Divide et Calibra: Multiclass Local Calibration via Vector Quantization

Cesare Barbera[*]
DI, University of Pisa
DISI, University of Trento

Lorenzo Perini
Meta Platforms, Inc.

Giovanni De Toni[†]
ETH AI Center & ETH Zurich

Andrea Passerini
DISI, University of Trento

Andrea Pugnana
DISI, University of Trento

Abstract

Accurate and well-calibrated Machine Learning (ML) models are mandatory in high-stakes settings, yet effective multiclass calibration remains challenging: global approaches assume calibration errors are homogeneous across the latent space, while local methods often rely on representation-space dimensionality reduction, discarding latent directions relevant to miscalibration. To address these issues, we propose a compositional approach to multiclass calibration, where region-specific calibration maps are constructed from shared factors. We instantiate this idea via Vector Quantization (VQ), which induces a structured partition of the representation space, and an indexed parameterization of Dirichlet concentrations that enables parameter sharing across regions. Our approach learns heterogeneous calibration maps that generalize well even to sparse regions of the latent space. Experiments on benchmark datasets show significant improvements in local calibration while maintaining competitive global calibration and predictive performance.

1 Introduction

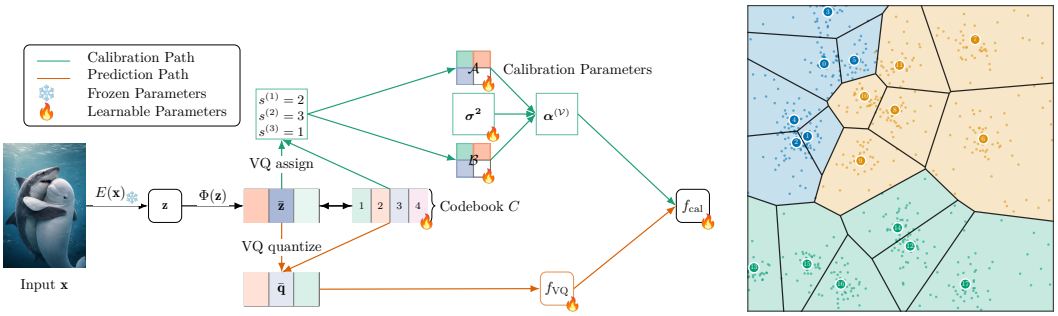
In high-stakes settings such as healthcare and finance, Machine Learning (ML) models must be not only accurate but also well-calibrated [1–3]. In these settings, the reliability of the *entire predicted probability vector* is critical: clinicians, judges, or automated policies act on relative likelihoods between multiple competing outcomes, rather than on the most likely class alone.

Consider an ML model that predicts cancer stage from cell images. Its probabilities can become systematically distorted for cells exhibiting unusual combinations of shape, texture and staining patterns because few similar examples were observed during training. This makes the model unreliable exactly in those cases where calibration is highly sought for trustworthy decision-making [4].

Such systematic distortions arising in low-density regions of the representation space are known as *proximity bias* [5]. To address proximity bias, recent work has moved towards *local calibration*, which requires probability estimates to be reliable not only on average, but within specific regions of the input or latent space [6]. However, local multiclass calibration has a critical statistical bottleneck: *data sparsity*. As the dimensionality of the representation space increases, data points become more isolated, making it harder to reliably estimate local corrections. Conversely, global calibration methods [7, 8] avoid data sparsity by applying a single correction map to all samples, but fail to correct *proximity bias* in low-density regions. This describes a key trade-off: global methods are stable but biased, while local methods are flexible but high-variance.

[*]Corresponding author: cesare.barbera@phd.unipi.it; cesare.barbera@unitn.it.
[†]Work done while at Fondazione Bruno Kessler.

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



(a) Toy example pipeline when $w = 3$ and $|\mathcal{C}| = 4$.

(b) Voronoi Tessellation Example.

Figure 1: We depict our pipeline (Figure 1a) and Voronoi assignment procedure (Figure 1b).

To address this challenge, we reinterpret local calibration as a compositional learning problem, where region-specific calibration functions are constructed from reusable latent components, rather than estimated independently for each region. Figure 1a showcases our proposed methodology: first, we consider a discretization of the representation space via vector quantization (VQ). VQ induces a Voronoi tessellation of the embedding space, mapping each continuous latent vector to a discrete sequence of elements in a shared *codebook* (*i.e.*, a finite set of prototype vectors, or "*codewords*"). This replaces a continuous representation with a structured composition of discrete components, drawn from frequently reused codewords. Intuitively, even if a particular index sequence corresponds to a rare region, its codewords are shared across many samples (assuming good codebook utilization), providing better statistical support than direct conditioning on rare continuous features. Second, we apply this same discretization principle also to calibration parameters, *i.e.*, we learn calibration maps for rare or isolated instances through combinations of well-estimated, frequently reused calibration factors. This yields an implicit form of density regularization: the model is locally adaptive, yet avoids learning fully instance-specific calibration maps in sparse regions.

Our Contributions. To sum up, our contributions are:

- (i) We propose a *compositional formulation of local multiclass calibration*, where calibration maps are defined over a Voronoi tessellation of the embedding space induced by Vector Quantization, enabling region-aware modeling of miscalibration (Section 3.1).
- (ii) To make this formulation tractable, we introduce an *indexed parameterization trick* that constructs region-specific calibration maps by reusing shared codeword-dependent factors. This yields a parameter-efficient model that scales to exponentially many regions, enables compositional generalization, and whose estimation error is governed by codeword frequency rather than by the number of samples in each region (Section 3.2).
- (iii) We empirically validate the approach showing large and significant improvements over existing baselines, particularly in low-support regions (Section 4).

2 Background

Preliminaries. We consider a multi-class classification task, where $\mathcal{X} \subseteq \mathbb{R}^m$ is the feature space and $\mathcal{Y} = \{0, \dots, |\mathcal{Y}| - 1\}$ is a finite target space with $|\mathcal{Y}|$ distinct labels. Let us assume we have access to a given dataset $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$ of input-output pairs drawn from an unknown joint distribution $\mathcal{P}$ over $\mathcal{X} \times \mathcal{Y}$. Each input $\mathbf{x}_i \in \mathcal{X}$ is a feature vector of m dimensions, and each label $y_i \in \mathcal{Y}$ has a corresponding one-hot encoded vector $\mathbf{y}_i$ indicating the correct class among the $|\mathcal{Y}|$ possible classes. We consider a probabilistic classifier $f \colon \mathcal{X} \to \Delta^{|\mathcal{Y}|}$ mapping inputs to the $(|\mathcal{Y}| - 1)$-dimensional probability simplex, with $f(\mathbf{x}) = \hat{\mathbf{p}}$ and $\hat{\mathbf{p}}_k = f_k(\mathbf{x})$ the predicted probability of class k.

Calibration notions and metrics. The strictest notion is *strong calibration* [9], which requires the target class conditional distribution on any prediction of the classifier to match that prediction:

$$\mathbb{P}(\mathbf{y}_k = 1 \mid \hat{\mathbf{p}}) = \hat{\mathbf{p}}_k \qquad \forall\, k \in \{0, \dots, |\mathcal{Y}| - 1\} \tag{1}$$

While strong calibration is defined as a population-level expectation, recent work has shifted attention to *local calibration* [6, 10, 11], which requires calibration to hold within localized regions of the input or representation space. A model is then *locally calibrated* if its predictions match consistent kernel estimates of the conditional label distribution.

We consider both *global* and *local* calibration metrics to assess calibration. Expected Calibration Error (ECE) [12] measures the discrepancy between predicted confidence and empirical accuracy across bins; in multiclass settings it extends to Class-wise ECE [7] (per-class average) and Multidimensional ECE (MECE), the latter impractical due to combinatorial simplex-binning. Expected Cumulative Calibration Error (ECCE) [13] aggregates errors cumulatively, providing a more robust assessment. We summarize these errors by averaging the absolute cumulative residuals across bins.

For local calibration metrics, we consider the multiclass extension of *Local Calibration Error* (LCE) [6] proposed by Barbera et al. [11]. This metric captures discrepancies between predicted probabilities and ground-truth labels for neighboring points of an anchor $\mathbf{x}_i$, weighted by a kernel function. We also consider the *Maximum Local Calibration Error* (MLCE), which measures the worst-case local calibration error over the dataset. See Appendix B for a detailed description of all metrics.

3 Methodology

Given a pre-trained classifier f and a calibration set $\mathcal{D}_{\text{cal}}$, our goal is to construct a region-aware calibration function f_{cal} that maps the model output $\hat{\mathbf{p}}$ to a calibrated posterior $\hat{\mathbf{p}}_{\text{cal}}$. Unlike global methods that apply a unique map to the model scores for all instances, learning *local* calibration functions requires access to a (frozen) representation space and replacing the original classification head, as in Kirichenko et al. [14] or other local calibration approaches [15, 11].

Note that obtaining a heterogeneous calibrator in high-dimensional settings is challenging for three main reasons. First, defining *"local regions"* in high-dimensional embedding spaces is non-trivial. Standard distance-based methods suffer from the curse of dimensionality [16–19], leading to isolated neighborhoods and unreliable density estimation. Second, even if we define discrete regions, the number of such regions typically grows *exponentially* with the dimensionality. Learning a distinct, unstructured calibration map for every region is computationally intractable and memory-heavy. Third, models that act locally are often prone to overfitting. Thus, we need to ensure that a local calibration map does not come at the cost of statistical instability, especially for rare instances.

In the following, we address these challenges. In Section 3.1 (*Divide*), we solve the first problem by partitioning the space using vector quantization, mapping each instance to a structured region defined by shared codewords. In Section 3.2 (*Calibra*), we first introduce a compositional Dirichlet calibration model based on shared codeword-dependent factors that assigns region-specific calibration maps while sharing parameters across regions. Then, we show how parameter sharing controls the estimation error of the calibration maps, even for rare or isolated instances.

3.1 *Divide:* Voronoi Tessellation of the Latent Space

The encoder representation space induced by modern neural networks is typically *high-dimensional* and *structured*, yet calibration behavior varies substantially across different regions of this space[5]. To reason about such locality in a principled way, we adopt a geometric perspective: partitioning the space into a collection of discrete regions, where the model's predictions are approximately similar. We construct this partition by independently quantizing contiguous sub-vectors of the latent representation using a shared codebook.

Let $\mathbf{x}$ be mapped by the encoder (*e.g.,* the penultimate layer of a Neural Network) to a feature vector $\mathbf{z} = E(\mathbf{x}) \in \mathbb{R}^{m'}$. Let us define a segmentation map as $\Phi : \mathbb{R}^{m'} \to (\mathbb{R}^d)^w$, where $d, w \in \mathbb{N}$. Such a map partitions $\mathbf{z}$ into w contiguous segments (*"slots"*) of size d, so that we have $m' = d \cdot w$ and:

$$\overline{\mathbf{z}} = \left[\mathbf{z}^{(1)}, \mathbf{z}^{(2)}, \ldots, \mathbf{z}^{(w)}\right], \qquad \mathbf{z}^{(i)} \in \mathbb{R}^d \tag{2}$$

We define a *codebook* as $\mathcal{C} = \{\mathbf{c}_1, \ldots, \mathbf{c}_{|\mathcal{C}|}\} \subset \mathbb{R}^d$ where $\mathbf{c}_i \in \mathbb{R}^d$ is a *codeword*. A codebook is a finite set of representative vectors (codewords) in $\mathbb{R}^d$ that serve as building blocks for the quantization procedure. Given the segmented vector $\overline{\mathbf{z}} = \Phi(\mathbf{z})$, we assign each i-th slot $\mathbf{z}^{(i)}$ independently to a

codebook entry, following a nearest-neighbor rule (*e.g.*, the ℓ_2 norm), via the indices:

$$s(i) = \underset{j \in \{1,\dots,|\mathcal{C}|\}}{\operatorname{argmin}} \|\mathbf{z}^{(i)} - \mathbf{c}_j\|_2 \tag{3}$$

We denote the full vector of quantization indices as $\mathbf{s} = (\, s(1), \dots, s(w)\,)$ and the fully quantized latent representation as $\overline{\mathbf{q}}_{\mathbf{s}} = [\, \mathbf{c}_{s(1)}, \dots, \mathbf{c}_{s(w)}\,] \in \mathbb{R}^{w \times d}$. We define the corresponding flattened representation (obtained by concatenating the w codewords) as $\mathbf{q}_{\mathbf{s}} \in \mathbb{R}^{m'}$. The set of all possible flattened quantized configurations is $\mathcal{Q} = \left\{ \mathbf{q}_{\mathbf{s}} : \mathbf{s} \in \{1, \dots, |\mathcal{C}|\}^w \right\} \subset \mathbb{R}^{m'}$. Even if we quantize locally at the level of individual slots, product quantization partitions the full latent space into a combinatorially large number of regions [20], as the following proposition illustrates:

Proposition 1. *A global minimizer of the Euclidean distance in the flattened space,*

$$\mathbf{q}^\star = \underset{\mathbf{q} \in \mathcal{Q}}{\operatorname{argmin}} \|\mathbf{z} - \mathbf{q}\|_2 \tag{4}$$

can be obtained by assigning each slot independently to its nearest codebook vector, namely:

$$s^\star(i) = \underset{j \in \{1,\dots,|\mathcal{C}|\}}{\operatorname{argmin}} \|\mathbf{z}^{(i)} - \mathbf{c}_j\|_2 \quad \forall i, \qquad \mathbf{q}^\star = \mathbf{q}_{\mathbf{s}^\star} \tag{5}$$

Proof. We provide the proof in Appendix A.1. □

Each element $\mathbf{q} \in \mathcal{Q}$ defines a *Voronoi cell* centroid in the flattened feature space, namely: $\mathcal{V}(\mathbf{q}) = \left\{ \mathbf{z} \in \mathbb{R}^{m'} : \|\mathbf{z} - \mathbf{q}\|_2 \le \|\mathbf{z} - \mathbf{q}'\|_2 \quad \forall\, \mathbf{q}' \in \mathcal{Q} \right\}$, with potential ties handled by random assignment. These Voronoi cells form a partition of $\mathbb{R}^{m'}$ (see Figure 1b), as

$$\mathbb{R}^{m'} = \bigcup_{\mathbf{q} \in \mathcal{Q}} \mathcal{V}(\mathbf{q}), \qquad \mathcal{V}(\mathbf{q}) \cap \mathcal{V}(\mathbf{q}') = \varnothing \text{ for } \mathbf{q} \neq \mathbf{q}' \tag{6}$$

Note that (i) the standard per-slot nearest-neighbor quantization is equivalent to assigning $\mathbf{z} = E(\mathbf{x})$ to the centroid $\mathbf{q}^\star \in \mathcal{Q}$ whose Voronoi region contains it; (ii) the global Voronoi tessellation in $\mathbb{R}^{m'}$ factorizes into $|\mathcal{Q}| = |\mathcal{C}|^w$ Voronoi cells. Thus, we can interpret the latent space as a tessellation of Voronoi cells, each associated with a distinct quantized representation. Notably, this partition provides the structural foundation to build locally adaptive yet parameter-efficient calibration maps.

3.2 *Calibra:* Learning Local Calibration Maps

In this subsection, we formalize our calibration approach from a probabilistic perspective. Following Kull et al. [7], we motivate the functional form of the calibration map with a generative argument. We model the distribution of predicted probabilities conditioned on the true label and Voronoi region, and derive the corresponding calibrated posterior over labels via Bayes' rule. The resulting posterior admits a tractable log-linear form presented in Proposition 2.

Proposition 2. *Assume that, conditioned on a cell $\mathcal{V}$ and the true label $y = j$, the predicted probability vector $\hat{\mathbf{p}}$ follows a Dirichlet distribution, i.e., $\hat{\mathbf{p}} \mid (y = j, \mathcal{V}) \sim \mathrm{Dir}\big(\boldsymbol{\alpha}^{(j,\mathcal{V})}\big)$. The calibration bias and weight vector for j and $\mathcal{V}$ are:*

$$\boldsymbol{\beta}_{j,\mathcal{V}} = \log \pi_{j|\mathcal{V}} - \log \mathrm{Beta}\big(\boldsymbol{\alpha}^{(j,\mathcal{V})}\big) \qquad \boldsymbol{\omega}_{j,\mathcal{V}} = \boldsymbol{\alpha}^{(j,\mathcal{V})} - \mathbf{1} \tag{7}$$

and the logarithm of posterior probabilities admits the form:

$$\log \mathbb{P}(y = j \mid \hat{\mathbf{p}}, \mathcal{V}) = \boldsymbol{\beta}_{j,\mathcal{V}} + \boldsymbol{\omega}_{j,\mathcal{V}}^\top \log \hat{\mathbf{p}} \; + \; \text{const} \tag{8}$$

Proof. We provide the proof in Appendix A.2. □

Here, $\mathrm{Beta}(\cdot)$ denotes the multivariate Beta function associated with the Dirichlet distribution, $\pi_{j|\mathcal{V}}$ a possibly cell-dependent prior probability and $\mathbf{1} \in \mathbb{R}^{|\mathcal{Y}|}$ is the all ones vector. This results in a linear calibration model where the per-cell parameter is determined by Dirichlet concentrations. Note that, in our pipeline, the scores $\hat{\mathbf{p}}$ fed to the calibrator are produced by the quantization-aware head from $\overline{\mathbf{q}}_{\mathbf{s}}$ (Appendix D.5) and are therefore constant within each cell. Hence, the Dirichlet assumption

cannot hold literally: similarly to Kull et al. [7], who fit the Dirichlet calibration map discriminatively, we use Proposition 2 to motivate a parametric family rather than as a generative model.

The formulation in Proposition 2 shows that learning a calibration map requires specifying the Dirichlet concentration parameters $\boldsymbol{\alpha}^{(\mathcal{V})}$ for each Voronoi cell $\mathcal{V}$. However, the number of such cells grows exponentially with the number of quantization slots, making an unconstrained per-cell parameterization computationally and statistically infeasible. To address this challenge, we seek a parameterization of $\boldsymbol{\alpha}^{(\mathcal{V})}$ that (i) captures class-dependent redistribution of confidence mass, while (ii) enabling parameter sharing across regions. We achieve this by constructing region-specific calibration maps compositionally from shared codeword-dependent factors, yielding a parameter-efficient representation that generalizes across exponentially many cells.

Indexed Parameterization Trick. Instead of learning independent calibration maps per region, we construct them compositionally from shared codeword-dependent factors, allowing information to be reused across exponentially many regions. To enable compositional generalization across regions, we introduce an *indexed parameterization trick*: discrete quantization indices are reused as selectors of calibration parameters. This induces a compositional parameterization that shares statistical strength across exponentially many regions while using only a linear number of parameters. We thereby obtain a parameter-efficient way to associate each region with a pair of *sender-receiver* calibration maps.

Let $\mathcal{C} = \{\mathbf{c}_1, \ldots, \mathbf{c}_{|\mathcal{C}|}\} \subset \mathbb{R}^d$ be the *embedding codebook*, and $\mathcal{A} = \{\mathbf{a}_1, \ldots, \mathbf{a}_{|\mathcal{C}|}\} \subset \mathbb{R}^{|\mathcal{Y}|}$, $\mathcal{B} = \{\mathbf{b}_1, \ldots, \mathbf{b}_{|\mathcal{C}|}\} \subset \mathbb{R}^{|\mathcal{Y}|}$ the corresponding *receiver* and *sender* calibration codebooks, respectively. All three codebooks share the same discrete indexing set $\{1, \ldots, |\mathcal{C}|\}$. We also introduce slot dependent scaling $\boldsymbol{\sigma}^2 = (\sigma_1^2, \ldots, \sigma_w^2)$. Each sub-vector $\mathbf{z}^{(i)}$ is assigned independently to its nearest embedding centroid, yielding an index sequence $\mathbf{s} = (s(1), \ldots, s(w))$, which uniquely identifies a region $\mathcal{V}$. The same index sequence is reused to select calibration parameters from $\mathcal{A}$ and $\mathcal{B}$. This induces local *sender* and *receiver* weight matrices of shape $w \times |\mathcal{Y}|$:

$$\mathbf{A}^{(\mathcal{V})} = [\mathbf{a}_{s(1)}, \ldots, \mathbf{a}_{s(w)}]^\top \quad \mathbf{B}^{(\mathcal{V})} = [\mathbf{b}_{s(1)}, \ldots, \mathbf{b}_{s(w)}]^\top \tag{9}$$

The matrices $\mathbf{A}^{(\mathcal{V})}$ and $\mathbf{B}^{(\mathcal{V})}$ define the Dirichlet concentration parameters through the bilinear form:

$$\boldsymbol{\alpha}^{(\mathcal{V})} \;=\; \phi\big(\mathbf{A}^{(\mathcal{V})\top} \mathrm{diag}(\boldsymbol{\sigma}^2)\, \mathbf{B}^{(\mathcal{V})}\big) + \mathbf{I} \tag{10}$$

where ϕ is a positive, monotone link function applied element-wise to ensure positivity and $\boldsymbol{\alpha}^{(\mathcal{V})}$ is a $|\mathcal{Y}| \times |\mathcal{Y}|$ matrix whose j-th column is the concentration vector $\boldsymbol{\alpha}^{(j,\mathcal{V})}$ of Proposition 2, *i.e.*, $\alpha_{ij}^{(\mathcal{V})}$ encodes the evidence toward predicted class i when the true class is j. Finally, the identity shift guarantees $\alpha_{jj}^{(\mathcal{V})} > 1$ and allows the map to be initialized close to the identity (Appendix D). We provide an illustrative example of the procedure in Figure 10 in the Appendix.

This construction admits a natural interpretation in terms of class-dependent evidence redistribution. Each codeword contributes to a pair of vectors: a *sender* $\mathbf{b}_k$, describing how confidence mass is emitted when a given class is true, and a *receiver* $\mathbf{a}_k$, describing how this mass is accumulated across predicted classes. The bilinear interaction combines these contributions across slots, capturing how evidence is transferred between classes in a region-specific manner. In this sense, the model represents miscalibration as structured, class-dependent redistribution of confidence mass. Moreover, the diagonal weighting $\mathrm{diag}(\boldsymbol{\sigma}^2)$ introduces slot-dependent scaling, allowing different positions in the index sequence to contribute unequally. When the entries of $\boldsymbol{\sigma}^2$ are distinct, this breaks permutation invariance over the sequence $\mathbf{s}$, allowing different index configurations to produce distinct calibration maps. Conversely, for $\boldsymbol{\sigma}^2 = \mathbf{1}$, the map then depends on $\mathbf{s}$ only through its codeword counts.

This construction changes the number of parameters from $(|\mathcal{Y}|^2 + |\mathcal{Y}|) \times |\mathcal{C}|^w$, when per-cell concentrations and priors are learned, to $2 \times |\mathcal{C}| \times |\mathcal{Y}| + |\mathcal{Y}| + w$. In practice, when $|\mathcal{C}|, w > 1$, our proposal substantially decreases the number of parameters. Finally, parameter sharing is key to statistical stability, as it leverages recurring structure across regions. As we show in the following, it keeps the estimation error of the calibration parameters under control even in low-density regions.

Statistical Stability. The key insight of our approach is the following: while individual Voronoi cells may be sparsely populated, the *codewords* that compose them are frequently reused. By tying calibration parameters to codewords rather than cells, we gain statistical stability. We support the benefits of this parameter sharing through two *local* statistical results and a corollary, conditional on the optimization trajectory entering a well-behaved neighborhood of a stationary point of the loss.

Let θ collect the calibration parameters (the codebooks $\mathcal{A}, \mathcal{B}$, the global prior, and $\boldsymbol{\sigma}^2$ when learned) and $N = |\mathcal{D}_{\text{cal}}|$ denote the number of calibration samples. Our analysis concerns the calibration stage, conditional on the codebook and the quantization-aware head (Appendix A.3). First, let us assume the following regularity conditions hold, i.e., (A1) *smoothness*, (A2) *uniform convergence*, (A3) *gradient concentration*, (A4) *local curvature*, and (A5) *approximate stationarity*[3]. Under such assumptions, we show that, despite the compositional structure, the estimation procedure is locally statistically consistent, as formally stated in the following Theorem:

Theorem 1 (Local Rate and Consistency)**.** *Let $\mathcal{L}_N(\theta)$ be the empirical cross-entropy loss and Θ_{local} be a compact convex neighborhood of a population local minimizer $\theta^\star$ (the pseudo-true parameter). Let $\hat{\theta}_N \in \Theta_{local}$ be an approximate stationary point satisfying $\|\nabla \mathcal{L}_N(\hat{\theta}_N)\|_2 \leq \varepsilon_N$, where $\varepsilon_N \xrightarrow{p} 0$. Under the assumptions (A1)-(A5), the estimator converges in probability to the population minimizer:*

$$\|\hat{\theta}_N - \theta^\star\|_2 = O_p\left(\frac{1}{\sqrt{N}}\right) + O_p(\varepsilon_N) \tag{11}$$

In particular, if $\varepsilon_N = o_p(1)$, then $\hat{\theta}_N \xrightarrow{p} \theta^\star$.

Proof. We provide the proof in Appendix A.4. □

While consistency ensures correctness in the limit, it does not describe behavior under data sparsity. We introduce additional assumptions - (A6) weak *cross-block coupling*, (A7) *bounded conditional influence between codewords*, (A8) *standard moment and concentration conditions* and (A9) *Local Hessian Lipschitzness* - to capture the mechanism by which parameter sharing induced by VQ can stabilize optimization empirically. If these assumptions hold, the leading term of the error of each codebook parameter scales with the number of occurrences of the corresponding codeword, not with the number of calibration samples falling in a given cell $\mathcal{V}$, as shown in the following Theorem:

Theorem 2 (Frequency-Weighted Convergence of Codeword Parameters)**.** *Assume (A1)-(A9), and condition on the global scale $\boldsymbol{\sigma}^2$, the codebook $\mathcal{C}$ and the assignments induced by quantization. Let $\boldsymbol{\pi}_k$ denote a codebook parameter block $\mathbf{a}_k \in \mathcal{A}$ or $\mathbf{b}_k \in \mathcal{B}$, and let $p_k = \sum_{i=1}^{w} \mathbb{P}(s(i) = k)$ denote the expected number of occurrences of codeword k per sample. For any active codeword with $p_k > 0$,*

$$\|\hat{\boldsymbol{\pi}}_k - \boldsymbol{\pi}_k^\star\|_2 = O_p\left(\frac{1}{\sqrt{N p_k}}\right) + O_p\left(\frac{1}{\sqrt{N}}\right) + O_p\left(\frac{\varepsilon_N}{p_k}\right) \tag{12}$$

and, since $N_k/N \xrightarrow{p} p_k$, the leading codeword-specific statistical term scales as $1/\sqrt{N_k}$.

Proof. We provide the proof in Appendix A.5. □

Note that Theorem 2 applies to each codeword individually and does not imply convergence over the entire codebook. Moreover, when the optimization error is negligible relative to the statistical term, i.e., $\varepsilon_N = o_p(\sqrt{N_k}/N)$, estimation depends on the actual number of occurrences of the corresponding codeword. The main benefit of parameter sharing is precisely that a codeword can be accurately estimated from all of its occurrences across the calibration set, independently of how frequently any particular combination of codewords is observed. Consequently, a Voronoi cell may be rare, or even absent from the calibration sample, but still associated with a well-estimated calibration map, provided that its constituent codewords are sufficiently prevalent. We now formalize this cell-level property by bounding the estimation error of the corresponding Dirichlet concentrations.

Corollary 1 (Cell-Level Estimation Error)**.** *Under the setting of Theorem 2, further assume that the link function ϕ is positive, monotone and 1-Lipschitz, e.g., the softplus. Since the codebook is fixed and finite, for every fixed cell $\mathcal{V}$ with index sequence $\mathbf{s}$ of active codewords, the Frobenius norm of the estimation error in the local Dirichlet concentrations satisfies:*

$$\left\|\hat{\boldsymbol{\alpha}}^{(\mathcal{V})} - \boldsymbol{\alpha}^{(\mathcal{V})\star}\right\|_F = O_p\left(\sum_{i=1}^{w} \sigma_i^2 \left[\frac{1}{\sqrt{N p_{s(i)}}} + \frac{1}{\sqrt{N}} + \frac{\varepsilon_N}{p_{s(i)}}\right]\right) \tag{13}$$

Proof. We provide the proof in Appendix A.6. □

[3]We provide a detailed description of all our assumptions for Theorem 1, Theorem 2 and Corollary 1 in Appendix A.3.

Hence, the estimation error of a cell depends on the estimation errors of its constituent codewords, rather than on the number of calibration samples falling in that cell. In particular, a cell may contain few or even no calibration samples while still admitting a well-estimated calibration map, provided that its constituent codewords are active and sufficiently well estimated. Because $\|\mathbf{a}_k^\star\|_2$ and $\|\mathbf{b}_k^\star\|_2$ are uniformly bounded over active codewords, combining the above bound with Theorem 2 yields a cell-level error controlled by the frequencies of the codewords appearing in the index sequence $\mathbf{s}$.

Learning procedure. In practice, we propose a two-stage procedure (shown in Algorithm 1 in the Appendix) to decouple representation learning from calibration:

1. *Quantization-Aware Representation Learning:* We freeze the backbone encoder and train the codebook $\mathcal{C}$ and a quantization-aware classification head. Codebooks are updated via Exponential Moving Average (EMA) to capture the density of the latent space [21].
2. *Region-Aware Calibration:* We freeze the codebook and quantized classifier, then optimize the calibration parameters $(\mathcal{A}, \mathcal{B})$ (and, optionally, $\boldsymbol{\sigma}^2$, which we keep fixed in our experiments) by minimizing the negative log-likelihood of the calibrated posterior.

Both stages are trained on the calibration set $\mathcal{D}_{\text{cal}}$ (Algorithm 1). The output is a codebook $\mathcal{C}$ defining the tessellation of the latent space and a calibrated predictor f_{cal}.

4 Experimental Evaluation

In our experiments, we address the following questions:

- **Q1:** Does our method improve over existing baselines on *local calibration* metrics?
- **Q2:** How does our method perform on low-support regions, where calibration is hardest?
- **Q3:** Is our method competitive with existing baselines on *global calibration* metrics?

The code[4] can be found at `https://github.com/Cesbar99/Divide-et-Calibra`.

4.1 Experimental Settings

Datasets. We evaluate the research questions over the same image multiclass datasets used by Barbera et al. [11], *i.e.*, classic benchmark data `Cifar10`, `Cifar100` [22], and real-world medical data `TissueMNIST` from the `MedMNIST` collection [23–25]. We also evaluate our approach on tabular data, using the `Weather` dataset [26]; for space reasons, tabular results are reported in Appendix C.1, where VQ attains the best LCE and is competitive on both $MLCE$ and global calibration metrics.

Methods and Architectures. We evaluate our approach (VQ) against both global and local calibrators. We consider publicly available calibration baselines, including Temperature Scaling (TS) [27], Isotonic Regression (IR) [28], and Platt Scaling (PS) [29]. In addition, we compare with multiclass parametric models, *i.e.*, Dirichlet Calibration (DC) [7] and Structured Matrix Scaling (SM) [8], and local approaches such as KCal (KC) [15], Local Nets (LN) [11] and ProCal (PC) [5]. Finally, we include the base model when no calibration is performed (NC).

For image data, we report results using `ResNet` architectures in the main paper, for a direct comparison with the Barbera et al. [11] benchmark. Results are largely consistent across backbones, with smaller gains for `ViT` on `Cifar10` and `Cifar100`. We provide values for `ConvNext` [30] and `ViT` [31] architectures in Appendix C.6 and Appendix C.7, respectively.

Implementation details. For all experiments, we set the number of slots $w = 64$ and codebook size $|\mathcal{C}| = 64$ for all VQ methods. Still, our proposal is robust to this choice of hyperparameters, showing good performance across a wide range of values. We provide results when varying w and $|\mathcal{C}|$ in Appendix C.3. Moreover, we fix $\text{diag}(\boldsymbol{\sigma}^2) = \mathbf{I}$ to avoid over-fitting due to limited calibration sets. Indeed, over-fitting is a well-known issue in post-hoc calibration, where simpler models and stronger regularization are often preferred [8]. This choice yields a regularized model that preserves the compositional structure induced by codeword reuse. We instead report a variant of our method with a regularized, learnable $\text{diag}(\boldsymbol{\sigma}^2)$ in Appendix C.9. This variant achieves performance comparable to

[4] Code developed by authors from University of Trento.

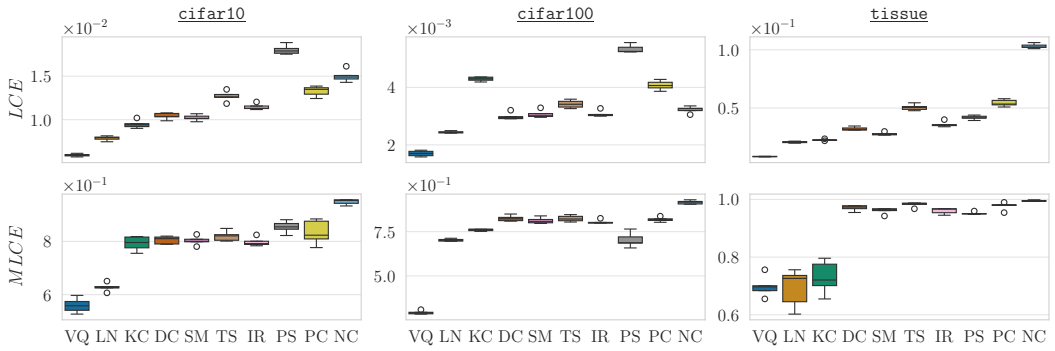


Figure 2: Local calibration metrics over five runs (`ResNet` backbone). The lower the better.

the fixed-scaling formulation presented here, with modest improvements on some metrics. Similarly, we use a global learned prior $\pi_{j|\mathcal{V}} = \pi_j$, so that locality enters the calibration bias $\boldsymbol{\beta}_{j,\mathcal{V}}$ only through $\log \mathrm{Beta}(\boldsymbol{\alpha}^{(j,\mathcal{V})})$. We detail all other hyperparameters in Appendix D.

Metrics. We evaluate local calibration using both LCE and $MLCE$ (see Section 2) as the main metrics of interest. We assess global calibration using $ECCE$ and the Negative Log-Likelihood (NLL), which is a proper scoring rule and captures both calibration and predictive performance. Moreover, we report in Appendix C also ECE and accuracy (ACC) for all datasets and baselines.

Experimental Setup. For image datasets, we consider the Barbera et al. [11] data splits, *i.e.*, (a) a training set; (b) a calibration set (with an internal $90\%/10\%$ calibration/validation split); and (c) a held-out test set used exclusively for evaluation. Then, we (i) train classifiers on the training set; (ii) train calibration methods on the calibration set; (iii) compute the metrics defined above on the test set. We repeat this procedure using 5 different seeds and average results. We provide further implementation details in Appendix D.

Additional Experiments. We include in Appendix C.2 further ablations for the components of our calibration pipeline (confirming quantization as the primary source of improvement, with consistent gains from our compositional parameterization) and for the calibration set size (showing consistent performance; see Appendix C.4). Finally, in Appendix C.8 we report codeword usage statistics, showing good codebook utilizations and lack of unused codewords. We further disentangle the effect of quantization from classification-head adaptation in Appendix C.10, investigate the role of the quantization partition itself in Appendix C.11, and evaluate sensitivity of LCE and MLCE to the kernel bandwidth in Appendix C.12. Finally, Appendix C.13 reports training-time measurements.

4.2 Experimental Results

Q1: VQ outperforms all baselines in local calibration metrics. Figure 2 shows the results for local calibration metrics, *i.e.*, LCE and $MLCE$.

Regarding LCE, our approach outperforms all existing baselines on all datasets: on `Cifar10`, VQ reaches $LCE \approx 0.0059 \pm 0.0002$, obtaining a $\approx 25\%$ drop with respect to LN, the closest competitor ($LCE \approx 0.0079 \pm 0.0003$); on `Cifar100`, VQ obtains $LCE \approx 0.0017 \pm 0.0001$, a $\approx 29\%$ drop compared to LN (the second-best with $LCE \approx 0.0024 \pm 0.0001$); on `tissue`, VQ's LCE is $\approx 0.0083 \pm 0.0002$, which is a $\approx 60\%$ drop with respect to LN ($LCE \approx 0.0208 \pm 0.0007$). Regarding $MLCE$, we see similar results, with VQ achieving the best results on `Cifar10` and `Cifar100` and ranking second best on `tissue`.

Overall, results indicate that our approach outperforms competitors on local calibration metrics.

Q2: VQ obtains most pronounced gains in low support regions. To better understand the behavior of calibration under data sparsity, we analyze local calibration error as a function of the Effective Sample Size (ESS) of each point neighborhood. Specifically, we partition samples into quantile-based bins according to their ESS, computed from kernel weights in the representation space, and report the average absolute LCE within each bin (Figure 3).

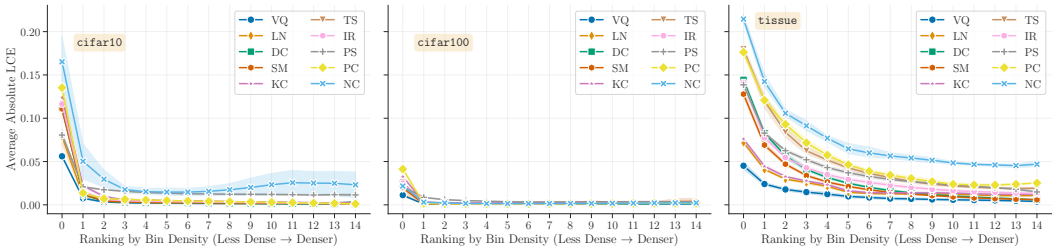


Figure 3: Local calibration in density-based sub-bins over five runs (`ResNet` backbone).

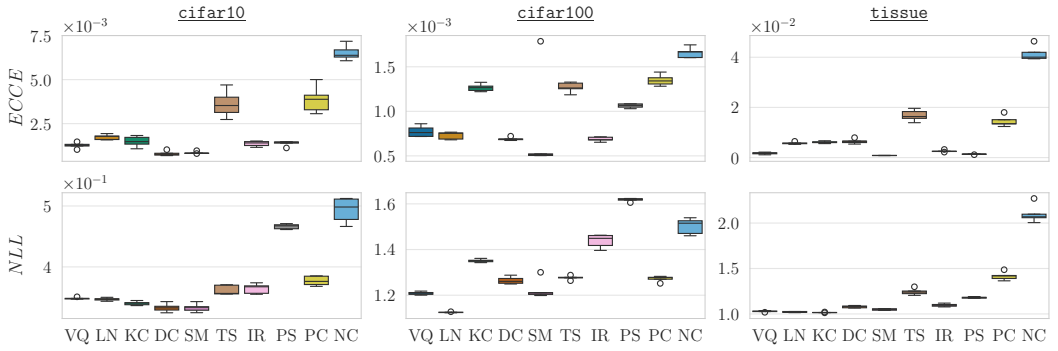


Figure 4: Global calibration metrics over five runs (`ResNet` backbone). The lower the better.

We observe similar trends for all datasets: local calibration error decreases as the effective sample size increases. This behavior is expected, as points with low ESS belong to low-density regions, which are harder to correct. Importantly, while all methods exhibit this trend, our VQ-based approach consistently achieves lower LCE across the entire ESS spectrum, with the most pronounced gains in low-support regions. This suggests that VQ effectively mitigates the impact of data sparsity. We attribute this behavior to the parameter sharing induced by vector quantization: although individual Voronoi regions may be sparsely populated, their constituent codewords are reused across many samples. Interestingly, this empirical observation is consistent with our theoretical analysis (Section 3.2), where the estimation error of the calibration parameters depends on codeword frequency rather than on the number of samples in each region.

Overall, these findings highlight that our method reduces the sensitivity of calibration error to data sparsity, enabling reliable calibration even in low-density regions of the representation space.

Q3: VQ reaches comparable results on other metrics. Figure 4 reports $ECCE$ and NLL for all methods and datasets (ECE and ACC are reported in Table 6 of the appendix).

Regarding $ECCE$, our approach improves upon other local approaches, such as KC, LN and PC, on `Cifar10` and `tissue`, while performing similarly on `Cifar100`. Similarly, when compared to SM, the strongest global calibration method, VQ achieves comparable results on `Cifar100` while closely following on the remaining datasets (see Table 6 for detailed results). For NLL, VQ improves over the base model, while remaining slightly behind the best results in some settings. As intended, VQ delivers significant improvements on local calibration metrics, with its clearest advantages on sparse regions. On global calibration and predictive metrics, it remains competitive with strong baselines, indicating that local gains are not obtained at the cost of severe global degradation.

5 Related Work

Global Calibration. Standard post-hoc calibration involves learning a single, global map to obtain calibrated probabilities. Parametric methods such as *Platt Scaling* [29], *Temperature Scaling* [27], and *Beta Calibration* [32] are widely used. These methods generally learn a monotonic function to rescale the logits (or log-probabilities), preserving the rank ordering of classes. Non-parametric approaches, such as *Isotonic Regression* [28] and *Histogram Binning* [33], offer greater flexibility

but require significantly more data to estimate the calibration map, as they rely on having sufficient samples in each bin to compute reliable statistics. In the multiclass setting, *Dirichlet Calibration* (DC) [7] generalizes the Beta calibration framework to the probability simplex. Recently, Berta et al. [8] introduced an effective regularization scheme showing promising results in mitigating over-fitting.

Local Calibration. To address the limitations of global maps, *local calibration* methods make the correction function dependent on the input features or embeddings. Approaches like *Local Temperature Scaling* [34], *KCal* [15], *LECE* [10] and *ProCal* [5] typically employ kernel density estimation or nearest-neighbor lookups to estimate local error rates. More recently, *Local Nets* [11] proposed using a secondary neural network to predict calibrated probabilities dynamically. Our VQ-based approach differs from these techniques by replacing continuous density estimation with Voronoi tessellations. This allows statistics to be pooled across regions through shared codewords, so that the estimation error of each local calibration map is controlled by codeword frequency (Section 3.2).

Multicalibration. Another calibration notion, known as *Multicalibration*, requires the model to be calibrated not just on the entire population, but on virtually any sub-population identifiable by a specific hypothesis class [35–40]. Thus, the goal of multicalibration approaches is to ensure that ML models do *not* make biased predictions on any of these sub-populations [41, 42]. While related, multicalibration differs from our setting in two key aspects. First, it is mostly studied for binary/regression tasks, whereas we address *multiclass* calibration and the complexities of the probability simplex. Second, it identifies semantically meaningful subgroups (*e.g.*, `User Age` $>$ `18`) using expressive features in tabular data.

Calibration and Conformal Prediction. Conformal prediction provides distribution-free prediction sets with finite-sample marginal coverage guarantees under exchangeability [43]. Mondrian conformal prediction extends this idea providing finer guarantees than global marginal validity conditioning on predefined categories or partitions of the data [44, 45]. More broadly, recent work has also investigated how post-hoc calibration affects the coverage and efficiency of conformal predictors [46, 47]. These approaches are complementary to our setting, which instead learns region-specific calibration maps.

6 Conclusion

In this work, we tackle the trade-off between stable but coarse global calibration techniques and flexible but high-variance local calibration approaches. We proposed a compositional approach to local multiclass calibration, instantiated via vector quantization. Empirical evidence shows improvements over existing baselines in terms of local calibration metrics, while retaining comparable results on global calibration and classification metrics.

Limitations and Future Work. Our method relies on a frozen representation space, and therefore its success depends on the extent to which that space captures locality relevant to miscalibration. If latent features are poorly aligned with the calibration structure the induced Voronoi partition may be less effective. Interestingly, such partition defines a finite family of groups in latent space, suggesting a promising direction for future work: bridging local calibration and multicalibration by deriving guarantees for input-space groups while operating over representation-defined subpopulations.

Moreover, our theory is local and conditional on regularity assumptions, stable quantization assignments, and convergence to a well-behaved stationary point. It covers the calibration stage only and treats the codebook and the quantization-aware head as fixed. Extending it to joint end-to-end learning via recent VQ differentiable methods [48] or growing-codebook regimes is left for future work.

We further stress that, while VQ consistently improves local calibration empirically, it does not uniformly outperform strong global baselines on $ECCE$ or NLL, and its advantages are clearest in low-support in-distribution regions. This suggests that a promising direction is to design calibration models that explicitly learn when to apply local corrections and when to rely on global adjustments.

Additionally, while our empirical evaluation includes multiple datasets, baselines, and architectures, it remains focused on classification settings. Extending the study to additional modalities, larger-scale tasks, and more diverse forms of distribution shift is an important direction for future work.

In conclusion, better-calibrated probabilities matter because they turn model scores into information that people can act on: investigating how reliable probability estimates interact with downstream decision-making and uncertainty-aware guarantees is a fundamental direction for future research.

Broader Impact

This work aims to improve the reliability of probabilistic predictions by reducing miscalibration in local regions of the data space, which is particularly relevant when models are used for risk-sensitive decisions (e.g., medical triage, content moderation, or autonomous systems). Better local calibration enables more trustworthy uncertainty estimates, possibly improving downstream decision-making.

Acknowledgements

We thank the Anonymous Reviewers and Area Chair for their feedback. Cesare Barbera, Andrea Pugnana and Andrea Passerini acknowledge the CINECA award under the ISCRA initiative for the availability of high-performance computing resources and support. This work was funded by the European Union. The views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union, the European Health and Digital Executive Agency (HaDEA) or the European Research Executive Agency. Neither the European Union nor the granting authority can be held responsible for them. Grant Agreement no. 101120763 - TANGO.

References

[1] Cheng Wang. Calibration in deep learning: A survey of the state-of-the-art. *CoRR*, abs/2308.01222, 2023.

[2] Abhishek Singh Sambyal, Usma Niyaz, Narayanan C. Krishnan, and Deepti R. Bathula. Understanding calibration of deep neural networks for medical image classification. *Comput. Methods Programs Biomed.*, 242:107816, 2023.

[3] Giuseppe Cianci, Roberto Goglia, Riccardo Guidotti, Matteo Kapllaj, Roberto Mosca, Andrea Pugnana, Franco Ricotti, and Salvatore Ruggieri. Applied data science for leasing score prediction. In *IEEE Big Data*, pages 1687–1696. IEEE, 2023.

[4] Salvatore Ruggieri and Andrea Pugnana. Things machine learning models know that they don't know. In *AAAI*, pages 28684–28693. AAAI Press, 2025.

[5] 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 *NeurIPS*, 2023.

[6] 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 *UAI*, volume 180 of *Proceedings of Machine Learning Research*, pages 1286–1295. PMLR, 2022.

[7] 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 *NeurIPS*, pages 12295–12305, 2019.

[8] Eugene Berta, David Holzmüller, Michael I. Jordan, and Francis Bach. Structured Matrix Scaling for Multi-Class Calibration. In *AISTATS*, 2026.

[9] Juozas Vaicenavicius, David Widmann, Carl R. Andersson, Fredrik Lindsten, Jacob Roll, and Thomas B. Schön. Evaluating model calibration in classification. In *AISTATS*, volume 89 of *Proceedings of Machine Learning Research*, pages 3459–3467. PMLR, 2019.

[10] Kaspar Valk and Meelis Kull. Assuming locally equal calibration errors for non-parametric multiclass calibration. *Transactions on Machine Learning Research*, 2023.

[11] Cesare Barbera, Lorenzo Perini, Giovanni De Toni, Andrea Passerini, and Andrea Pugnana. Multiclass Local Calibration With the Jensen-Shannon Distance. In *AISTATS*, 2026.

[12] Mahdi Pakdaman Naeini, Gregory F. Cooper, and Milos Hauskrecht. Obtaining well calibrated probabilities using bayesian binning. In *AAAI*, pages 2901–2907. AAAI Press, 2015.

[13] Imanol Arrieta Ibarra, Paman Gujral, Jonathan Tannen, Mark Tygert, and Cherie Xu. Metrics of calibration for probabilistic predictions. *J. Mach. Learn. Res.*, 23:351:1–351:54, 2022.

[14] Polina Kirichenko, Pavel Izmailov, and Andrew Gordon Wilson. Last layer re-training is sufficient for robustness to spurious correlations. In *ICLR*. OpenReview.net, 2023.

[15] Zhen Lin, Shubhendu Trivedi, and Jimeng Sun. Taking a step back with kcal: Multi-class kernel-based calibration for deep neural networks. In *ICLR*. OpenReview.net, 2023.

[16] Kevin Beyer, Jonathan Goldstein, Raghu Ramakrishnan, and Uri Shaft. When is nearest neighbor meaningful? In *International conference on database theory*, pages 217–235. Springer, 1999.

[17] Larry Wasserman. *All of nonparametric statistics*. Springer, 2006.

[18] Trevor Hastie, Robert Tibshirani, Jerome Friedman, et al. The elements of statistical learning, 2009.

[19] Vladislav Polianskii, Giovanni Luca Marchetti, Alexander Kravberg, Anastasiia Varava, Florian T Pokorny, and Danica Kragic. Voronoi density estimator for high-dimensional data: Computation, compactification and convergence. In *Uncertainty in Artificial Intelligence*, pages 1644–1653. PMLR, 2022.

[20] Hervé Jégou, Matthijs Douze, and Cordelia Schmid. Product quantization for nearest neighbor search. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(1):117–128, 2011.

[21] Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *Advances in neural information processing systems*, 30, 2017.

[22] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[23] Jiancheng Yang, Rui Shi, and Bingbing Ni. Medmnist classification decathlon: A lightweight automl benchmark for medical image analysis. In *ISBI*, pages 191–195. IEEE, 2021.

[24] Jiancheng Yang, Rui Shi, Donglai Wei, Zequan Liu, Lin Zhao, Bilian Ke, Hanspeter Pfister, and Bingbing Ni. Medmnist v2-a large-scale lightweight benchmark for 2d and 3d biomedical image classification. *Scientific Data*, 10(1):41, 2023.

[25] Vebjorn Ljosa, Katherine L Sokolnicki, and Anne E Carpenter. Annotated high-throughput microscopy image sets for validation. *Nature methods*, 9(7):637, 2012.

[26] Andrey Malinin, Neil Band, Yarin Gal, Mark J. F. Gales, Alexander Ganshin, German Chesnokov, Alexey Noskov, Andrey Ploskonosov, Liudmila Prokhorenkova, Ivan Provilkov, Vatsal Raina, Vyas Raina, Denis Roginskiy, Mariya Shmatova, Panagiotis Tigas, and Boris Yangel. Shifts: A dataset of real distributional shift across multiple large-scale tasks. In *NeurIPS Datasets and Benchmarks*, 2021.

[27] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In *ICML*, volume 70 of *Proceedings of Machine Learning Research*, pages 1321–1330. PMLR, 2017.

[28] Bianca Zadrozny and Charles Elkan. Transforming classifier scores into accurate multiclass probability estimates. In *KDD*, pages 694–699. ACM, 2002.

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

[30] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A ConvNet for the 2020s. In *CVPR*, pages 11966–11976. IEEE, 2022.

[31] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *ICLR*. OpenReview.net, 2021.

[32] Meelis Kull, Telmo Silva Filho, and Peter Flach. Beta calibration: a well-founded and easily implemented improvement on logistic calibration for binary classifiers. In *Artificial intelligence and statistics*, pages 623–631. PMLR, 2017.

[33] Bianca Zadrozny and Charles Elkan. Obtaining calibrated probability estimates from decision trees and naive bayesian classifiers. In *ICML*, pages 609–616. Morgan Kaufmann, 2001.

[34] Zhipeng Ding, Xu Han, Peirong Liu, and Marc Niethammer. Local temperature scaling for probability calibration. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6889–6899, 2021.

[35] Ursula Hébert-Johnson, Michael Kim, Omer Reingold, and Guy Rothblum. Multicalibration: Calibration for the (computationally-identifiable) masses. In *International Conference on Machine Learning*, pages 1939–1948. PMLR, 2018.

[36] Christopher Jung, Changhwa Lee, Mallesh Pai, Aaron Roth, and Rakesh Vohra. Moment multicalibration for uncertainty estimation. In *Conference on Learning Theory*, pages 2634–2678. PMLR, 2021.

[37] Riccardo Colini Baldeschi, Simone Di Gregorio, Simone Fioravanti, Federico Fusco, Ido Guy, Daniel Haimovich, Stefano Leonardi, Fridolin Linder, Lorenzo Perini, Matteo Russo, Cem Sirin, and Niek Tax. Multicalibration yields better matchings. In *International Conference on Machine Learning*, 2026. URL `https://openreview.net/forum?id=G1bzbJtcHD`.

[38] Hongyi Henry Jin, Zijun Ding, Dung Daniel Ngo, and Zhiwei Steven Wu. Discretization-free multicalibration through loss minimization over tree ensembles. *arXiv preprint arXiv:2505.17435*, 2025.

[39] Niek Tax, Lorenzo Perini, Fridolin Linder, Daniel Haimovich, Dima Karamshuk, Nastaran Okati, Milan Vojnovic, and Pavlos Athanasios Apostolopoulos. Mcgrad: Multicalibration at web scale. In *KDD (1)*, pages 2470–2481. ACM, 2026.

[40] Daniel Haimovich, Fridolin Linder, Lorenzo Perini, Niek Tax, and Milan Vojnovic. On the convergence of multicalibration gradient boosting. *CoRR*, abs/2602.06773, 2026.

[41] Ira Globus-Harris, Varun Gupta, Christopher Jung, Michael Kearns, Jamie Morgenstern, and Aaron Roth. Multicalibrated regression for downstream fairness. In *Proceedings of the 2023 AAAI/ACM Conference on AI, Ethics, and Society*, pages 259–286, 2023.

[42] Georgy Noarov and Aaron Roth. The statistical scope of multicalibration. In *International Conference on Machine Learning*, pages 26283–26310. PMLR, 2023.

[43] Vladimir Vovk, Alexander Gammerman, and Glenn Shafer. *Algorithmic learning in a random world*. Springer, 2005.

[44] Henrik Boström, Ulf Johansson, and Tuwe Löfström. Mondrian conformal predictive distributions. In *COPA*, volume 152 of *Proceedings of Machine Learning Research*, pages 24–38. PMLR, 2021.

[45] Lars van der Laan and Ahmed M. Alaa. Generalized venn and venn-abers calibration with applications in conformal prediction. In *ICML*, volume 267 of *Proceedings of Machine Learning Research*. PMLR / OpenReview.net, 2025.

[46] Lahav Dabah and Tom Tirer. On temperature scaling and conformal prediction of deep classifiers. In *ICML*, volume 267 of *Proceedings of Machine Learning Research*. PMLR / OpenReview.net, 2025.

[47] Huajun Xi, Jianguo Huang, Kangdao Liu, Lei Feng, and Hongxin Wei. Does confidence calibration improve conformal prediction? *Trans. Mach. Learn. Res.*, 2025, 2025.

[48] Mohammad Hassan Vali, Tom Bäckström, and Arno Solin. Diveq: Differentiable vector quantization using the reparameterization trick. 2026.

[49] Aad W Van der Vaart. *Asymptotic statistics*, volume 3. Cambridge university press, 2000.

[50] Halbert White. Maximum likelihood estimation of misspecified models. *Econometrica: Journal of the econometric society*, pages 1–25, 1982.

[51] Teodora Popordanoska, Raphael Sayer, and Matthew B. Blaschko. A consistent and differentiable lp canonical calibration error estimator. In *NeurIPS*, 2022.

[52] Teodora Popordanoska, Sebastian Gregor Gruber, Aleksei Tiulpin, Florian Büttner, and Matthew B. Blaschko. Consistent and asymptotically unbiased estimation of proper calibration errors. In *AISTATS*, volume 238 of *Proceedings of Machine Learning Research*, pages 3466–3474. PMLR, 2024.

[53] Yury Gorishniy, Ivan Rubachev, Valentin Khrulkov, and Artem Babenko. Revisiting deep learning models for tabular data. In *NeurIPS*, pages 18932–18943, 2021.

[54] Yunye Gong, Xiao Lin, Yi Yao, Thomas G. Dietterich, Ajay Divakaran, and Melinda T. Gervasio. Confidence calibration for domain generalization under covariate shift. In *ICCV*, pages 8938–8947. IEEE, 2021.

[55] Ross Wightman et al. Pytorch image models. 2019.

[56] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR (Poster)*, 2015.

A Proofs

A.1 Proof of Proposition 1

Statement: A global minimizer of the Euclidean distance in the flattened space,

$$\mathbf{q}^\star = \operatorname*{argmin}_{\mathbf{q}\in\mathcal{Q}} \|\mathbf{z}-\mathbf{q}\|_2 \tag{14}$$

can be obtained by assigning each slot independently to its nearest codebook vector, namely:

$$s^\star(i) = \operatorname*{argmin}_{j\in\{1,\dots,|\mathcal{C}|\}} \|\mathbf{z}^{(i)}-\mathbf{c}_j\|_2 \quad \forall i, \qquad \mathbf{q}^\star = \mathbf{q}_{\mathbf{s}^\star} \tag{15}$$

Proof. It suffices to show that:

$$\min_{\mathbf{q}\in\mathcal{Q}} \|\mathbf{z}-\mathbf{q}\|^2 = \sum_{i=1}^{w} \min_{j\in\{1,\dots,|\mathcal{C}|\}} \|\mathbf{z}^{(i)}-\mathbf{c}_j\|^2.$$

Take any index vector $\mathbf{s}$,

$$\|\mathbf{z}-\mathbf{q}_\mathbf{s}\|^2 = \left\|(\mathbf{z}^{(1)}-\mathbf{c}_{s(1)},\dots,\mathbf{z}^{(w)}-\mathbf{c}_{s(w)})\right\|^2 = \sum_{i=1}^{w} \|\mathbf{z}^{(i)}-\mathbf{c}_{s(i)}\|^2.$$

Because the Euclidean norm is additive over the w blocks, minimizing over all vectors $\mathbf{s}$ separates into w independent minimizations, each solved by:

$$s^\star(i) = \arg\min_j \|\mathbf{z}^{(i)}-\mathbf{c}_j\|^2.$$

The vector $\mathbf{s}^\star$ containing those indices therefore corresponds to the flattened codeword $\mathbf{q}^\star$ that attains the global minimum. □

A.2 Proof of Proposition 2

Statement: Assume that, conditioned on a cell $\mathcal{V}$ and the true label $y=j$, the predicted probability vector $\hat{\mathbf{p}}$ follows a Dirichlet distribution, i.e., $\hat{\mathbf{p}} \mid (y=j,\,\mathcal{V}) \;\sim\; \mathrm{Dir}\big(\boldsymbol{\alpha}^{(j,\mathcal{V})}\big)$. The calibration bias and weight vector for j and $\mathcal{V}$ are:

$$\boldsymbol{\beta}_{j,\mathcal{V}} = \log \pi_{j|\mathcal{V}} - \log \mathrm{Beta}\big(\boldsymbol{\alpha}^{(j,\mathcal{V})}\big) \qquad \boldsymbol{\omega}_{j,\mathcal{V}} = \boldsymbol{\alpha}^{(j,\mathcal{V})} - \mathbf{1} \tag{16}$$

and the logarithm of posterior probabilities admits the form:

$$\log \mathbb{P}(y=j \mid \hat{\mathbf{p}}, \mathcal{V}) = \boldsymbol{\beta}_{j,\mathcal{V}} + \boldsymbol{\omega}_{j,\mathcal{V}}^\top \log \hat{\mathbf{p}} \;+\; \mathrm{const} \tag{17}$$

Proof. Conditioned on the true label $y=j$ and the Voronoi cell $\mathcal{V}$, we assume the model's output probability vector $\hat{\mathbf{p}}$ is distributed according to a Dirichlet:

$$\hat{\mathbf{p}} \mid (y=j,\,\mathcal{V}) \;\sim\; \mathrm{Dir}\big(\boldsymbol{\alpha}^{(j,\mathcal{V})}\big), \tag{18}$$

where $\boldsymbol{\alpha}^{(j,\mathcal{V})}$ encodes the model's belief structure over classes conditioned on the true label and the cell. Then the Dirichlet density is

$$\mathbb{P}\big(\hat{\mathbf{p}} \mid y=j,\mathcal{V}\big) = \frac{1}{\mathrm{Beta}\big(\boldsymbol{\alpha}^{(j,\mathcal{V})}\big)} \prod_{i=1}^{|\mathcal{Y}|} \hat{\mathbf{p}}_i^{\boldsymbol{\alpha}_i^{(j,\mathcal{V})}-1}, \tag{19}$$

where $\mathrm{Beta}(\cdot)$ is the multivariate Beta function. Assuming a (possibly cell-dependent) prior $\pi_{j|\mathcal{V}} = \mathbb{P}(y=j \mid \mathcal{V})$, the posterior over labels given the observed $\hat{\mathbf{p}}$ and cell $\mathcal{V}$ is

$$\begin{aligned}\mathbb{P}(y=j \mid \hat{\mathbf{p}}, \mathcal{V}) &\propto \pi_{j|\mathcal{V}}\; p\big(\hat{\mathbf{p}} \mid y=j,\mathcal{V}\big) \\ &= \pi_{j|\mathcal{V}}\; \frac{1}{\mathrm{Beta}\big(\boldsymbol{\alpha}^{(j,\mathcal{V})}\big)} \prod_{i=1}^{|\mathcal{Y}|} \hat{\mathbf{p}}_i^{\boldsymbol{\alpha}_i^{(j,\mathcal{V})}-1}.\end{aligned} \tag{20}$$

Log-linear calibration form. Taking logs in (20) gives

$$\log p(y = j \mid \hat{\mathbf{p}}, \mathcal{V}) = \log \pi_{j|\mathcal{V}} - \log \mathrm{Beta}\big(\boldsymbol{\alpha}^{(j,\mathcal{V})}\big) + \sum_{i=1}^{|\mathcal{Y}|} \big(\boldsymbol{\alpha}_i^{(j,\mathcal{V})} - 1\big) \log \hat{\mathbf{p}}_i + \text{const}, \tag{21}$$

where the additive constant enforces normalization across j. The calibration bias and weight vector for j and $\mathcal{V}$ are

$$\boldsymbol{\beta}_{j,\mathcal{V}} = \log \pi_{j|\mathcal{V}} - \log \mathrm{Beta}\big(\boldsymbol{\alpha}^{(j,\mathcal{V})}\big) \quad \boldsymbol{\omega}_{j,\mathcal{V}} = \boldsymbol{\alpha}^{(j,\mathcal{V})} - \mathbf{1} \tag{22}$$

where $\mathbf{1} \in \mathbb{R}^{|\mathcal{Y}|}$ is the all-ones vector. Then, (21) simplifies:

$$\log \mathbb{P}(y = j \mid \hat{\mathbf{p}}, \mathcal{V}) = \boldsymbol{\beta}_{j,\mathcal{V}} + \boldsymbol{\omega}_{j,\mathcal{V}}^{\top} \log \hat{\mathbf{p}} \; + \; \text{const}.$$

□

A.3 Assumptions for Theorem 1 and Theorem 2

In the following we discuss the assumptions underlying our statistical results. Assumptions (A1)-(A5) are standard local regularity conditions for M-estimators and Z-estimators, see *e.g.*, [49, 50]. They ensure well-defined population gradients and Hessians, local identifiability of the statistical target, uniform convergence of the empirical objective and curvature in a neighborhood of the solution, and stability under approximate stationarity of the optimization procedure.

Assumptions (A6)-(A7) impose a weak blockwise coupling condition through Schur dominance of the population Hessian. Intuitively, this rules out pathological over-parameterization in which changes in one block can be arbitrarily compensated by others, and is realistic in compositional models where codewords contribute additively and appear with heterogeneous frequencies.

Assumption (A8) collects mild moment and dependence conditions needed to control empirical gradients and Hessians via standard concentration and law-of-large-numbers arguments.

Finally, Assumption (A9) imposes a local Lipschitz continuity condition on the empirical Hessian. This is a standard smoothness requirement ensuring that curvature does not vary too abruptly within the neighborhood Θ_{local}. Such conditions are mild and are typically satisfied when the loss is sufficiently smooth in a local neighborhood of the optimum.

Throughout, we work in a local fixed-dimension regime (with $|\mathcal{C}|, w$ treated as fixed and independent of N), and all results are local in nature, characterizing statistical behavior once optimization reaches the basin of a well-specified solution. Theorem 2 additionally conditions on the global scale parameters $\boldsymbol{\sigma}^2$ being locally stable within the same basin of attraction as the pseudo-true parameter $\theta^\star$ to isolate the statistical behavior of the compositional codebook parameters. Extending the analysis beyond this assumption is out of the scope of this work.

We stress that the bilinear calibration parameterization is invariant to reciprocal rescaling of the sender and receiver factors. In particular, for each active codeword k,

$$(\mathbf{a}_k, \mathbf{b}_k) \mapsto (t_k \mathbf{a}_k, \mathbf{b}_k / t_k), \qquad t_k \neq 0, \tag{23}$$

leaves the induced calibration map unchanged. Our statistical analysis is therefore understood on a locally identifiable parameterization obtained by fixing this scale ambiguity (equivalently, by imposing a local normalization constraint on one factor of each active codeword). All gradients and Hessians below are taken with respect to the resulting free local coordinates. In the experiments, we instead resolve this degeneracy operationally through the small weight-decay penalty used during optimization. The regularization used in training is not part of the statistical objective analyzed below.

More precisely, we assume there exists a neighborhood $\mathcal{N}(\theta^\star)$ and a convex compact set $\Theta_{\mathrm{local}} \subset \mathcal{N}(\theta^\star)$ with $\theta^\star \in \Theta_{\mathrm{local}}$ such that the following holds:

- (A1) **Smoothness and exchangeability of expectation.** The loss $\mathcal{L}(z;\theta)$ is twice continuously differentiable in a neighborhood of $\theta^\star$, with derivatives dominated by integrable envelopes so that differentiation may be interchanged with expectation for both the gradient and Hessian.
- (A2) **Uniform convergence.** $\sup_{\theta\in\Theta_{\mathrm{local}}} |\mathcal{L}_N(\theta) - \mathcal{L}(\theta)| \xrightarrow{p} 0$, $\sup_{\theta\in\Theta_{\mathrm{local}}} \big\|\nabla\mathcal{L}_N(\theta) - \nabla\mathcal{L}(\theta)\big\|_2 \xrightarrow{p} 0$ and $\sup_{\theta\in\Theta_{\mathrm{local}}} \big\|\nabla^2\mathcal{L}_N(\theta) - \nabla^2\mathcal{L}(\theta)\big\|_{\mathrm{op}} \xrightarrow{p} 0$

(A3) **Uniform stochastic equicontinuity and gradient concentration.** In addition to uniform convergence of the Hessian, assume

$$\|\nabla L_N(\theta^\star) - \nabla L(\theta^\star)\|_2 = O_p\left(\frac{1}{\sqrt{N}}\right) \tag{24}$$

(A4) **Local curvature.** Under the locally identifiable parameterization above, $\theta^\star$ is the unique minimizer of L on Θ_{local}, and $\nabla^2 L(\theta^\star)$ is positive definite. In particular, there exists $\lambda > 0$ such that

$$\lambda_{\min}\big(\nabla^2 L(\theta)\big) \geq \lambda \qquad \forall \theta \in \Theta_{\text{local}} \tag{25}$$

Hence, for all sufficiently large N, $\nabla^2 L_N(\theta)$ is invertible on Θ_{local} with

$$\sup_{\theta \in \Theta_{\text{local}}} \|\nabla^2 L_N(\theta)^{-1}\|_{\text{op}} = O_p(1) \tag{26}$$

(A5) **Approximate stationarity.** The optimization output satisfies $\|\nabla \mathcal{L}_N(\hat{\theta}_N)\|_2 \leq \varepsilon_N$ with $\varepsilon_N \to 0$ in probability, and $\hat{\theta}_N \in \Theta_{\text{local}}$ w.p. $\to 1$.

(A6) **Blockwise Schur dominance (weak coupling).** For each codeword block $\boldsymbol{\pi}_k$, let S_k denote the population Schur complement

$$S_k = \nabla^2_{\boldsymbol{\pi}_k \boldsymbol{\pi}_k} \mathcal{L}(\theta^\star) - \nabla^2_{\boldsymbol{\pi}_k, -k} \mathcal{L}(\theta^\star) \big[\nabla^2_{-k,-k} \mathcal{L}(\theta^\star)\big]^{-1} \nabla^2_{-k, \boldsymbol{\pi}_k} \mathcal{L}(\theta^\star) \tag{27}$$

Assume there exists $c \in (0, 1]$ such that for all k,

$$S_k \succeq c\, \nabla^2_{\boldsymbol{\pi}_k \boldsymbol{\pi}_k} \mathcal{L}(\theta^\star) \tag{28}$$

It guarantees that the effective curvature along $\boldsymbol{\pi}_k$ remains non-degenerate even after accounting for interactions with other parameters.

(A7) **Bounded conditional cross-block influence.** Let $\bar{H}_{k,-k} = \mathbb{E}\Big[H^{(k)}_{n,i,-k}(\theta^\star) \;\Big|\; s_n(i) = k\Big]$ be the expected Hessian cross-terms conditional on codeword appearance. We assume there exists $M \in \mathbb{R}_{>0}$ such that:

$$\sup_k \left\| \bar{H}_{k,-k} \big[\nabla^2_{-k,-k} \mathcal{L}(\theta^\star)\big]^{-1} \right\|_{\text{op}} < M < \infty \tag{29}$$

This ensures that while codewords may interact, the strength of cross-block curvature contributed by each occurrence of a codeword is uniformly bounded relative to the global curvature of the remaining parameters.

(A8) **Moment and dependence conditions.** For each block k, the per-occurrence gradient contributions $\{g^{(k)}_{n,i}(\theta^\star) : (n,i) \in \mathcal{I}_k\}$ are mean-zero and satisfy $\sup_k \mathbb{E}\|g^{(k)}_{n,i}(\theta^\star)\|_2^2 < \infty$. Similarly, let $H^{(k)}_{n,i}(\theta^\star)$ denote the per-occurrence Hessian contribution to the $(\boldsymbol{\pi}_k, \boldsymbol{\pi}_k)$ block and let $H^{(k)}_{n,i,-k}(\theta^\star)$ denote the corresponding per-occurrence contribution to the $(\boldsymbol{\pi}_k, \theta_{-k})$ cross-block. Assume the uniform moment bounds $\sup_k \mathbb{E}\|H^{(k)}_{n,i}(\theta^\star)\|_{\text{op}} < \infty$ and $\sup_k \mathbb{E}\|H^{(k)}_{n,i,-k}(\theta^\star)\|_{\text{op}} < \infty$. Conditional on the codeword assignment, with the quantization rule held fixed during calibration and in the asymptotic analysis, the empirical averages of these per-occurrence contributions concentrate around their conditional expectations (*i.e.*, a conditional LLN with $\sqrt{N_k}$-rate concentration holds).

(A9) **Local Hessian Lipschitzness.** There exists $L_H < \infty$ such that, with probability tending to one,

$$\sup_{\theta, \theta' \in \Theta_{\text{local}}} \frac{\|\nabla^2 L_N(\theta) - \nabla^2 L_N(\theta')\|_{\text{op}}}{\|\theta - \theta'\|_2} \leq L_H \tag{30}$$

A.4 Proof of Theorem 1

Statement: Let $\mathcal{L}_N(\theta)$ be the empirical cross-entropy loss and Θ_{local} be a compact convex neighborhood of a population local minimizer $\theta^\star$ (the pseudo-true parameter). Let $\hat{\theta}_N \in \Theta_{\text{local}}$ be an

approximate stationary point satisfying $\|\nabla \mathcal{L}_N(\hat{\theta}_N)\|_2 \leq \varepsilon_N$, where $\varepsilon_N \xrightarrow{p} 0$. Under the assumptions (A1)-(A5), the estimator converges in probability to the population minimizer:

$$\|\hat{\theta}_N - \theta^\star\|_2 = O_p\left(\frac{1}{\sqrt{N}}\right) + O_p(\varepsilon_N) \tag{31}$$

In particular, if $\varepsilon_N = o_p(1)$, then $\hat{\theta}_N \xrightarrow{p} \theta^\star$.

Proof. In the following proof, $\|\cdot\|_2$ denotes the Euclidean norm for vectors, and $\|\mathbf{A}\|_{\mathrm{op}} = \sup_{\|x\|_2=1} \|\mathbf{A}x\|_2$ denotes the induced operator (spectral) norm for matrices. Derivatives are taken with respect to the free local coordinates of the identifiable parameterization introduced in Appendix A.3. Let

$$\Delta_N = \hat{\theta}_N - \theta^\star \tag{32}$$

By the fundamental theorem of calculus,

$$\nabla L_N(\hat{\theta}_N) - \nabla L_N(\theta^\star) = \bar{H}_N \Delta_N, \tag{33}$$

where

$$\bar{H}_N = \int_0^1 \nabla^2 L_N(\theta^\star + t\Delta_N)\, dt \tag{34}$$

Hence,

$$\Delta_N = \bar{H}_N^{-1}\left(\nabla L_N(\hat{\theta}_N) - \nabla L_N(\theta^\star)\right), \tag{35}$$

so that

$$\Delta_N = -\bar{H}_N^{-1}\nabla L_N(\theta^\star) + \bar{H}_N^{-1}\nabla L_N(\hat{\theta}_N) \tag{36}$$

Since $\theta^\star$ is an interior population minimizer, $\nabla L(\theta^\star) = 0$. Hence, by (A3),

$$\|\nabla L_N(\theta^\star)\|_2 = \|\nabla L_N(\theta^\star) - \nabla L(\theta^\star)\|_2 = O_p\left(\frac{1}{\sqrt{N}}\right) \tag{37}$$

By (A4) and uniform Hessian convergence (A2), all Hessians along the segment joining $\theta^\star$ and $\hat{\theta}_N$ are uniformly positive definite with probability tending to one. Hence,

$$\|\bar{H}_N^{-1}\|_{\mathrm{op}} = O_p(1) \tag{38}$$

Combining:

$$\|\Delta_N\|_2 \leq O_p(1) \cdot O_p\left(\frac{1}{\sqrt{N}}\right) + O_p(1) \cdot \varepsilon_N = O_p\left(\frac{1}{\sqrt{N}}\right) + O_p(\varepsilon_N) \tag{39}$$

if $\varepsilon_N = o_p(1)$, this proves $\hat{\theta}_N \to_p \theta^\star$.

To interpret $\theta^\star$, we write the population risk as the expected negative log-likelihood:

$$\mathcal{L}(\theta) = \mathbb{E}_{P_{\mathrm{true}}}[-\log p_\theta(\mathbf{y} \mid \mathbf{x})] \tag{40}$$

Then, for each $\mathbf{x}$, the conditional cross-entropy decomposes as

$$\mathbb{E}[-\log p_\theta(\mathbf{y} \mid \mathbf{x})] = H(P_{\mathrm{true}}(\cdot \mid \mathbf{x})) + D_{\mathrm{KL}}(P_{\mathrm{true}}(\cdot \mid \mathbf{x}) \,\|\, p_\theta(\cdot \mid \mathbf{x})), \tag{41}$$

where the entropy term H does not depend on θ. Taking expectation over $\mathbf{x}$ yields

$$\mathcal{L}(\theta) = \text{const} + \mathbb{E}_X[D_{\mathrm{KL}}(P_{\mathrm{true}}(\cdot \mid \mathbf{x}) \,\|\, p_\theta(\cdot \mid \mathbf{x}))] \tag{42}$$

Hence $\theta^\star$ minimizes the (conditional) Kullback-Leibler divergence within Θ_{local}, *i.e.*, it is the pseudo-true parameter [50].

□

A.5 Proof of Theorem 2

Statement: Assume (A1)-(A9), and condition on the global scale $\boldsymbol{\sigma}^2$, the codebook $\mathcal{C}$ and the assignments induced by quantization. Let $\boldsymbol{\pi}_k$ denote a codebook parameter block $\mathbf{a}_k \in \mathcal{A}$ or $\mathbf{b}_k \in \mathcal{B}$, and let $p_k = \sum_{i=1}^{w} \mathbb{P}(s(i) = k)$ denote the expected number of occurrences of codeword k per sample. For any active codeword with $p_k > 0$,

$$\|\hat{\boldsymbol{\pi}}_k - \boldsymbol{\pi}_k^\star\|_2 = O_p\left(\frac{1}{\sqrt{N p_k}}\right) + O_p\left(\frac{1}{\sqrt{N}}\right) + O_p\left(\frac{\varepsilon_N}{p_k}\right) \tag{43}$$

Since $N_k/N \xrightarrow{p} p_k$, the leading codeword-specific statistical term scales as $1/\sqrt{N_k}$.

Proof. In the following proof, $\|\cdot\|_2$ denotes the Euclidean norm for vectors, and $\|\mathbf{A}\|_{\mathrm{op}} = \sup_{\|x\|_2=1} \|\mathbf{A}x\|_2$ denotes the induced operator (spectral) norm for matrices. All parameter blocks, gradients, and Hessians are expressed in the free local coordinates of the identifiable parametrization.

Proof strategy. We bound the error of a fixed codeword block $\Delta_k = \hat{\pi}_k - \pi_k^\star$. We first linearize the empirical first-order condition around $\theta^\star$, and then isolate Δ_k through the Schur complement of the empirical Hessian. This expresses Δ_k as the sum of three contributions: a direct block-gradient term, a cross-block leakage term, and an optimization residual term. We then bound each contribution separately. In particular, codeword k appears N_k times, which yields a block-gradient of order $O_p(\sqrt{N_k}/N)$, while the inverse effective curvature scales as $O_p(1/p_k)$. Their product therefore gives the codeword-specific rate $O_p(1/\sqrt{N p_k})$. The cross-block contribution is shown to be $O_p(1/\sqrt{N})$, while approximate stationarity contributes $O_p(\varepsilon_N/p_k)$, up to lower-order terms. Combining these three bounds gives the stated result. More precisely, these are the steps required:

Step 1: Linearization around $\theta^\star$. Let

$$\Delta_N = \hat{\theta}_N - \theta^\star, \qquad H_N^\star = \nabla^2 L_N(\theta^\star) \tag{44}$$

and define

$$\bar{H}_N = \int_0^1 \nabla^2 L_N(\theta^\star + t\Delta_N)\, dt \tag{45}$$

By the fundamental theorem of calculus,

$$\nabla L_N(\hat{\theta}_N) - \nabla L_N(\theta^\star) = \bar{H}_N \Delta_N \tag{46}$$

Adding and subtracting $H_N^\star \Delta_N$ yields

$$H_N^\star \Delta_N = -\nabla L_N(\theta^\star) + r_N \tag{47}$$

where

$$r_N = \nabla L_N(\hat{\theta}_N) + \left(H_N^\star - \bar{H}_N\right) \Delta_N \tag{48}$$

By (A5),

$$\|\nabla L_N(\hat{\theta}_N)\|_2 \le \varepsilon_N \tag{49}$$

Moreover, by the triangle inequality for the operator norm and then Assumption (A9),

$$\|H_N^\star - \bar{H}_N\|_{\mathrm{op}} = \left\| \int_0^1 \left[\nabla^2 L_N(\theta^\star) - \nabla^2 L_N(\theta^\star + t\Delta_N)\right] dt \right\|_{\mathrm{op}} \tag{50}$$

$$\le \int_0^1 \left\|\nabla^2 L_N(\theta^\star) - \nabla^2 L_N(\theta^\star + t\Delta_N)\right\|_{\mathrm{op}} dt \tag{51}$$

$$\le \int_0^1 L_H \left\|\theta^\star - (\theta^\star + t\Delta_N)\right\|_2 dt \tag{52}$$

$$= \int_0^1 L_H\, t \|\Delta_N\|_2\, dt \tag{53}$$

$$= \frac{L_H}{2} \|\Delta_N\|_2 \tag{54}$$

Therefore,

$$\|(H_N^\star - \bar{H}_N)\Delta_N\|_2 \le \frac{L_H}{2}\|\Delta_N\|_2^2 \tag{55}$$

By Theorem 1,

$$\|\Delta_N\|_2 = O_p\left(\frac{1}{\sqrt{N}}\right) + O_p(\varepsilon_N) \tag{56}$$

and hence

$$\|r_N\|_2 \le \varepsilon_N + O_p\left(\left(\frac{1}{\sqrt{N}} + \varepsilon_N\right)^2\right) \tag{57}$$

Step 2: Codeword-frequency scaling. Consider the linearized system at $\theta^\star$ with optimization residual

$$\nabla^2\mathcal{L}_N(\theta^\star)\,\Delta_N = -\nabla\mathcal{L}_N(\theta^\star) + r_N, \tag{58}$$

and recall that

$$\Delta_N = \hat{\theta}_N - \theta^\star, \qquad \|r_N\|_2 \le \varepsilon_N + O_p\left(\left(\frac{1}{\sqrt{N}} + \varepsilon_N\right)^2\right)$$

To simplify notation, in this paragraph block Hessians H and gradients g are evaluated at $\theta^\star$ and, together with residuals, are empirical unless stated otherwise. Outside of this paragraph these quantities will instead feature the N subscript to signal empirical. Now partition $\theta^\star = (\boldsymbol{\pi}_k, \theta_{-k})$ and write the gradient, residual, and Hessian in block form:

$$\Delta_N = \begin{pmatrix}\Delta_k \\ \Delta_{-k}\end{pmatrix}, \quad \nabla\mathcal{L}_N(\theta^\star) = \begin{pmatrix}g_k \\ g_{-k}\end{pmatrix}, \quad r_N = \begin{pmatrix}r_k \\ r_{-k}\end{pmatrix}, \quad \nabla^2\mathcal{L}_N(\theta^\star) = \begin{pmatrix}H_{kk} & H_{k,-k} \\ H_{-k,k} & H_{-k,-k}\end{pmatrix} \tag{59}$$

where $g_k = \nabla_{\boldsymbol{\pi}_k}\mathcal{L}_N(\theta^\star)$ and $H_{kk} = \nabla^2_{\boldsymbol{\pi}_k\boldsymbol{\pi}_k}\mathcal{L}_N(\theta^\star)$. Equation (58) is equivalent to

$$H_{kk}\Delta_k + H_{k,-k}\Delta_{-k} = -g_k + r_k, \tag{60}$$

$$H_{-k,k}\Delta_k + H_{-k,-k}\Delta_{-k} = -g_{-k} + r_{-k} \tag{61}$$

Under (A4) and (A2), for large N the principal submatrix $H_{-k,-k}$ is also invertible w.p. $\to 1$ and we solve (61) for Δ_{-k}:

$$\Delta_{-k} = -H_{-k,-k}^{-1}g_{-k} + H_{-k,-k}^{-1}r_{-k} - H_{-k,-k}^{-1}H_{-k,k}\Delta_k \tag{62}$$

Substituting (62) into (60) yields

$$\Big(H_{kk} - H_{k,-k}H_{-k,-k}^{-1}H_{-k,k}\Big)\Delta_k = -g_k + H_{k,-k}H_{-k,-k}^{-1}g_{-k} + r_k - H_{k,-k}H_{-k,-k}^{-1}r_{-k} \tag{63}$$

Define the empirical Schur complement and leakage term

$$S_{k,N} = H_{kk} - H_{k,-k}H_{-k,-k}^{-1}H_{-k,k}, \qquad R_{k,N} = H_{k,-k}H_{-k,-k}^{-1}g_{-k} \tag{64}$$

Then

$$\Delta_k = -S_{k,N}^{-1}\Big(g_k - R_{k,N}\Big) + S_{k,N}^{-1}\Big(r_k - H_{k,-k}H_{-k,-k}^{-1}r_{-k}\Big) \tag{65}$$

Here, for large N, the empirical Schur complement $S_{k,N}$ is invertible with probability tending to one, since $S_{k,N} \to_p S_k$ by (A2) and S_k is positive definite by (A6).

Step 3: Express in terms of the Schur complement. Consider again the linearized system induced by the Hessian at $\theta^\star$:

$$\nabla^2\mathcal{L}_N(\theta^\star)\,\Delta_N = -\nabla\mathcal{L}_N(\theta^\star) + r_N, \qquad \|r_N\|_2 \le \varepsilon_N + O_p\Big((N^{-1/2} + \varepsilon_N)^2\Big) \tag{66}$$

Taking the $\boldsymbol{\pi}_k$-block and solving via the Schur complement gives (65):

$$\boldsymbol{\pi}_k - \boldsymbol{\pi}_k^\star = -S_{k,N}^{-1}\Big(\nabla_{\boldsymbol{\pi}_k}\mathcal{L}_N(\theta^\star) - R_{k,N}\Big) + S_{k,N}^{-1}\Big(r_k - H_{k,-k}H_{-k,-k}^{-1}r_{-k}\Big) \tag{67}$$

we take norms and apply the triangle inequality:

$$\|\boldsymbol{\pi}_k - \boldsymbol{\pi}_k^\star\|_2 \le \|S_{k,N}^{-1}\|_2\left(\|\nabla_{\boldsymbol{\pi}_k}\mathcal{L}_N(\theta^\star)\|_2 + \|R_{k,N}\|_2\right) + \|S_{k,N}^{-1}\|_2\left\|r_k - H_{k,-k}H_{-k,-k}^{-1}r_{-k}\right\|_2 \tag{68}$$

Step 4: Gradient scaling. Let $\mathcal{I}_k = \{(n,i) : s_n(i) = k\}$ denote the set of sample-slot pairs in which codeword k appears, with $|\mathcal{I}_k| = N_k$, where N is the number of samples and w the number of slots. The empirical gradient block with respect to $\boldsymbol{\pi}_k$ admits the decomposition

$$\nabla_{\boldsymbol{\pi}_k} \mathcal{L}_N(\theta^\star) = \frac{1}{N} \sum_{n=1}^{N} \sum_{i=1}^{w} \mathbf{1}\{s_n(i) = k\} g_{n,i}^{(k)}(\theta^\star) \tag{69}$$

By (A8) and a standard Chebyshev bound for empirical averages,

$$\|\nabla_{\boldsymbol{\pi}_k} \mathcal{L}_N(\theta^\star)\|_2 = O_p\left(\frac{\sqrt{N_k}}{N}\right) \tag{70}$$

Step 5: Hessian scaling. Define the pooled conditional Hessian contribution

$$\bar{H}_k = \frac{1}{p_k} \sum_{i=1}^{w} \mathbb{P}(s(i) = k)\, \mathbb{E}\Big[H_{n,i}^{(k)}(\theta^\star) \mid s(i) = k\Big] \tag{71}$$

where $p_k = \sum_{i=1}^{w} \mathbb{P}(s(i) = k)$. This definition allows the expected Hessian contribution of codeword k to vary across slots. The corresponding Hessian block satisfies

$$\nabla^2_{\boldsymbol{\pi}_k \boldsymbol{\pi}_k} \mathcal{L}_N(\theta^\star) = \frac{1}{N} \sum_{n=1}^{N} \sum_{i=1}^{w} \mathbf{1}\{s_n(i) = k\} H_{n,i}^{(k)}(\theta^\star) \tag{72}$$

By Assumption (A8) and the conditional law of large numbers (as $N \to \infty$), the empirical Hessian becomes:

$$\nabla^2_{\boldsymbol{\pi}_k \boldsymbol{\pi}_k} \mathcal{L}_N(\theta^\star) = \frac{N_k}{N}\Big(\bar{H}_k + o_p(1)\Big) \tag{73}$$

Since $\frac{N_k}{N} \xrightarrow{p} p_k$, for any active codeword k with $p_k > 0$ we have

$$\nabla^2_{\boldsymbol{\pi}_k \boldsymbol{\pi}_k} L_N(\theta^\star) = p_k(\bar{H}_k + o_p(1)) \tag{74}$$

The corresponding population Hessian block is therefore

$$\nabla^2_{\boldsymbol{\pi}_k \boldsymbol{\pi}_k} L(\theta^\star) = p_k \bar{H}_k \tag{75}$$

Since $\nabla^2 L(\theta^\star)$ is positive definite by Assumption (A4), $\bar{H}_k \succ 0$ for every active codeword. Hence

$$\left[\nabla^2_{\boldsymbol{\pi}_k \boldsymbol{\pi}_k} L_N(\theta^\star)\right]^{-1} = \frac{1}{p_k} \bar{H}_k^{-1} + o_p\left(\frac{1}{p_k}\right) \tag{76}$$

and therefore

$$\left\| \left[\nabla^2_{\boldsymbol{\pi}_k \boldsymbol{\pi}_k} L_N(\theta^\star)\right]^{-1} \right\|_{\mathrm{op}} = O_p\left(\frac{1}{p_k}\right) \tag{77}$$

Step 6: Controlling the Schur inverse. By block-wise identifiability the (population) Schur complement

$$S_k = \nabla^2_{\boldsymbol{\pi}_k \boldsymbol{\pi}_k} \mathcal{L}(\theta^\star) - \nabla^2_{\boldsymbol{\pi}_k, -k} \mathcal{L}(\theta^\star) \left[\nabla^2_{-k,-k} \mathcal{L}(\theta^\star)\right]^{-1} \nabla^2_{-k, \boldsymbol{\pi}_k} \mathcal{L}(\theta^\star) \tag{78}$$

satisfies

$$S_k \succeq c\, H_{kk}, \qquad H_{kk} = \nabla^2_{\boldsymbol{\pi}_k \boldsymbol{\pi}_k} \mathcal{L}(\theta^\star). \tag{79}$$

Then S_k is positive definite and:

$$\|S_k^{-1}\|_2 \le \frac{1}{c} \|H_{kk}^{-1}\|_2 \tag{80}$$

(A2) implies $S_{k,N} \to_p S_k$ and $H_{kk,N} \to_p H_{kk}$ in operator norm with probability tending to one. (A4) implies $H_{-k,-k}$ is invertible, hence $H_{kk,N}$ also is, in probability. Therefore, for N large enough, the inequality holds,

$$\|S_{k,N}^{-1}\|_2 \le \frac{2}{c} \|H_{kk,N}^{-1}\|_2, \qquad \text{and hence} \qquad \|S_{k,N}^{-1}\|_2 = O_p\Big(\|H_{kk,N}^{-1}\|_2\Big) \tag{81}$$

Using the result of Step 5 and Assumption (A6),

$$\|S_{k,N}^{-1}\|_{\mathrm{op}} = O_p\left(\frac{1}{p_k}\right) \tag{82}$$

Step 7: Controlling the gradient and leakage terms. By (70), the gradient contribution satisfies

$$\|\nabla_{\boldsymbol{\pi}_k}\mathcal{L}_N(\theta^\star)\|_2 = O_p\left(\frac{\sqrt{N_k}}{N}\right) \tag{83}$$

Since $N_k/N \xrightarrow{p} p_k$,

$$\|\nabla_{\boldsymbol{\pi}_k}L_N(\theta^\star)\|_2 = O_p\left(\sqrt{\frac{p_k}{N}}\right) \tag{84}$$

Combining this with $\|S_{k,N}^{-1}\|_{\mathrm{op}} = O_p(p_k^{-1})$ gives

$$\|S_{k,N}^{-1}\nabla_{\boldsymbol{\pi}_k}L_N(\theta^\star)\|_2 = O_p\left(\frac{1}{\sqrt{Np_k}}\right) \tag{85}$$

Since $N_k/N \xrightarrow{p} p_k$,

$$\|H_{k,-k,N}\|_{\mathrm{op}} = O_p(p_k) \tag{86}$$

By (A4),

$$\|H_{-k,-k,N}^{-1}\|_{\mathrm{op}} = O_p(1) \tag{87}$$

and, as before,

$$\|\nabla_{\theta_{-k}}L_N(\theta^\star)\|_2 = O_p\left(\frac{1}{\sqrt{N}}\right) \tag{88}$$

Hence

$$\|R_{k,N}\|_2 = O_p\left(\frac{p_k}{\sqrt{N}}\right) \tag{89}$$

Therefore,

$$\|S_{k,N}^{-1}R_{k,N}\|_2 = O_p\left(\frac{1}{\sqrt{N}}\right) \tag{90}$$

Step 8: Controlling the residuals. From Step 1 (Equation (57)),

$$\|r_N\|_2 \le \varepsilon_N + O_p\left(\left(\frac{1}{\sqrt{N}} + \varepsilon_N\right)^2\right) \tag{91}$$

Moreover by Equation (86) and Equation (87),

$$\|H_{k,-k,N}H_{-k,-k,N}^{-1}\|_{\mathrm{op}} \le \|H_{k,-k,N}\|_{\mathrm{op}}\|H_{-k,-k,N}^{-1}\|_{\mathrm{op}} = O_p(p_k)O_p(1) = O_p(p_k) \tag{92}$$

Then

$$\left\|r_{k,N} - H_{k,-k,N}H_{-k,-k,N}^{-1}r_{-k,N}\right\|_2 \le \|r_{k,N}\|_2 + \|H_{k,-k,N}H_{-k,-k,N}^{-1}\|_{\mathrm{op}}\|r_{-k,N}\|_2 \tag{93}$$

$$\le \|r_N\|_2 + O_p(p_k)\|r_N\|_2 \tag{94}$$

Since k and $p_k > 0$ are fixed and $1 + O_p(p_k) = O_p(1)$,

$$\left\|r_{k,N} - H_{k,-k,N}H_{-k,-k,N}^{-1}r_{-k,N}\right\|_2 = O_p(\|r_N\|_2) \tag{95}$$

Combining:

$$\left\|r_{k,N} - H_{k,-k,N}H_{-k,-k,N}^{-1}r_{-k,N}\right\|_2 = O_p(\varepsilon_N) + O_p\left(\left(\frac{1}{\sqrt{N}} + \varepsilon_N\right)^2\right) \tag{96}$$

Using $\|S_{k,N}^{-1}\|_{\mathrm{op}} = O_p(p_k^{-1})$,

$$\left\|S_{k,N}^{-1}\left(r_{k,N} - H_{k,-k,N}H_{-k,-k,N}^{-1}r_{-k,N}\right)\right\|_2 = O_p\left(\frac{\varepsilon_N}{p_k}\right) + O_p\left(\frac{\left(1/\sqrt{N} + \varepsilon_N\right)^2}{p_k}\right) \tag{97}$$

Because $p_k > 0$ is fixed pointwise in k and $\varepsilon_N = o_p(1)$,

$$O_p\left(\frac{(N^{-1/2}+\varepsilon_N)^2}{p_k}\right) = O_p\left(\frac{1}{\sqrt{N}}\right) + o_p\left(\frac{\varepsilon_N}{p_k}\right) \tag{98}$$

and can therefore be absorbed into the leading terms.

Step 9: Combining the bounds. From the Schur-complement representation in (68) the estimation error decomposes into the intrinsic block-gradient term, the cross-block leakage term, and the optimization residual term.

By Steps 6-8,

$$\left\|S_{k,N}^{-1}\nabla_{\boldsymbol{\pi}_k} L_N(\theta^\star)\right\|_2 = O_p\left(\frac{1}{\sqrt{Np_k}}\right), \qquad \left\|S_{k,N}^{-1}R_{k,N}\right\|_2 = O_p\left(\frac{1}{\sqrt{N}}\right) \tag{99}$$

and

$$\left\|S_{k,N}^{-1}\left(r_{k,N} - H_{k,-k,N}H_{-k,-k,N}^{-1}r_{-k,N}\right)\right\|_2 = O_p\left(\frac{\varepsilon_N}{p_k}\right) + o_p\left(\frac{1}{\sqrt{N}} + \frac{\varepsilon_N}{p_k}\right) \tag{100}$$

Therefore, for every fixed active codeword k with $p_k > 0$,

$$\|\hat{\boldsymbol{\pi}}_k - \boldsymbol{\pi}_k^\star\|_2 = O_p\left(\frac{1}{\sqrt{Np_k}}\right) + O_p\left(\frac{1}{\sqrt{N}}\right) + O_p\left(\frac{\varepsilon_N}{p_k}\right) \tag{101}$$

Finally, since

$$\frac{N_k}{N} \xrightarrow{p} p_k, \tag{102}$$

we have

$$\frac{1}{\sqrt{(Np_k)}} = \frac{1+o_p(1)}{\sqrt{N_k}} \tag{103}$$

so the leading codeword-specific statistical term scales with the effective number of occurrences of codeword k. This concludes the proof. □

A.6 Proof of Corollary 1

Statement: Under the setting of Theorem 2, further assume that the link function ϕ is positive, monotone and 1-Lipschitz, *e.g.*, the softplus. Since the codebook is fixed and finite, for every fixed cell $\mathcal{V}$ with index sequence $\mathbf{s}$ of active codewords, the Frobenius norm of the estimation error in the local Dirichlet concentrations satisfies:

$$\left\|\hat{\boldsymbol{\alpha}}^{(\mathcal{V})} - \boldsymbol{\alpha}^{(\mathcal{V})\star}\right\|_F = O_p\left(\sum_{i=1}^{w}\sigma_i^2\left[\frac{1}{\sqrt{Np_{s(i)}}} + \frac{1}{\sqrt{N}} + \frac{\varepsilon_N}{p_{s(i)}}\right]\right) \tag{104}$$

Proof. Throughout this proof, $\|\cdot\|_F$ denotes the Frobenius norm. For a cell $\mathcal{V}$ with index sequence $\mathbf{s}$, define the pre-link matrix

$$\mathbf{M}^{(\mathcal{V})} = \sum_{i=1}^{w}\sigma_i^2\,\mathbf{a}_{s(i)}\mathbf{b}_{s(i)}^\top, \tag{105}$$

so that

$$\boldsymbol{\alpha}^{(\mathcal{V})} = \phi\left(\mathbf{M}^{(\mathcal{V})}\right) + \mathbf{I} \tag{106}$$

Analogously, let $\hat{\mathbf{M}}^{(\mathcal{V})}$ denote the corresponding matrix obtained from the estimated codeword parameters. Since ϕ is applied element-wise and is 1-Lipschitz, it holds that

$$\left\|\hat{\boldsymbol{\alpha}}^{(\mathcal{V})} - \boldsymbol{\alpha}^{(\mathcal{V})\star}\right\|_F \le \left\|\hat{\mathbf{M}}^{(\mathcal{V})} - \mathbf{M}^{(\mathcal{V})\star}\right\|_F \tag{107}$$

Moreover, by definition,

$$\hat{\mathbf{M}}^{(\mathcal{V})} - \mathbf{M}^{(\mathcal{V})\star} = \sum_{i=1}^{w}\sigma_i^2\left(\hat{\mathbf{a}}_{s(i)}\hat{\mathbf{b}}_{s(i)}^\top - \mathbf{a}_{s(i)}^\star\mathbf{b}_{s(i)}^{\star\top}\right) \tag{108}$$

Let us denote for each active codeword k the difference of parameters with respect to the optimal values as $\Delta\mathbf{a}_k = \hat{\mathbf{a}}_k - \mathbf{a}_k^\star$ and $\Delta\mathbf{b}_k = \hat{\mathbf{b}}_k - \mathbf{b}_k^\star$, for $\mathbf{a}$ and $\mathbf{b}$ respectively. Therefore, we have that

$$\hat{\mathbf{a}}_k\hat{\mathbf{b}}_k^\top - \mathbf{a}_k^\star\mathbf{b}_k^{\star\top} = \Delta\mathbf{a}_k\mathbf{b}_k^{\star\top} + \mathbf{a}_k^\star\Delta\mathbf{b}_k^\top + \Delta\mathbf{a}_k\Delta\mathbf{b}_k^\top \tag{109}$$

Note that for any vectors $\mathbf{u}, \mathbf{v} \in \mathbb{R}^d$, with $d \geq 1$, the following holds: $\|\mathbf{u}\mathbf{v}^\top\|_F = \|\mathbf{u}\|_2\|\mathbf{v}\|_2$. Therefore, by applying the triangle inequality, we have that:

$$\left\|\hat{\boldsymbol{\alpha}}^{(\mathcal{V})} - \boldsymbol{\alpha}^{(\mathcal{V})\star}\right\|_F \leq \sum_{i=1}^{w} \sigma_i^2 \Big(\|\Delta\mathbf{a}_{s(i)}\|_2\|\mathbf{b}_{s(i)}^\star\|_2 + \|\mathbf{a}_{s(i)}^\star\|_2\|\Delta\mathbf{b}_{s(i)}\|_2 + \|\Delta\mathbf{a}_{s(i)}\|_2\|\Delta\mathbf{b}_{s(i)}\|_2 \Big) \tag{110}$$

Now, let us define

$$\rho_k = \frac{1}{\sqrt{Np_k}} + \frac{1}{\sqrt{N}} + \frac{\varepsilon_N}{p_k} \tag{111}$$

Since the codebook is fixed and finite, there exists a constant $C < \infty$ such that

$$\|\mathbf{a}_k^\star\|_2 \leq C, \qquad \|\mathbf{b}_k^\star\|_2 \leq C \tag{112}$$

for every active codeword k. Moreover, by Theorem 2,

$$\|\Delta\mathbf{a}_k\|_2 = O_p(\rho_k), \qquad \|\Delta\mathbf{b}_k\|_2 = O_p(\rho_k) \tag{113}$$

Therefore, for each slot i,

$$\|\Delta\mathbf{a}_{s(i)}\|_2\|\mathbf{b}_{s(i)}^\star\|_2 = O_p(\rho_{s(i)}), \qquad \|\mathbf{a}_{s(i)}^\star\|_2\|\Delta\mathbf{b}_{s(i)}\|_2 = O_p(\rho_{s(i)}) \tag{114}$$

while

$$\|\Delta\mathbf{a}_{s(i)}\|_2\|\Delta\mathbf{b}_{s(i)}\|_2 = O_p(\rho_{s(i)}^2) \tag{115}$$

Substituting these bounds into the previous inequality and using that w is fixed yields

$$\left\|\hat{\boldsymbol{\alpha}}^{(\mathcal{V})} - \boldsymbol{\alpha}^{(\mathcal{V})\star}\right\|_F = O_p\left(\sum_{i=1}^{w} \sigma_i^2 \big(\rho_{s(i)} + \rho_{s(i)}^2\big) \right) \tag{116}$$

For every active codeword, $p_k > 0$ is fixed and $\varepsilon_N = o_p(1)$, so that $\rho_k = o_p(1)$. Therefore $\rho_k^2 = o_p(\rho_k)$, and the quadratic term can be absorbed into the leading rate. Thus,

$$\left\|\hat{\boldsymbol{\alpha}}^{(\mathcal{V})} - \boldsymbol{\alpha}^{(\mathcal{V})\star}\right\|_F = O_p\left(\sum_{i=1}^{w} \sigma_i^2 \left[\frac{1}{\sqrt{Np_{s(i)}}} + \frac{1}{\sqrt{N}} + \frac{\varepsilon_N}{p_{s(i)}} \right] \right) \tag{117}$$

which proves the claim. □

B Metrics

We consider both *global* and *local* metrics to assess the calibration of a classifier.

Global metrics. Expected Calibration Error (ECE) [12] is a standard metric for binary calibration, measuring the discrepancy between predicted confidence ($\text{conf}(\cdot)$) and empirical accuracy ($\text{acc}(\cdot)$) across confidence bins:

$$ECE = \sum_{b=1}^{m_b} \frac{|B_b|}{n} \left|\text{acc}(B_b) - \text{conf}(B_b)\right| \tag{118}$$

where B_b is the set of instances in the b-th bin and m_b the number of bins. In multiclass settings, this idea extends to Class-wise ECE [7], which averages per-class ECEs:

$$\textit{Class-wise } ECE = \frac{1}{|\mathcal{Y}|} \sum_{c=1}^{|\mathcal{Y}|} ECE_c \tag{119}$$

Following Ibarra et al. [13], we construct cumulative calibration residuals. In our experiments, we use a binned variant that averages the absolute cumulative residuals across bins. More precisely, we summarize residuals using their empirically weighted average absolute magnitude:

$$\text{ECCE}_c = \sum_{b=1}^{m_b} \frac{|B_b|}{n} \left| \sum_{i=1}^{b} \frac{|B_{i,c}|}{n} \left(\text{freq}_c(B_i) - \text{conf}_c(B_i)\right) \right| \tag{120}$$

where $\mathrm{freq}_c(B_i)$ and $\mathrm{conf}_c(B_i)$ denote, respectively, the empirical frequency and the average predicted probability of class c within bin B_i.

Local metrics. Beyond traditional binning-based metrics, recent work has proposed consistent estimators of canonical multiclass calibration error [51] and proper calibration errors associated with proper scoring rules [52]. In this work, we consider the multiclass extension of *Local Calibration Error* [6] proposed by Barbera et al. [11]:

$$LCE = \frac{1}{|\mathcal{Y}|} \sum_{b=1}^{m_b} \frac{1}{n} \sum_{i \in B_b} \left\| \frac{\sum_{j \in B_b} \left(\hat{\mathbf{p}}_j - \mathbf{y}_j\right) k_\gamma(\mathbf{x}_i, \mathbf{x}_j)}{\sum_{j \in B_b} k_\gamma(\mathbf{x}_i, \mathbf{x}_j)} \right\|_1 \tag{121}$$

where $\|\cdot\|_1$ is the ℓ^1 norm and $k_\gamma(\mathbf{x}_i, \mathbf{x}_j)$ is a kernel function that weights the influence of neighboring points of the anchor $\mathbf{x}_i$ on its individual LCE score. In practice, this metric captures the differences in the predicted probabilities and the corresponding ground truths for neighbors of an anchor point $\mathbf{x}_i$.

Moreover, we also consider the *Maximum Local Calibration Error*, defined as the worst-case per-anchor LCE residual:

$$MLCE = \max_{\mathbf{x}_i \in D} \left\| \frac{\sum_{j \in B_{b(i)}} \left(\hat{\mathbf{p}}_j - \mathbf{y}_j\right) k_\gamma(\mathbf{x}_i, \mathbf{x}_j)}{\sum_{j \in B_{b(i)}} k_\gamma(\mathbf{x}_i, \mathbf{x}_j)} \right\|_1 \tag{122}$$

Intuitively, this metric captures the largest local calibration error the ML model can make, making it particularly relevant in high-stakes settings.

C Additional Experiments

C.1 Tabular data

Setup. We evaluate our method on the `Weather` dataset, a tabular benchmark for multi-class classification. For tabular data, we leverage the same data-splitting criterion of image data *i.e.*, (a) a training set ($\approx 100k$ samples); (b) we use the development set as a calibration set ($\approx 40k$ samples) with an internal $90\%/10\%$ calibration/validation split; and (c) a held-out test set used exclusively for evaluation ($\approx 100k$ samples). Due to the extreme class imbalance we restrict the problem to 5 classes which present a similar imbalance to `tissue`. As the underlying predictor, we use an FT-Transformer (`FTT`) model [53], a state-of-the-art neural network architecture for tabular data.

Calibration and performance. We report results for `Weather` data in Table 1. VQ achieves the best performance in terms of local calibration, obtaining the lowest LCE (.0053) and significantly improving over all baselines. While VQ does not achieve the best $MLCE$, it remains competitive and substantially improves over global methods, which exhibit much higher worst-case local errors. Regarding global calibration, VQ attains competitive $ECCE$ (.0010), slightly worse than the best-performing global methods (SM and DC), but still within the same order of magnitude. Finally, VQ maintains comparable predictive performance ($ACC \approx .603$) to all baselines.

Under data sparsity. For `Weather` data, we report in Figure 5 local calibration error as a function of neighborhood density, measured via ESS. As expected, all methods exhibit a clear trend: calibration error is highest in low-density regions (low ESS) and decreases as the local support increases. However, VQ consistently achieves the lowest LCE across the entire ESS spectrum, with the most pronounced improvements in the lowest-density bins. In particular, in the sparsest regions (leftmost bins), VQ outperforms all baselines with local methods like LN and KC closely following. As density increases, the gap narrows, and all methods converge to similar performance, indicating that calibration is easier when sufficient local data is available. These results highlight that VQ is particularly effective in mitigating calibration errors in low-support regions, where traditional methods struggle, while remaining competitive in high-density regimes.

Out of Distribution. The `Weather` dataset naturally supports out-of-distribution (OOD) evaluation, as it provides a dedicated test split containing data from different climatic regions. To assess

Table 1: Results on the `Weather` dataset with an `FT-Transformer` classifier.

Data	Method	$LCE \downarrow$	$MLCE \downarrow$	$ECCE \downarrow$	$ECE \downarrow$	$ACC \uparrow$	$NLL \downarrow$
weather	VQ	**.0053 ± .0007**	.1782 ± .0344	.0010 ± .0004	.0030 ± .0007	.603 ± .001	.943 ± .003
	DC	.0221 ± .0018	.4509 ± .0102	.0007 ± .0001	.0022 ± .0001	.607 ± .002	.933 ± .003
	KC	.0101 ± .0007	**.0998 ± .0208**	.0028 ± .0003	.0091 ± .0005	.606 ± .002	.941 ± .002
	LN	.0107 ± .0014	.1021 ± .0138	.0028 ± .0003	.0100 ± .0015	.605 ± .001	.945 ± .004
	SM	.0221 ± .0018	.4495 ± .0105	**.0005 ± .0001**	**.0015 ± .0002**	**.607 ± .001**	**.933 ± .003**
	PS	.0242 ± .0018	.4379 ± .0104	.0009 ± .0000	.0068 ± .0003	.607 ± .001	.948 ± .003
	IR	.0223 ± .0018	.4505 ± .0109	.0006 ± .0001	.0020 ± .0001	.606 ± .001	.936 ± .003
	TS	.0234 ± .0018	.4429 ± .0145	.0019 ± .0004	.0045 ± .0005	.607 ± .001	.936 ± .003
	PC	.0227 ± .0018	.4471 ± .0165	.0013 ± .0004	.0029 ± .0002	.606 ± .001	.935 ± .003
	NC	.0226 ± .0017	.4528 ± .0108	.0019 ± .0005	.0047 ± .0004	.607 ± .001	.936 ± .003

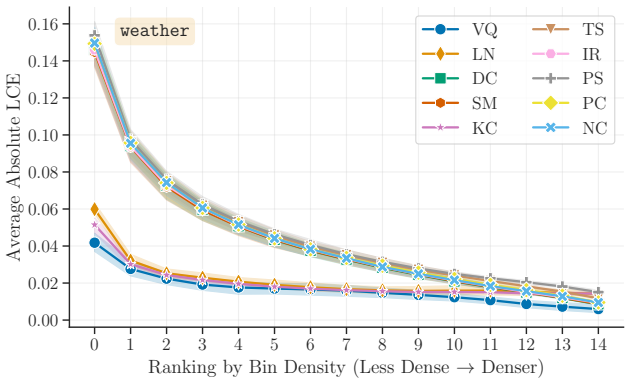


Figure 5: Local calibration in density-based sub-bins over five runs on `Weather` (FTT backbone).

robustness, we train the base model on in-distribution data and perform calibration using only in-distribution samples for all methods, then evaluate performance on the OOD split. This setting reflects a distribution shift at test time, while preserving the standard calibration protocol.

As expected, calibration performance degrades under distribution shift for all methods. Non-parametric local methods such as KC or hybrid methods like LN, which rely on kernel-based or neighborhood adaptations, achieve the best results in this setting, benefiting from their ability to adapt more flexibly to local test-time structure. VQ remains competitive, outperforming all global calibration methods in terms of both LCE and $MLCE$, but does not match the performance of its local competitors.

Previous work has investigated confidence calibration under distribution shift, including domain generalization approaches that leverage multiple source domains to improve calibration on unseen target distributions [54]. While addressing a different setting, we complement this line of research by evaluating the robustness of our local calibration approach under distribution shifts induced by image corruptions. We emphasize that calibration approaches are designed for in-distribution settings, where the calibration and test distributions coincide. Therefore, we argue OOD experiments should be interpreted as a robustness check rather than a core requirement. Still, we observe that VQ provides a favourable trade-off, maintaining strong performance under distribution shift while retaining its advantages in the standard in-distribution setting. More importantly, the method shows consistent improvements in local calibration across the main in-distribution experiments, ablations, and additional architectures.

C.2 Ablating the role of quantization and calibration

We consider two main ablations of VQ: (i) VQ-NC only employs vector quantization with no calibration procedure afterwards; (ii) VQ-DC does not use VQ's bilinear factorization, but employs the standard Dirichlet calibration on the quantization head. For completeness, we also include NC and DC in the comparison. Table 3 shows the results.

Table 2: VQ results on OOD `Weather` data with an `FT-Transformer` classifier.

Data	Method	$LCE \downarrow$	$MLCE \downarrow$	$ECCE \downarrow$	$ECE \downarrow$	$ACC \uparrow$	$NLL \downarrow$
weather	VQ	.0144 ± .0017	.3388 ± .0327	.0038 ± .0003	.0122 ± .0020	.474 ± .005	1.175 ± .011
	LN	**.0089 ± .0005**	**.2029 ± .0699**	**.0027 ± .0005**	**.0059 ± .0004**	.478 ± .005	1.162 ± .008
	KC	.0108 ± .0004	.2829 ± .0292	.0028 ± .0004	.0075 ± .0007	.477 ± .004	**1.159 ± .007**
	SM	.0252 ± .0018	.4886 ± .0047	.0034 ± .0008	.0094 ± .0011	**.481 ± .005**	1.164 ± .011
	DC	.0261 ± .0018	.4891 ± .0047	.0038 ± .0013	.0108 ± .0011	.480 ± .005	1.167 ± .011
	PS	.0283 ± .0022	.4742 ± .0043	.0047 ± .0011	.0142 ± .0012	.480 ± .006	1.175 ± .012
	TS	.0258 ± .0023	.4858 ± .0048	.0044 ± .0021	.0087 ± .0023	.479 ± .006	1.163 ± .011
	IR	.0263 ± .0024	.4898 ± .0053	.0038 ± .0017	.0105 ± .0017	.479 ± .006	1.168 ± .012
	PC	.0266 ± .0024	.4879 ± .0038	.0040 ± .0015	.0111 ± .0017	.478 ± .006	1.169 ± .012
	NC	.0285 ± .0023	.4898 ± .0042	.0050 ± .0017	.0135 ± .0017	.478 ± .006	1.174 ± .012

Table 3: Ablation results for VQ (highlighted in blue) when evaluating variants of our pipeline components.

Data	Method	$LCE \downarrow$	$MLCE \downarrow$	$ECCE \downarrow$	$ACC \uparrow$	$NLL \downarrow$
cifar10	VQ	**.0059 ± .0002**	.5595 ± .0276	.0013 ± .0002	.889 ± .001	.350 ± .001
	VQ-DC	.0062 ± .0003	**.5534 ± .0311**	.0014 ± .0004	.889 ± .001	.350 ± .001
	VQ-NC	.0065 ± .0002	.5586 ± .0253	.0022 ± .0003	.889 ± .001	.351 ± .001
	DC	.0099 ± .0003	.8127 ± .0194	**.0011 ± .0001**	**.890 ± .002**	**.332 ± .007**
	NC	.0154 ± .0005	.8729 ± .0235	.0066 ± .0004	.884 ± .001	.494 ± .0018
cifar100	VQ	**.0017 ± .0001**	**.2932 ± .0099**	.0008 ± .0001	.681 ± .001	1.237 ± .010
	VQ-DC	.0020 ± .0001	.3050 ± .0110	.0012 ± .0001	.674 ± .004	1.210 ± .005
	VQ-NC	.0021 ± .0001	.3199 ± .0074	.0013 ± .0001	.681 ± .001	1.227 ± .007
	DC	.0027 ± .0001	.8113 ± .0121	**.0007 ± .0001**	**.690 ± .002**	**1.154 ± .010**
	NC	.0032 ± .0001	.9155 ± .0104	.0017 ± .0001	.670 ± .003	1.502 ± .036
tissue	VQ	.0088 ± .0004	**.5760 ± .0349**	**.0017 ± .0004**	.624 ± .002	**1.028 ± .007**
	VQ-DC	**.0087 ± .0005**	.6631 ± .0401	.0023 ± .0005	.624 ± .001	1.029 ± .007
	VQ-NC	.0112 ± .0016	.6557 ± .0649	.0049 ± .0017	**.625 ± .001**	1.033 ± .009
	DC	.0284 ± .0013	.9613 ± .0097	.0065 ± .0009	.615 ± .004	1.079 ± .012
	NC	.0739 ± .0018	.9741 ± .0086	.0415 ± .0029	.603 ± .008	2.100 ± .010
weather	VQ	**.0053 ± .0007**	.1782 ± .0344	.0010 ± .0004	.603 ± .001	.943 ± .003
	VQ-DC	.0056 ± .0006	.1806 ± .0203	.0012 ± .0004	.603 ± .002	.943 ± .003
	VQ-NC	.0084 ± .0016	**.1736 ± .0156**	.0039 ± .0015	.601 ± .001	.948 ± .003
	DC	.0221 ± .0018	.4509 ± .0102	**.0007 ± .0001**	.607 ± .002	**.933 ± .003**
	NC	.0226 ± .0017	.4528 ± .0108	.0019 ± .0005	**.607 ± .001**	.936 ± .003

For local metrics, we see that vector quantization is the main driver of improvements: all the ablated versions of our approach outperform both NC and DC, with impressive gains in terms of $MLCE$ (*e.g.*, for `Cifar100` we pass from $\approx .8248 \pm .0161$ of DC to $\approx .2932 \pm .0099$ of VQ). Still, the choice of parametrization matters: while we do not always observe statistically significant differences between VQ and VQ-DC, there are gains over VQ-NC.

When looking at $ECCE$, we can see that VQ is better compared to both VQ-DC and VQ-NC, suggesting that our parametrization offers advantages in terms of global calibration. This is expected, as the bilinear parametrization regularizes calibration maps across regions. More precisely, it constrains regional differences to lie in a shared, low-dimensional space of miscalibration factors, allowing region-specific behavior while preserving global coherence. Finally, regarding NLL, VQ achieves competitive values across datasets, with only minor gaps to globally optimized methods such as DC. Not surprisingly, the main driver of NLL gains is the quantization step as the new learned classification-head optimizes this quantity explicitly.

C.3 Ablating the number of slots and codebook size

We study the sensitivity of VQ to the number of slots w and the codebook size $|\mathcal{C}|$. Results are reported in Table 4 by varying $w \in \{16, 32, 64 \text{ (default)}, 128, 256\}$ for $|\mathcal{C}| \in \{32, 64\}$, and in Table 5 by varying $|\mathcal{C}|$ while fixing $w = 64$ (the best-performing value).

Across datasets, performance exhibits a clear U-shape trend for w: small values (*e.g.*, $w = 16$) provide too coarse a partition of the representation space and limit the benefit of locality, while very large values (*e.g.*, $w \geq 128$) make the region assignment and the resulting local estimates less stable. Consistently, the strongest local calibration performance is achieved for intermediate values, with

Table 4: VQ results when varying the number of slots (w) and codebook size ($|\mathcal{C}|$). Blue row reports main-paper parameters.

Data	w	$\lvert\mathcal{C}\rvert$	$LCE \downarrow$	$MLCE \downarrow$	$ECCE \downarrow$	$ECE \downarrow$	$ACC \uparrow$	$NLL \downarrow$
cifar10	16	32	.0104 ± .0005	.8247 ± .0058	.0012 ± .0002	.0044 ± .0004	.878 ± .001	.412 ± .005
	16	64	.0091 ± .0002	.8040 ± .0141	.0011 ± .0002	.0044 ± .0003	.879 ± .002	.422 ± .004
	32	32	.0081 ± .0002	.6852 ± .0327	**.0010 ± .0002**	.0039 ± .0002	.885 ± .001	.374 ± .002
	32	64	.0076 ± .0003	.6365 ± .0276	.0013 ± .0005	.0043 ± .0003	.884 ± .001	.380 ± .004
	64	32	.0065 ± .0002	.5665 ± .0163	.0012 ± .0001	.0038 ± .0002	.889 ± .001	.352 ± .001
	64	64	**.0059 ± .0002**	**.5595 ± .0276**	.0013 ± .0002	.0037 ± .0002	.889 ± .001	.348 ± .002
	128	32	.0063 ± .0002	.6365 ± .0363	.0012 ± .0001	.0039 ± .0003	.890 ± .002	.341 ± .003
	128	64	.0066 ± .0001	.6552 ± .0317	.0014 ± .0004	.0039 ± .0004	.889 ± .001	.341 ± .003
	256	32	.0075 ± .0001	.7306 ± .0241	.0011 ± .0001	**.0036 ± .0002**	**.892 ± .001**	**.333 ± .003**
	256	64	.0082 ± .0003	.7290 ± .0298	.0016 ± .0003	.0040 ± .0004	.891 ± .001	.334 ± .002
cifar100	16	32	.0032 ± .0002	.7790 ± .0337	.0007 ± .0001	.0017 ± .0001	.638 ± .003	1.428 ± .020
	16	64	.0023 ± .0001	.6163 ± .0245	.0008 ± .0001	.0018 ± .0001	.645 ± .002	1.395 ± .006
	32	32	.0023 ± .0005	.6206 ± .1806	.0008 ± .0001	.0017 ± .0001	.666 ± .002	1.289 ± .006
	32	64	.0017 ± .0002	**.2758 ± .0104**	.0009 ± .0001	.0018 ± .0001	.670 ± .001	1.276 ± .007
	64	32	**.0016 ± .0001**	.3274 ± .0133	.0007 ± .0001	.0017 ± .0001	.680 ± .003	1.213 ± .004
	64	64	.0017 ± .0001	.2932 ± .0099	.0008 ± .0001	.0018 ± .0001	.681 ± .001	1.208 ± .007
	128	32	.0017 ± .0001	.3882 ± .0104	.0006 ± .0001	.0016 ± .0001	.688 ± .001	1.167 ± .008
	128	64	.0018 ± .0001	.4280 ± .0208	.0007 ± .0001	.0017 ± .0001	.688 ± .001	1.159 ± .006
	256	32	.0019 ± .0001	.5717 ± .0054	**.0005 ± .0001**	**.0015 ± .0001**	**.692 ± .001**	1.137 ± .004
	256	64	.0021 ± .0001	.6285 ± .0092	.0006 ± .0001	.0016 ± .0001	.692 ± .001	**1.135 ± .005**
tissue	16	32	.0178 ± .0011	.8854 ± .0248	.0011 ± .0004	.0043 ± .0003	.589 ± .005	1.117 ± .012
	16	64	.0147 ± .0006	.8420 ± .0424	.0010 ± .0002	.0042 ± .0003	.598 ± .002	1.105 ± .004
	32	32	.0113 ± .0005	.6746 ± .0607	.0011 ± .0004	.0037 ± .0005	.616 ± .003	1.051 ± .007
	32	64	.0097 ± .0005	.6498 ± .0528	**.0010 ± .0003**	**.0036 ± .0002**	.618 ± .003	1.047 ± .008
	64	32	**.0082 ± .0003**	**.5091 ± .0653**	.0012 ± .0002	.0038 ± .0003	.623 ± .003	1.032 ± .007
	64	64	.0083 ± .0002	.6984 ± .0368	.0017 ± .0004	.0043 ± .0002	.624 ± .002	1.028 ± .007
	128	32	.0095 ± .0004	.7525 ± .0382	.0016 ± .0007	.0040 ± .0007	.628 ± .002	1.018 ± .006
	128	64	.0105 ± .0002	.8273 ± .0244	.0017 ± .0003	.0043 ± .0003	.628 ± .002	1.014 ± .005
	256	32	.0133 ± .0002	.8543 ± .0203	.0021 ± .0003	.0046 ± .0006	.632 ± .001	1.006 ± .005
	256	64	.0153 ± .0004	.8799 ± .0146	.0021 ± .0004	.0048 ± .0005	**.632 ± .002**	**1.004 ± .005**

$w \in \{32, 64\}$ yielding the best or comparable results. Finally, Table 5 indicates that the effect of $|\mathcal{C}|$ is secondary once w is fixed: once again we observe that the best results are achieved at $|\mathcal{C}| \in \{32, 64\}$.

Table 5: VQ results when changing codebook size ($|\mathcal{C}|$) for a fixed number of slots ($w = 64$). Blue row reports main-paper parameters.

Data	w	$\lvert\mathcal{C}\rvert$	$LCE \downarrow$	$MLCE \downarrow$	$ECCE \downarrow$	$ECE \downarrow$	$ACC \uparrow$	$NLL \downarrow$
cifar10	64	16	.0082 ± .0001	.6662 ± .0122	**.0011 ± .0002**	**.0037 ± .0003**	.887 ± .001	.359 ± .003
	64	32	.0065 ± .0002	.5665 ± .0173	.0012 ± .0001	.0038 ± .0002	.889 ± .001	.352 ± .001
	64	64	**.0059 ± .0002**	.5595 ± .0276	.0013 ± .0002	.0037 ± .0002	**.889 ± .001**	**.348 ± .002**
	64	128	.0061 ± .0002	.5650 ± .0478	.0014 ± .0003	.0040 ± .0005	.888 ± .001	.352 ± .004
	64	256	.0060 ± .0001	**.5358 ± .0218**	.0011 ± .0003	.0038 ± .0004	.888 ± .001	.354 ± .004
cifar100	64	16	.0023 ± .0002	.6288 ± .0573	**.0006 ± .0001**	**.0016 ± .0001**	.677 ± .002	1.221 ± .004
	64	32	**.0016 ± .0001**	.3274 ± .0141	.0007 ± .0001	.0017 ± .0001	.680 ± .003	1.213 ± .004
	64	64	.0017 ± .0001	**.2932 ± .0099**	.0008 ± .0001	.0018 ± .0001	.681 ± .001	1.208 ± .007
	64	128	.0017 ± .0002	.3299 ± .0068	.0008 ± .0001	.0018 ± .0001	.681 ± .002	1.212 ± .011
	64	256	.0017 ± .0001	.3546 ± .0127	.0008 ± .0001	.0018 ± .0001	**.682 ± .001**	**1.204 ± .006**
tissue	64	16	.0115 ± .0005	.5393 ± .0397	.0012 ± .0004	.0039 ± .0005	.621 ± .003	1.039 ± 0.007
	64	32	**.0082 ± .0003**	**.5091 ± .0653**	**.0012 ± .0002**	**.0038 ± .0003**	.623 ± .003	1.032 ± 0.007
	64	64	.0083 ± .0002	.6984 ± .0368	.0017 ± .0004	.0043 ± .0002	.624 ± .002	1.028 ± 0.007
	64	128	.0086 ± .0003	.7504 ± .0304	.0015 ± .0004	.0038 ± .0004	.625 ± .002	1.026 ± 0.006
	64	256	.0088 ± .0004	.7341 ± .0508	.0013 ± .0004	.0038 ± .0007	**.625 ± .003**	**1.024 ± .007**

C.4 Ablating the calibration-set size

We report the effect of calibration set size on local calibration (LCE) and negative log-likelihood (NLL) for `tissue` in Figure 6. As expected, all methods benefit from increased calibration data, with both LCE and NLL generally decreasing as the calibration set grows with the sole exception of LN which shows rather unstable performance. However, VQ consistently outperforms all baselines across the entire data regime. Notably, VQ achieves strong performance even in the low-data setting, where it already attains the lowest LCE and competitive NLL, indicating superior data efficiency.

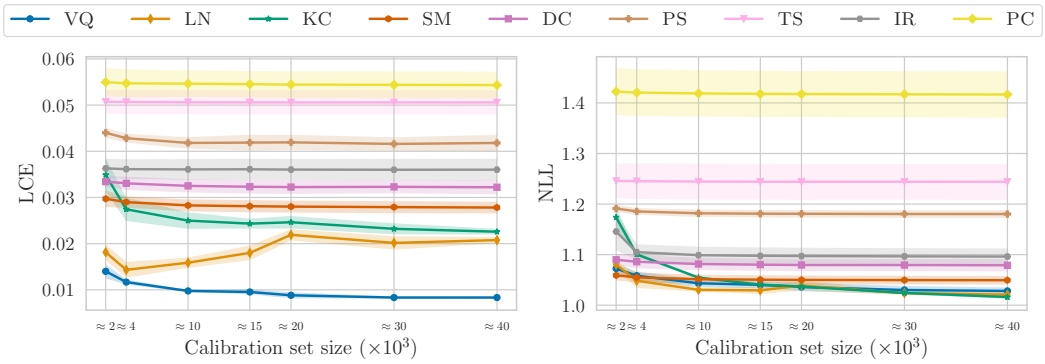


Figure 6: LCE (left) and NLL (right) for various calibration-set sizes on `TissueMNIST`.

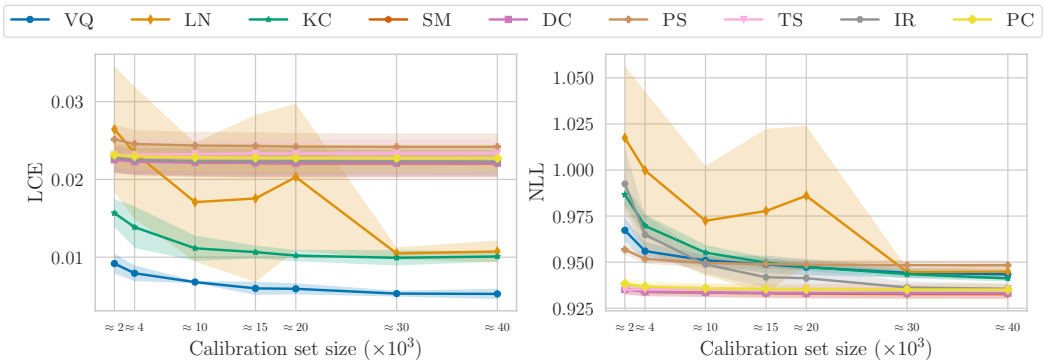


Figure 7: LCE (left) and NLL (right) for various calibration-set sizes on `Weather`.

As the calibration set size increases, VQ continues to improve and maintains a clear margin in LCE, while remaining among the best methods in terms of NLL. In contrast, several baselines exhibit slower improvements or plateau early, particularly in low-data regimes. These results suggest that VQ is not only more accurate in well-sampled settings but is also substantially more robust under data scarcity, highlighting the effectiveness of parameter sharing in leveraging limited calibration data.

We report the effect of calibration set size on local calibration (LCE) and negative log-likelihood (NLL) for `Weather` in Figure 7. Consistent with the previous experiment, all methods generally benefit from increased calibration data, with both LCE and NLL decreasing as the calibration set grows. However, VQ consistently achieves the lowest LCE across all calibration sizes, with a clear margin over competing methods. Notably, VQ exhibits strong data efficiency, attaining low LCE even in the smallest data regime and continuing to improve steadily as more calibration data becomes available. In terms of NLL, all methods converge to similar performance as the calibration set increases, with only minor differences in the large-data regime. While VQ is not always the best in NLL for very small calibration sizes, it quickly becomes competitive and remains among the top-performing methods as more data is available. Overall, these results reinforce that VQ's primary advantage lies in robust and data-efficient local calibration, particularly in low-data settings, while maintaining competitive negative log-likelihood performance.

C.5 Additional metrics

We additionally evaluate our method and all baselines according to expected calibration error (ECE) and report accuracy (ACC) (Table 6).

Table 6: Results for our approach (highlighted in blue) vs baselines (ResNet backbone).

Data	Method	$LCE \downarrow$	$MLCE \downarrow$	$ECCE \downarrow$	$ECE \downarrow$	$ACC \uparrow$	$NLL \downarrow$
cifar10	VQ	**.0059 ± .0002**	**.5595 ± .0276**	.0013 ± .0002	.0037 ± .0002	.889 ± .001	.348 ± .002
	SM	.0102 ± .0003	.8027 ± .0166	**.0008 ± .0001**	**.0037 ± .0001**	.890 ± .002	.333 ± .007
	DC	.0099 ± .0003	.8127 ± .0194	.0011 ± .0001	.0037 ± .0001	.890 ± .002	**.332 ± .007**
	KC	.0095 ± .0004	.7923 ± .0241	.0015 ± .0003	.0054 ± .0004	**.893 ± .001**	.340 ± .002
	LN	.0079 ± .0003	.6279 ± .0157	.0017 ± .0001	.0048 ± .0003	.888 ± .002	.347 ± .002
	PS	.0180 ± .0005	.8542 ± .0224	.0014 ± .0001	.0085 ± .0003	.884 ± .001	.466 ± .004
	IR	.0115 ± .0003	.7978 ± .0162	.0014 ± .0002	.0047 ± .0001	.884 ± .001	.364 ± .008
	TS	.0127 ± .0006	.8200 ± .0191	.0036 ± .0008	.0072 ± .0010	.884 ± .001	.362 ± .008
	PC	.0133 ± .0006	.8339 ± .0456	.0039 ± .0008	.0085 ± .0009	.880 ± .002	.377 ± .008
	NC	.0154 ± .0005	.8729 ± .0235	.0066 ± .0004	.0153 ± .0006	.8838 ± .0014	.4943 ± .0180
cifar100	VQ	**.0017 ± .0001**	**.2932 ± .0099**	.0008 ± .0001	**.0018 ± .0001**	.681 ± .001	1.208 ± .007
	SM	.0031 ± .0002	.8121 ± .0190	.0008 ± .0006	.0024 ± .0015	.677 ± .013	1.225 ± .043
	DC	.0027 ± .0001	.8113 ± .0121	**.0007 ± .0001**	.0019 ± .0001	**.690 ± .002**	1.154 ± .009
	KC	.0043 ± .0001	.7575 ± .0101	.0013 ± .0001	.0044 ± .0001	.679 ± .003	1.352 ± .009
	LN	.0024 ± .0001	.7022 ± .0070	**.0007 ± .0001**	.0018 ± .0001	.688 ± .001	**1.125 ± .002**
	PS	.0053 ± .0001	.7031 ± .0409	.0011 ± .0001	.0033 ± .0001	.670 ± .002	1.618 ± .007
	IR	.0031 ± .0001	.8049 ± .0116	**.0007 ± .0001**	.0020 ± .0001	.670 ± .002	1.437 ± .029
	TS	.0034 ± .0001	.8239 ± .0173	.0013 ± .0001	.0024 ± .0001	.670 ± .002	1.277 ± .009
	PC	.0041 ± .0002	.8187 ± .0127	.0013 ± .0001	.0026 ± .0001	.662 ± .003	1.272 ± .012
	NC	.0032 ± .0001	.9155 ± .0104	.0017 ± .0001	.0039 ± .0001	.670 ± .002	1.502 ± .035
tissue	VQ	**.0083 ± .0002**	.6984 ± .0368	.0017 ± .0004	.0043 ± .0002	.624 ± .002	1.028 ± .007
	SM	.0278 ± .0013	.9609 ± .0106	**.0008 ± .0001**	**.0037 ± .0001**	.618 ± .003	1.049 ± .009
	DC	.0322 ± .0015	.9706 ± .0098	.0065 ± .0009	.0183 ± .0010	.615 ± .004	1.079 ± .012
	KC	.0226 ± .0006	.7296 ± .0570	.0061 ± .0004	.0195 ± .0004	**.638 ± .001**	**1.016 ± .003**
	LN	.0208 ± .0007	**.6932 ± .0661**	.0058 ± .0004	.0181 ± .0006	.634 ± .002	1.021 ± .006
	PS	.0418 ± .0018	.9518 ± .0045	.0013 ± .0001	.0135 ± .0012	.611 ± .004	1.180 ± .008
	IR	.0360 ± .0024	.9602 ± .0103	.0026 ± .0004	.0095 ± .0005	.612 ± .004	1.096 ± .016
	TS	.0506 ± .0026	.9822 ± .0084	.0168 ± .0023	.0410 ± .0022	.603 ± .008	1.244 ± .036
	PC	.0543 ± .0029	.9768 ± .0133	.0145 ± .0022	.0384 ± .0025	.575 ± .008	1.417 ± .046
	NC	.1031 ± .0020	.9953 ± .0021	.0415 ± .0029	.0977 ± .0020	.603 ± .008	2.100 ± .101

Across all three datasets, VQ consistently attains very strong results in terms of ECE. It ties for first place with DC and SM on Cifar10, is best on Cifar100 and second best on tissue. This result further validates the effectiveness of our approach in jointly achieving strong local and global calibration. For ACC, VQ yields improvements over the NC baseline. Overall, its performance is competitive with the other approaches, though it never attains the best individual score.

C.6 More architectures: ConvNext

Calibration and performance. We report results for ConvNext architecture in Table 7. Results with a ConvNext backbone confirm the trends observed with ResNet. Across all datasets, VQ consistently achieves the best local calibration, attaining the lowest LCE and MLCE by a clear margin. On cifar10 and cifar100, VQ substantially reduces both average and worst-case local calibration error compared to all baselines, with particularly large improvements in $MLCE$, highlighting its robustness in hard regions. Similar behavior is observed on tissue, where VQ again achieves the lowest LCE and $MLCE$, while competing local methods exhibit significantly higher errors.

In terms of global calibration, VQ remains competitive but does not always achieve the best $ECCE$, with global methods such as DC and SM slightly outperforming it in some cases. However, these differences are small and come at the cost of significantly worse local calibration. Finally, predictive performance remains comparable across methods: VQ achieves accuracy on par with other calibrators, with only minor variations across datasets, indicating that improvements in local calibration do not come at the expense of classification performance.

Under data sparsity. For ConvNext architecture we report in Figure 8 local calibration error as a function of neighborhood density, measured via ESS. As in the ResNet setting, all methods exhibit a consistent trend: calibration error is highest in low-density regions and decreases as the effective sample size increases. However, VQ consistently achieves the lowest LCE across all datasets and ESS bins, with the largest gains in the sparsest regions. On cifar10 and cifar100, VQ significantly reduces local calibration error in low-density bins and quickly reaches near-zero error as density increases, outperforming both global and local baselines. A similar pattern is observed on tissue, where VQ maintains a clear advantage across the entire ESS spectrum, particularly in the low-support regime where other methods exhibit substantially higher errors.

Table 7: VQ results when the classifier backbone is a `ConvNext` model.

Data	Method	$LCE \downarrow$	$MLCE \downarrow$	$ECCE \downarrow$	$ECE \downarrow$	$ACC \uparrow$	$NLL \downarrow$
cifar10	VQ	**.0024 ± .0002**	**.3326 ± .0353**	.0006 ± .0003	.0021 ± .0004	.958 ± .002	.128 ± .006
	DC	.0034 ± .0006	.5810 ± .0829	**.0006 ± .0001**	**.0020 ± .0002**	.958 ± .002	.124 ± .003
	KC	.0038 ± .0004	.5501 ± .0854	.0008 ± .0001	.0025 ± .0003	**.960 ± .001**	**.122 ± .005**
	LN	.0032 ± .0006	.5615 ± .1347	.0007 ± .0003	.0023 ± .0004	.960 ± .002	.122 ± .002
	SM	.0036 ± .0006	.5791 ± .0867	.0006 ± .0001	.0021 ± .0001	.959 ± .002	.124 ± .003
	PS	.0077 ± .0005	.7295 ± .0649	.0013 ± .0001	.0059 ± .0007	.956 ± .002	.187 ± .005
	IR	.0040 ± .0007	.5589 ± .0676	.0006 ± .0001	.0024 ± .0002	.957 ± .002	.148 ± .001
	TS	.0047 ± .0004	.5893 ± .0908	.0014 ± .0004	.0033 ± .0006	.956 ± .003	.135 ± .007
	PC	.0045 ± .0006	.5987 ± .1074	.0014 ± .0003	.0032 ± .0003	.955 ± .002	.139 ± .005
	NC	.0043 ± .0003	.6400 ± .1017	.0017 ± .0003	.0039 ± .0007	.956 ± .003	.143 ± .007
cifar100	VQ	**.0012 ± .0001**	**.2778 ± .0078**	.0005 ± .0001	.0014 ± .0001	.799 ± .003	.690 ± .009
	DC	.0013 ± .0001	.4576 ± .0056	.0004 ± .0001	.0012 ± .0001	.817 ± .002	**.609 ± .006**
	KC	.0023 ± .0001	.5495 ± .0085	.0009 ± .0001	.0019 ± .0001	.789 ± .001	.729 ± .004
	LN	.0015 ± .0001	.6542 ± .0277	.0005 ± .0001	.0013 ± .0001	**.819 ± .001**	.616 ± .006
	SM	.0013 ± .0001	.4611 ± .0059	**.0004 ± .0001**	**.0012 ± .0001**	.811 ± .001	.631 ± .004
	PS	.0037 ± .0001	.4288 ± .0202	.0012 ± .0001	.0036 ± .0001	.801 ± .001	.989 ± .008
	IR	.0014 ± .0001	.4560 ± .0095	.0004 ± .0001	.0014 ± .0001	.805 ± .002	.837 ± .020
	TS	.0020 ± .0001	.4876 ± .0135	.0011 ± .0001	.0020 ± .0001	.791 ± .003	.702 ± .012
	PC	.0019 ± .0001	.4851 ± .0159	.0011 ± .0001	.0019 ± .0001	.790 ± .003	.696 ± .011
	NC	.0018 ± .0001	.5137 ± .0082	.0010 ± .0001	.0020 ± .0001	.791 ± .003	.711 ± .013
tissue	VQ	**.0041 ± .0003**	**.1553 ± .0323**	.0011 ± .0002	.0033 ± .0003	.671 ± .004	.903 ± .011
	DC	.0060 ± .0004	.4298 ± .0422	.0011 ± .0002	.0031 ± .0003	.674 ± .003	.894 ± .008
	KC	.0200 ± .0012	.2108 ± .0127	.0066 ± .0005	.0191 ± .0013	**.678 ± .003**	.913 ± .008
	LN	.0171 ± .0003	.1858 ± .0084	.0061 ± .0003	.0166 ± .0004	.677 ± .004	.907 ± .009
	SM	.0060 ± .0003	.4325 ± .0424	**.0006 ± .0001**	**.0029 ± .0001**	.674 ± .003	**.894 ± .008**
	PS	.0175 ± .0005	.4316 ± .0305	.0023 ± .0001	.0152 ± .0007	.672 ± .003	.953 ± .006
	IR	.0075 ± .0006	.4405 ± .0550	.0011 ± .0002	.0041 ± .0004	.672 ± .003	.903 ± .009
	TS	.0127 ± .0025	.4303 ± .0458	.0068 ± .0020	.0111 ± .0029	.668 ± .002	.909 ± .006
	PC	.0124 ± .0023	.4331 ± .0497	.0066 ± .0019	.0109 ± .0027	.668 ± .002	.910 ± .005
	NC	.0159 ± .0036	.4537 ± .0480	.0084 ± .0033	.0144 ± .0041	.668 ± .002	.910 ± .004

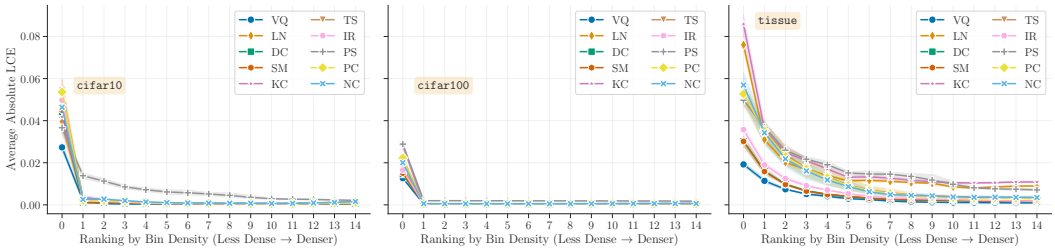


Figure 8: Local calibration in density-based sub-bins over five runs (`ConvNext` backbone).

As density increases, the performance gap narrows and all methods converge to similar LCE values, indicating that calibration becomes easier with sufficient local data. These results confirm that the benefits of VQ generalize across different convolutional architectures.

C.7 More Architectures: ViT

Calibration and performance. We report results for `ViT` architecture in Table 8. Results with a `ViT` backbone further confirm the trends observed across other architectures. VQ consistently achieves the best or near-best local calibration, attaining the lowest LCE across all datasets (tied on `cifar10`, second-best but within noise on `cifar100`, and best on `tissue`). More importantly, VQ clearly dominates in terms of $MLCE$, achieving the lowest worst-case local calibration error on all datasets by a substantial margin with the sole exception of `Cifar10` where differences are not significant. This highlights its robustness in challenging regions of the representation space, where competing methods exhibit significantly higher variability.

In terms of global calibration, VQ remains competitive but does not consistently achieve the best $ECCE$, with global methods such as DC and SM occasionally outperforming it. However, these gains in global calibration are accompanied by substantially worse local calibration. Finally, VQ

Table 8: VQ results when the classifier backbone is a `ViT` model.

Data	Method	$LCE \downarrow$	$MLCE \downarrow$	$ECCE \downarrow$	$ECE \downarrow$	$ACC \uparrow$	$NLL \downarrow$
cifar10	VQ	**.0016 ± .0001**	.2808 ± .0210	.0004 ± .0001	.0016 ± .0001	.974 ± .001	.086 ± .002
	DC	**.0016 ± .0001**	.4789 ± .0354	.0003 ± .0001	**.0013 ± .0001**	.977 ± .001	.073 ± .001
	KC	.0032 ± .0002	.7915 ± .0214	.0006 ± .0001	.0018 ± .0002	**.979 ± .001**	**.071 ± .002**
	LN	.0025 ± .0003	.9027 ± .0709	.0013 ± .0005	.0032 ± .0011	.977 ± .001	.132 ± .039
	SM	.0017 ± .0001	.4389 ± .0408	**.0003 ± .0001**	.0014 ± .0001	.978 ± .001	.071 ± .002
	PS	.0043 ± .0002	**.2219 ± .1137**	.0011 ± .0001	.0035 ± .0005	.977 ± .001	.118 ± .007
	IR	.0018 ± .0001	.4422 ± .0467	.0004 ± .0001	.0016 ± .0001	.977 ± .001	.087 ± .004
	TS	.0020 ± .0001	.4728 ± .0424	.0008 ± .0002	.0019 ± .0003	.977 ± .001	.078 ± .004
	PC	.0022 ± .0003	.5402 ± .0483	.0009 ± .0002	.0024 ± .0004	.975 ± .001	.086 ± .006
	NC	.0021 ± .0002	.5239 ± .0750	.0012 ± .0003	.0031 ± .0007	.977 ± .001	.106 ± .018
cifar100	VQ	.0012 ± .0001	**.2680 ± .0075**	.0005 ± .0001	.0014 ± .0001	.773 ± .002	.838 ± .003
	DC	.0023 ± .0002	.4558 ± .0088	.0009 ± .0001	.0026 ± .0002	.786 ± .004	1.013 ± .083
	KC	.0049 ± .0001	.8920 ± .0029	.0010 ± .0001	.0022 ± .0001	.782 ± .001	.818 ± .005
	LN	.0028 ± .0001	.9671 ± .0017	.0006 ± .0001	.0017 ± .0001	.819 ± .002	.717 ± .008
	SM	**.0011 ± .0001**	.3590 ± .0090	**.0003 ± .0001**	**.0011 ± .0001**	**.819 ± .002**	**.660 ± .005**
	PS	.0030 ± .0001	.3317 ± .0149	.0011 ± .0001	.0031 ± .0001	.818 ± .002	.969 ± .007
	IR	**.0011 ± .0001**	.3439 ± .0073	.0004 ± .0001	.0013 ± .0001	.816 ± .002	.868 ± .011
	TS	.0012 ± .0001	.3715 ± .0048	.0005 ± .0001	.0013 ± .0001	.816 ± .002	.677 ± .007
	PC	.0014 ± .0001	.3550 ± .0094	.0005 ± .0001	.0015 ± .0001	.812 ± .002	.700 ± .005
	NC	.0017 ± .0001	.4098 ± .0056	.0008 ± .0001	.0020 ± .0001	.816 ± .002	.768 ± .008
tissue	VQ	**.0042 ± .0006**	**.1064 ± .0101**	.0014 ± .0004	.0037 ± .0005	.662 ± .001	.925 ± .001
	DC	.0062 ± .0003	.5737 ± .0755	.0011 ± .0002	**.0034 ± .0002**	.677 ± .002	**.886 ± .005**
	KC	.0225 ± .0015	.6667 ± .0451	.0066 ± .0005	.0193 ± .0018	**.678 ± .002**	.911 ± .002
	LN	.0189 ± .0007	.5906 ± .0351	.0059 ± .0005	.0159 ± .0008	.676 ± .002	.906 ± .003
	SM	.0066 ± .0005	.5603 ± .0670	**.0009 ± .0001**	.0036 ± .0002	.677 ± .002	.885 ± .005
	PS	.0180 ± .0008	.4960 ± .0901	.0025 ± .0002	.0166 ± .0010	.674 ± .002	.951 ± .008
	IR	.0077 ± .0004	.5725 ± .0937	.0014 ± .0002	.0042 ± .0002	.675 ± .002	.896 ± .007
	TS	.0132 ± .0006	.5615 ± .0898	.0066 ± .0007	.0108 ± .0008	.672 ± .002	.902 ± .005
	PC	.0126 ± .0006	.5676 ± .0732	.0062 ± .0006	.0110 ± .0007	.672 ± .001	.905 ± .006
	NC	.0169 ± .0033	.6226 ± .0826	.0076 ± .0016	.0164 ± .0044	.672 ± .002	.911 ± .011

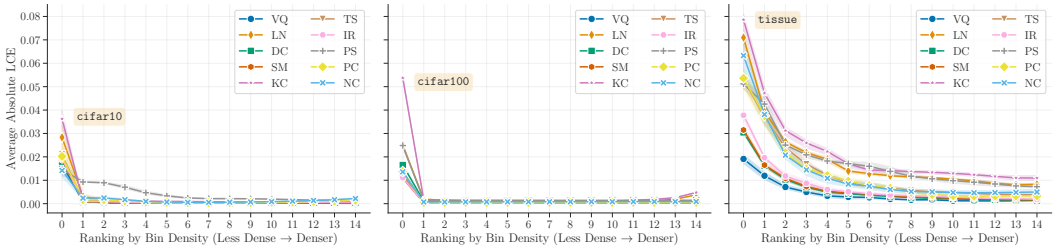


Figure 9: Local calibration in density-based sub-bins over five runs (`ViT` backbone).

shows a slight decrease in ACC in some ViT settings, likely due to interactions between quantization and the geometry of the representation space. However, this effect is small and does not impact its strong gains in local calibration.

Under data sparsity. For `ViT` architecture we report in Figure 9 local calibration error as a function of neighborhood density, measured via ESS. As in previous architectures, all methods exhibit the expected trend: calibration error is highest in low-density regions and decreases as ESS increases. However, the relative improvements of VQ vary across datasets. On `cifar10` and `cifar100`, differences between methods are relatively small, and VQ performs comparably to strong baselines across most ESS bins, with no substantial advantage in low-density regions.

In contrast, on `tissue`, VQ consistently achieves lower LCE across the entire ESS spectrum, with particularly pronounced gains in the lowest-density bins. This indicates that VQ remains effective in mitigating calibration errors under data sparsity when the representation space exhibits stronger heterogeneity. Overall, these results suggest that the benefits of VQ depend on the structure of the underlying representation space: while gains may be limited when features are already well-behaved (as in `Cifar10` or `Cifar100` with `ViT`), VQ provides clear advantages in more challenging settings.

Table 9: VQ codewords usage statistics across all datasets and model architectures on the test set.

Data	Model	Min Usage	Max Usage	Usage Std	# Assign. ($\times 10^6$)
cifar10	RESNET	1790 ± 208	$126\,583 \pm 10\,273$	$17\,974 \pm 1123$	1.296
	CONVNEXT	2874 ± 1200	$42\,831 \pm 8268$	9245 ± 2427	1.296
	VIT	4968 ± 1425	$27\,955 \pm 1325$	5252 ± 580	1.296
cifar100	RESNET	2481 ± 1167	$106\,616 \pm 12\,547$	$14\,896 \pm 683$	1.296
	CONVNEXT	4294 ± 792	$31\,017 \pm 3162$	5207 ± 602	1.296
	VIT	5435 ± 1635	$23\,984 \pm 599$	3565 ± 425	1.296
tissue	RESNET	4825 ± 897	$503\,999 \pm 17\,512$	$71\,517 \pm 1148$	4.765
	CONVNEXT	9233 ± 2048	$254\,738 \pm 32\,372$	$50\,291 \pm 4460$	4.765
	VIT	$14\,126 \pm 2716$	$116\,994 \pm 7445$	$23\,582 \pm 1222$	4.765
weather	FTT	$28\,983 \pm 5381$	$177\,686 \pm 21\,865$	$41\,796 \pm 3553$	5.760

C.8 Codewords usage statistics

We report quantization codewords usage statistics across datasets and architectures in Table 9.

Min Usage and Max Usage report the minimum and maximum number of assignments received by any codeword, respectively, while Usage Std denotes the standard deviation of codeword assignments, capturing the variability in usage across the codebook. # Assign. indicates the total number of assignments (in millions), obtained as the number of quantization slots times the test-set cardinality. Each value is assigned a standard deviation across the 5 used seeds. These statistics provide a global view of how evenly the codebook is utilized and allow us to detect potential issues such as underutilized (dead) codewords.

We observe that all codewords are consistently assigned a non-zero number of samples, with minimum usage well above zero in all settings. This confirms that no dead codewords are present and that the learned codebooks are fully utilized. While usage is naturally imbalanced, with some codewords receiving more assignments than others, the standard deviation remains moderate relative to the total number of assignments, indicating a healthy distribution of mass across the codebook. Overall, these results show that the quantization scheme effectively leverages the full capacity of the codebook across different architectures and datasets.

C.9 Regularisation of the parameter codebooks and of the diagonal scaling component.

In our experiments, we keep the diagonal scaling $\boldsymbol{\sigma}^2$ fixed. In what follows we explore a regularisation strategy to improve the flexibility of the model allowing it to freely learn the diagonal scaling component while preventing overfitting. Interestingly, Corollary 1 already highlights the role of the diagonal scaling. The cell-level estimation error grows proportionally to the slot weights $\boldsymbol{\sigma}_i^2$, so excessively large scaling coefficients may amplify estimation noise from the corresponding codeword parameters. This provides additional motivation for regularizing $\boldsymbol{\sigma}^2$ when it is learned. Following [7], we propose to add the following regularisation component to the cross-entropy loss:

$$\begin{aligned}\mathcal{R}(\mathcal{A}, \mathcal{B}, \boldsymbol{\sigma}) = \lambda_{\mathcal{A}} \sum_{c=1}^{|\mathcal{C}|} \frac{|\mathcal{Y}|}{n_c + \eta} \|\mathbf{a}_c - \mathbf{a}_c^0\|_2^2 \\ + \lambda_{\mathcal{B}} \sum_{c=1}^{|\mathcal{C}|} \frac{|\mathcal{Y}|}{n_c + \eta} \|\mathbf{b}_c - \mathbf{b}_c^0\|_2^2 \\ + \lambda_{\boldsymbol{\sigma}} \sum_{j=1}^{w} \frac{1}{n_{\text{cal}} + \eta} (\sigma_j^2 - 1)^2 \end{aligned} \tag{123}$$

Table 10: Results with a `ResNet` classifier. $\lambda_{\mathcal{A}}, \lambda_{\mathcal{B}}, \lambda_{\boldsymbol{\sigma}} = 1e^{-4}$

Data	Method	$LCE \downarrow$	$MLCE \downarrow$	$ECCE \downarrow$	$ECE \downarrow$	$ACC \uparrow$	$NLL \downarrow$
cifar10	VQ	.0059 ± .0002	.5595 ± .0276	.0013 ± .0002	.0037 ± .0002	.889 ± .001	.348 ± .002
	VQ_λ	**.0058 ± .0002**	**.5368 ± .0591**	.0011 ± .0003	.0038 ± .0003	.890 ± .002	.342 ± .006
	SM	.0102 ± .0003	.8027 ± .0166	**.0008 ± .0001**	**.0037 ± .0001**	.890 ± .002	.333 ± .007
	DC	.0099 ± .0003	.8127 ± .0194	.0011 ± .0001	.0037 ± .0001	.890 ± .002	**.332 ± .007**
	KC	.0095 ± .0004	.7923 ± .0241	.0015 ± .0003	.0054 ± .0004	**.893 ± .001**	.340 ± .002
	LN	.0079 ± .0003	.6279 ± .0157	.0017 ± .0001	.0048 ± .0003	.888 ± .002	.347 ± .002
	PS	.0180 ± .0005	.8542 ± .0224	.0014 ± .0001	.0085 ± .0003	.884 ± .001	.466 ± .004
	IR	.0115 ± .0003	.7978 ± .0162	.0014 ± .0002	.0047 ± .0001	.884 ± .001	.364 ± .008
	TS	.0127 ± .0006	.8200 ± .0191	.0036 ± .0008	.0072 ± .0010	.884 ± .001	.362 ± .008
	PC	.0133 ± .0006	.8339 ± .0456	.0039 ± .0008	.0085 ± .0009	.880 ± .002	.377 ± .008
	NC	.0154 ± .0005	.8729 ± .0235	.0066 ± .0004	.0153 ± .0006	.8838 ± .0014	.4943 ± .0180
cifar100	VQ	**.0017 ± .0001**	**.2932 ± .0099**	.0008 ± .0001	**.0018 ± .0001**	.681 ± .001	1.208 ± .007
	VQ_λ	**.0017 ± .0001**	.3073 ± .0115	.0008 ± .0001	**.0018 ± .0001**	.679 ± .002	1.227 ± .013
	SM	.0031 ± .0002	.8121 ± .0190	.0008 ± .0006	.0024 ± .0015	.677 ± .013	1.225 ± .043
	DC	.0027 ± .0001	.8113 ± .0121	**.0007 ± .0001**	.0019 ± .0001	**.690 ± .002**	1.154 ± .009
	KC	.0043 ± .0001	.7575 ± .0101	.0013 ± .0001	.0044 ± .0001	.679 ± .003	1.352 ± .009
	LN	.0024 ± .0001	.7022 ± .0070	**.0007 ± .0001**	.0018 ± .0001	.688 ± .001	**1.125 ± .002**
	PS	.0053 ± .0001	.7031 ± .0409	.0011 ± .0001	.0033 ± .0001	.670 ± .002	1.618 ± .007
	IR	.0031 ± .0001	.8049 ± .0116	**.0007 ± .0001**	.0020 ± .0001	.670 ± .002	1.437 ± .029
	TS	.0034 ± .0001	.8239 ± .0173	.0013 ± .0001	.0024 ± .0001	.670 ± .002	1.277 ± .009
	PC	.0041 ± .0002	.8187 ± .0127	.0013 ± .0001	.0026 ± .0001	.662 ± .003	1.272 ± .012
	NC	.0032 ± .0001	.9155 ± .0104	.0017 ± .0001	.0039 ± .0001	.670 ± .002	1.502 ± .035
tissue	VQ	.0083 ± .0002	.6984 ± .0368	.0017 ± .0004	.0043 ± .0002	.624 ± .002	1.028 ± .007
	VQ_λ	**.0080 ± .0004**	.6945 ± .0472	.0012 ± .0004	.0037 ± .0006	.625 ± .002	1.028 ± .007
	SM	.0278 ± .0013	.9609 ± .0106	**.0008 ± .0001**	**.0037 ± .0001**	.618 ± .003	1.049 ± .009
	DC	.0322 ± .0015	.9706 ± .0098	.0065 ± .0009	.0183 ± .0010	.615 ± .004	1.079 ± .012
	KC	.0226 ± .0006	.7296 ± .0570	.0061 ± .0004	.0195 ± .0004	**.638 ± .001**	**1.016 ± .003**
	LN	.0208 ± .0007	**.6932 ± .0661**	.0058 ± .0004	.0181 ± .0006	.634 ± .002	1.021 ± .006
	PS	.0418 ± .0018	.9518 ± .0045	.0013 ± .0001	.0135 ± .0012	.611 ± .004	1.180 ± .008
	IR	.0360 ± .0024	.9602 ± .0103	.0026 ± .0004	.0095 ± .0005	.612 ± .004	1.096 ± .016
	TS	.0506 ± .0026	.9822 ± .0084	.0168 ± .0023	.0410 ± .0022	.603 ± .008	1.244 ± .036
	PC	.0543 ± .0029	.9768 ± .0133	.0145 ± .0022	.0384 ± .0025	.575 ± .008	1.417 ± .046
	NC	.1031 ± .0020	.9953 ± .0021	.0415 ± .0029	.0977 ± .0020	.603 ± .008	2.100 ± .101

Table 11: Results on the `Weather` dataset with an `FT-Transformer`. $\lambda_{\mathcal{A}}, \lambda_{\mathcal{B}}, \lambda_{\boldsymbol{\sigma}} = 1e^{-4}$

Data	Method	$LCE \downarrow$	$MLCE \downarrow$	$ECCE \downarrow$	$ECE \downarrow$	$ACC \uparrow$	$NLL \downarrow$
weather	VQ	.0053 ± .0007	.1782 ± .0344	.0010 ± .0004	.0030 ± .0007	.603 ± .001	.943 ± .003
	VQ_λ	**.0047 ± .0003**	.1968 ± .0325	.0007 ± .0001	.0024 ± .0002	.603 ± .001	.943 ± .003
	DC	.0221 ± .0018	.4509 ± .0102	.0007 ± .0001	.0022 ± .0001	.607 ± .002	.933 ± .003
	KC	.0101 ± .0007	**.0998 ± .0208**	.0028 ± .0003	.0091 ± .0005	.606 ± .002	.941 ± .002
	LN	.0107 ± .0014	.1021 ± .0138	.0028 ± .0003	.0100 ± .0015	.605 ± .001	.945 ± .004
	SM	.0221 ± .0018	.4495 ± .0105	**.0005 ± .0001**	**.0015 ± .0002**	**.607 ± .001**	**.933 ± .003**
	PS	.0242 ± .0018	.4379 ± .0104	.0009 ± .0000	.0068 ± .0003	.607 ± .001	.948 ± .003
	IR	.0223 ± .0018	.4505 ± .0109	.0006 ± .0001	.0020 ± .0001	.606 ± .001	.936 ± .003
	TS	.0234 ± .0018	.4429 ± .0145	.0019 ± .0004	.0045 ± .0005	.607 ± .001	.936 ± .003
	PC	.0227 ± .0018	.4471 ± .0165	.0013 ± .0004	.0029 ± .0002	.606 ± .001	.935 ± .003
	NC	.0226 ± .0017	.4528 ± .0108	.0019 ± .0005	.0047 ± .0004	.607 ± .001	.936 ± .003

The factor, $|\mathcal{Y}|$ denotes the number of classes and appears in the penalty because each calibration parameter $\mathbf{a}_c \in \mathcal{A}$ and $\mathbf{b}_c \in \mathcal{B}$ is a vector in $\mathbb{R}^{|\mathcal{Y}|}$, so the regularization strength is scaled by the dimensionality of each factor. The quantity n_c denotes the empirical usage count of the c-th codeword in the calibration set. If each calibration example i is associated with a sequence of VQ indices $\mathbf{s}_i = \big(s_i(1), \ldots, s_i(w)\big)$, then $n_c = \sum_{i=1}^{n_{\text{cal}}} \sum_{j=1}^{w} \mathbf{1}\{s_i(j) = c\}$ where n_{cal} denotes the cardinality of the available calibration set. Thus, n_c counts how often the c-th codeword appears across all calibration examples and all positions. Frequent codewords receive a weaker penalty, whereas rare codewords are kept closer to their reference values. The constant $\eta > 0$ is a small stabilizing term that prevents division by zero and avoids excessively large weights for codewords with very small support. We use the simple default $\eta = 1$. The vectors $\mathbf{a}_c^0$ and $\mathbf{b}_c^0$ denote the initialisation values. These reference values are chosen so that the calibrator starts close to its default, identity-preserving behavior. Therefore, the terms

$$\|\mathbf{a}_c - \mathbf{a}_c^0\|_2^2 \qquad \text{and} \qquad \|\mathbf{b}_c - \mathbf{b}_c^0\|_2^2 \tag{124}$$

Table 12: Results with a `ConvNext` classifier. $\lambda_{\mathcal{A}}, \lambda_{\mathcal{B}}, \lambda_{\boldsymbol{\sigma}} = 1e^{-4}$

Data	Method	$LCE \downarrow$	$MLCE \downarrow$	$ECCE \downarrow$	$ECE \downarrow$	$ACC \uparrow$	$NLL \downarrow$
cifar10	VQ	.0024 ± .0002	.3326 ± .0353	.0006 ± .0003	.0021 ± .0004	.958 ± .002	.128 ± .006
	VQ_λ	**.0024 ± .0001**	**.3259 ± .0244**	**.0006 ± .0001**	**.0020 ± .0001**	.958 ± .001	.127 ± .004
	DC	.0034 ± .0006	.5810 ± .0829	**.0006 ± .0001**	.0020 ± .0002	.958 ± .002	.124 ± .003
	KC	.0038 ± .0004	.5501 ± .0854	.0008 ± .0001	.0025 ± .0003	**.960 ± .001**	**.122 ± .005**
	LN	.0032 ± .0006	.5615 ± .1347	.0007 ± .0003	.0023 ± .0004	.960 ± .002	.122 ± .002
	SM	.0036 ± .0006	.5791 ± .0867	.0006 ± .0001	.0021 ± .0001	.959 ± .002	.124 ± .003
	PS	.0077 ± .0005	.7295 ± .0649	.0013 ± .0001	.0059 ± .0007	.956 ± .002	.187 ± .005
	IR	.0040 ± .0007	.5589 ± .0676	.0006 ± .0001	.0024 ± .0002	.957 ± .002	.148 ± .001
	TS	.0047 ± .0004	.5893 ± .0908	.0014 ± .0004	.0033 ± .0006	.956 ± .003	.135 ± .007
	PC	.0045 ± .0006	.5987 ± .1074	.0014 ± .0003	.0032 ± .0003	.955 ± .002	.139 ± .005
	NC	.0043 ± .0003	.6400 ± .1017	.0017 ± .0003	.0039 ± .0007	.956 ± .003	.143 ± .007
cifar100	VQ	**.0012 ± .0001**	.2778 ± .0078	.0005 ± .0001	.0014 ± .0001	.799 ± .003	.690 ± .009
	VQ_λ	**.0012 ± .0001**	**.2743 ± .0080**	.0005 ± .0001	.0013 ± .0001	.799 ± .004	.687 ± .009
	DC	.0013 ± .0001	.4576 ± .0056	.0004 ± .0001	.0012 ± .0001	.817 ± .002	**.609 ± .006**
	KC	.0023 ± .0001	.5495 ± .0085	.0009 ± .0001	.0019 ± .0001	.789 ± .001	.729 ± .004
	LN	.0015 ± .0001	.6542 ± .0277	.0005 ± .0001	.0013 ± .0001	**.819 ± .001**	.616 ± .006
	SM	.0013 ± .0001	.4611 ± .0059	**.0004 ± .0001**	**.0012 ± .0001**	.811 ± .001	.631 ± .004
	PS	.0037 ± .0001	.4288 ± .0202	.0012 ± .0001	.0036 ± .0001	.801 ± .001	.989 ± .008
	IR	.0014 ± .0001	.4560 ± .0095	.0004 ± .0001	.0014 ± .0001	.805 ± .002	.837 ± .020
	TS	.0020 ± .0001	.4876 ± .0135	.0011 ± .0001	.0020 ± .0001	.791 ± .003	.702 ± .012
	PC	.0019 ± .0001	.4851 ± .0159	.0011 ± .0001	.0019 ± .0001	.790 ± .003	.696 ± .011
	NC	.0018 ± .0001	.5137 ± .0082	.0010 ± .0001	.0020 ± .0001	.791 ± .003	.711 ± .013
tissue	VQ	**.0041 ± .0003**	**.1553 ± .0323**	.0011 ± .0002	.0033 ± .0003	.671 ± .004	.903 ± .011
	VQ_λ	.0043 ± .0003	.1421 ± .0336	.0009 ± .0003	.0035 ± .0003	.672 ± .004	.904 ± .011
	DC	.0060 ± .0004	.4298 ± .0422	.0011 ± .0002	.0031 ± .0003	.674 ± .003	.894 ± .008
	KC	.0200 ± .0012	.2108 ± .0127	.0066 ± .0005	.0191 ± .0013	**.678 ± .003**	.913 ± .008
	LN	.0171 ± .0003	.1858 ± .0084	.0061 ± .0003	.0166 ± .0004	.677 ± .004	.907 ± .009
	SM	.0060 ± .0003	.4325 ± .0424	**.0006 ± .0001**	**.0029 ± .0001**	.674 ± .003	**.894 ± .008**
	PS	.0175 ± .0005	.4316 ± .0305	.0023 ± .0001	.0152 ± .0007	.672 ± .003	.953 ± .006
	IR	.0075 ± .0006	.4405 ± .0550	.0011 ± .0002	.0041 ± .0004	.672 ± .003	.903 ± .009
	TS	.0127 ± .0025	.4303 ± .0458	.0068 ± .0020	.0111 ± .0029	.668 ± .002	.909 ± .006
	PC	.0124 ± .0023	.4331 ± .0497	.0066 ± .0019	.0109 ± .0027	.668 ± .002	.910 ± .005
	NC	.0159 ± .0036	.4537 ± .0480	.0084 ± .0033	.0144 ± .0041	.668 ± .002	.910 ± .004

penalize deviations of the learned codeword factors from their initial calibration state. Overall, the factor $|\mathcal{Y}|/(n_c+\eta)$ makes the penalty stronger for rarely observed codewords. The regularizer therefore allows well-supported codewords to adapt more freely, while keeping poorly supported codewords close to their initial values. Across `ResNet`, `FT-Transformer`, `ConvNeXt`, and `ViT`, the regularized variant VQ_λ behaves very similarly to the main VQ formulation, while often yielding small improvements on several calibration metrics. On Weather (Table 11), VQ_λ improves LCE, ECCE, ECE, and NLL over VQ, while preserving accuracy, although MLCE slightly worsens. With ResNet and ConvNext backbones (Tables 10 and 12), the two variants remain closely matched, with VQ_λ typically improving or preserving local calibration and occasionally reducing NLL, while retaining the same overall advantage over competing local methods. The same pattern is observed with ViT (Table 13), where regularization produces only modest changes relative to VQ. Overall, these results indicate that learning $\mathrm{diag}(\boldsymbol{\sigma}^2)$ under regularization is a competitive alternative, but does not materially alter the conclusions obtained with the simpler fixed-scaling formulation.

C.10 Disentangling quantization and head adaptation

We finetune a classifier head on the calibration set (FT), to clearly isolate the gains of our proposal with respect to simple fine-tuning of the classification head on the calibration set. We further test the proposed pipeline with a frozen classification head ($\mathrm{VQ}_{\mathrm{frz}}$), where only the quantization module and the corresponding calibration maps can be learned. Results (Table 14) indicate that, although FT is effective in terms of NLL and ACC, simply refitting the head severely underperforms on both local metrics (LCE and $MLCE$) and global metrics like ECE and $ECCE$ with respect to VQ and its variant $\mathrm{VQ}_{\mathrm{frz}}$. Moreover, $\mathrm{VQ}_{\mathrm{frz}}$ has similar performance in terms of local metrics to VQ, suggesting that local gains do not primarily arise from replacing the classifier head. Overall, the full pipeline provides a stronger balance between local calibration and global performance than these approaches.

Table 13: Results with a `ViT` classifier. $\lambda_{\mathcal{A}}, \lambda_{\mathcal{B}}, \lambda_{\boldsymbol{\sigma}} = 1e^{-4}$

Data	Method	$LCE \downarrow$	$MLCE \downarrow$	$ECCE \downarrow$	$ECE \downarrow$	$ACC \uparrow$	$NLL \downarrow$
cifar10	VQ	**.0016 ± .0001**	.2808 ± .0210	.0004 ± .0001	.0016 ± .0001	.974 ± .001	.086 ± .002
	VQ_{λ}	**.0016 ± .0001**	.2849 ± .0002	.0004 ± .0000	.0016 ± .0000	.974 ± .000	.086 ± .001
	DC	**.0016 ± .0001**	.4789 ± .0354	.0003 ± .0001	**.0013 ± .0001**	.977 ± .001	.073 ± .001
	KC	.0032 ± .0002	.7915 ± .0214	.0006 ± .0001	.0018 ± .0002	**.979 ± .001**	**.071 ± .002**
	LN	.0025 ± .0003	.9027 ± .0709	.0013 ± .0005	.0032 ± .0011	.977 ± .001	.132 ± .039
	SM	.0017 ± .0001	.4389 ± .0408	**.0003 ± .0001**	.0014 ± .0001	.978 ± .001	.071 ± .002
	PS	.0043 ± .0002	**.2219 ± .1137**	.0011 ± .0001	.0035 ± .0005	.977 ± .001	.118 ± .007
	IR	.0018 ± .0001	.4422 ± .0467	.0004 ± .0001	.0016 ± .0001	.977 ± .001	.087 ± .004
	TS	.0020 ± .0001	.4728 ± .0424	.0008 ± .0002	.0019 ± .0003	.977 ± .001	.078 ± .004
	PC	.0022 ± .0003	.5402 ± .0483	.0009 ± .0002	.0024 ± .0004	.975 ± .001	.086 ± .006
	NC	.0021 ± .0002	.5239 ± .0750	.0012 ± .0003	.0031 ± .0007	.977 ± .001	.106 ± .018
cifar100	VQ	.0012 ± .0001	.2680 ± .0075	.0005 ± .0001	.0014 ± .0001	.773 ± .002	.838 ± .003
	VQ_{λ}	.0012 ± .0001	**.2636 ± .0001**	.0005 ± .0000	.0014 ± .0000	.774 ± .000	.834 ± .001
	DC	.0023 ± .0002	.4558 ± .0088	.0009 ± .0001	.0026 ± .0002	.786 ± .004	1.01 ± .083
	KC	.0049 ± .0001	.8920 ± .0029	.0010 ± .0001	.0022 ± .0001	.782 ± .001	.818 ± .005
	LN	.0028 ± .0001	.9671 ± .0017	.0006 ± .0001	.0017 ± .0001	.819 ± .002	.717 ± .008
	SM	**.0011 ± .0001**	.3590 ± .0090	**.0003 ± .0001**	**.0011 ± .0001**	**.819 ± .002**	**.660 ± .005**
	PS	.0030 ± .0001	.3317 ± .0149	.0011 ± .0001	.0031 ± .0001	.818 ± .002	.969 ± .007
	IR	**.0011 ± .0001**	.3439 ± .0073	.0004 ± .0001	.0013 ± .0001	.816 ± .002	.868 ± .011
	TS	.0012 ± .0001	.3715 ± .0048	.0005 ± .0001	.0013 ± .0001	.816 ± .002	.677 ± .007
	PC	.0014 ± .0001	.3550 ± .0094	.0005 ± .0001	.0015 ± .0001	.812 ± .002	.700 ± .005
	NC	.0017 ± .0001	.4098 ± .0056	.0008 ± .0001	.0020 ± .0001	.816 ± .002	.768 ± .008
tissue	VQ	.0042 ± .0006	.1064 ± .0101	.0014 ± .0004	.0037 ± .0005	.662 ± .001	.925 ± .001
	VQ_{λ}	**.0041 ± .0001**	**.1044 ± .0002**	.0011 ± .0001	.0035 ± .0001	.662 ± .001	.925 ± .001
	DC	.0062 ± .0003	.5737 ± .0755	.0011 ± .0002	**.0034 ± .0002**	.677 ± .002	**.886 ± .005**
	KC	.0225 ± .0015	.6667 ± .0451	.0066 ± .0005	.0193 ± .0018	**.678 ± .002**	.911 ± .002
	LN	.0189 ± .0007	.5906 ± .0351	.0059 ± .0005	.0159 ± .0008	.676 ± .002	.906 ± .003
	SM	.0066 ± .0005	.5603 ± .0670	**.0009 ± .0001**	.0036 ± .0002	.677 ± .002	.885 ± .005
	PS	.0180 ± .0008	.4960 ± .0901	.0025 ± .0002	.0166 ± .0010	.674 ± .002	.951 ± .008
	IR	.0077 ± .0004	.5725 ± .0937	.0014 ± .0002	.0042 ± .0002	.675 ± .002	.896 ± .007
	TS	.0132 ± .0006	.5615 ± .0898	.0066 ± .0007	.0108 ± .0008	.672 ± .002	.902 ± .005
	PC	.0126 ± .0006	.5676 ± .0732	.0062 ± .0006	.0110 ± .0007	.672 ± .001	.905 ± .006
	NC	.0169 ± .0033	.6226 ± .0826	.0076 ± .0016	.0164 ± .0044	.672 ± .002	.911 ± .011

Table 14: Finetune (FT) and frozen classification head (VQ_{frz}) performance with a `ResNet`.

Data	Method	$LCE \downarrow$	$MLCE \downarrow$	$ECCE \downarrow$	$ECE \downarrow$	$ACC \uparrow$	$NLL \downarrow$
cifar10	VQ	**.0059 ± .0002**	.5595 ± .0276	.0013 ± .0002	**.0037 ± .0002**	.889 ± .001	.348 ± .002
	FT	.0107 ± .0005	.8214 ± .0266	.0022 ± .0004	.0048 ± .0003	**.8942 ± .0013**	**.3173 ± .0051**
	VQ_{frz}	.0063 ± .0003	**.5268 ± .0413**	**.0012 ± .0002**	.0038 ± .0004	.888 ± .0012	.355 ± .006
cifar100	VQ	**.0017 ± .0001**	**.2932 ± .0099**	**.0008 ± .0001**	**.0018 ± .0001**	.681 ± .001	1.208 ± .007
	FT	.0032 ± .0001	.8124 ± .0118	.0013 ± .0001	.0023 ± .0001	**.6827 ± .0029**	**1.173 ± .007**
	VQ_{frz}	.0023 ± .0001	.3488 ± .0138	.0012 ± .0001	.0023 ± .0001	.661 ± .0010	1.380 ± .014
tissue	VQ	**.0083 ± .0002**	**.6984 ± .0368**	**.0017 ± .0004**	**.0043 ± .0002**	.624 ± .002	1.028 ± .007
	FT	.0282 ± .0011	.9597 ± .0158	.0042 ± .0012	.0074 ± .0013	**.6382 ± .0014**	**.9900 ± .0049**
	VQ_{frz}	.0097 ± .0007	.7143 ± .0382	.0022 ± .0007	.0048 ± .0007	.6132 ± .0028	1.0640 ± .0087

C.11 Role of the quantization partition

Our product-quantization scheme discretizes contiguous feature slots $\bar{\mathbf{z}}$, although there is no requirement that features grouped within the same slot be adjacent in the original representation. To assess the sensitivity of the method to this operational choice, we introduce a random permutation π of the feature indices and apply it to the representation before slotting, i.e.,

$$\mathbf{z}_{\pi} = \big(z_{\pi(1)}, \ldots, z_{\pi(m)}\big). \tag{125}$$

We then partition $\mathbf{z}_{\pi}$ into contiguous slots and apply the same VQ calibration pipeline. We denote this variant by VQ_{π}.

Table 15: Performance after random permutations of inputs $\mathbf{z}$ with a `ResNet`.

Data	Method	$LCE \downarrow$	$MLCE \downarrow$	$ECCE \downarrow$	$ECE \downarrow$	$ACC \uparrow$	$NLL \downarrow$
cifar10	VQ	**.0059 ± .0002**	.5595 ± .0276	.0013 ± .0002	**.0037 ± .0002**	.889 ± .001	.348 ± .002
	VQ_π	.0060 ± .0002	**.5584 ± .0305**	**.0012 ± .0002**	.0039 ± .0002	**.890 ± .0011**	**.342 ± .007**
cifar100	VQ	**.0017 ± .0001**	**.2932 ± .0099**	**.0008 ± .0001**	**.0018 ± .0001**	**.681 ± .001**	**1.208 ± .007**
	VQ_π	**.0017 ± .0001**	.2996 ± .0056	**.0008 ± .0001**	.0018 ± .0001	.6774 ± .0020	1.2300 ± .0120
tissue	VQ	.0083 ± .0002	.6984 ± .0368	.0017 ± .0004	.0043 ± .0002	**.624 ± .002**	1.028 ± .007
	VQ_π	**.0079 ± .0003**	**.6943 ± .0409**	**.0012 ± .0002**	**.0036 ± .0004**	**.624 ± .002**	**1.027 ± .007**

Results (Table 15) indicate that the random feature permutation has little effect on performance overall, with VQ_π closely matching the original VQ formulation and even improving several metrics. On `CIFAR-10`, VQ_π slightly improves $MLCE$, $ECCE$, ACC, and NLL, while on `CIFAR-100`, the two variants remain essentially comparable, with only small differences. On `tissue`, VQ_π improves all calibration metrics and marginally improves NLL at unchanged accuracy. This suggests that the method is not particularly sensitive to grouping contiguous features within slots. More importantly, the occasional improvements obtained after random permutation indicate that the choice of feature grouping itself may be worth optimizing: relevant directions of miscalibration may involve combinations of non-contiguous representation features, and a learned grouping strategy could therefore provide a more effective partition than the fixed contiguous one.

We further investigate the role of quantization in our pipeline by considering partitions that are not learned. To this end, we evaluate two strategies for initializing the codebook and keeping it fixed throughout calibration. In this setting, the codewords serve only to define the discrete indices used to assemble the compositional calibration maps, while also fixing the model scores within each cell. The first strategy ($\mathrm{VQ}_{\mathcal{N}}$) initializes codewords by sampling from a standard normal. Since these codewords may lie far from the representation manifold, the resulting partition can be highly lossy and may poorly reflect the geometry of the learned representation space. We therefore consider a second strategy ($\mathrm{VQ}_{\text{data}}$) that initializes the codebook directly from calibration representations. Specifically, each calibration representation is partitioned into w contiguous slots, yielding $w \cdot n_{\text{cal}}$ candidate slot vectors, from which $|\mathcal{C}|$ are sampled uniformly at random to form the codebook $\mathcal{C}$.

The results (Table 16) highlight the importance of learning a partition that captures meaningful local structure. When codewords are sampled from a standard normal distribution, the resulting assignments are largely unrelated to the geometry of the representation space, and performance consistently deteriorates. This provides an informative control: simply learning different calibration maps over an arbitrary partition is not sufficient to obtain the gains of VQ. Intuitively, if the induced cells do not correspond to meaningful differences in miscalibration, the optimal cell-specific corrections should become increasingly similar, effectively approaching the behavior of a global calibrator. In contrast, sampling codewords directly from calibration representations substantially improves performance and often approaches that of learned VQ, indicating that even a fixed partition benefits from being aligned with the representation manifold. Learning the codebook further allows the partition to adapt to the data, identifying local structure that is informative for calibration.

C.12 Sensitivity to the kernel bandwidth

We report in Table 17 and Table 18 results for LCE and $MLCE$ for different choices of the kernel bandwidth γ. Default for all results in the paper is 10 and the column is highlighted in blue. Across

Table 16: Performance when the quantization partition is not learned with a `ResNet`.

Data	Method	$LCE \downarrow$	$MLCE \downarrow$	$ECCE \downarrow$	$ECE \downarrow$	$ACC \uparrow$	$NLL \downarrow$
cifar10	VQ	$\mathbf{.0059 \pm .0002}$	$.5595 \pm .0276$	$.0013 \pm .0002$	$\mathbf{.0037 \pm .0002}$	$\mathbf{.889 \pm .001}$	$\mathbf{.348 \pm .002}$
	$\text{VQ}_{\mathcal{N}}$	$.0121 \pm .0022$	$.6997 \pm .0999$	$.0014 \pm .0005$	$.0051 \pm .0024$	$.877 \pm .012$	$.395 \pm .054$
	VQ_{data}	$.0066 \pm .0002$	$\mathbf{.4863 \pm .0512}$	$\mathbf{.0012 \pm .0003}$	$.0038 \pm .0003$	$.887 \pm .001$	$.353 \pm .006$
cifar100	VQ	$\mathbf{.0017 \pm .0001}$	$\mathbf{.2932 \pm .0099}$	$\mathbf{.0008 \pm .0001}$	$\mathbf{.0018 \pm .0001}$	$\mathbf{.681 \pm .001}$	$\mathbf{1.208 \pm .007}$
	$\text{VQ}_{\mathcal{N}}$	$.0026 \pm .0001$	$.5659 \pm .0334$	$.0009 \pm .0001$	$.0021 \pm .0002$	$.638 \pm .007$	$1.465 \pm .039$
	VQ_{data}	$.0022 \pm .0001$	$.4011 \pm .0327$	$\mathbf{.0008 \pm .0001}$	$.0020 \pm .0001$	$.6485 \pm .0043$	$1.422 \pm .040$
tissue	VQ	$\mathbf{.0083 \pm .0002}$	$.6984 \pm .0368$	$\mathbf{.0017 \pm .0004}$	$\mathbf{.0043 \pm .0002}$	$\mathbf{.624 \pm .002}$	$\mathbf{1.028 \pm .007}$
	$\text{VQ}_{\mathcal{N}}$	$.0342 \pm .0101$	$.8889 \pm .0925$	$.0041 \pm .0009$	$.0120 \pm .0014$	$.577 \pm .019$	$1.185 \pm .050$
	VQ_{data}	$.0093 \pm .0006$	$\mathbf{.5330 \pm .0577}$	$.0021 \pm .0007$	$.0043 \pm .0008$	$.616 \pm .003$	$1.056 \pm .008$

Table 17: LCE with different γ values with a `ResNet`.

Data	Method	$\gamma = 0.1$	$\gamma = 1.0$	$\gamma = 5.0$	$\gamma = 7.0$	$\gamma = 10.0$	$\gamma = 100.0$
cifar10	VQ	0.0316 ± 0.0004	0.0316 ± 0.0004	0.0192 ± 0.0002	$\mathbf{0.0104 \pm 0.0001}$	$\mathbf{0.0059 \pm 0.0002}$	0.0039 ± 0.0002
	SM	0.0316 ± 0.0005	0.0316 ± 0.0005	0.0284 ± 0.0003	0.0203 ± 0.0004	0.0102 ± 0.0003	$\mathbf{0.0037 \pm 0.0001}$
	DC	0.0305 ± 0.0005	0.0305 ± 0.0005	0.0274 ± 0.0004	0.0196 ± 0.0003	0.0099 ± 0.0003	$\mathbf{0.0037 \pm 0.0001}$
	KC	0.0339 ± 0.0002	0.0339 ± 0.0002	0.0313 ± 0.0004	0.0236 ± 0.0005	0.0095 ± 0.0004	0.0057 ± 0.0001
	LN	0.0345 ± 0.0003	0.0344 ± 0.0003	$\mathbf{0.0181 \pm 0.0006}$	0.0117 ± 0.0004	0.0079 ± 0.0003	0.0050 ± 0.0003
	PS	0.0387 ± 0.0004	0.0387 ± 0.0004	0.0342 ± 0.0003	0.0253 ± 0.0006	0.0180 ± 0.0005	0.0085 ± 0.0002
	IR	0.0317 ± 0.0004	0.0317 ± 0.0004	0.0286 ± 0.0002	0.0210 ± 0.0004	0.0115 ± 0.0003	0.0047 ± 0.0001
	TS	0.0317 ± 0.0023	0.0317 ± 0.0023	0.0289 ± 0.0015	0.0219 ± 0.0004	0.0127 ± 0.0006	0.0075 ± 0.0010
	PC	$\mathbf{0.0298 \pm 0.0004}$	$\mathbf{0.0298 \pm 0.0004}$	0.0273 ± 0.0003	0.0213 ± 0.0003	0.0133 ± 0.0005	0.0084 ± 0.0004
cifar100	VQ	0.0062 ± 0.0002	0.0062 ± 0.0002	$\mathbf{0.0034 \pm 0.0001}$	$\mathbf{0.0022 \pm 0.0001}$	$\mathbf{0.0017 \pm 0.0001}$	0.0012 ± 0.0001
	SM	0.0061 ± 0.0001	0.0061 ± 0.0001	0.0059 ± 0.0001	0.0052 ± 0.0001	0.0031 ± 0.0002	$\mathbf{0.0010 \pm 0.0000}$
	DC	$\mathbf{0.0056 \pm 0.0001}$	$\mathbf{0.0056 \pm 0.0001}$	0.0055 ± 0.0001	0.0048 ± 0.0001	0.0027 ± 0.0001	0.0014 ± 0.0001
	KC	0.0085 ± 0.0001	0.0085 ± 0.0001	0.0083 ± 0.0001	0.0072 ± 0.0001	0.0043 ± 0.0001	0.0030 ± 0.0001
	LN	0.0066 ± 0.0000	0.0066 ± 0.0000	0.0062 ± 0.0000	0.0046 ± 0.0000	0.0024 ± 0.0001	0.0012 ± 0.0000
	PS	0.0100 ± 0.0001	0.0100 ± 0.0001	0.0096 ± 0.0001	0.0084 ± 0.0002	0.0053 ± 0.0001	0.0028 ± 0.0000
	IR	0.0060 ± 0.0000	0.0060 ± 0.0000	0.0058 ± 0.0000	0.0051 ± 0.0001	0.0031 ± 0.0001	0.0012 ± 0.0000
	TS	0.0063 ± 0.0001	0.0063 ± 0.0001	0.0062 ± 0.0001	0.0055 ± 0.0001	0.0034 ± 0.0001	0.0016 ± 0.0001
	PC	0.0068 ± 0.0001	0.0068 ± 0.0001	0.0066 ± 0.0001	0.0059 ± 0.0001	0.0041 ± 0.0002	0.0021 ± 0.0001
tissue	VQ	$\mathbf{0.1242 \pm 0.0002}$	$\mathbf{0.1242 \pm 0.0002}$	$\mathbf{0.0488 \pm 0.0004}$	$\mathbf{0.0183 \pm 0.0003}$	$\mathbf{0.0083 \pm 0.0002}$	0.0044 ± 0.0004
	SM	0.1278 ± 0.0008	0.1278 ± 0.0008	0.1236 ± 0.0005	0.0847 ± 0.0019	0.0278 ± 0.0013	$\mathbf{0.0037 \pm 0.0002}$
	DC	0.1282 ± 0.0008	0.1282 ± 0.0008	0.1240 ± 0.0004	0.0851 ± 0.0019	0.0322 ± 0.0015	0.0045 ± 0.0003
	KC	0.1315 ± 0.0006	0.1315 ± 0.0006	0.0766 ± 0.0030	0.0313 ± 0.0008	0.0226 ± 0.0006	0.0195 ± 0.0004
	LN	0.1263 ± 0.0008	0.1263 ± 0.0008	0.0597 ± 0.0009	0.0241 ± 0.0009	0.0208 ± 0.0007	0.0200 ± 0.0015
	PS	0.1381 ± 0.0007	0.1381 ± 0.0007	0.1322 ± 0.0005	0.0889 ± 0.0018	0.0418 ± 0.0018	0.0133 ± 0.0012
	IR	0.1286 ± 0.0016	0.1286 ± 0.0016	0.1244 ± 0.0013	0.0871 ± 0.0022	0.0360 ± 0.0024	0.0096 ± 0.0005
	TS	0.1307 ± 0.0023	0.1307 ± 0.0023	0.1267 ± 0.0020	0.0907 ± 0.0022	0.0506 ± 0.0026	0.0407 ± 0.0036
	PC	0.1287 ± 0.0022	0.1287 ± 0.0022	0.1256 ± 0.0019	0.0985 ± 0.0020	0.0543 ± 0.0029	0.0390 ± 0.0023

datasets and for both LCE and $MLCE$, we observe a consistent trend. For very small bandwidths, neighborhoods contain few effective samples, placing the metrics in a highly local and high-variance regime in which differences between methods are less informative. VQ performs particularly well across the broad intermediate range of bandwidths that is most relevant in practice. Conversely, for very large bandwidths, neighborhoods become increasingly global, reducing the sensitivity of the metrics to fine-grained local calibration differences and causing performance gaps to narrow or occasionally reverse. Overall, the results indicate that the observed advantages of VQ are robust across a broad range of practically meaningful bandwidth choices.

Table 18: $MLCE$ with different γ values with a `ResNet`.

Data	Method	$\gamma = 0.1$	$\gamma = 1.0$	$\gamma = 5.0$	$\gamma = 7.0$	$\gamma = 10.0$	$\gamma = 100.0$
cifar10	VQ	0.9996 ± 0.0001	0.9996 ± 0.0001	$\mathbf{0.9215 \pm 0.0153}$	$\mathbf{0.7948 \pm 0.0178}$	$\mathbf{0.5595 \pm 0.0276}$	0.1434 ± 0.0067
	SM	0.9990 ± 0.0002	0.9990 ± 0.0002	0.9886 ± 0.0042	0.9336 ± 0.0164	0.8027 ± 0.0166	$\mathbf{0.1381 \pm 0.0138}$
	DC	0.9995 ± 0.0002	0.9995 ± 0.0002	0.9917 ± 0.0032	0.9438 ± 0.0136	0.8127 ± 0.0194	0.1469 ± 0.0075
	KC	0.9993 ± 0.0002	0.9993 ± 0.0002	0.9929 ± 0.0025	0.9470 ± 0.0104	0.7923 ± 0.0243	0.1405 ± 0.0094
	LN	0.9997 ± 0.0001	0.9997 ± 0.0001	0.9280 ± 0.0129	0.8253 ± 0.0179	0.6279 ± 0.0157	0.1457 ± 0.0138
	PS	$\mathbf{0.9970 \pm 0.0002}$	$\mathbf{0.9970 \pm 0.0002}$	0.9933 ± 0.0008	0.9620 ± 0.0122	0.8542 ± 0.0224	0.3557 ± 0.0201
	IR	0.9996 ± 0.0002	0.9996 ± 0.0002	0.9920 ± 0.0021	0.9406 ± 0.0163	0.7978 ± 0.0162	0.1609 ± 0.0185
	TS	0.9986 ± 0.0011	0.9986 ± 0.0011	0.9878 ± 0.0040	0.9387 ± 0.0144	0.8200 ± 0.0191	0.1914 ± 0.0227
	PC	0.9999 ± 0.0001	0.9999 ± 0.0001	0.9926 ± 0.0028	0.9439 ± 0.0171	0.8339 ± 0.0456	0.1852 ± 0.0286
cifar100	VQ	0.9994 ± 0.0001	0.9994 ± 0.0001	$\mathbf{0.8465 \pm 0.0063}$	$\mathbf{0.5972 \pm 0.0104}$	$\mathbf{0.2932 \pm 0.0099}$	0.1265 ± 0.0027
	SM	0.9989 ± 0.0001	0.9989 ± 0.0001	0.9964 ± 0.0003	0.9749 ± 0.0042	0.8121 ± 0.0191	0.1277 ± 0.0072
	DC	0.9999 ± 0.0000	0.9999 ± 0.0000	0.9991 ± 0.0003	0.9831 ± 0.0027	0.8113 ± 0.0121	$\mathbf{0.1252 \pm 0.0054}$
	KC	$\mathbf{0.9938 \pm 0.0009}$	$\mathbf{0.9938 \pm 0.0009}$	0.9899 ± 0.0006	0.9525 ± 0.0026	0.7575 ± 0.0101	0.1482 ± 0.0062
	LN	0.9983 ± 0.0001	0.9983 ± 0.0001	0.9905 ± 0.0005	0.9159 ± 0.0028	0.7022 ± 0.0070	0.1319 ± 0.0068
	PS	0.9976 ± 0.0001	0.9976 ± 0.0001	0.9969 ± 0.0001	0.9827 ± 0.0027	0.7031 ± 0.0409	0.2090 ± 0.0029
	IR	0.9997 ± 0.0001	0.9997 ± 0.0001	0.9981 ± 0.0004	0.9783 ± 0.0042	0.8049 ± 0.0116	0.1357 ± 0.0068
	TS	0.9990 ± 0.0001	0.9990 ± 0.0001	0.9966 ± 0.0004	0.9764 ± 0.0037	0.8239 ± 0.0173	0.1801 ± 0.0045
	PC	0.9994 ± 0.0001	0.9994 ± 0.0001	0.9978 ± 0.0005	0.9778 ± 0.0041	0.8187 ± 0.0127	0.1905 ± 0.0064
tissue	VQ	0.9991 ± 0.0001	0.9991 ± 0.0001	$\mathbf{0.9727 \pm 0.0042}$	$\mathbf{0.8825 \pm 0.0076}$	0.6984 ± 0.0368	$\mathbf{0.0533 \pm 0.0064}$
	SM	0.9993 ± 0.0002	0.9993 ± 0.0002	0.9989 ± 0.0004	0.9953 ± 0.0014	0.9609 ± 0.0106	0.0710 ± 0.0098
	DC	0.9992 ± 0.0003	0.9992 ± 0.0003	0.9988 ± 0.0004	0.9951 ± 0.0014	0.9706 ± 0.0098	0.0826 ± 0.0085
	KC	0.9967 ± 0.0005	0.9967 ± 0.0005	0.9842 ± 0.0034	0.9327 ± 0.0125	0.7296 ± 0.0570	0.1814 ± 0.0072
	LN	0.9992 ± 0.0001	0.9992 ± 0.0001	0.9812 ± 0.0021	0.9141 ± 0.0102	$\mathbf{0.6932 \pm 0.0661}$	0.1008 ± 0.0061
	PS	$\mathbf{0.9890 \pm 0.0012}$	$\mathbf{0.9890 \pm 0.0012}$	0.9882 ± 0.0010	0.9803 ± 0.0015	0.9518 ± 0.0045	0.1246 ± 0.0094
	IR	0.9931 ± 0.0005	0.9931 ± 0.0005	0.9929 ± 0.0004	0.9909 ± 0.0010	0.9602 ± 0.0103	0.1472 ± 0.0115
	TS	0.9993 ± 0.0001	0.9993 ± 0.0001	0.9990 ± 0.0001	0.9959 ± 0.0011	0.9822 ± 0.0084	0.1608 ± 0.0255
	PC	1.0000 ± 0.0001	1.0000 ± 0.0001	1.0000 ± 0.0001	0.9996 ± 0.0001	0.9768 ± 0.0133	0.2452 ± 0.0177

C.13 Training Time

We report the average training time for a single epoch for VQ with a `ResNet` backbone on all three datasets. For both `Cifar10` and `Cifar100`, the *quantization* head takes on average ≈ 1 second to complete an epoch. The *calibrator* also takes ≈ 1 second to complete a single epoch. For `tissue`, the *quantization* head takes on average ≈ 3 seconds to complete an epoch. The *calibrator* takes ≈ 2.5 seconds to complete a single epoch.

D Implementation Details

In this section we provide a comprehensive description of all the procedures underlying our empirical results. We followed the same setting as in Barbera et al. [11].

D.1 Training of Classifiers

Here we describe hyperparameters and the training procedure for the baseline classifiers.

For `CIFAR-10` and `tissue` we leverage a `ResNet-50` architecture with `IMAGENET1K_V2` pre-trained weights. We additionally add a dropout layer to the backbone with 0.2 rate and a linear classification head.

For `CIFAR-100` we instead rely on a `ResNet-152` model, still with `IMAGENET1K_V2` pre-trained weights. We again add a dropout layer to the backbone, but with 0.5 rate in this case.

All `ResNet` architectures are trained with early stopping on validation NLL, `Adam` optimizer and a learning rate of 3×10^{-4}.

For all datasets we employ a `ConvNext-tiny` architecture. We again add a dropout layer to the backbone with 0.2 rate.

For all datasets we also employ a `vit-base-patch16-224` from the timm library [55], still with `IMAGENET1K_V2` pre-trained weights.

Both `ConvNext` and `ViT` architectures are trained with early stopping on validation NLL, `Adam` optimizer and a learning rate of 5×10^{-5}.

Finally, for `Weather` we leverage an `FTT` classifier. The model is trained with early stopping on validation NLL, `Adam` optimizer and a learning rate of 1×10^{-4}.

All classifiers are trained with Categorical Cross-Entropy.

D.2 Local calibration maps

Quantized classifier. For all of our experiments we segment latent representations into $w = 64$ slots, yielding $d = 32$ per slot for `ResNet` architectures and $d = 12$ for `ConvNext`, and `ViT` architectures and we instantiate the codebook with $|\mathcal{C}| = 64$ prototype vectors. The codebook $\mathcal{C}$ is initialized with random samples drawn from the segmented representations of the classifier. During learning, updates are performed via EMA with a decay rate of 0.99 as in [21]. The new classification head $f_{\text{VQ}} : \mathcal{Q} \to \Delta^{|\mathcal{Y}|}$ is a linear layer followed by softmax that maps quantized representations to the $|\mathcal{Y}| - 1$ probability simplex and is trained with `Adam` and 1×10^{-3} learning rate and weight decay.

Calibration function. The *local* calibrator we propose requires two sets of parameter vectors $\mathcal{A} = \{\mathbf{a}_1, \dots, \mathbf{a}_{|\mathcal{C}|}\} \subset \mathbb{R}^{|\mathcal{Y}|}, \mathcal{B} = \{\mathbf{b}_1, \dots, \mathbf{b}_{|\mathcal{C}|}\} \subset \mathbb{R}^{|\mathcal{Y}|}$ and a third parameter vector $\boldsymbol{\sigma}^2 \in \mathbb{R}^w$. $\mathcal{A}$ and $\mathcal{B}$ are initialised such that for any induced submatrices $\mathbf{A}$, $\mathbf{B}$ (as in Section 3.2):

$$\phi(\mathbf{A}^\top \mathbf{B}) = \mathbf{1}\mathbf{1}^\top \tag{126}$$

where $\phi(\cdot)$ is the softplus and $\mathbf{1} \in \mathbb{R}^{|\mathcal{Y}|}$ is the all ones vector. The calibration parameters then are:

$$\boldsymbol{\alpha}^{(\mathcal{V})} - \mathbf{1}\mathbf{1}^\top = \Big(\phi\big(\mathbf{A}^\top \operatorname{diag}(\boldsymbol{\sigma}^2)\mathbf{B}\big) + \mathbf{I}\Big) - \mathbf{1}\mathbf{1}^\top \tag{127}$$

This modeling choice initializes each region's calibration map close to the identity, providing a strong starting point for optimization and improving convergence. Lastly, our main results are presented fixing $\boldsymbol{\sigma}^2 = \mathbf{1}$ (the all-ones vector) and do not learn it. We refrain from using the full calibrator model, as it introduces additional capacity that empirically increases the risk of overfitting. This is a well-known issue in post-hoc calibration, where simpler models and stronger regularization are often preferred [8]. We present results with a regularized variant of our approach that learns $\boldsymbol{\sigma}^2$ in Appendix C.9. Training leverages `Adam` [56] with learning rate and weight decay set to 1×10^{-3}.

D.3 Training of Local Methods

In what follows we illustrate the technical details regarding implementation of local calibration methods in our experiments.

For `Cifar10` we use a fully connected network with a single hidden layer of size 64 and a dropout rate of 0.3. We operate in a PCA-reduced feature representation space of size 50. We use `Adam` for training with learning rate 1×10^{-3}, for 22 epochs (early stopping) and a batch size of 1024. For `Cifar100` we leverage the same architecture and hyperparameters except for the hidden layer, which now has size 128, and the dropout rate, which is set to 0.5. Training lasts 30 epochs with early stopping. The same applies to `tissue`, but we set the hidden dimension to 256 and the dropout rate back to 0.3. Learning instead lasts 60 epochs with early stopping.

D.4 Metrics

We follow the implementation of [11]. More precisely, for both *global* and *local* metrics we partitioned $f_k(\mathbf{x})$ into 15 bins based on predicted confidence scores. Moreover, we picked a value of 10 for the kernel-bandwidth hyperparameter consistently with the learning procedure of the local methods.

D.5 Calibration Algorithm

In this section we provide a comprehensive description of the algorithm underlying our proposal. The method can be summarized in two stages: (i) a discretization step (*divide*) and (ii) a *calibration* (*calibra*) step.

Algorithm 1: Calibration algorithm

Input: EMA decay γ; learning rates $\eta_{\mathrm{vq}}, \eta_{\mathrm{cal}}$.
Output: Learned codebook $\mathcal{C}$ and calibrated predictor f_{cal}.

Stage 1: Quantization-aware representation learning
Initialize codebook $\mathcal{C}$ and quantization-aware head f_{VQ};
repeat
  Sample minibatch $\mathcal{D}_{\mathrm{mini}} \subset \mathcal{D}_{\mathrm{cal}}$;
  $\mathbf{z} \leftarrow E(x)$ for $(x, y) \in \mathcal{D}_{\mathrm{mini}}$;
  $(z^{(1)}, \ldots, z^{(w)}) \leftarrow \Phi(\mathbf{z})$;
  $\mathbf{s} \leftarrow \mathrm{Assign}((z^{(i)})_{i=1}^{w}, \mathcal{C})$;
  $\overline{\mathbf{q}} \leftarrow \mathrm{Select}(\mathbf{s}, \mathcal{C})$;
  $\hat{\mathbf{p}}_{\mathrm{VQ}} \leftarrow f_{\mathrm{VQ}}(\overline{\mathbf{q}})$;
  $\mathcal{L}_{\mathrm{ce}} \leftarrow \mathrm{CE}(\hat{\mathbf{p}}_{\mathrm{VQ}}, y)$;
  $\theta_{\mathrm{VQ}} \leftarrow \theta_{\mathrm{VQ}} - \eta_{\mathrm{vq}} \nabla_{\theta_{\mathrm{VQ}}} \mathcal{L}_{\mathrm{ce}}$;
  $\mathcal{C} \leftarrow \mathrm{EMAUpdate}(\mathcal{C}, (z^{(i)})_{i=1}^{w}, \mathbf{s}; \gamma)$;
until *convergence*;

Stage 2: Region-aware calibration
Freeze $\mathcal{C}$ and f_{VQ}. Initialize calibration parameters $(\mathcal{A}, \mathcal{B}, \boldsymbol{\sigma}^2)$ and $f_{\mathrm{cal}}(\hat{\mathbf{p}}_{\mathrm{VQ}}, \mathbf{s})$;
repeat
  Sample minibatch $\mathcal{D}_{\mathrm{mini}} \subset \mathcal{D}_{\mathrm{cal}}$;
  $\mathbf{A}^{(\mathcal{V})}, \mathbf{B}^{(\mathcal{V})} \leftarrow \mathrm{Select}(\mathbf{s}, \mathcal{A}), \mathrm{Select}(\mathbf{s}, \mathcal{B})$ for $(\hat{\mathbf{p}}_{\mathrm{VQ}}, \mathbf{s}) \in \mathcal{D}_{\mathrm{mini}}$;
  $\boldsymbol{\alpha}^{(\mathcal{V})} \leftarrow \phi\big(\mathbf{A}^{(\mathcal{V})\top} \mathrm{diag}(\boldsymbol{\sigma}^2) \mathbf{B}^{(\mathcal{V})}\big) + \mathbf{I}$;
  $\hat{\mathbf{p}}_{\mathrm{cal}} \leftarrow f_{\mathrm{cal}}(\hat{\mathbf{p}}_{\mathrm{VQ}})$;
  $\mathcal{L}_{\mathrm{cal}} \leftarrow \mathrm{CE}(\hat{\mathbf{p}}_{\mathrm{cal}}, y)$;
  if $\boldsymbol{\sigma}^2$ *is learnable* **then**
    $(\mathcal{A}, \mathcal{B}, \boldsymbol{\sigma}^2) \leftarrow (\mathcal{A}, \mathcal{B}, \boldsymbol{\sigma}^2) - \eta_{\mathrm{cal}} \nabla_{(\mathcal{A}, \mathcal{B}, \boldsymbol{\sigma}^2)} \mathcal{L}_{\mathrm{cal}}$
  else
    $(\mathcal{A}, \mathcal{B}) \leftarrow (\mathcal{A}, \mathcal{B}) - \eta_{\mathrm{cal}} \nabla_{(\mathcal{A}, \mathcal{B})} \mathcal{L}_{\mathrm{cal}}$
until *convergence*;
return $\mathcal{C}$, f_{cal}.

Learning procedure **Quantization-Aware Representation Learning.** The first step discretizes the continuous latent representation of the baseline classifier. Learning proceeds as follows:

- **line 2-3:** A latent encoding $\mathbf{z}$ is extracted from the frozen encoder. $\mathbf{z}$ is then segmented into w equal-sized slots $\mathbf{z}^{(i)}$.
- **line 4-5:** Each slot $z^{(i)}$ is discretized by nearest neighbor assignment with respect to the vectors in $\mathcal{C}$. $\mathrm{Assign}(\cdot, \cdot)$ yields the corresponding index sequence $\mathbf{s}$ over all slots. The sequence is used to extract the discrete representation $\overline{\mathbf{q}}$ from $\mathcal{C}$ via $\mathrm{Select}(\cdot, \cdot)$. This, for the i-th slot, takes the value from $\mathcal{C}$ at index $s^{(i)}$. These values are then concatenated to form $\overline{\mathbf{q}}$.
- **lines 6-9:** A new classification head takes as input $\overline{\mathbf{q}}$ and produces new scores. These values are used to compute the categorical cross entropy between predicted probabilities and ground truth. The weights of f_{VQ} are updated via gradient descent, whereas the codewords are updated via exponential moving average.

Training proceeds until convergence.

Region-Aware Calibration. This step takes as input the quantization head scores and the index sequence $\mathbf{s}$. Its goal is to produce new *locally* calibrated probabilities. Given scores $\mathbf{p}_{\mathrm{VQ}}$ and indices $\mathbf{s}$, the calibration head does the following:

- **line 1:** Produce the *receiver* basis $\mathbf{A}^{(\mathcal{V})}$ from $\mathcal{A}$ and the *sender* basis $\mathbf{B}^{(\mathcal{V})}$ from $\mathcal{B}$ via $\mathrm{Select}(\mathbf{s}, \mathcal{A})$ and $\mathrm{Select}(\mathbf{s}, \mathcal{B})$;
- **line 2-3:** Compute $\boldsymbol{\alpha}^{(\mathcal{V})}$ and produce new calibrated scores (Equation (8)).

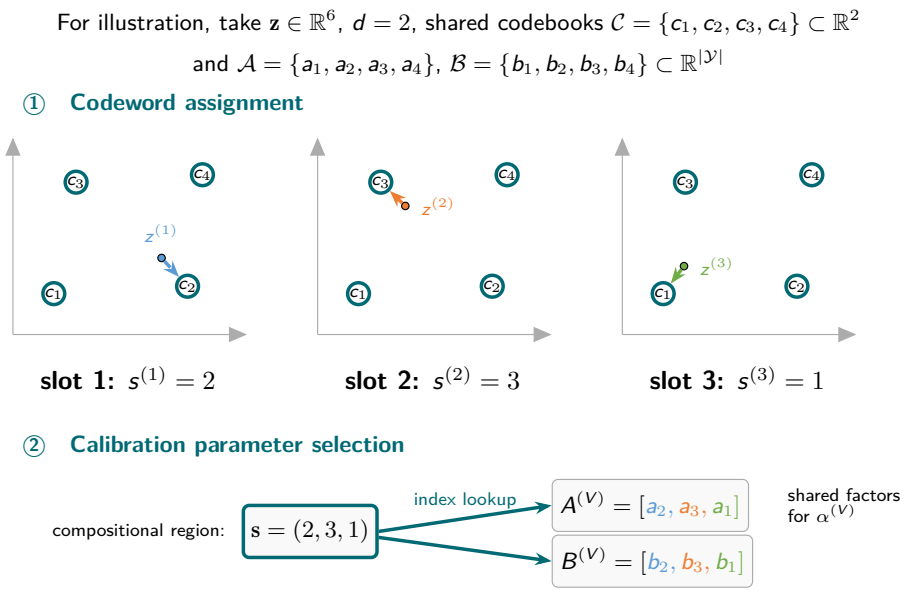


Figure 10: Illustration of the indexed parametrization trick. An instance $\mathbf{z} \in \mathbb{R}^6$ is first segmented in 2-dimensional slots $\mathbf{z}^{(1)}, \mathbf{z}^{(2)}, \mathbf{z}^{(3)}$. Each slot is assigned to the closest codeword $\mathbf{c}_j \in \mathcal{C}$. The resulting index sequence $\mathbf{s}$, identifies a unique Voronoi cell in $\mathbb{R}^6$ and is used to select parameters from $\mathcal{A}$ and $\mathcal{B}$. The matrices $A^{(\mathcal{V})}$, $B^{(\mathcal{V})}$ yield Dirichlet concentration parameters via Equation (10).

- **line 4-8:** Updated calibration parameters (potentially including $\text{diag}(\boldsymbol{\sigma}^2)$) via gradient descent on the categorical cross entropy until convergence.

The output is a codebook $\mathcal{C}$ defining the tessellation of the latent space and a calibrated predictor f_{cal}.

D.6 Hardware

To perform the experiments, we used three machines: (i) a 16-core machine with an AMD Ryzen 9 7950X CPU and 2 NVIDIA GeForce RTX 4090 GDDR6X with 24GB of memory, OS Ubuntu 22.04.4 LTS; (ii) a small cluster with 1 single master node and 2 nodes with 8 x GPU A100 80GB, OS Rocky Linux 9.6; (iii) the Leonardo supercomputer hosted by CINECA, using its Booster partition, whose compute nodes feature a 32-core Intel Xeon Platinum 8358 CPU, 512 GB of system RAM, and four NVIDIA A100 GPUs (64 GB each).

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The paper's main claims are well reflected by our theoretical (see Section 3) and empirical experiments (Section 4).

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: Please refer to Section 6 for an extended description of the paper limitations.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

   Answer: [Yes]

Justification: All the proofs for our theory in Section 3 are provided in Appendix A.

Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: We detail thoroughly all implementation details for base-models, methods and metrics in Appendix D.

   Guidelines:

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

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

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

Answer: [Yes]

Justification: The code is attached as a zip file and will be released upon acceptance.

Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: The details can be found in Appendix D.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [Yes]

   Justification: We provide all standard deviations in Tables and report either boxplots or confidence intervals at 95% in all the Figures.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.

- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: We provide hardware details in Appendix D.6.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: We follow the NeurIPS Code of Ethics.

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [Yes]

    Justification: We provide a dedicated Broader Impact section.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

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

    Answer: [N/A].

    Justification: Our proposal does not feature models that present risk of misuse.

    Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: We cite all the relevant information.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

    Answer: [N/A]

    Justification: We do not provide new assets.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: We do not involve humans.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: We do not involve humans.

    Guidelines:

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

16. **Declaration of LLM usage**

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

Answer: [N/A]

Justification: We used LLMs only for polishing the text.

Guidelines:

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