $z(x)$ on which distances are computed. Our two instantiations differ in how $z(x)$ is constructed: RGC_L aggregates information from sampled blocks using pooling and random projection, while RGC_C forms a sparse embedding by sampling activation coordinates. For clarity, we write $z_\text{L}(x)$ for RGC_L and $z_\text{C}(x)$ for RGC_C, and use $z(x)$ when the choice is immaterial.

RGC_L: Layer-Based Construction
RGC_L constructs an embedding that aggregates multi-scale spatial information from each sampled layer. For $\ell \in \mathcal{L}$, let $\phi_\ell(x) \in \mathbb{R}^{C_\ell \times H_\ell \times W_\ell}$ denote the activation tensor.

To obtain fixed-length vectors across heterogeneous spatial resolutions, we apply Spatial Pyramid Pooling (SPP) [He *et al.*, 2015] with pyramid levels 4×4, 2×2, and 1×1. SPP max-pools in each cell and concatenates pooled values, producing

$$v_\ell(x) = \text{SPP}(\phi_\ell(x)) \in \mathbb{R}^{d_\ell}, \qquad d_\ell = 21 \cdot C_\ell. \tag{6}$$

This preserves coarse-to-fine spatial structure while yielding a consistent representation length per layer.

Directly concatenating $\{v_\ell(x)\}_{\ell\in\mathcal{L}}$ can be prohibitively high dimensional. We therefore compress each pooled vector and aggregate without materializing the concatenation. For each $\ell \in \mathcal{L}$, we sample a projection matrix $\mathbf{P}_\ell \in \mathbb{R}^{d\times d_\ell}$ with i.i.d. entries from $\mathcal{N}(0, 1/d)$ and define

$$z_{\mathrm{L,raw}}(x) = \sum_{\ell\in\mathcal{L}} \mathbf{P}_\ell\, v_\ell(x). \tag{7}$$

This *project–then–sum* construction is equivalent to projecting the full concatenation with a block-diagonal random matrix, but is more memory efficient.

Because d_ℓ scales with C_ℓ, deeper layers (which typically have larger channel counts) contribute more coordinates to the pooled representation and therefore more signal to the aggregate. Random projection approximately preserves relative geometry, inducing a natural bias toward semantically richer late-stage representations without explicit layer weighting.

Finally, we L_2 normalize the aggregated vector:

$$z_{\mathrm{L}}(x) = \frac{z_{\mathrm{L}}^{\mathrm{raw}}(x)}{\|z_{\mathrm{L}}^{\mathrm{raw}}(x)\|_2}. \tag{8}$$

Distances are therefore dominated by angular similarity rather than scale. We use $d = 256$ in all experiments.

$\mathrm{RGC_C}$: Coordinate-Based Construction

$\mathrm{RGC_C}$ replaces pooling and dense projection with coordinate sampling, yielding a lightweight embedding with substantially lower feature extraction overhead. Uniform sampling over coordinates implicitly weights blocks in proportion to their activation volume $C_\ell H_\ell W_\ell$, which can place more emphasis on earlier high-resolution layers. The resulting lower-complexity feature space can also act as implicit regularization when calibration data are limited, a trade-off we observe empirically in Section 5.

3.3 Geometric Separation Score

Given a normalized embedding $z(x)$, RGC computes a scalar score that reflects local geometric separation relative to the training set $\mathcal{D}_{\mathrm{tr}}$. Following Chouraqui et al. [Chouraqui *et al.*, 2022], we compare the distance from a query point to its nearest *same-class* training representation versus its nearest *different-class* training representation, where class membership is defined with respect to the model's predicted class.

Let $\hat{y}(x)$ be the predicted class for x. Define

$$\mathcal{Z}_{\mathrm{friend}}(x) = \{z(x_i) : (x_i, y_i) \in \mathcal{D}_{\mathrm{tr}},\ y_i = \hat{y}(x)\}, \tag{9}$$

$$\mathcal{Z}_{\mathrm{non}}(x) = \{z(x_i) : (x_i, y_i) \in \mathcal{D}_{\mathrm{tr}},\ y_i \neq \hat{y}(x)\}. \tag{10}$$

Let $d_{\mathrm{friend}}(x) = \min_{z'\in\mathcal{Z}_{\mathrm{friend}}(x)} \|z(x)-z'\|_2$ and $d_{\mathrm{non}}(x) = \min_{z'\in\mathcal{Z}_{\mathrm{non}}(x)} \|z(x) - z'\|_2$. The separation score is

$$s(x) = \frac{d_{\mathrm{non}}(x) - d_{\mathrm{friend}}(x)}{2}. \tag{11}$$

Positive values indicate that the nearest neighbor from the predicted class is closer than the nearest neighbor from other classes, corresponding to high geometric confidence. Negative values indicate proximity to competing classes and often correlate with ambiguity or error.

3.4 Calibration Mapping

Separation scores are not probabilities and can vary in scale across architectures and feature constructions. We convert scores into calibrated probabilities using rank-based normalization followed by isotonic regression.

The normalization maps scores to a uniform $[0, 1]$ scale. We adopt rank-based normalization (empirical cumulative distribution function) rather than linear scaling. Linear methods such as percentile clipping can be sensitive to changes in the score distribution, since samples outside the calibration range may be mapped to boundary values, reducing the resolution of the geometric signal. Rank-based normalization mitigates this saturation by mapping scores to their relative rank, preserving ordinal information across the full range.

Let $\mathcal{S}_{\mathrm{val}} = \{s(x_i) : (x_i, y_i) \in \mathcal{D}_{\mathrm{val}}\}$ be the set of separation scores computed on the validation set, and let $N = |\mathcal{D}_{\mathrm{val}}|$. Define

$$\tilde{s}(x) = \frac{1}{N} \sum_{s'\in\mathcal{S}_{\mathrm{val}}} \mathbf{1}[s' \leq s(x)]. \tag{12}$$

We then fit a one-dimensional isotonic regressor $g : [0, 1] \to [0, 1]$ on $\mathcal{D}_{\mathrm{val}}$, minimizing squared error between $g(\tilde{s}(x_i))$ and the correctness indicator $\mathbf{1}[\hat{y}(x_i) = y_i]$ [Zadrozny and Elkan, 2002]. The final calibrated confidence is

$$\hat{p}(x) = g(\tilde{s}(x)). \tag{13}$$

4 Experimental Setup

Datasets. We evaluate on three standard datasets: CIFAR-10 [Krizhevsky, 2009] (60K images, 10 classes), CIFAR-100 [Krizhevsky, 2009] (60K images, 100 classes), and Tiny-ImageNet [Le and Yang, 2015] (200 classes, 500 images per class at 64×64 resolution). For all datasets, we reserve 10% of the training data for calibration fitting, the remaining 90% serves as the training set for geometric methods.

Architectures and Training. We used the following well-known neural networks for evaluation: ResNet-18/50/101/152 [He *et al.*, 2016], DenseNet-121 [Huang *et al.*, 2017], and DINOv2-Large [Oquab *et al.*, 2024]. ResNet and DenseNet are trained from scratch with stochastic gradient descent (SGD). For ResNet models we follow the original training protocol [He *et al.*, 2016], reducing learning rate when validation accuracy plateaus. For DenseNet we use a warmup-to-cosine schedule [Loshchilov and Hutter, 2017]. Following the standard evaluation protocol for self-supervised models [Oquab *et al.*, 2024], we use linear probing with inputs resized to 224×224, training only the classification head while keeping the backbone frozen. Our primary evaluation uses cross-entropy loss.

Implementation and Evaluation. We evaluated sensitivity to the main $\mathrm{RGC_L}$ hyperparameters: the number of layers L and the embedding dimension d. Across 30 configurations spanning $L \in \{2, 4, 6, 8, 10, 12\}$ and $d \in \{64, 128, 256, 512, 1024\}$, mean ECE remains within $[1.10, 1.22]\%$, and per-configuration standard deviations fall within $[0.71, 0.97]$. The chosen configuration ($L = 6, d = 256$) achieves a mean ECE of $1.14 \pm 0.74\%$, which reflects

Dataset	Model (Acc)	Uncal	TS	Platt	Isotonic	Beta	DAC	Fast Separation	TULIP	D.Ens	$\mathrm{RGC_L}$	$\mathrm{RGC_C}$
CIFAR10	densenet121 (94.83±0.59)	3.59±0.10	1.51±0.07	2.08±0.32	0.67±0.04	1.25±0.04	1.30±0.08	**0.64±0.08**	—	1.47±0.07	**0.59±0.07**	**0.39±0.06**
CIFAR10	resnet101 (94.44±0.15)	4.34±0.20	1.34±0.12	2.21±0.18	0.82±0.07	1.33±0.07	1.10±0.09	**0.77±0.06**	2.50±1.12	2.22±0.19	**0.69±0.07**	**0.65±0.07**
CIFAR10	resnet152 (94.37±0.15)	4.38±0.10	1.46±0.06	2.21±0.09	**0.77±0.06**	1.38±0.05	1.06±0.06	0.82±0.07	2.60±0.50	1.82±0.10	**0.72±0.06**	**0.75±0.08**
CIFAR10	resnet18 (93.84±0.67)	3.86±0.15	1.60±0.10	2.74±0.40	**0.92±0.06**	1.45±0.09	1.19±0.09	0.94±0.08	—	3.05±0.63	**0.81±0.10**	**0.83±0.10**
CIFAR10	resnet50 (94.15±0.49)	3.99±0.13	1.42±0.07	2.53±0.40	**0.80±0.05**	1.40±0.06	0.92±0.05	0.89±0.05	3.00±0.24	1.44±0.16	**0.67±0.05**	**0.79±0.07**
CIFAR100	densenet121 (78.91±0.24)	13.65±0.44	3.75±0.16	8.72±0.25	1.99±0.21	5.67±0.14	3.52±0.13	**1.27±0.16**	—	4.42±0.33	**1.36±0.18**	**1.12±0.17**
CIFAR100	dinov2(L) (84.57±1.21)	33.04±5.16	2.43±0.50	13.30±1.04	1.31±0.17	5.00±0.76	—	**1.12±0.72**	—	—	**0.91±0.43**	**1.15±0.61**
CIFAR100	resnet101 (76.83±0.33)	11.48±0.87	4.41±0.12	7.87±0.17	2.06±0.12	5.78±0.18	3.77±0.10	**1.58±0.16**	6.71±0.10	4.24±0.46	**1.49±0.15**	**1.50±0.11**
CIFAR100	resnet152 (76.64±0.38)	10.76±1.11	4.45±0.23	7.91±0.29	2.01±0.14	5.67±0.18	3.74±0.21	**1.65±0.17**	6.62±0.32	5.63±1.41	**1.46±0.17**	**1.66±0.16**
CIFAR100	resnet18 (75.94±0.31)	5.51±0.43	3.10±0.18	6.90±0.15	**1.39±0.11**	4.56±0.19	3.40±0.18	1.54±0.22	—	6.27±1.64	**1.59±0.12**	**1.57±0.30**
CIFAR100	resnet50 (77.25±0.38)	12.18±0.49	4.11±0.13	8.15±0.13	1.88±0.14	5.94±0.11	3.89±0.14	**1.60±0.13**	6.58±0.02	4.46±0.45	**1.40±0.11**	**1.38±0.18**
TINY	resnet101 (59.35±0.40)	2.91±1.00	2.29±0.53	23.74±0.48	1.61±0.23	2.06±0.48	1.84±0.34	**1.48±0.64**	—	5.61±1.64	**1.30±0.16**	**1.11±0.43**
TINY	resnet152 (57.90±2.09)	2.25±0.41	2.01±0.33	24.30±0.40	**1.52±0.21**	1.58±0.19	1.82±0.29	1.60±0.22	—	14.12±2.36	**1.24±0.23**	**0.84±0.23**
TINY	resnet50 (57.46±0.37)	3.29±0.86	2.42±0.35	23.68±0.49	1.60±0.50	2.22±0.33	2.07±0.22	**1.51±0.49**	—	5.36±1.71	**1.32±0.53**	**0.90±0.26**

Table 1: ECE (%) comparison across datasets and models. Values are shown as mean ± 95% CI. Bold entries highlight the top-three results in each row: best in blue, second-best in teal, and third-best in olive.

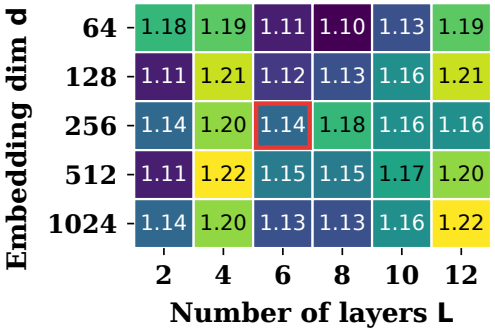

Figure 1: RGC hyperparameter sensitivity: mean ECE (%) across L and d, averaged over all dataset/model configurations. The red box marks the chosen configuration $L = 6$, $d = 256$. The narrow value range demonstrates ranking stability across the grid.

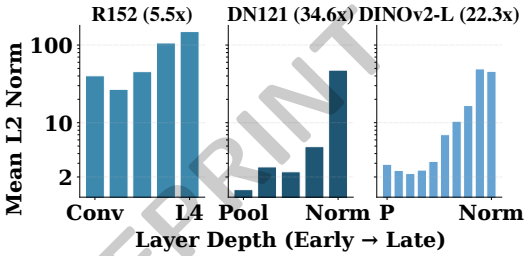

Figure 2: Mean L_2 norm of pooled feature vectors across depth on CIFAR-100, shown on a log scale.

an arbitrary selection of hyperparameters to avoid overfitting to the evaluation data. Figure 1 summarizes the results of the mean ECE for the entire grid. Notice that the relative ranking against the baselines remains stable across the tested grid, and that the selected configuration is not optimized. All experiments were conducted on a single NVIDIA RTX 3090 GPU with 24GB VRAM and 32GB system RAM using PyTorch. We report ECE with M=15 bins, as used in [Guo *et al.*, 2017], averaged over multiple seeds with 95% confidence intervals, as standard.

5 Results

This section demonstrates that: (1) $\mathrm{RGC_L}$ achieves strong top-label calibration across architectures and datasets (Section 5.1), (2) random layer sampling matches or exceeds optimized selection (Section 5.2), and (3) $\mathrm{RGC_C}$ provides an efficient alternative with competitive results (Section 5.3).

5.1 $\mathrm{RGC_L}$ Achieves Strong Top-Label Calibration

To evaluate the calibration performance of $\mathrm{RGC_L}$, we compare it against two families of single-model post-hoc calibrators. *Logit-based methods*: Temperature Scaling (TS) [Guo *et al.*, 2017], Top Label Isotonic Regression [Zadrozny and Elkan, 2002; Gupta and Ramdas, 2022], Platt Scaling [Platt, 2000], and Beta Calibration [Kull *et al.*, 2017]. *Feature-based methods*: Fast Separation [Chouraqui *et al.*, 2022] (geometric separation in pixel space), DAC [Tomani *et al.*, 2023] (density-aware calibration with architecture-specific layers), and TULIP [Benkert *et al.*, 2024] (layered intermediate predictions)[1]. We additionally report Deep Ensembles [Lakshminarayanan *et al.*, 2017] as a high-cost uncertainty reference.

Table 1 shows the ECE of these algorithms on the evaluated models and datasets. As can be observed, $\mathrm{RGC_L}$ consistently achieves the lowest or near-lowest ECE across the considered datasets and architectures. Specifically, $\mathrm{RGC_L}$ achieves an average rank of ∼1.71 across all configurations (∼1.21 when $\mathrm{RGC_C}$ is excluded), ranking first or second in 12 of 14 settings. Relative to uncalibrated predictions, $\mathrm{RGC_L}$ reduces ECE by 78% on average, and by up to 97% on CIFAR-100 DINOv2. Against the best non-RGC baseline, $\mathrm{RGC_L}$ achieves an average improvement of 12% and up to 19%.

Beyond ECE. Since RGC calibrates top-label confidence [Gupta and Ramdas, 2022], ECE is our primary metric. However, we also evaluate Adaptive ECE [Nixon *et al.*, 2019] and top-label MCE to assess whether the conclusions remain stable across related binning-based calibration metrics. On Adaptive ECE, both $\mathrm{RGC_L}$ and $\mathrm{RGC_C}$ achieve an average rank of approximately 1.57 against non-RGC baselines, with top-three placement in 13 and 12 out of 14 evaluated configurations, respectively. On MCE, $\mathrm{RGC_C}$ slightly improves over $\mathrm{RGC_L}$ in average rank (2.79 vs. 3.00), and both methods rank in the top-three in 9 of 14 configurations. We conclude that the selected metric is representative, and we obtain similar results with other metrics.

[1]The experimental results for TULIP are taken directly from [Benkert *et al.*, 2024].

			vs GC(DAC)				vs $\mathrm{RGC_C}$			
Dataset	Model	$\mathrm{RGC_L}$	GC(DAC)	Diff (pp)	TOST p	Result	$\mathrm{RGC_C}$	Diff (pp)	TOST p	Result
CIFAR10	densenet121	0.59±0.26	0.58±0.19	-0.01	**<0.001**	Eq.	0.39±0.24	-0.20	**<0.001**	Eq.
CIFAR10	resnet101	0.69±0.25	0.67±0.22	-0.02	**<0.001**	Eq.	0.65±0.24	-0.04	**<0.001**	Eq.
CIFAR10	resnet152	0.72±0.30	0.66±0.30	-0.06	**<0.001**	Eq.	0.75±0.39	+0.03	**<0.001**	Eq.
CIFAR10	resnet18	0.81±0.32	0.81±0.40	-0.00	**<0.001**	Eq.	0.83±0.33	+0.02	**<0.001**	Eq.
CIFAR10	resnet50	0.69±0.23	0.67±0.24	-0.04	**<0.001**	Eq.	0.79±0.28	+0.10	**<0.001**	Eq.
CIFAR100	densenet121	1.36±0.45	1.29±0.27	-0.06	**0.007**	Eq.	1.12±0.41	-0.23	0.056	-
CIFAR100	resnet101	1.49±0.59	1.45±0.50	-0.04	**<0.001**	Eq.	1.50±0.45	+0.02	**<0.001**	Eq.
CIFAR100	resnet152	1.46±0.52	1.33±0.37	-0.12	**0.017**	Eq.	1.66±0.46	+0.20	**0.014**	Eq.
CIFAR100	resnet18	1.59±0.31	1.35±0.26	-0.24	**0.021**	Eq.	1.57±0.75	-0.02	**0.011**	Eq.
CIFAR100	resnet50	1.40±0.36	1.30±0.41	-0.10	**0.002**	Eq.	1.38±0.57	-0.04	**0.009**	Eq.
TINY	resnet101	1.37±0.10	1.39±0.45	+0.02	0.086	-	1.19±0.41	-0.18	0.129	-
TINY	resnet152	1.02±0.30	1.37±0.29	+0.35	0.232	-	0.61±0.44	-0.42	0.388	-
TINY	resnet50	1.16±0.15	1.22±0.52	+0.06	0.102	-	0.61±0.29	-0.54	0.583	-

Table 2: TOST equivalence tests for calibration (ECE %; lower is better). We report mean±std over paired trials and test equivalence using paired TOST with margin $\delta = 0.5$ pp. TOST $p < 0.05$ indicates the mean difference falls significantly within $[-\delta, +\delta]$.

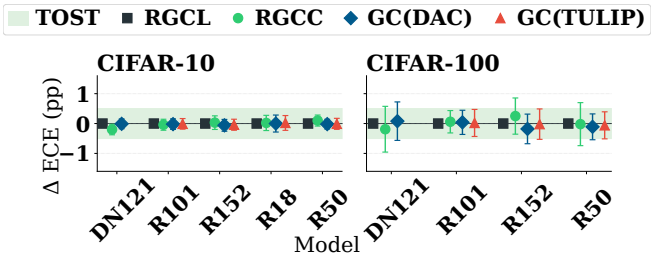

Figure 3: ECE normalized relative to $\mathrm{RGC_L}$; values below zero indicate better performance than $\mathrm{RGC_L}$, while values above zero indicate worse performance. The green square denotes the TOST equivalence region at the 95% confidence level. Methods whose confidence intervals lie entirely within the square are statistically equivalent to $\mathrm{RGC_L}$, while others are inconclusive.

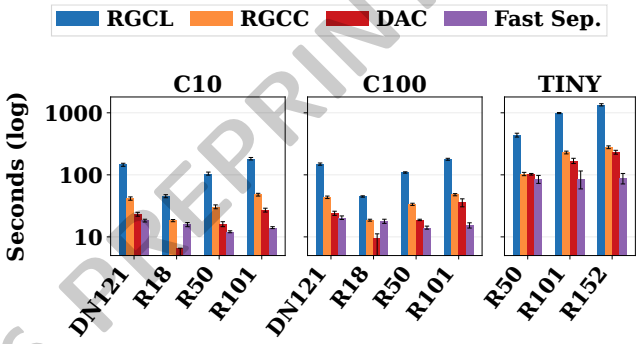

Figure 4: Preprocessing time required before deploying the geometric calibrator, compared with prior methods. This preprocessing step is performed only once during calibration setup.

5.2 Sampling Matches Optimal Layer Selection

We compare $\mathrm{RGC_L}$ and $\mathrm{RGC_C}$ with geometric calibrators that rely on optimized layer selection, namely DAC and TULIP, denoted GC(DAC) and GC(TULIP). Figure 3 reports ECE normalized relative to $\mathrm{RGC_L}$. The differences cluster near zero and mostly fall within the $\delta = 0.5$ percentage-point equivalence band suggesting that $\mathrm{RGC_L}$ and $\mathrm{RGC_C}$ achieve equivalent calibration performance to carefully selecting layer configurations.

Table 2 reports the corresponding numerical results. To formally assess equivalence, we perform two one-sided tests (TOST) [Schuirmann, 1987] with margin δ=0.5 percentage points. The test establishes equivalence in 10 of 13 model–dataset configurations, and in no configuration is $\mathrm{RGC_L}$ found to be significantly worse than DAC- or TULIP-based layer selection. Taken together, these results show that our randomized sampling strategy can replace carefully tuned layer selection without a statistically significant degradation in calibration performance.

5.3 $\mathrm{RGC_C}$ Provides Comparable Calibration at Lower Preprocessing Cost

This section compares $\mathrm{RGC_L}$ and $\mathrm{RGC_C}$ using TOST equivalence testing, with results reported in Table 2. Of the 13 dataset and model combinations, 9 establish formal equivalence at margin $\delta = 0.5$ percentage points. The point estimates mildly favor $\mathrm{RGC_L}$ on CIFAR-10 and CIFAR-100 on average, whereas $\mathrm{RGC_C}$ achieves lower ECE on average on Tiny-ImageNet. The remaining configurations do not establish equivalence, primarily due to higher variance or fewer paired trials. Overall, the differences are small, and combined with $\mathrm{RGC_C}$'s lower preprocessing cost (Section 5.5), this makes $\mathrm{RGC_C}$ an attractive option for resource-constrained deployment.

5.4 Why $\mathrm{RGC_L}$ Works

At first glance, the strong performance of $\mathrm{RGC_L}$ and $\mathrm{RGC_C}$ may seem surprising. Unlike prior work, our method does not rely on explicit per-layer weighting, even though uniformly weighting layers is generally suboptimal when many layers are uninformative. We show that layers exhibiting strong geometric separation also tend to have disproportionately large L_2 norms. As a result, once such a layer is sampled, it naturally dominates the aggregation.

Figure 2 plots the average L_2 norm of pooled feature vectors, $\|v_\ell(x)\|_2$, across layers for three architectures on CIFAR-100. We observe a consistent increase in norm magnitude from early to late layers: 5.5× for ResNet-152, 22.3× for DINOv2-Large (despite a constant embedding dimension), and 34.6× for DenseNet-121 (likely driven by dense

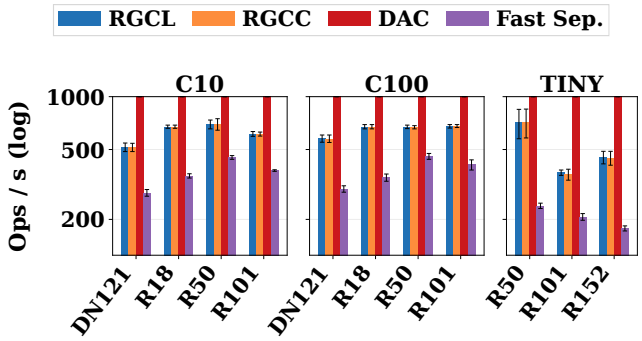

Figure 5: Throughput, measured in confidence estimates per second.

feature concatenation). In DINOv2-Large, the embedding dimension is fixed, and we still see larger norms in important layers. When projected vectors are summed, layers with larger norms therefore contribute disproportionately to the induced Euclidean geometry, effectively emphasizing later blocks without requiring explicit layer weighting. We emphasize that this is an empirical explanation rather than a worst-case guarantee: in architectures where late-layer norms do not dominate, the implicit weighting effect may be weaker or distributed more uniformly across layers.

This implicit emphasis on a small number of informative layers motivates the second design choice of our method. Once the layers are selected, we compress them into a compact representation with drastically fewer coordinates. The effectiveness of this compression is consistent with growing evidence that learned representations exhibit substantial distributed redundancy. For example, Nanda et al. [Nanda *et al.*, 2023] demonstrate that randomly selected neuron subsets preserve much of a layer's representational power.

5.5 Computational Efficiency

In this section, we evaluate the computational overhead of our proposed methods. We consider two benchmarks: the offline preprocessing time and the online inference throughput.

First, we evaluate the offline preprocessing time required by our methods. These results are shown in Figure 4. As expected, $\mathrm{RGC}_{\mathrm{L}}$ requires more preprocessing time than the alternative methods. $\mathrm{RGC}_{\mathrm{C}}$ substantially reduces this cost; however, the preprocessing time remains higher than that of prior work. This overhead constitutes a limitation of our approach, but it is incurred only once during calibration setup and does not affect inference-time performance.

Second, we evaluate the number of confidence estimates produced per second on each dataset. Figure 5 reports these results. We observe that DAC is considerably faster than both $\mathrm{RGC}_{\mathrm{L}}$ and $\mathrm{RGC}_{\mathrm{C}}$. The two variants attain nearly identical inference speeds, as they execute the same geometric scoring procedure on slightly different feature representations. Fast Separation [Chouraqui *et al.*, 2022] is slower still, and its throughput depends directly on the input dimensionality. In contrast, our methods project inputs into a fixed-dimensional random subspace, yielding a constant computational cost independent of the original input size.

The main takeaway from these experiments is the trade-off inherent in our approach and in geometric calibration methods more broadly. Unlike purely logit-based calibrators, $\mathrm{RGC}_{\mathrm{L}}$ and $\mathrm{RGC}_{\mathrm{C}}$ require an offline preprocessing stage for feature extraction and nearest-neighbor index construction, which can be costly when the calibration set or data distribution changes frequently. However, once this index is built, both variants support several hundred confidence estimates per second, making them compatible with real-time vision workloads such as video-rate inference. Thus, RGC is best suited to settings with relatively stable calibration data, where improved top-label calibration justifies additional offline preprocessing and moderate online overhead.

6 Discussion

Our work contributes to the ongoing line of research on geometric confidence calibration. We demonstrate that explicit layer selection and weighting can be avoided by adopting layer-based or coordinate-based sampling schemes, and that the dimensionality of the geometric feature space can be drastically reduced without sacrificing performance. These findings reinforce observations by Nanda et al. [Nanda *et al.*, 2023] regarding the high degree of redundancy present in late-stage feature representations.

Building on these insights, we introduce a geometric confidence calibration method. The method assesses how close an input representation lies to the boundary of the class-induced geometry.

We further show that our approach outperforms several recently proposed geometric confidence calibrators, including DAC [Tomani *et al.*, 2023], TULIP [Benkert *et al.*, 2024], and Separation [Chouraqui *et al.*, 2022], establishing a strong empirical contribution to the literature on confidence estimation. Through extensive evaluation, we analyze both the strengths and limitations of our method, provide insight into its effectiveness, and position its computational overhead within the broader literature, exposing the trade-offs involved.

Our evaluation is limited to in-distribution image-classification benchmarks.

Evaluating RGC under corruptions, domain shift, and non-vision modalities is an important direction for future work.

Looking forward, our results suggest that future geometric calibrators may benefit from placing less emphasis on explicit layer selection and weighting. Such design choices can introduce additional adaptivity to specific architectures or datasets, potentially limiting transfer across models. In contrast, randomized sampling strategies, such as the two variants proposed in this work, offer a simple and architecture-agnostic route for constructing informative geometric feature spaces.

An additional promising direction is the approximation of reduced semantic feature spaces induced by successful geometric calibrators using smaller neural networks. We hypothesize that such networks could produce embeddings better suited for semantic caching, potentially leading to improved efficiency and performance.

References

[Benkert *et al.*, 2024] Ryan Benkert, Mohit Prabhushankar, and Ghassan AlRegib. Transitional uncertainty with layered intermediate predictions. In Ruslan Salakhutdinov, Zico Kolter, Katherine A. Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp, editors, *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*, Proceedings of Machine Learning Research, pages 3484–3505. PMLR / OpenReview.net, 2024.

[Caruana and Niculescu-Mizil, 2006] Rich Caruana and Alexandru Niculescu-Mizil. An empirical comparison of supervised learning algorithms. In William W. Cohen and Andrew W. Moore, editors, *Machine Learning, Proceedings of the Twenty-Third International Conference (ICML 2006), Pittsburgh, Pennsylvania, USA, June 25-29, 2006*, ACM International Conference Proceeding Series, pages 161–168. ACM, 2006.

[Chouraqui *et al.*, 2022] Gabriella Chouraqui, Liron Cohen, Gil Einziger, and Liel Leman. A geometric method for improved uncertainty estimation in real-time. In James Cussens and Kun Zhang, editors, *Uncertainty in Artificial Intelligence, Proceedings of the Thirty-Eighth Conference on Uncertainty in Artificial Intelligence, UAI 2022, 1-5 August 2022, Eindhoven, The Netherlands*, Proceedings of Machine Learning Research, pages 422–432. PMLR, 2022.

[Gal and Ghahramani, 2016] Yarin Gal and Zoubin Ghahramani. Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. In Maria-Florina Balcan and Kilian Q. Weinberger, editors, *Proceedings of the 33rd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, JMLR Workshop and Conference Proceedings, pages 1050–1059. JMLR.org, 2016.

[Guo *et al.*, 2017] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, Proceedings of Machine Learning Research, pages 1321–1330. PMLR, 2017.

[Gupta and Ramdas, 2021] Chirag Gupta and Aaditya Ramdas. Distribution-free calibration guarantees for histogram binning without sample splitting. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, Proceedings of Machine Learning Research, pages 3942–3952. PMLR, 2021.

[Gupta and Ramdas, 2022] Chirag Gupta and Aaditya Ramdas. Top-label calibration and multiclass-to-binary reductions. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022.

[He *et al.*, 2015] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 37(9):1904–1916, 2015.

[He *et al.*, 2016] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 770–778. IEEE Computer Society, 2016.

[Hendrycks *et al.*, 2019] Dan Hendrycks, Kimin Lee, and Mantas Mazeika. Using pre-training can improve model robustness and uncertainty. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, Proceedings of Machine Learning Research, pages 2712–2721. PMLR, 2019.

[Huang *et al.*, 2017] Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 2261–2269. IEEE Computer Society, 2017.

[Jiang *et al.*, 2018] Heinrich Jiang, Been Kim, Melody Y. Guan, and Maya R. Gupta. To trust or not to trust a classifier. In Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 5546–5557, 2018.

[Krizhevsky, 2009] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

[Kull *et al.*, 2017] Meelis Kull, Telmo de Menezes e Silva Filho, and Peter A. Flach. Beta calibration: a well-founded and easily implemented improvement on logistic calibration for binary classifiers. In Aarti Singh and Xiaojin (Jerry) Zhu, editors, *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, AISTATS 2017, 20-22 April 2017, Fort Lauderdale, FL, USA*, Proceedings of Machine Learning Research, pages 623–631. PMLR, 2017.

[Kull *et al.*, 2019] Meelis Kull, Miquel Perelló-Nieto, Markus Kängsepp, Telmo de Menezes e Silva Filho, Hao Song, and Peter A. Flach. Beyond temperature scaling: Obtaining well-calibrated multi-class probabilities with Dirichlet calibration. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 12295–12305, 2019.

[Lakshminarayanan *et al.*, 2017] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and

scalable predictive uncertainty estimation using deep ensembles. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 6402–6413, 2017.

[Le and Yang, 2015] Ya Le and Xuan Yang. Tiny ImageNet visual recognition challenge, 2015. Stanford University CS231n Course Project.

[Lee *et al.*, 2018] Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. In Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 7167–7177, 2018.

[Loshchilov and Hutter, 2017] Ilya Loshchilov and Frank Hutter. SGDR: stochastic gradient descent with warm restarts. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017.

[Mandelbaum and Weinshall, 2017] Amit Mandelbaum and Daphna Weinshall. Distance-based confidence score for neural network classifiers. *CoRR*, abs/1709.09844, 2017.

[Nanda *et al.*, 2023] Vedant Nanda, Till Speicher, John P. Dickerson, Krishna P. Gummadi, Soheil Feizi, and Adrian Weller. Diffused redundancy in pre-trained representations. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine, editors, *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023.

[Nixon *et al.*, 2019] Jeremy Nixon, Michael W. Dusenberry, Linchuan Zhang, Ghassen Jerfel, and Dustin Tran. Measuring calibration in deep learning. In *Uncertainty and Robustness in Deep Visual Learning Workshop, CVPR 2019, Long Beach, CA, USA, June 17, 2019*, 2019.

[Oquab *et al.*, 2024] Maxime Oquab, Timothée Darcet, Théo Moutakanni, et al. DINOv2: Learning robust visual features without supervision. *Trans. Mach. Learn. Res.*, 2024.

[Papernot and McDaniel, 2018] Nicolas Papernot and Patrick D. McDaniel. Deep k-nearest neighbors: Towards confident, interpretable and robust deep learning. *CoRR*, abs/1803.04765, 2018.

[Platt, 2000] John C. Platt. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In Alexander J. Smola, Peter L. Bartlett, Bernhard Schölkopf, and Dale Schuurmans, editors, *Advances in Large Margin Classifiers*, pages 61–74. MIT Press, Cambridge, MA, 2000.

[Schuirmann, 1987] Donald J Schuirmann. A comparison of the two one-sided tests procedure and the power approach for assessing the equivalence of average bioavailability. *Journal of pharmacokinetics and biopharmaceutics*, 15(6):657–680, 1987.

[Thulasidasan *et al.*, 2019] 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 Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 13888–13899, 2019.

[Tomani *et al.*, 2023] Christian Tomani, Futa Kai Waseda, Yuesong Shen, and Daniel Cremers. Beyond in-domain scenarios: Robust density-aware calibration. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, Proceedings of Machine Learning Research, pages 34344–34368. PMLR, 2023.

[Xiong *et al.*, 2023] Miao Xiong, Ailin Deng, Pang Wei W. Koh, Jiaying Wu, Shen Li, Jianqing Xu, and Bryan Hooi. Proximity-informed calibration for deep neural networks. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine, editors, *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023.

[Yüksekgönül *et al.*, 2023] Mert Yüksekgönül, Linjun Zhang, James Y. Zou, and Carlos Guestrin. Beyond confidence: Reliable models should also consider atypicality. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine, editors, *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023.

[Zadrozny and Elkan, 2002] Bianca Zadrozny and Charles Elkan. Transforming classifier scores into accurate multiclass probability estimates. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, July 23-26, 2002, Edmonton, Alberta, Canada*, pages 694–699. ACM, 2002.

[Zhang and Haghani, 2015] Yanru Zhang and Ali Haghani. A gradient boosting method to improve travel time prediction. *Transportation Research Part C-emerging Technologies*, 58:308–324, 2015.