Dirichlet Calibration Goes Local

Cesare Barbera[◇,*,†] **Lorenzo Perini**[⋆] **Giovanni De Toni**[‡]

Andrea Passerini[*] **Andrea Pugnana**[◇,*]

[*]University of Trento [†]University of Pisa [⋆]Meta [‡]Fondazione Bruno Kessler

Abstract

Accurate and well-calibrated Machine Learning (ML) models are mandatory in high-stakes settings, yet effective calibration remains challenging: global approaches assume calibration errors are homogeneous across the space, while local methods often rely on latent-space dimensionality reduction, which leads to information loss. We address these issues by injecting a notion of locality in Dirichlet Calibration via Vector Quantization (VQ). We further introduce an efficient parametrization of the Dirichlet concentrations that prevents a combinatorial explosion of calibration parameters. Our approach allows us to learn 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 preserving global calibration and predictive performance.

1 INTRODUCTION

In complex industrial systems and automated logistics, Machine Learning (ML) models must be not only accurate but also well-calibrated (Wang, 2023; Sambyal et al., 2023). The strictest notion of calibration is *strong calibration* (Vaicenavicius et al., 2019), which requires the target class conditional distribution on any classifier prediction to match that prediction:

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

Workshop "Towards Trustworthy Predictions: Theory and Applications of Calibration for Modern AI" at AISTATS 2026, Tangier, Morocco.

In practice, satisfying this form of calibration poses significant challenges. Perez-Lebel et al. (2023) recently showed that whenever the classifier's decision boundary is complex, it can lead to poor calibration of the scores predicted for less likely instances. This phenomenon is due to (i) the *cancellation effect*, *i.e.*, miscalibration errors within a confidence group offset one another, and (ii) *proximity bias*, *i.e.*, disparities arise in calibration quality for instances in sparsely populated regions of the decision space. To address this problem, recent work has moved towards *local calibration*, which requires probability estimates to be reliable within specific regions of the input or latent space (Luo et al., 2022). However, local multiclass calibration has a critical statistical bottleneck: *data sparsity*. As the dimensionality of the space increases, points become more isolated, making it harder to estimate local corrections. Conversely, global calibration methods, such as Dirichlet Calibration (Kull et al., 2019), avoid data sparsity by applying a single correction map to all samples, but fail to correct *proximity bias*. This leads to a key trade-off: global methods are stable but biased, while local methods are flexible but have high variance.

In this work, we address the aforementioned challenge by considering 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. 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.

2 BACKGROUND

Let us consider a multi-class classification setting, where $\mathcal{X} \subseteq \mathbb{R}^m$ is the feature space and $\mathcal{Y} = \{0, \ldots, |\mathcal{Y}| - 1\}$ is a finite target space with $|\mathcal{Y}|$ distinct labels. We have access to a given dataset $D = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$ 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 $f\colon \mathcal{X} \to \Delta^{|\mathcal{Y}|}$, where $\Delta^{|\mathcal{Y}|}$ is the $(|\mathcal{Y}| - 1)$-dimensional probability simplex. In words, a probabilistic classifier maps an input $\mathbf{x}$ to a probability distribution over classes, *i.e.*, $f(\mathbf{x}) = \hat{\mathbf{p}} \in \Delta^{|\mathcal{Y}|}$, where each entry $\hat{\mathbf{p}}_k = f_k(\mathbf{x})$ of the predicted probability vector $\hat{\mathbf{p}}$ denotes the predicted probability of class k.

3 VORONOI TESSELATION

The representation space induced by modern neural networks is typically *high-dimensional* and *structured*, yet calibration behavior varies substantially across different regions of this space. 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 (*i.e.* 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 $m' = dw$ and: $\overline{\mathbf{z}} = \left[\mathbf{z}^{(1)}, \mathbf{z}^{(2)}, \ldots, \mathbf{z}^{(w)}\right]$

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 $s(i) = \operatorname{argmin}_{j \in \{1,\ldots,|\mathcal{C}|\}} \|\mathbf{z}^{(i)} - \mathbf{c}_j\|_2$. We define the full vector of quantization indices as $\mathbf{s} = (s(1), \ldots, s(w))$ and the fully quantized latent representation as $\overline{\mathbf{q}}_{\mathbf{s}} = [\mathbf{c}_{s(1)}, \ldots, \mathbf{c}_{s(w)}] \in \mathbb{R}^{w \times d}$. We denote the set of all possible flattened quantized configurations with:

$$\mathcal{Q} = \left\{\mathbf{q}_{\mathbf{s}} : (\mathbf{c}_{s(1)}, \ldots, \mathbf{c}_{s(w)}) \, \forall \, \mathbf{s} \in \{1, \ldots, |\mathcal{C}|\}^w\right\} \subset \mathbb{R}^{m'}. \tag{2}$$

Although this operation is performed locally at the level of individual slots, it induces a global partition of the full latent space into a combinatorially large number of regions, as the following proposition illustrates:

Proposition 1. *A global minimizer of the squared distance in the flattened space,* $\mathbf{q}^\star = \operatorname{argmin}_{\mathbf{q} \in \mathcal{Q}} \|\mathbf{z} - \mathbf{q}\|_2$ *can be obtained by assigning each slot independently to its nearest codebook vector:*

$$s^\star(i) = \operatorname*{argmin}_{j \in \{1,\ldots,|\mathcal{C}|\}} \|\mathbf{z}^{(i)} - \mathbf{c}_j\|_2 \qquad \mathbf{q}^\star = \mathbf{q}_{\mathbf{s}^\star} \tag{3}$$

Proof can be found in Appendix B.1. Each element $\mathbf{q} \in \mathcal{Q}$ defines a *Voronoi cell* centroid in the flattened feature space, namely: $\mathcal{V}(\mathbf{q}) = \big\{\, \mathbf{z} \in \mathbb{R}^{m'} : \|\mathbf{z} - \mathbf{q}\|_2 \leq \|\mathbf{z} - \mathbf{q}'\|_2 \quad \forall\, \mathbf{q}' \in \mathcal{Q} \,\big\}$. Note that these Voronoi cells form a partition of $\mathbb{R}^{m'}$ (see Fig. 1b), as:

$$\mathbb{R}^{m'} = \bigcup_{\mathbf{q} \in \mathcal{Q}} \mathcal{V}(\mathbf{q}), \qquad \mathcal{V}(\mathbf{q}) \cap \mathcal{V}(\mathbf{q}') = \varnothing \text{ for } \mathbf{q} \neq \mathbf{q}' \tag{4}$$

Moreover, we highlight that (i) standard per-slot nearest-neighbour 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 tesselation of Voronoi cells, each associated with a distinct quantized representation and, later, a potentially distinct calibration map.

4 HOW TO LOCALLY CALIBRATE

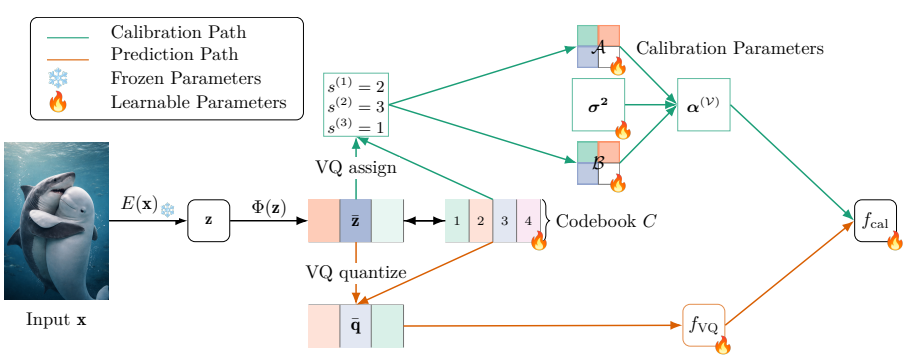
In this section, we formalize our calibration approach from a probabilistic perspective. We model the distribution of predicted probabilities conditioned on the true label and Voronoi region, and derive the corresponding calibrated posterior via Bayes' rule.

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:*

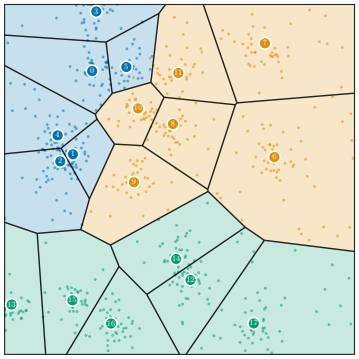
$$\mathbf{b}_{j,\mathcal{V}} = \log \pi_{j|\mathcal{V}} - \log B\big(\boldsymbol{\alpha}^{(j,\mathcal{V})}\big) \quad \mathbf{w}_{j,\mathcal{V}} = \boldsymbol{\alpha}^{(j,\mathcal{V})} - \mathbf{1} \tag{5}$$

and the logarithm of posterior probabilities is:

$$\log p(y = j \mid \hat{\mathbf{p}}, \mathcal{V}) = \mathbf{b}_{j,\mathcal{V}} + \mathbf{w}_{j,\mathcal{V}}^\top \log \hat{\mathbf{p}} \; + \; \text{const.} \tag{6}$$



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



(b) Voronoi Tesselation Example.

Figure 1: Fig. 1a depicts our approach; Fig. 1b shows how Voronoi tesselation works, assigning points to the closest centroid.

Proof is reported in Appendix B.2 Here, $B(\cdot)$ denotes the multivariate Beta function associated with the Dirichlet distribution, and $\pi_{j|\mathcal{V}}$ a possibly cell-dependent prior probability. This results in a linear calibration model where the per-cell parameter is determined by Dirichlet concentrations.

Modelling Dirichlet Concentrations per Voronoi cell. We now introduce a parametric model to represent the concentration of the Dirichlet distribution, conditioned on the Voronoi space. We construct a *bilinear map* that measures how confidence mass is redistributed between predicted and true classes, given a Voronoi region $\mathcal{V}$. Moreover, we introduce *latent miscalibration factors* that enable the bilinear map to represent phenomena by which confidence mass is redistributed across classes (*e.g.* systematic overconfidence or class-wise confusion).

First, consider two matrices $\mathbf{A}^{(\mathcal{V})}, \mathbf{B}^{(\mathcal{V})} \in \mathbb{R}^{w \times |\mathcal{Y}|}$, where w denotes the number of latent factors $\mathbf{h} = (\mathbf{h}_1, \dots, \mathbf{h}_w)^\top$ influencing miscalibration, for each Voronoi region $\mathcal{V}$. $\mathbf{B}^{(\mathcal{V})}$ defines the *sender basis*, describing how latent factors are expressed when the true class is j, and $\mathbf{A}^{(\mathcal{V})}$ defines the *receiver basis*, describing how the same factors are interpreted as evidence for each predicted class i. Within this framework, instance-level miscalibration can be viewed as arising from the *latent bilinear interaction* $\mathbf{A}^{(\mathcal{V})\top}\mathbf{h}\mathbf{h}^\top\mathbf{B}^{(\mathcal{V})}$.

We now make explicit the key consequence of vector quantization for calibration: instance-level interactions must be aggregated at the level of the Voronoi cell. More precisely, under VQ, each instance is assigned to $\mathbf{q}^\star$ (Proposition 1). Consequently, calibration parameters must be defined with respect to the centroid of $\mathcal{V}$. This implies that the region-level *latent bilinear interaction* is formed by aggregating the instance-level bilinear interactions over all instances assigned to $\mathcal{V}$, resulting in the conditional expectation:

$$\mathbb{E}\left[\mathbf{A}^{(\mathcal{V})\top}\mathbf{h}\mathbf{h}^\top\mathbf{B}^{(\mathcal{V})} \,\middle|\, \mathcal{V}\right] = \mathbf{A}^{(\mathcal{V})\top}\,\mathbb{E}[\mathbf{h}\mathbf{h}^\top \mid \mathcal{V}]\,\mathbf{B}^{(\mathcal{V})} \quad (7)$$

We summarize this variability under the assumption of region-invariant and diagonal second moments of the latent factors, *i.e.,* $\mathbb{E}[\mathbf{h}\mathbf{h}^\top \mid \mathcal{V}] = \mathrm{diag}(\boldsymbol{\sigma}^2)$. Intuitively, $\boldsymbol{\sigma}^2$ encodes the *global strength* of each latent factor, modulating its contribution uniformly across regions. This allows us to define local Dirichlet concentrations:

$$\boldsymbol{\alpha}^{(\mathcal{V})} := \phi\left(\mathbf{A}^{(\mathcal{V})\top}\mathrm{diag}(\boldsymbol{\sigma}^2)\mathbf{B}^{(\mathcal{V})}\right) \quad (8)$$

where $\boldsymbol{\alpha}^{(\mathcal{V})}$ is a $|\mathcal{Y}| \times |\mathcal{Y}|$ matrix whose (i,j)-th entry encodes evidence mass toward predicted class i when the true class is j, and ϕ is a positive, monotone link function applied element-wise to ensure positivity. Notably, Eq. (8) represents a lower-order latent interaction model that captures class-dependent redistribution of confidence. The matrix $\mathrm{diag}(\boldsymbol{\sigma}^2)$ fixes the latent coordinate system for the miscalibration factors, while expressivity is preserved through the region-specific $\mathbf{A}^{(\mathcal{V})}$ and $\mathbf{B}^{(\mathcal{V})}$.

However, naively learning a separate set of parameters for every region is *computationally infeasible*: the number of Voronoi cells grows on the order of $|\mathcal{C}|^w$. In the next section, we introduce a parameter-efficient learning scheme that avoids this exponential blow-up while remaining empirically effective.

5 LEARNING MAPS

We propose a parameter-efficient way to quantise the encoder space and to associate each Voronoi centroid with a corresponding pair of *sender–receiver* maps.

Let $\mathcal{C} = \{\mathbf{c}_1, \dots, \mathbf{c}_{|\mathcal{C}|}\} \subset \mathbb{R}^d$ be the *embedding codebook*, and $\mathcal{A} = \{\mathbf{a}_1, \dots, \mathbf{a}_{|\mathcal{C}|}\} \subset \mathbb{R}^{|\mathcal{Y}|}$, $\mathcal{B} =$

Table 1: Results for our approach (highlighted in blue) vs baselines.

Dataset	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
	LN	.0079 ± .0003	.6279 ± .0157	.0017 ± .0001	.0048 ± .0003	.888 ± .002	.347 ± .002
	DC	.0099 ± .0003	.8133 ± .0195	**.0011 ± .0001**	**.0037 ± .0001**	.890 ± .002	**.332 ± .007**
	KC	.0095 ± .0005	.7923 ± .0269	.0015 ± .0003	.0054 ± .0004	**.893 ± .001**	.340 ± .003
	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
	NC	.0154 ± .0005	.8738 ± .0238	.0065 ± .0004	.0153 ± .0006	.884 ± .001	.494 ± .018
cifar100	VQ	**.0017 ± .0001**	**.2932 ± .0099**	.0008 ± .0001	**.0018 ± .0001**	.681 ± .001	1.208 ± .007
	LN	.0024 ± .0001	.7022 ± .0070	.0007 ± .0001	.0018 ± .0001	.688 ± .001	**1.125 ± .002**
	DC	.0027 ± .0001	.8117 ± .0121	**.0007 ± .0001**	.0019 ± .0001	**.690 ± .002**	1.154 ± .009
	KC	.0043 ± .0001	.7594 ± .0055	.0013 ± .0001	.0044 ± .0002	.679 ± .004	1.351 ± .007
	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
	NC	.0032 ± .0001	.8250 ± .0176	.0017 ± .0001	.0039 ± .0002	.670 ± .002	1.502 ± .036
tissue	VQ	**.0088 ± .0004**	**.5760 ± .0349**	.0016 ± .0003	**.0043 ± .0003**	.618 ± .002	1.042 ± .004
	LN	.0144 ± .0012	.7293 ± .0359	.0050 ± .0011	.0100 ± .0015	.630 ± .001	**1.012 ± .003**
	DC	.0276 ± .0013	.9638 ± .0104	.0021 ± .0005	.0062 ± .0007	.617 ± .003	1.052 ± .009
	KC	.0165 ± .0008	.7179 ± .0250	.0043 ± .0006	.0126 ± .0010	**.632 ± .001**	1.021 ± .002
	PS	.0418 ± .0018	.9521 ± .0043	**.0013 ± .0001**	.0135 ± .0012	.603 ± .008	1.180 ± .008
	IR	.0360 ± .0024	.9605 ± .0101	.0026 ± .0004	.0095 ± .0005	.603 ± .008	1.096 ± .016
	TS	.0413 ± .0034	.9620 ± .0107	.0127 ± .0025	.0196 ± .0036	.603 ± .008	1.112 ± .023
	NC	.0768 ± .0019	.9695 ± .0127	.0308 ± .0016	.0725 ± .0020	.603 ± .008	2.100 ± .101

$\{\mathbf{b}_1, \ldots, \mathbf{b}_{|\mathcal{C}|}\} \subset \mathbb{R}^{|\mathcal{Y}|}$ the corresponding *receiver* and *sender* calibration codebooks, respectively. All three codebooks share the same discrete indexing set $\{1, \ldots, |\mathcal{C}|\}$. We also introduce the second moments $\boldsymbol{\sigma}^2 = (\sigma_1^2, \ldots, \sigma_w^2)$. 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 $\mathcal{V}$. Crucially, the same index sequence is reused to select calibration parameters from $\mathcal{A}$ and $\mathcal{B}$. This induces local 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 *receiver* matrix $\mathbf{A}^{(\mathcal{V})}$ and *sender* matrix $\mathbf{B}^{(\mathcal{V})}$ jointly define the Dirichlet concentration parameters through the bilinear form:

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

This modelling approach requires only $2 \times |\mathcal{C}| \times |\mathcal{Y}| + w$ parameters, thus providing a cheap yet expressive representation of class-conditional evidence geometry. In Appendix C we analyse the statistical stability induced by this modelling choice.

Learning procedure. We propose a two-stage procedure (further details in Appendix F.5) 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 similarly to Van Den Oord et al. (2017).
2. *Region-Aware Calibration:* We freeze the codebook and quantized classifier, then learn the calibration parameters $(\mathcal{A}, \mathcal{B}, \boldsymbol{\sigma}^2)$.

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

6 EXPERIMENTAL EVALUATION

Methods. We evaluate our approach (VQ) against the following baselines: Temperature Scaling (TS) (Guo et al., 2017), Isotonic Regression (IR) (Zadrozny and Elkan, 2002), and Platt Scaling (PS) (Platt et al., 1999). In addition, we compare with Dirichlet Calibration (DC) (Kull et al., 2019), non-parametric methods such as KCal (KC) (Lin et al., 2023) and local approaches such as Local Nets (LN) (Barbera et al., 2025). Finally, we include the uncalibrated base model (NC). We report ablation studies in Appendix D.

Datasets. We consider three datasets, *i.e.*, `cifar10`, `cifar100` (Krizhevsky et al., 2009), and real world medical data `tissuemnist` from the `MedMNIST` collection (Yang et al., 2023).

Metrics. We evaluate local calibration (LCE and $MLCE$) and global calibration ($ECCE$ and ECE). We capture the models predictive performance using accuracy (ACC) and negative log-likelihood (NLL). All metrics are discussed in detail in Appendix E.

Results. As shown in Table 1, VQ outperforms all existing methods in terms of local calibration metrics, while retaining comparable results on global calibration metrics and predictive performance.

7 CONCLUSIONS

We proposed a novel local calibration approach based on Vector Quantization. Empirical evidence validates the effectiveness of our approach.

References

Baldeschi, R. C., Di Gregorio, S., Fioravanti, S., Fusco, F., Guy, I., Haimovich, D., Leonardi, S., Linder, F., Perini, L., Russo, M., et al. (2025). Multicalibration yields better matchings. *arXiv preprint arXiv:2511.11413*.

Barbera, C., Perini, L., Toni, G. D., Passerini, A., and Pugnana, A. (2025). Multiclass local calibration with the jensen-shannon distance. *CoRR*, abs/2510.26566.

Ding, Z., Han, X., Liu, P., and Niethammer, M. (2021). Local temperature scaling for probability calibration. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6889–6899.

Globus-Harris, I., Gupta, V., Jung, C., Kearns, M., Morgenstern, J., and Roth, A. (2023). Multicalibrated regression for downstream fairness. In *Proceedings of the 2023 AAAI/ACM Conference on AI, Ethics, and Society*, pages 259–286.

Guo, C., Pleiss, G., Sun, Y., and Weinberger, K. Q. (2017). On calibration of modern neural networks. In *ICML*, volume 70 of *Proceedings of Machine Learning Research*, pages 1321–1330. PMLR.

Hébert-Johnson, U., Kim, M., Reingold, O., and Rothblum, G. (2018). Multicalibration: Calibration for the (computationally-identifiable) masses. In *International Conference on Machine Learning*, pages 1939–1948. PMLR.

Ibarra, I. A., Gujral, P., Tannen, J., Tygert, M., and Xu, C. (2022). Metrics of calibration for probabilistic predictions. *J. Mach. Learn. Res.*, 23:351:1–351:54.

Jin, H. H., Ding, Z., Ngo, D. D., and Wu, Z. S. (2025). Discretization-free multicalibration through loss minimization over tree ensembles. *arXiv preprint arXiv:2505.17435*.

Jung, C., Lee, C., Pai, M., Roth, A., and Vohra, R. (2021). Moment multicalibration for uncertainty estimation. In *Conference on Learning Theory*, pages 2634–2678. PMLR.

Kingma, D. P. and Ba, J. (2015). Adam: A method for stochastic optimization. In *ICLR (Poster)*.

Krizhevsky, A., Hinton, G., et al. (2009). Learning multiple layers of features from tiny images.

Kull, M., Perelló-Nieto, M., Kängsepp, M., de Menezes e Silva Filho, T., Song, H., and Flach, P. A. (2019). Beyond temperature scaling: Obtaining well-calibrated multi-class probabilities with dirichlet calibration. In *NeurIPS*, pages 12295–12305.

Kull, M., Silva Filho, T., and Flach, P. (2017). Beta calibration: a well-founded and easily implemented improvement on logistic calibration for binary classifiers. In *Artificial intelligence and statistics*, pages 623–631. PMLR.

Lin, Z., Trivedi, S., and Sun, J. (2022). Taking a step back with kcal: Multi-class kernel-based calibration for deep neural networks. *arXiv preprint arXiv:2202.07679*.

Lin, Z., Trivedi, S., and Sun, J. (2023). Taking a step back with kcal: Multi-class kernel-based calibration for deep neural networks. In *ICLR*. OpenReview.net.

Luo, R., Bhatnagar, A., Bai, Y., Zhao, S., Wang, H., Xiong, C., Savarese, S., Ermon, S., Schmerling, E., and Pavone, M. (2022). Local calibration: metrics and recalibration. In *UAI*, volume 180 of *Proceedings of Machine Learning Research*, pages 1286–1295. PMLR.

Naeini, M. P., Cooper, G. F., and Hauskrecht, M. (2015). Obtaining well calibrated probabilities using bayesian binning. In *AAAI*, pages 2901–2907. AAAI Press.

Noarov, G. and Roth, A. (2023). The statistical scope of multicalibration. In *International Conference on Machine Learning*, pages 26283–26310. PMLR.

Perez-Lebel, A., Morvan, M. L., and Varoquaux, G. (2023). Beyond calibration: estimating the grouping loss of modern neural networks. In *ICLR*. OpenReview.net.

Perini, L., Haimovich, D., Linder, F., Tax, N., Karamshuk, D., Vojnovic, M., Okati, N., and Apostolopoulos, P. A. (2025). Mcgrad: Multicalibration at web scale. *arXiv preprint arXiv:2509.19884*.

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

Sambyal, A. S., Niyaz, U., Krishnan, N. C., and Bathula, D. R. (2023). Understanding calibration

of deep neural networks for medical image classification. *Comput. Methods Programs Biomed.*, 242:107816.

Vaicenavicius, J., Widmann, D., Andersson, C. R., Lindsten, F., Roll, J., and Schön, T. B. (2019). Evaluating model calibration in classification. In *AISTATS*, volume 89 of *Proceedings of Machine Learning Research*, pages 3459–3467. PMLR.

Valk, K. and Kull, M. (2023). Assuming locally equal calibration errors for non-parametric multiclass calibration. *Transactions on Machine Learning Research*.

Van Den Oord, A., Vinyals, O., et al. (2017). Neural discrete representation learning. *Advances in neural information processing systems*, 30.

Van der Vaart, A. W. (2000). *Asymptotic statistics*, volume 3. Cambridge university press.

Wang, C. (2023). Calibration in deep learning: A survey of the state-of-the-art. *CoRR*, abs/2308.01222.

White, H. (1982). Maximum likelihood estimation of misspecified models. *Econometrica: Journal of the econometric society*, pages 1–25.

Yang, J., Shi, R., Wei, D., Liu, Z., Zhao, L., Ke, B., Pfister, H., and Ni, B. (2023). Medmnist v2-a large-scale lightweight benchmark for 2d and 3d biomedical image classification. *Scientific Data*, 10(1):41.

Zadrozny, B. and Elkan, C. (2001). Obtaining calibrated probability estimates from decision trees and naive bayesian classifiers. In *ICML*, pages 609–616. Morgan Kaufmann.

Zadrozny, B. and Elkan, C. (2002). Transforming classifier scores into accurate multiclass probability estimates. In *KDD*, pages 694–699. ACM.

Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [**Yes**] We clearly state the settings and assumptions of our method in Section **??**.
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [**Not Applicable**] We do not study these properties in our work.
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [**Yes**] The code to reproduce our results can be found at `https://anonymous.4open.science/r/local-calibration-25A3`.

2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. [**Yes**] All our theorems and definitions in Sections 4, 3 and Appendix C clearly state assumptions.
   (b) Complete proofs of all theoretical results. [**Yes**] The proofs can be found in Appendix B, C
   (c) Clear explanations of any assumptions. [Yes] See Appendix C.1.

3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [**Yes**] The code to reproduce our results can be found at `https://anonymous.4open.science/r/local-calibration-25A3`.
   (b) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [**Yes**] We provide results using boxplots and $avg \pm std$.
   (c) All the training details (e.g., data splits, hyperparameters, how they were chosen). [**Yes**] See Appendix F.
   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [**Yes**] We provide this information in Appendix (Section Experimental Details).

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
   (a) Citations of the creator If your work uses existing assets. [**Yes**] We cite all the datasets owners.
   (b) The license information of the assets, if applicable. [**Not Applicable**]
   (c) New assets either in the supplemental material or as a URL, if applicable. [**Not Applicable**]
   (d) Information about consent from data providers/curators. [**Not Applicable**]
   (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [**Not Applicable**]

5. If you used crowdsourcing or conducted research with human subjects, check if you include:

(a) The full text of instructions given to participants and screenshots. [**Not Applicable**]

(b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [**Not Applicable**]

(c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [**Not Applicable**]

A Related Work

Global Calibration. Standard post-hoc calibration involves learning a single, global map that transforms the uncalibrated outputs of a model into calibrated probabilities. For binary classification, parametric methods such as *Platt Scaling* (Platt et al., 1999), *Temperature Scaling* (Guo et al., 2017), and *Beta Calibration* (Kull et al., 2017) are widely used. These methods generally learn a monotonic function to rescale the logits (or probabilities), preserving the rank ordering of classes. Non-parametric approaches, such as *Isotonic Regression* (Zadrozny and Elkan, 2002) and *Histogram Binning* (Zadrozny and Elkan, 2001), 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) (Kull et al., 2019) generalizes the Beta calibration framework to the probability simplex, allowing for interactions between classes.

Local Calibration. To address the limitations of global maps, *local calibration* methods make the correction function dependent on the input features or embedding unlike global post-hoc methods applied to model scores. Approaches like *Local Temperature Scaling* (Ding et al., 2021), *KCal* (Lin et al., 2022) and *LECE* (Valk and Kull, 2023) typically employ kernel density estimation or nearest-neighbor lookups to estimate local error rates. More recently, *Local Nets* (Barbera et al., 2025) 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 us to pool statistics within learned regions (codewords), ensuring that the local calibration maps are statistically stable even in high-dimensional settings.

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 identifyiable by a specific hypothesis class (Hébert-Johnson et al., 2018; Jung et al., 2021; Baldeschi et al., 2025; Jin et al., 2025; Perini et al., 2025). Thus, the goal of multicalibration approaches is to ensure that ML models do *not* make biased predictions on any of these sub-populations (Globus-Harris et al., 2023; Noarov and Roth, 2023).

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. In contrast, our approach operates in the high-dimensional representation space of neural networks, where dimensions are abstract and "auditing" for subgroups via standard multicalibration techniques is computationally intractable.

B Proofs

B.1 Proof of Proposition 1

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 $\mathbf{s}$,

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

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. □

B.2 Proof of Proposition 2

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:

$$\hat{\mathbf{p}} \mid (y = j,\, \mathcal{V}) \;\sim\; \mathrm{Dir}\big(\boldsymbol{\alpha}^{(j,\mathcal{V})}\big), \tag{11}$$

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

$$p\big(\hat{\mathbf{p}} \mid y = j, \mathcal{V}\big) = \frac{1}{B\big(\boldsymbol{\alpha}^{(j,\mathcal{V})}\big)} \prod_{i=1}^{|\mathcal{Y}|} \hat{\mathbf{p}}_i^{\boldsymbol{\alpha}_i^{(j,\mathcal{V})}-1}, \tag{12}$$

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

$$\begin{aligned} p(y = j \mid \hat{\mathbf{p}}, \mathcal{V}) &\propto \pi_{j|\mathcal{V}}\, p\big(\hat{\mathbf{p}} \mid y = j, \mathcal{V}\big) \\ &= \pi_{j|\mathcal{V}}\, \frac{1}{B\big(\boldsymbol{\alpha}^{(j,\mathcal{V})}\big)} \prod_{i=1}^{|\mathcal{Y}|} \hat{\mathbf{p}}_i^{\boldsymbol{\alpha}_i^{(j,\mathcal{V})}-1}. \end{aligned} \tag{13}$$

Log-linear calibration form. Taking logs in (13) gives

$$\log p(y = j \mid \hat{\mathbf{p}}, \mathcal{V}) = \log \pi_{j|\mathcal{V}} - \log B\big(\boldsymbol{\alpha}^{(j,\mathcal{V})}\big) + \sum_{i=1}^{|\mathcal{Y}|} \big(\boldsymbol{\alpha}_i^{(j,\mathcal{V})} - 1\big) \log \hat{\mathbf{p}}_i + \text{const}, \tag{14}$$

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

$$\mathbf{b}_{j,\mathcal{V}} = \log \pi_{j|\mathcal{V}} - \log B\big(\boldsymbol{\alpha}^{(j,\mathcal{V})}\big) \quad \mathbf{w}_{j,\mathcal{V}} = \boldsymbol{\alpha}^{(j,\mathcal{V})} - \mathbf{1} \tag{15}$$

where $\mathbf{1} \in \mathbb{R}^{|\mathcal{Y}|}$ is the all-ones vector. Then, (14) simplifies:

$$\log p(y = j \mid \hat{\mathbf{p}}, \mathcal{V}) = \mathbf{b}_{j,\mathcal{V}} + \mathbf{w}_{j,\mathcal{V}}^{\top} \log \hat{\mathbf{p}} \;+\; \text{const}. \tag{16}$$

□

C Discussion on Statistical Stability

The key insight of our approach is the following. While individual Voronoi cells are sparse, 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, conditional on the optimization trajectory entering a well-behaved neighbourhood of a stationary point of the loss. First, we show that, despite the compositional structure, the estimation procedure is locally statistically consistent. This result relies on standard regularity conditions, including (A1) *smoothness*, (A2) *uniform convergence*, (A3) *local curvature*, and (A4) *approximate stationarity*. A detailed description is provided in the Appendix C.1.

Theorem 1 (Local Consistency)**.** *Let $\mathcal{L}_N(\theta)$ be the empirical cross-entropy loss and Θ_{local} be a compact convex neighbourhood of a population local minimizer $\theta^\star$ (the pseudo-true parameter). Let $\hat{\theta}_N \in \Theta_{local}$ be an approximate stationary point satisfying $\|\nabla \mathcal{L}_N(\hat{\theta}_N)\|_2 \le \varepsilon_N$, where $\varepsilon_N \xrightarrow{p} 0$. Under the regularity assumptions (A1)-(A4), the estimator converges in probability to the population minimizer:*

$$\hat{\theta}_N \xrightarrow{p} \theta^\star.$$

Proof provided in Appendix C.2.

While consistency ensures correctness in the limit, it does not describe behaviour under data sparsity. If the following additional assumptions[1] (discussed in the Appendix C.1) - *i.e.,* (A5) weak *cross-block coupling*, (A6)

[1]These assumptions capture the mechanism by which parameter sharing induced by VQ can stabilize optimization empirically.

bounded conditional influence between codewords, and (A7) *standard moment* and *concentration conditions* - hold, then the convergence rate of the calibration parameters depends on the frequency of codeword usage, not on the density within $\mathcal{V}$.

Theorem 2 (Frequency-Weighted Convergence of Codeword Parameters)**.** *Assume (A1)-(A7) and condition on the global scale parameter $\boldsymbol{\sigma}^2$ and on the (fixed) codeword assignments induced by quantization. For any active codebook parameter $\pi_k \in \{\mathcal{A} \cup \mathcal{B}\}$ with effective occurrence count $N_k = \Theta_p(N)$ (poitwise in k), we have:*

$$\|\pi_k - \pi_k^\star\|_2 = O_p\Big(N_k^{-1/2}\Big) + O_p(\varepsilon_N) + o_p(1)$$

Proof provided in Appendix C.3. The result is pointwise in k, meaning that the stated rate applies to each active codeword individually and does not imply uniform convergence across the entire codebook. Moreover, in regimes where optimization error is controlled ($\varepsilon_N = o_p(N_k^{-1/2})$), the error is dominated by statistical variability up to vanishing higher order terms. This result provides theoretical insight into why we can learn reliable calibration maps for rare, isolated regions (rare combinations of indices) as long as their constituent codewords are observed frequently enough elsewhere.

C.1 Assumptions for Theorem 1 and Theorem 2

In the following we discuss the underlying assumptions our statistical result. Assumptions (A1)–(A4) are standard local regularity conditions for M-estimators and Z-estimators e.g., (Van der Vaart, 2000; White, 1982). 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. Such assumptions are routinely invoked in local asymptotic analyses of nonconvex empirical risk minimization, where consistency and rates are established conditional on convergence to a neighborhood of a well-behaved stationary point.

Assumptions (A5)–(A6) impose a weak blockwise coupling condition through Schur dominance of the population Hessian. These conditions ensure that each codeword block π_k is locally identifiable given the remaining parameters, and that cross-block curvature does not dominate the intrinsic curvature associated with π_k. 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.

Finally, Assumption (A7) collects mild moment and dependence conditions needed to control empirical gradients and Hessians via standard concentration and law-of-large-numbers arguments.

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. More precisely, we assume there exists a neighborhood $\mathcal{N}(\theta^\star)$ and a convex compact set $\Theta_{\text{local}} \subset \mathcal{N}(\theta^\star)$ with $\theta^\star \in \Theta_{\text{local}}$ 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.** $\sup_{\theta\in\Theta_{\text{local}}} |\mathcal{L}_N(\theta) - \mathcal{L}(\theta)| \xrightarrow{p} 0$, $\sup_{\theta\in\Theta_{\text{local}}} \big\|\nabla\mathcal{L}_N(\theta) - \nabla\mathcal{L}(\theta)\big\|_2 \xrightarrow{p} 0$ and $\sup_{\theta\in\Theta_{\text{local}}} \big\|\nabla^2\mathcal{L}_N(\theta) - \nabla^2\mathcal{L}(\theta)\big\|_{\text{op}} \xrightarrow{p} 0$.

(A3) **Local curvature.** $\theta^\star$ is the unique minimizer of $\mathcal{L}$ on Θ_{local}, and $\nabla^2\mathcal{L}(\theta^\star)$ is positive definite. In particular, there exists $\underline{\lambda} > 0$ such that

$$\lambda_{\min}\big(\nabla^2\mathcal{L}(\theta)\big) \geq \underline{\lambda} \qquad \forall\, \theta \in \Theta_{\text{local}}. \tag{17}$$

Hence, for all large N, $\nabla^2\mathcal{L}_N(\theta)$ is invertible on Θ_{local} with $\sup_{\theta\in\Theta_{\text{local}}} \|\nabla^2\mathcal{L}_N(\theta)^{-1}\|_{\text{op}} = O_p(1)$.

Remark. In our bilinear calibration parameterization, the unregularized loss can be invariant under the rescaling $(A, B) \mapsto (tA, B/t)$, which violates strict identifiability and can introduce a flat direction (zero curvature). In practice this is easily addressed by adding a small quadratic regularizer (weight decay), or equivalently by fixing a normalization constraint, which restores local strong convexity for the resulting objective in the neighborhood of the optimum.

(A4) **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. $\to 1$.

(A5) **Blockwise Schur dominance (weak coupling).** For each codeword block π_k, let S_k denote the population Schur complement

$$S_k := \nabla^2_{\pi_k \pi_k} \mathcal{L}(\theta^\star) - \nabla^2_{\pi_k, -k} \mathcal{L}(\theta^\star) \big[\nabla^2_{-k,-k} \mathcal{L}(\theta^\star)\big]^{-1} \nabla^2_{-k, \pi_k} \mathcal{L}(\theta^\star). \tag{18}$$

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

$$S_k \succeq c\, \nabla^2_{\pi_k \pi_k} \mathcal{L}(\theta^\star). \tag{19}$$

It guarantees that the effective curvature along π_k remains non-degenerate even after accounting for interactions with other parameters.

(A6) **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 exists $M \in \mathbb{R}^+$:

$$\sup_k \Big\| \bar{H}_{k,-k} \big[\nabla^2_{-k,-k} \mathcal{L}(\theta^\star)\big]^{-1} \Big\|_{\text{op}} < M < \infty. \tag{20}$$

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.

(A7) **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 (π_k, π_k) block and let $H^{(k)}_{n,i,-k}(\theta^\star)$ denote the corresponding per-occurrence contribution to the (π_k, θ_{-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 possibly data dependent but fixed during calibration codeword assignment (which is also independent of N), 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).

C.2 Proof of Theorem 1

Statement: Let $\mathcal{L}_N(\theta)$ be the empirical cross-entropy loss and Θ_{local} be a compact convex neighbourhood 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 regularity assumptions (A1)-(A4), the estimator converges in probability to the population minimizer:

$$\hat{\theta}_N \xrightarrow{p} \theta^\star.$$

Proof. In the following proof, $\|\cdot\|_2$ denotes the Euclidean norm for vectors, and $\|\mathbf{A}\|_{\text{op}} := \sup_{\|x\|_2 = 1} \|\mathbf{A}x\|_2$ denotes the induced operator (spectral) norm for matrices.

Let $\Delta_N := \hat{\theta}_N - \theta^\star$. By a first-order Taylor expansion of the empirical gradient around $\theta^\star$, there exists a point $\tilde{\theta}_N \in \Theta_{\text{local}}$ on the line segment joining $\hat{\theta}_N$ and $\theta^\star$ such that

$$\nabla \mathcal{L}_N(\hat{\theta}_N) = \nabla \mathcal{L}_N(\theta^\star) + \nabla^2 \mathcal{L}_N(\tilde{\theta}_N)\, \Delta_N. \tag{21}$$

Rearranging (21) yields the exact identity

$$\Delta_N = \left[\nabla^2 \mathcal{L}_N(\tilde{\theta}_N)\right]^{-1} \left(\nabla \mathcal{L}_N(\hat{\theta}_N) - \nabla \mathcal{L}_N(\theta^\star)\right) = \\ = -\left[\nabla^2 \mathcal{L}_N(\tilde{\theta}_N)\right]^{-1} \nabla \mathcal{L}_N(\theta^\star) + \left[\nabla^2 \mathcal{L}_N(\tilde{\theta}_N)\right]^{-1} \nabla \mathcal{L}_N(\hat{\theta}_N) \quad (22)$$

By (A3), $\nabla \mathcal{L}(\theta^\star) = 0$. Combined with (A2):

$$\|\nabla \mathcal{L}_N(\theta^\star)\|_2 = \|\nabla \mathcal{L}_N(\theta^\star) - \nabla \mathcal{L}(\theta^\star)\|_2 \leq \sup_{\theta \in \Theta_{\text{local}}} \|\nabla \mathcal{L}_N(\theta) - \nabla \mathcal{L}(\theta)\|_2 = o_p(1) \quad (23)$$

By (A3) and uniform Hessian convergence (A2):

$$\left\|\nabla^2 \mathcal{L}_N(\tilde{\theta}_N)^{-1}\right\|_2 \leq \sup_{\theta \in \Theta_{\text{local}}} \left\|\nabla^2 \mathcal{L}_N(\theta)^{-1}\right\|_2 \leq O_p(1), \quad (24)$$

Combining:

$$\|\Delta_N\|_2 \leq O_p(1) o_p(1) + O_p(1)\varepsilon_N = o_p(1) \quad (25)$$

this proves $\hat{\theta}_N \to_p \theta^\star$.

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

$$\mathcal{L}(\theta) = \mathbb{E}_{P_{\text{true}}}[-\log p_\theta(\mathbf{y} \mid \mathbf{x})]\,. \quad (26)$$

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

$$\mathbb{E}[-\log p_\theta(\mathbf{y} \mid \mathbf{x})] = H(P_{\text{true}}(\cdot \mid \mathbf{x})) + D_{\text{KL}}(P_{\text{true}}(\cdot \mid \mathbf{x}) \,\|\, p_\theta(\cdot \mid \mathbf{x}))\,, \quad (27)$$

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

$$\mathcal{L}(\theta) = \text{const} + \mathbb{E}_X[D_{\text{KL}}(P_{\text{true}}(\cdot \mid \mathbf{x}) \,\|\, p_\theta(\cdot \mid \mathbf{x}))]\,. \quad (28)$$

Hence $\theta^\star$ minimizes the (conditional) Kullback–Leibler divergence within Θ_{local}, i.e. it is the pseudo-true parameter (White, 1982).

□

C.3 Proof of Theorem 2

Statement: Under assumptions (A1)-(A7) condition on the global scale parameter $\boldsymbol{\sigma}^2$ and on the (fixed) codeword assignments induced by quantization. For any active codebook parameter $\pi_k \in \{\mathcal{A} \cup \mathcal{B}\}$ with effective occurrence count $N_k = \Theta_p(N)$ (pointwise in k), we have:

$$\|\pi_k - \pi_k^\star\|_2 = O_p\left(N_k^{-1/2}\right) + O_p(\varepsilon_N) + o_p(1)$$

Proof. In the following proof, $\|\cdot\|_2$ denotes the Euclidean norm for vectors, and $\|\mathbf{A}\|_{\text{op}} := \sup_{\|x\|_2=1} \|\mathbf{A}x\|_2$ denotes the induced operator (spectral) norm for matrices.

Step 1: Replacing $\tilde{\theta}_N$ by $\theta^\star$. Building on Eq. (21):

$$\nabla^2 \mathcal{L}(\tilde{\theta})\Delta_N = \nabla \mathcal{L}(\hat{\theta}_N) - \nabla \mathcal{L}(\theta^\star) = \nabla \mathcal{L}(\hat{\theta}_N) - \nabla \mathcal{L}(\theta^\star) + \nabla^2 \mathcal{L}(\theta^\star)\Delta_N - \nabla^2 \mathcal{L}(\theta^\star)\Delta_N, \quad (29)$$

where we obtain the second equality by adding and subtracting $\nabla^2 \mathcal{L}(\theta^\star)\Delta_N$. By rearranging, we obtain that

$$\nabla^2 \mathcal{L}(\theta^\star)\Delta_N = \nabla^2 \mathcal{L}(\tilde{\theta})\Delta_N + (\nabla^2 \mathcal{L}(\theta^\star)\Delta_N - \nabla^2 \mathcal{L}(\tilde{\theta})\Delta_N) = -\nabla \mathcal{L}(\theta^\star) + \nabla \mathcal{L}(\hat{\theta}_N) + (\nabla^2 \mathcal{L}(\theta^\star) - \nabla^2 \mathcal{L}(\tilde{\theta}))\Delta_N \quad (30)$$

Let us call $r_N := \nabla \mathcal{L}(\hat{\theta}_N) + (\nabla^2 \mathcal{L}(\theta^\star) - \nabla^2 \mathcal{L}(\tilde{\theta}))\Delta_N$.

By (A4) we have $\|\nabla\mathcal{L}(\hat{\theta}_N)\|_2 \leq \varepsilon_N$. By Theorem 1, $\Delta_N = o_p(1)$ and since $\tilde{\theta}_N$ lies on the segment between $\hat{\theta}_N$ and $\theta^\star$, also $\tilde{\theta}_N \to_p \theta^\star$. Using (A1)–(A2), it follows that $\|\nabla^2\mathcal{L}(\theta^\star) - \nabla^2\mathcal{L}(\tilde{\theta})\|_2 = o_p(1)$, and therefore

$$\|(\nabla^2\mathcal{L}(\theta^\star) - \nabla^2\mathcal{L}(\tilde{\theta}))\Delta_N\|_2 \leq \|\nabla^2\mathcal{L}(\theta^\star) - \nabla^2\mathcal{L}(\tilde{\theta})\|_2\|\Delta_N\|_2 = o_p(1). \tag{31}$$

Hence,

$$\|r_N\|_2 \leq \|\nabla\mathcal{L}(\hat{\theta}_N)\|_2 + \|(\nabla^2\mathcal{L}(\theta^\star) - \nabla^2\mathcal{L}(\tilde{\theta}))\Delta_N\|_2 \leq \varepsilon_N + o_p(1). \tag{32}$$

Step2: Codeword-frequency scaling. Consider the linearized system at $\theta^\star$ with optimization residual

$$\nabla^2\mathcal{L}_N(\theta^\star)\,\Delta_N = -\nabla\mathcal{L}_N(\theta^\star) + r_N, \qquad \Delta_N := \hat{\theta}_N - \theta^\star, \qquad \|r_N\|_2 \leq \varepsilon_N + o_p(1). \tag{33}$$

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 $\theta^\star = (\pi_k, \theta_{-k})$ and write the gradient, residual, and Hessian in block form:

$$\Delta_N = \begin{pmatrix}\Delta_k \\ \Delta_{-k}\end{pmatrix}, \qquad \nabla\mathcal{L}_N(\theta^\star) = \begin{pmatrix}g_k \\ g_{-k}\end{pmatrix}, \qquad r_N = \begin{pmatrix}r_k \\ r_{-k}\end{pmatrix}, \qquad \nabla^2\mathcal{L}_N(\theta^\star) = \begin{pmatrix}H_{kk} & H_{k,-k} \\ H_{-k,k} & H_{-k,-k}\end{pmatrix}, \tag{34}$$

where $g_k := \nabla_{\pi_k}\mathcal{L}_N(\theta^\star)$ and $H_{kk} := \nabla^2_{\pi_k\pi_k}\mathcal{L}_N(\theta^\star)$. Equation (33) is equivalent to

$$H_{kk}\Delta_k + H_{k,-k}\Delta_{-k} = -g_k + r_k, \tag{35}$$

$$H_{-k,k}\Delta_k + H_{-k,-k}\Delta_{-k} = -g_{-k} + r_{-k}. \tag{36}$$

Under (A3) and (A2), for large N the principal submatrix $H_{-k,-k}$ is also invertible w.p. $\to 1$ and we solve (36) for Δ_{-k}:

$$\Delta_{-k} = -H^{-1}_{-k,-k}g_{-k} + H^{-1}_{-k,-k}r_{-k} - H^{-1}_{-k,-k}H_{-k,k}\Delta_k. \tag{37}$$

Substituting (37) into (35) yields

$$\Big(H_{kk} - H_{k,-k}H^{-1}_{-k,-k}H_{-k,k}\Big)\Delta_k = -g_k + H_{k,-k}H^{-1}_{-k,-k}g_{-k} + r_k - H_{k,-k}H^{-1}_{-k,-k}r_{-k}. \tag{38}$$

Define the empirical Schur complement and leakage term

$$S_{k,N} := H_{kk} - H_{k,-k}H^{-1}_{-k,-k}H_{-k,k}, \qquad R_{k,N} := H_{k,-k}H^{-1}_{-k,-k}g_{-k}. \tag{39}$$

Then

$$\Delta_k = -S^{-1}_{k,N}\Big(g_k - R_{k,N}\Big) + S^{-1}_{k,N}\Big(r_k - H_{k,-k}H^{-1}_{-k,-k}r_{-k}\Big). \tag{40}$$

Where 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 (A5).

Step 3: 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 $|\mathcal{I}_k| = N_k$, N number of samples and w slots. The empirical gradient block with respect to π_k admits the decomposition

$$\nabla_{\pi_k}\mathcal{L}_N(\theta^\star) = \frac{1}{N}\sum_{n=1}^{N}\sum_{i=1}^{w}\mathbf{1}\{s_n(i) = k\}g^{(k)}_{n,i}(\theta^\star), \tag{41}$$

By (A7) and a standard Chebyshev bound for empirical averages,

$$\|\nabla_{\pi_k}\mathcal{L}_N(\theta^\star)\|_2 = O_p\left(\frac{\sqrt{N_k}}{N}\right). \tag{42}$$

Step 4: Hessian scaling. Similarly, the corresponding Hessian block satisfies

$$\nabla^2_{\pi_k\pi_k}\mathcal{L}_N(\theta^\star) = \frac{1}{N}\sum_{n=1}^{N}\sum_{i=1}^{w}\mathbf{1}\{s_n(i) = k\}H^{(k)}_{n,i}(\theta^\star), \tag{43}$$

with $\bar{H}_k := \mathbb{E}[H_{n,i}^{(k)}(\theta^\star) \mid s_n(i) = k]$, by the law of large numbers (as $N \to \infty$), the empirical Hessian is:

$$\nabla^2_{\pi_k \pi_k} \mathcal{L}_N(\theta^\star) = \frac{N_k}{N}\Big(\bar{H}_k + o_p(1)\Big), \tag{44}$$

and for any active codeword k with $N_k = \Theta_p(N)$, the corresponding population Hessian block satisfies:

$$\nabla^2_{\pi_k \pi_k} \mathcal{L}(\theta^\star) = w \cdot P(s = k)\, \bar{H}_k. \tag{45}$$

Since $\nabla^2 \mathcal{L}(\theta^\star)$ is positive definite by Assumption (A3), every principal submatrix is positive definite, and therefore $\bar{H}_k \succ 0$ for all active codewords and:

$$\left[\nabla^2_{\pi_k \pi_k} \mathcal{L}_N(\theta^\star)\right]^{-1} = \frac{N}{N_k}\, \bar{H}_k^{-1} + o_p\left(\frac{N}{N_k}\right). \tag{46}$$

Step 5: Express in terms of the Schur complement. Consider again the linearized system induced by the Hessian at $\theta^\star$:

$$\nabla^2 \mathcal{L}_N(\theta^\star)\, \Delta_N = -\nabla \mathcal{L}_N(\theta^\star) + r_N, \qquad \|r_N\|_2 \le \varepsilon_N + o_p(1). \tag{47}$$

Taking the π_k-block and solving via the Schur complement gives (40):

$$\pi_k - \pi_k^\star = -S_{k,N}^{-1}\Big(\nabla_{\pi_k} \mathcal{L}_N(\theta^\star) - R_{k,N}\Big) + S_{k,N}^{-1}\Big(r_k - H_{k,-k} H_{-k,-k}^{-1} r_{-k}\Big), \tag{48}$$

we take norms and apply the triangle inequality:

$$\|\pi_k - \pi_k^\star\|_2 \le \|S_{k,N}^{-1}\|_2 \left(\|\nabla_{\pi_k} \mathcal{L}_N(\theta^\star)\|_2 + \|R_{k,N}\|_2\right) + \|S_{k,N}^{-1}\|\Big\|\Big(r_k - H_{k,-k} H_{-k,-k}^{-1} r_{-k}\Big)\Big\|_2 \tag{49}$$

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

$$S_k := \nabla^2_{\pi_k \pi_k} \mathcal{L}(\theta^\star) - \nabla^2_{\pi_k, -k} \mathcal{L}(\theta^\star) \left[\nabla^2_{-k,-k} \mathcal{L}(\theta^\star)\right]^{-1} \nabla^2_{-k, \pi_k} \mathcal{L}(\theta^\star) \tag{50}$$

satisfies

$$S_k \succeq c\, H_{kk}, \qquad H_{kk} := \nabla^2_{\pi_k \pi_k} \mathcal{L}(\theta^\star). \tag{51}$$

Then S_k is positive definite and:

$$\|S_k^{-1}\|_2 \le \frac{1}{c}\, \|H_{kk}^{-1}\|_2. \tag{52}$$

Assumption A(2) implies $S_{k,N} \to_p S_k$ and $H_{kk,N} \to_p H_{kk}$ in operator norm with probability tending to one. Assumption A(3) implies $H_{-k,-k}$ is invertible hence $H_{kk,N}$ also is in probability. Therefore, for N large enough, the inequality holds,

$$\|S_{k,N}^{-1}\|_2 \le \frac{2}{c}\, \|H_{kk,N}^{-1}\|_2, \qquad \text{and hence} \qquad \|S_{k,N}^{-1}\|_2 = O_p\Big(\|H_{kk,N}^{-1}\|_2\Big). \tag{53}$$

Using (46), we have

$$\|H_{kk,N}^{-1}\|_2 = \left\| \frac{N}{N_k}\, \bar{H}_k^{-1} + o_p\left(\frac{N}{N_k}\right) \right\|_2, \qquad \text{and therefore} \qquad \|S_{k,N}^{-1}\|_2 = O_p\left(\frac{N}{N_k}\right). \tag{54}$$

Step 7: Controlling the gradient and leakage terms. By (42), the intrinsic gradient contribution satisfies

$$\|\nabla_{\pi_k} \mathcal{L}_N(\theta^\star)\|_2 = O_p\left(\frac{\sqrt{N_k}}{N}\right). \tag{55}$$

We now bound the leakage term

$$R_{k,N} = H_{k,-k,N}\, H_{-k,-k,N}^{-1}\, \nabla_{\theta_{-k}} \mathcal{L}_N(\theta^\star). \tag{56}$$

The empirical cross-block Hessian admits the decomposition

$$H_{k,-k,N} = \frac{1}{N} \sum_{(n,i)\in\mathcal{I}_k} H^{(k)}_{n,i,-k}(\theta^\star) = \frac{N_k}{N}\big(\bar{H}_{k,-k} + o_p(1)\big), \tag{57}$$

where the convergence follows from a law of large numbers (A7) conditional on $s_n(i) = k$. By (A6) and boundedness of $\|\nabla^2_{-k,-k}\mathcal{L}(\theta^\star)\|_{\mathrm{op}}$, we have $\|\bar{H}_{k,-k}\|_{\mathrm{op}} = O(1)$ uniformly in k, and therefore $\|H_{k,-k,N}\|_{\mathrm{op}} = O_p(N_k/N)$. By (A3), for large N $\|H^{-1}_{-k,-k,N}\|_{\mathrm{op}} = O_p(1)$. Moreover, since $\theta^\star$ minimizes the population risk, $\nabla_{\theta_{-k}}\mathcal{L}_N(\theta^\star)$ is a mean-zero empirical average with finite second moments (with $\nabla\mathcal{L}(\theta^\star) = 0$ and exchangeability by (A1)), and hence

$$\|\nabla_{\theta_{-k}}\mathcal{L}_N(\theta^\star)\|_2 = O_p\left(\frac{1}{\sqrt{N}}\right). \tag{58}$$

Combining these bounds yields

$$\|R_{k,N}\|_2 \le \|H_{k,-k,N}\|_{\mathrm{op}}\,\|H^{-1}_{-k,-k,N}\|_{\mathrm{op}}\,\|\nabla_{\theta_{-k}}\mathcal{L}_N(\theta^\star)\|_2 = O_p\left(\frac{N_k}{N\sqrt{N}}\right). \tag{59}$$

Step 8: Controlling the residuals. Note that $\|r_N\|_2 \le \varepsilon_N + o_p(1)$ and $\|H_{k,-k,N}H^{-1}_{-k,-k,N}\|_2 = O_p(1)$ by Assumption A(6). Hence:

$$\left\| r_{k,N} - H_{k,-k,N}H^{-1}_{-k,-k,N}r_{-k,N} \right\|_2 = O_p(\varepsilon_N) + o_p(1) \tag{60}$$

Therefore,

$$\left\| S^{-1}_{k,N}\big(r_{k,N} - H_{k,-k,N}H^{-1}_{-k,-k,N}r_{-k,N}\big) \right\|_2 = O_p\Big(\|S^{-1}_{k,N}\|_2\,(\varepsilon_N + o_p(1))\Big) = O_p\left(\frac{N}{N_k}\varepsilon_N\right) + o_P(1) \tag{61}$$

Since $N_k = \Theta_p(N)$ (pointwise in k) and w is fixed, there exists $c_1(k) > 0$ such that $N_k \ge c_1(k)N$ w.p. $\to 1$, hence $Nw/N_k = O_p(1)$. Thus:

$$O_p\left(\frac{N}{N_k}\varepsilon_N\right) = O_p(\varepsilon_N). \tag{62}$$

Step 9: Combining the bounds. Substituting (54), (55), (59) and (62) into (49) yields

$$\begin{aligned}\|\pi_k - \pi^\star_k\|_2 &\le O_p\left(\frac{N}{N_k}\right)\left[O_p\left(\frac{\sqrt{N_k}}{N}\right) + O_p\left(\frac{N_k}{N\sqrt{N}}\right)\right] + O_p\left(\varepsilon_N\right) + o_p(1)\\ &= O_p\left(\frac{1}{\sqrt{N_k}}\right) + O_p\left(\frac{1}{\sqrt{N}}\right) + O_p\left(\varepsilon_N\right) + o_p(1)\end{aligned}$$

Since $N_k = \Theta_p(N)$ (pointwise in k), there exists $c_2(k) > 0$ such that $N_k \le c_2(k)N$ w.p. $\to 1$, and hence $\sqrt{N} \le \sqrt{c_2(k)}\,\sqrt{N_k}$ w.p. $\to 1$. Therefore $O_p(N^{-1/2}) = O_p(N_k^{-1/2})$. Combining:

$$\|\pi_k - \pi^\star_k\|_2 \le O_p\left(\frac{1}{\sqrt{N_k}}\right) + O_p\left(\varepsilon_N\right) + o_p(1) \tag{63}$$

which concludes the codeword-frequency scaling claim. □

D Additional Experiments

D.1 Ablation of VQ components

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 2 shows the results.

Table 2: Ablation results for VQ (highlighted in blue) when evaluating variants of our pipeline components.

Dataset	Method	$LCE \downarrow$	$MLCE \downarrow$	$ECCE \downarrow$	$ACC \uparrow$
cifar10	VQ	**.0059 ± .0002**	.5595 ± .0276	.0013 ± .0002	**.889 ± .001**
	VQ-DC	.0062 ± .0003	**.5534 ± .0311**	.0014 ± .0004	.889 ± .001
	VQ-NC	.0065 ± .0002	.5586 ± .0253	.0022 ± .0003	**.889 ± .001**
	DC	.0104 ± .0004	.8052 ± .0145	**.0008 ± .0001**	.884 ± .001
	NC	.0150 ± .0007	.9485 ± .0101	.0065 ± .0004	.884 ± .001
cifar100	VQ	**.0017 ± .0001**	**.2932 ± .0099**	.0008 ± .0001	**.681 ± .001**
	VQ-DC	.0020 ± .0001	.3050 ± .0110	.0012 ± .0001	.674 ± .004
	VQ-NC	.0021 ± .0001	.3199 ± .0074	.0013 ± .0001	**.681 ± .001**
	DC	.0030 ± .0001	.8248 ± .0161	**.0007 ± .0000**	.670 ± .002
	NC	.0032 ± .0001	.9155 ± .0104	.0017 ± .0001	.670 ± .002
tissue	VQ	.0088 ± .0004	**.5760 ± .0349**	.0016 ± .0003	.618 ± .002
	VQ-DC	**.0087 ± .0005**	.6631 ± .0401	.0023 ± .0005	.624 ± .001
	VQ-NC	.0112 ± .0016	.6557 ± .0649	.0049 ± .0017	**.625 ± .001**
	DC	.0284 ± .0013	.9613 ± .0097	**.0014 ± .0001**	.603 ± .008
	NC	.0739 ± .0018	.9741 ± .0086	.0308 ± .0016	.603 ± .008

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 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 behaviour while preserving global coherence. Regarding ACC, we can see that all VQ-based approaches slightly improve compared to NC and DC.

D.2 Ablating the Number of Slots in the Codebook 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 3 by varying $w \in \{16, 32, 64 \text{ (default)}, 128, 256\}$ for $|\mathcal{C}| \in \{32, 64\}$, and in Table 4 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 a too coarse 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 $w \in \{32, 64\}$, yielding the best or comparable results. Finally, Table 4 indicates that the effect of $|\mathcal{C}|$ is secondary once w is fixed: once again we observe that the best results are achieved at $| \mathcal{C} | \in \{32, 64\}$.

E Metrics

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

Global metrics. Expected Calibration Error (ECE) (Naeini et al., 2015) 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:

$$ECE = \sum_{b=1}^{m_b} \frac{|B_b|}{n} \left|\text{acc}(B_b) - \text{conf}(B_b)\right| \tag{64}$$

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

Table 3: VQ results when changing number of slots (w) and codebook size ($|\ C\ |$). Blue line reports main-paper parameters.

Dataset	w	$\|\mathcal{C}\|$	$LCE \downarrow$	$MLCE \downarrow$	$ECCE \downarrow$	$ECE \downarrow$	$ACC \uparrow$	$NLL \downarrow$
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 ± .0001**	**.5595 ± .0260**	.0013 ± .0002	.0037 ± .0002	.889 ± .001	.348 ± .001
	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 ± .0094	.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	.0172 ± .0009	.8997 ± .0198	.0011 ± .0004	.0044 ± .0004	.584 ± .004	1.124 ± .005
	16	64	.0144 ± .0007	.8174 ± .0421	.0009 ± .0001	.0047 ± .0004	.599 ± .003	1.102 ± .005
	32	32	.0116 ± .0003	.5976 ± .0540	.0009 ± .0002	.0036 ± .0002	.618 ± .001	1.049 ± .002
	32	64	.0097 ± .0003	.5774 ± .0356	.0010 ± .0001	.0035 ± .0001	.619 ± .001	1.046 ± .001
	64	32	**.0081 ± .0003**	**.4465 ± .0173**	**.0009 ± .0002**	**.0034 ± .0003**	.623 ± .001	1.032 ± .001
	64	64	.0088 ± .0003	.5760 ± .0329	.0016 ± .0003	.0043 ± .0003	.618 ± .002	1.042 ± .003
	128	32	.0094 ± .0005	.7147 ± .0320	.0012 ± .0005	.0040 ± .0003	.627 ± .001	1.019 ± .002
	128	64	.0099 ± .0002	.7893 ± .0195	.0010 ± .0002	.0036 ± .0004	.629 ± .000	1.014 ± .001
	256	32	.0130 ± .0002	.8448 ± .0053	.0014 ± .0002	.0044 ± .0002	.632 ± .001	1.008 ± .002
	256	64	.0146 ± .0003	.8857 ± .0098	.0012 ± .0003	.0041 ± .0006	**.632 ± .001**	**1.004 ± .001**

extends to Class-wise ECE (Kull et al., 2019), which averages per-class ECEs:

$$\textit{Class-wise ECE} = \frac{1}{|\mathcal{Y}|}\sum_{c=1}^{|\mathcal{Y}|} ECE_c \tag{65}$$

A more stable metric is Expected Cumulative Calibration Error (ECCE) (Ibarra et al., 2022) which instead aggregates calibration errors cumulatively across bins, providing a more robust global assessment:

$$\text{ECCE}_c = \sum_{b=1}^{m_b}\left|\sum_{i=1}^{b}\frac{|B_{i,c}|}{n}\left(\text{freq}_c(B_i) - \text{conf}_c(B_i)\right)\right| \tag{66}$$

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

Local metrics. Regarding local calibration metrics, we consider the multiclass extension of *Local Calibration Error* (Luo et al., 2022) by (Barbera et al., 2025):

$$LCE = \frac{1}{|\mathcal{Y}|}\sum_{b=1}^{m_b}\frac{1}{n}\sum_{i\in B_b}\left\|\frac{\sum_{j\in B_b}\left(\hat{\mathbf{p}}_j - \mathbf{y}_j\right)k_\gamma(\mathbf{x}_i,\mathbf{x}_j)}{\sum_{j\in B_b}k_\gamma(\mathbf{x}_i,\mathbf{x}_j)}\right\|_1 \tag{67}$$

where $||\cdot||_1$ is an appropriate 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 $\mathbf{x}_i$ to its individual LCE score. In practice, this metric captures the differences in the predicted probabilities and the corresponding ground truths for neighbours of an anchor point $\mathbf{x}_i$.

Moreover, we also consider the *Maximum Local Calibration Error*

$$MLCE = \max_{i\in D}\left\|\frac{\sum_{j\in B_b}\left(\hat{\mathbf{p}}_j - \mathbf{y}_j\right)k_\gamma(\mathbf{x}_i,\mathbf{x}_j)}{\sum_{j\in B_b}k_\gamma(\mathbf{x}_i,\mathbf{x}_j)}\right\|_1 \tag{68}$$

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

Table 4: VQ results when changing codebook size (| C |) for a fixed number of slots ($w = 64$). Blue line is main-paper parameters.

Dataset	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 ± .0003	.5680 ± .0193	**.0008 ± .0002**	**.0034 ± .0003**	.621 ± .001	1.038 ± .002
	64	32	**.0081 ± .0003**	**.4465 ± .0184**	.0009 ± .0002	.0034 ± .0003	.623 ± .001	1.032 ± .001
	64	64	.0088 ± .0004	.5760 ± .0349	.0016 ± .0003	.0043 ± .0003	.618 ± .002	1.042 ± .004
	64	128	.0087 ± .0003	.7409 ± .0237	.0012 ± .0004	.0039 ± .0005	.625 ± .001	1.026 ± .004
	64	256	.0092 ± .0002	.7032 ± .0176	.0010 ± .0003	.0037 ± .0006	**.626 ± .001**	**1.024 ± .002**

F 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. (2025) .

F.1 Training of Classifiers

Here we describe hyper-parameters and the training procedure for the baseline classifiers.

For `CIFAR-10` we leverage a `ResNet-50` architecture with `IMAGENET1K_V2` pre-trained weights. We additionally add a dropout layer to the backcone with 0.2 rate and a linear classification head. The model is trained for 9 epochs using the `Adam` optimizer (Kingma and Ba, 2015) with a learning rate of 3×10^{-4}. For `CIFAR-100` we instead rely on a `ResNet-152` model, still with `IMAGENET1K_V2` pre-trained weights. We again a dropout layer to the backbone, but with 0.5 rate in this case. Learning lasts 9 epochs with `Adam` and a learning rate of 3×10^{-4}. Finally, we fine-tune a `ResNet-50` architecture initialized with `IMAGENET1K_V2` weights on `TissueMNIST`. We again insert a dropout layer with a rate 0.2. After 10 epochs with `Adam` optimizer and a learning rate of 3×10^{-4} training concludes. All classifiers are trained with Categorical Cross-Entropy.

F.2 Local Dirichlet Calibration

Quantized classifier. For all of our experiments we segment latent representations in $w = 64$ slots, leaving $d = 32$, and we instantiate the codebook with $|\mathcal{C}| = 64$ prototype vectors. The codebook $\mathcal{C}$ is initialized with random samples of the segmented representations of the classifier. During learning, updates are performed via EMA with a weight decay of 0.99. The new classification head $f_{\text{VQ}} : \mathcal{Q} \to \mathcal{Y}$ is a linear layer that maps quantized representations to logits and is trained with Adam and $1 \times 10^{-}3$ for both learning rate and weight decay.

Calibration function. The *local* adaptation of Dirichlet Calibration we propose requires two sets of parameters vectors $\mathcal{A} = \{\mathbf{a}_1, \ldots, \mathbf{a}_{|\mathcal{C}|}\} \subset \mathbb{R}^{|\mathcal{Y}|}, \mathcal{B} = \{\mathbf{b}_1, \ldots, \mathbf{b}_{|\mathcal{C}|}\} \subset \mathbb{R}^{|\mathcal{Y}|}$ and a third parameters vector $\boldsymbol{\sigma}^2 \in \mathbb{R}^w$. $\mathcal{A}$ and $\mathcal{B}$ are initialised such that for any submatrices $\mathbf{A}$, $\mathbf{B}$:

$$\phi(A^\top B) = \mathbf{1} \tag{69}$$

where $\phi(\cdot)$ is the softplus and $\mathbf{1}$ is a matrix of all ones. The calibration parameters $\boldsymbol{\alpha}^{(\mathcal{V})} - \mathbf{1}$ are then obtained as:

$$\boldsymbol{\alpha}^{(\mathcal{V})} := \phi(A^\top B) - \mathbf{1} + \mathbf{I} \tag{70}$$

where $\mathbf{I}$ is the identity matrix. This modelling choice, combined with random standard normal initialisation of $\boldsymbol{\sigma}^2$, initializes each region's calibration map close to the identity, providing a strong starting point for optimization and improving convergence. The bias parameters are instead obtained learning a global parameter $\pi_{j|\mathcal{V}}$ under the assumption $\pi_{j|\mathcal{V}} \approx \pi_j$. This choice does not hinder model capacity as locality is introduce in the bias term via $\log B(\alpha^{(j,\mathcal{V})})$.

Algorithm 1 Calibration algorithm

Input: Frozen encoder $E(\cdot)$; segmentation map $\Phi(\cdot)$; dataset $\mathcal{D}_{\text{cal}}$; EMA decay γ; learning rates $\eta_{\text{vq}}, \eta_{\text{cal}}$.
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 $\mathcal{B} \subset \mathcal{D}_{\text{cal}}$
  $z \leftarrow E(x)$ for $(x, y) \in \mathcal{B}$; $(z^{(1)}, \dots, z^{(w)}) \leftarrow \Phi(z)$
  $\mathbf{s} \leftarrow \text{Assign}((z^{(i)})_{i=1}^{w}, \mathcal{C})$ $\overline{\mathbf{q}} \leftarrow \text{Select}(\mathbf{s}, \mathcal{C})$
  $\hat{\mathbf{p}}_{\text{VQ}} \leftarrow f_{\text{VQ}}(\overline{\mathbf{q}})$
  $\mathcal{L}_{\text{ce}} \leftarrow \text{CE}(\hat{\mathbf{p}}_{\text{VQ}}, y)$
  $\theta_{\text{VQ}} \leftarrow \theta_{\text{VQ}} - \eta_{\text{vq}} \nabla_{\theta_{\text{VQ}}} \mathcal{L}_{\text{ce}}$
  $\mathcal{C} \leftarrow \text{EMAUpdate}(\mathcal{C}, (z^{(i)})_{i=1}^{w}, \mathbf{s}; \gamma)$
until *convergence*;
Freeze $\mathcal{C}$ and f_{VQ}.

Stage 2: Region-aware Dirichlet calibration
Initialize calibration parameters $(\mathcal{A}, \mathcal{B}, \sigma)$ and $f_{\text{cal}}(\hat{\mathbf{p}}_{\text{VQ}}, \mathbf{s})$
repeat
  Sample minibatch $\mathcal{B} \subset \mathcal{D}_{\text{cal}}$
  $\hat{\mathbf{p}}_{\text{cal}} \leftarrow f_{\text{cal}}(\hat{\mathbf{p}}_{\text{VQ}}, \mathbf{s})$ for $(\hat{\mathbf{p}}_{\text{VQ}}, \mathbf{s}) \in \mathcal{B}$
  $\mathcal{L}_{\text{cal}} \leftarrow \text{CE}(\hat{\mathbf{p}}_{\text{cal}}, y)$
  $(\mathcal{A}, \mathcal{B}, \sigma) \leftarrow (\mathcal{A}, \mathcal{B}, \sigma) - \eta_{\text{cal}} \nabla_{(\mathcal{A},\mathcal{B},\sigma)} \mathcal{L}_{\text{cal}}$
until *convergence*;
return $\mathcal{C}$, f_{cal}

Training leverages Adam with learning rate and weight decay set to 1×10^{-3}.

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

F.3 Training of Local Methods

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

For `CIFAR-10` we use a fully connected network with a single hidden layer of size 64 and dropout rate of 0.3. We operate in a PCA-reduced feature representations 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 `CIFAR-100` we leverage the same architecture and hyper-parameters expect for the hidden layer, which now has size 128, and dropout rate the, set to 0.5. Training lasts 30 epochs with early stopping. The same applies to `TissueMNIST.` but we set hidden dimension to 256 and dropout rate back to 0.3. Learning instead lasts 60 with early stopping.

F.4 Metrics

For both *global* and *local* metrics we partitioned $f_k(\mathbf{x})$ into 15 bins based on predicted confidence scores. We accounted for class imbalance in `TissueMNIST` setting class weights accordingly. Moreover, we picked a value of 10 for the kernel-bandwidth hyper-paremter consistently with the learning procedure of the local methods.

F.5 Calibration Algorithm

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

Discretization step. The first step regards assigning discrete representation to the continuous latent represen-

tation of the baseline classifier. Learning proceeds as follows:

- **line 1:** A latent encoding $\mathbf{z}$ i extracted from the frozen encoder. $\mathbf{z}$ is segmented in w equal-sized slots $\mathbf{z}^{(i)}$.
- **line 2:** Each slot $\mathbf{z}^{(i)}$ is discretized by nearest neighbour assignment with respect to the vectors in $\mathcal{C}$. Assign$(\cdot, \cdot)$ yields an indices vector $\mathbf{s}$ used to select vectors from $\mathcal{C}$ and obtain the discrete representation $\overline{\mathbf{q}}$ via Select$(\cdot, \cdot)$.
- **line 3-6:** A new classification head takes as input $\overline{\mathbf{q}}$ and produces new scores. These values are used to compute categorical cross entropy between predicted probabilities and ground truth. The weights of f_{VQ} are updated via gradient descend whereas the codebook prototype vectors are updated via exponential moving average.

Training proceeds until convergence.

Calibration step. 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 $\mathbf{p}_{\text{VQ}}$ and indices $\mathbf{s}$, the calibration head does the following (**line 1** of **Stage 2**):

- Produces *sender* basis $\mathbf{A}$ from $\mathcal{A}^{(\mathcal{V})}$ and *receiver* basis $\mathbf{B}^{(\mathcal{V})}$ from $\mathcal{B}$ via Select$(\mathbf{s}, \mathcal{A})$ and Select$(\mathbf{s}, \mathcal{B})$;
- Computes $\boldsymbol{\alpha}^{\mathcal{V}}$ (Eq. (8)) and produces new calibrated scores (Eq. (16)).

Calibration parameters are updated via gradient descend on the categorical cross entropy until convergence.

F.6 Hardware

We used 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.