RETHINKING THE HARMONIC LOSS VIA NON-EUCLIDEAN DISTANCE LAYERS

Anonymous authors
Paper under double-blind review

ABSTRACT

Cross-entropy is the standard objective for training deep networks, yet it gives no inherent meaning to the learned weight vectors and drives them to grow without bound. The *harmonic loss* is a distance-based alternative that improves interpretability and mitigates *grokking*, but it has been studied only with the Euclidean distance, and never for its computational or energy cost. We replace that distance with a broad spectrum of metrics and evaluate the resulting *distance-tailored harmonic losses* on vision backbones and language models along three axes: *model performance*, *interpretability*, and *sustainability*. Across vision backbones, replacing the Euclidean distance is close to free: Bray-Curtis, cosine and Mahalanobis heads improve on Euclidean harmonic loss in most of the dataset–backbone settings we measure, and on the deeper backbones also on cross-entropy, at comparable measured emissions; where the harmonic family trails cross-entropy, on shallow backbones, the choice of distance still recovers most of the gap. On language modeling tasks, for smaller models, cosine-based harmonic losses improve gradient and learning stability, strengthen representation structure, and reduce emissions relative to cross-entropy and Euclidean harmonic loss. For a larger, modern decoder (SmolLM2), Euclidean harmonic loss attains the lowest perplexity and a Hamming-based harmonic loss the most structured representations at matched perplexity, at a higher training cost. On a pretrained 7B model, whose head spans only a few hundred classes, harmonic heads match cross-entropy's accuracy while improving representation structure and out-of-distribution detection at comparable measured training cost. Our code is available at: `https://anonymous.4open.science/r/rethinking-harmonic-loss/`.

1 INTRODUCTION

Cross-entropy is the *de facto* loss function for classification, but it is weak on model interpretability and training dynamics. It gives no inherent meaning to the learned weight vectors, which act as abstract parameters, and can drive them to grow without bound in pursuit of confident predictions (Baek et al., 2025), which is associated with phenomena such as *grokking*: a delayed generalization in which the train–test gap closes only after extensive overtraining (Power et al., 2022). In high-stakes applications where transparency is critical (*e.g.*, healthcare or finance), this opacity further complicates trust and error diagnosis. These issues motivate alternative loss functions that may yield more *interpretable*, *efficient*, and *robust* model behavior.

The *harmonic loss* was recently proposed to address some of these concerns (Baek et al., 2025). It replaces inner-product logits and softmax normalization with distances to learned *class prototypes*, encouraging each representation to approach its correct prototype. For the original Euclidean formulation, Baek et al. (2025) establish joint-scale-invariant probabilities and finite solutions under their assumptions; these results neither identify prototypes with empirical class centers nor extend automatically to other distances. Empirically, Baek et al. (2025) showed that harmonic loss closes the train–test gap faster, requires less data to generalize, and mitigates grokking, while remaining competitive on vision and language benchmarks.

That work, however, used Euclidean distance exclusively, and did not examine computational efficiency or energy consumption. Elsewhere the choice of metric is known to matter: for change point detection under concept drift, Coil et al. (2026) found that simple alternatives to a costly metric

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

(*e.g.*, Wasserstein) detect shifts as well at a fraction of the cost. This raises a natural question: *might other distance measures offer advantages over Euclidean in a harmonic loss setting?* No work has yet evaluated harmonic loss beyond Euclidean, nor benchmarked such choices across domains.

We present the first systematic study of the *distance* as a design choice inside the harmonic loss, integrating a rich set of metrics – Manhattan, Euclidean, Chebyshev, Minkowski, and cosine distance, as well as Hamming, Canberra, Bray-Curtis, and Mahalanobis – as drop-in replacements for Euclidean distance in the harmonic loss formulation. We evaluate them on two heterogeneous task families, *image classification* (MLP, ResNet, PVT) and *language modeling* with transformer-based LLMs (GPT-2, BERT, and others, including SmolLM2), which lets us assess whether certain distance-based losses consistently outperform cross-entropy and Euclidean harmonic loss on metrics of *effectiveness*, *efficiency*, and *explainability*. Specifically, we pursue the following research questions:

RQ1 (Model Performance): Do non-Euclidean harmonic loss functions offer higher accuracy or faster convergence compared to cross-entropy and Euclidean harmonic loss?

RQ2 (Interpretability): Do models trained with non-Euclidean harmonic losses exhibit more interpretable representations than those trained with cross-entropy?

RQ3 (Efficiency & Sustainability): If a custom non-Euclidean harmonic loss outperforms cross-entropy on a given downstream task, does it do so without incurring a higher computational cost? We track training time, resource utilization, and energy consumption to assess the *Green AI* perspective (Schwartz et al., 2020).

Together, these questions probe a *three-way trade-off* between *accuracy, interpretability, and sustainability* in model training. Previous work has typically optimized one or two of these aspects in isolation, improving accuracy at the cost of enormous compute, known as "Red AI" (Schwartz et al., 2020), or simplifying models for interpretability while losing accuracy. We seek solutions that improve predictive performance while also reducing energy consumption and yielding more transparent models.

Contributions. This paper introduces *distance-tailored harmonic losses* and provides an extensive empirical and analytical evaluation of their merits. To our knowledge, this is the first work to: i) extend the harmonic loss beyond Euclidean distance, and benchmark a wide spectrum of metrics on both vision and NLP tasks; ii) assess the carbon footprint and resource usage of different loss functions under a fixed, shared training budget; iii) investigate interpretability outcomes of distance-based losses; and iv) evaluate distance-based heads on a larger, modern decoder (SmolLM2) with repeated seeds, comparing interpretability at matched perplexity to control for its dependence on model quality; and v) show that, on a pretrained 7B model, harmonic heads improve representation structure and out-of-distribution detection at unchanged accuracy and training cost. We also offer theoretical insights into how the metric shapes the learned geometry: for the attraction component, L_1 losses induce a weighted-median condition and squared L_2 a weighted-mean condition (see Appendix P), which could inform the selection of a loss for a given objective.

2 HARMONIC LOSS

Harmonic loss replaces inner-product logits and softmax normalization with distances between a representation and learned class prototypes.

From Baek et al. (2025), given the training set $D = \{(x_i, y_i)\}_{i=1}^{n}$ with $y_i \in \{1, ..., K\}$ and class prototypes $W = \{\mathbf{w}_k\}_{k=1}^{K}$ with $\mathbf{w}_k \in \mathbb{R}^d$, the harmonic logit is the L_2 distance between the prototype $\mathbf{w}_k$ and the instance representation $\mathbf{h} \in \mathbb{R}^d$, *i.e.*, $d_k = \|\mathbf{h} - \mathbf{w}_k\|_2$. Then, the harmonic probabilities are given by:

$$p_W(y_k \mid x) \;=\; \frac{d_k^{-n}}{\sum_{j=1}^{K} d_j^{-n}}, \tag{1}$$

where the harmonic exponent n is a hyperparameter that controls the heavy-tailedness of the probability distribution. The Harmonic loss is then given by:

$$\mathcal{L}(W) = -\sum_{(x,y)\in D} \log p_W(y \mid x). \tag{2}$$

This approach endows the learning process with two key properties: i) *ratio invariance:* for 1-homogeneous distances (*e.g.*, Euclidean, Manhattan), the class probabilities $p_W(y \mid x)$ are invariant to uniform rescaling of both representations and prototypes, since factors of c^{-n} cancel in the ratio (see Proposition B.2); this contrasts with cross-entropy, where scaling logits changes the softmax distribution; [1] and ii) *finite-scale behavior in the original Euclidean formulation:* under the assumptions of Baek et al. (2025), optimization admits finite solutions; we do not extend this result to every distance considered here.

3 Non-Euclidean Harmonic Losses

Our framework introduces *non-Euclidean harmonic losses* as a generalization of the harmonic loss, and as a replacement for conventional cross-entropy training. The idea is that, in Equation (1), the Euclidean distance $d_k = \|\mathbf{h} - \mathbf{w}_k\|_2$ is replaced by a non-Euclidean distance.

3.1 Class Prototypes, Distances, and non-Euclidean Harmonic Loss functions

Each class $k \in \{1, \ldots, K\}$ is associated with a *prototype vector* $\mathbf{w} \in \mathbb{R}^d$. Given an instance representation $\mathbf{h}$, we compute its distance to all prototypes via a chosen metric $d(\cdot, \cdot)$.

Prototypes are learned parameters, just like the weight matrix in linear classification. Here, a *head* is the final output layer that maps a representation to scores over classes or vocabulary items; it is distinct from a Transformer attention head. A parameter-free distance head uses the same $K \times d$ prototype matrix as a bias-free linear layer; learned Mahalanobis variants add precision or covariance parameters.

We extend the Euclidean formulation of harmonic loss (Baek et al., 2025) with the distances of Table 1, each chosen for a specific property:

Their variants (three Hamming relaxations, three Bray–Curtis and Mahalanobis forms, the Minkowski exponents we search) and the numerical stabilizations every head needs are described in Appendix G.

From distances to probabilities. Every head turns its distance into the harmonic probabilities of Equation (1), so the only architectural change is the output layer, but two implementation choices shape the results (see Appendix C for integration details). The first is the exponent n: we use $n{=}1$ for the vision heads, $n{=}d$ for the language model heads (applied, following Baek et al. (2025), to the squared Euclidean distance), and $n{=}\sqrt{d}$ for the classification heads of Section 4.3. Because the exponent is applied to the *squared* Euclidean distance but to the unsquared Minkowski one, the "Euclidean" and "Minkowski ($p{=}2$)" language heads share a metric and differ only in effective exponent. The second is the numerical implementation. The language-model heads form normalized inverse-distance probabilities directly and apply a small numerical floor. At vocabulary scale, this floor can suppress gradients for some distances and make them difficult to train (see Section 4.3); we therefore treat such non-convergence as specific to this implementation. The classification heads instead use the equivalent log-space logits $-n \log(d_k + \varepsilon)$, avoiding explicit inverse powers and probability clamping. We evaluate the computational overhead of these heads at different output-space scales in Section 4.3.

Main implications. The distance also fixes where a prototype settles: for the attraction term alone, stationarity gives an inverse-distance weighted *mean* of the class features for squared Euclidean and a weighted coordinate-wise *median* for L_1, with Mahalanobis giving the mean in whitened coordinates (see Appendix P). Appendix B proves joint scale invariance for 1-homogeneous distances and notes the corresponding invariance for 0-homogeneous cosine, Bray–Curtis and Canberra. We note that we do not claim a finite-minimizer or generalization result for this wider family. Instead, the behaviour of soft Hamming is established empirically.

[1] *Individual* scale invariance—invariance to scaling $\mathbf{h}$ alone—holds only for angular distances such as cosine; for Euclidean distance, $\|\alpha\mathbf{h} - \mathbf{w}\|_2 \neq \|\mathbf{h} - \mathbf{w}\|_2$ in general.

Table 1: Catalog of distances adopted in this work for the harmonic loss. Details, variants and numerical stabilizations are given in Appendix G. We use their differentiable relaxations where the exact form has no usable gradient.

Distance	$d(\mathbf{h}, \mathbf{w})$	Why it might help
Euclidean	$\|\mathbf{h} - \mathbf{w}\|_2$	The formulation harmonic loss was defined with (Baek et al., 2025).
Manhattan (L_1)	$\|\mathbf{h} - \mathbf{w}\|_1$	Large deviations count linearly, which is more robust to outliers (Keeling & Kunisch, 2016; Brooks et al., 2013; Giloni & Padberg, 2004); it also needs no multiplications, which saves energy (Chen et al., 2020).
Chebyshev (L_∞)	$\|\mathbf{h} - \mathbf{w}\|_\infty$	Reads off the single most discriminative coordinate, and is cheap to compute.
Minkowski (L_p)	$\|\mathbf{h} - \mathbf{w}\|_p$	Interpolates between the three above; fractional p stays discriminative in high dimension (Aggarwal et al., 2001; Hu et al., 2016).
Cosine	$1 - \frac{\mathbf{h}^\top \mathbf{w}}{\|\mathbf{h}\|_2 \, \|\mathbf{w}\|_2}$	Compares directions only, so it ignores magnitude; its bounded output lowers activation variance (Luo et al., 2018) and underpins retrieval and face-recognition objectives (Reimers & Gurevych, 2019; Deng et al., 2019; Wang et al., 2018; Karpukhin et al., 2020; Sun et al., 2016).
Hamming	$\frac{1}{d} \sum_{i=1}^{d} \mathbf{1}_{\{h_i \neq w_i\}}$	Counts mismatching coordinates, one unit of distance per feature; made differentiable with sigmoid or Gumbel relaxations (Jang et al., 2017; Maddison et al., 2017).
Canberra	$\sum_{i=1}^{d} \frac{\|h_i - w_i\|}{\|h_i\| + \|w_i\| + \varepsilon}$	Normalizes each coordinate by its own magnitude, so small features weigh as much as large ones (Lance & Williams, 1967).
Bray–Curtis	$\frac{\sum_{i=1}^{d} \|h_i - w_i\|}{\sum_{i=1}^{d} (\|h_i\| + \|w_i\|) + \varepsilon}$	Measures proportional differences, the standard choice for compositional data (Bray & Curtis, 1957; Fuschi et al., 2025; Song et al., 2020).
Mahalanobis	$\sqrt{(\mathbf{h} - \mathbf{w})^\top \Sigma^{-1} (\mathbf{h} - \mathbf{w})}$	Accounts for correlated and unequally scaled features (Pang et al., 2018; Lee et al., 2018; Gómez-Silva et al., 2021; Omara et al., 2021).

4 Experiments

4.1 Training and Evaluation

Datasets. We evaluate on five *vision* benchmarks (MNIST, CIFAR-10, CIFAR-100, MarathiSign-Language, TinyImageNet) and one *language* corpus (OpenWebText), plus MiniPile (Kaddour, 2023) for the SmolLM2 study and four text classification benchmarks (SST-2, AG News, Banking77, CLINC150) for the 7B study.

Vision. We consider a two-layer MLP, a small CNN, ResNet-50 (with a stride-1 stem for small inputs) and PVTv2-B0; Appendix D.1 gives the full configurations.

Language. We study three Transformer families: GPT-style (decoder-only causal LM), BERT (encoder-only masked LM with 15% masking), and Qwen2-style decoders. Section 4.3 also pretrains a larger, modern decoder, SmolLM2, and Section 4.3 fine-tunes a pretrained 7B model, Qwen2.5-7B, for text classification. Complete configurations for the three smaller Transformer families are provided separately (see Appendix D.2).

Baselines. Besides cross-entropy and Euclidean harmonic loss, the vision studies compare against Focal loss, label smoothing, center loss, confidence penalty and ArcFace (see Appendix L).

Optimization. Unless noted, models are trained from scratch with Adam/AdamW-style optimizers (weight decay, (β_1, β_2) as configured), cosine learning-rate decay with linear warmup, mixed precision (FP16/BF16 when available), and gradient accumulation. We apply gradient clipping, dataset-specific schedulers, and early stopping with *dataset-specific patience* and a minimum improvement threshold ($\Delta_{\min}$). For fairness, all harmonic layers and the baseline share the same

backbone, batch size, scheduler, and data order. Additional details about optimization are reported in Appendix D.1.

Model Performance. For vision tasks, we report average Accuracy and F1. For language, we report *perplexity*, the exponential of the mean token-level negative log-likelihood; *gradient stability* (GS), which compares the variance of the gradient norm against the cross-entropy baseline, so that $\mathrm{GS} > 0$ means smoother training (Faghri et al., 2020; Lei & Ying, 2020); and the *effective rank* of the representation covariance, the exponential of the entropy of its eigenvalue spectrum, which detects dimensional collapse (Jing et al., 2022; Bardes et al., 2022; Garrido et al., 2023; Roy & Vetterli, 2007). Training curves and the modulo-addition case study are provided separately (see Appendices M and N). Appendix F further elaborates on their formulation. Effective rank and variance concentration pull in opposite directions by construction: we use concentration to compare heads of comparable quality, and effective rank to detect collapse. In the radar plots, metrics for which lower is better are inverted and every axis is normalized to $[0, 10]$ relative to the Euclidean harmonic baseline, so a larger area means better relative behaviour; the radial values carry no absolute meaning.

Interpretability. We adopt proxies to quantify how concentrated and separable a representation is. To this end, we probe whether learned prototypes/weights act as class centers and whether features become more structured by computing PCA explained variance on the penultimate features: i) *PC2 EV* (variance explained by the top two PCs), and ii) *PCA@90%* (dimensions required to reach 90% variance). Lower PCA@90% and higher PC2 EV indicate more concentrated, low-dimensional structure. For language, we report *PCA5*: Δ variance explained by the top 5 principal components of final hidden states (causal LM: last token; MLM: masked positions); higher values implies more concentrated, low-dimensional structure. For the 7B classification study, we also report the silhouette score computed from the test features and true class labels; higher values indicate greater within-class cohesion relative to between-class overlap. For vision, PCA is fitted with at most 50 components. Consequently, PCA@90% is capped at 50: a reported value of 50 means that 50 or more components are required. Any reduction relative to a baseline at this cap is therefore a lower bound (see Appendix A). The language studies use full-rank PCA.

Sustainability. We use *CodeCarbon* during training to log *duration*, *energy*, and *CO_2 emissions*. Emissions are reported per run and as differences from the cross-entropy baseline (grams CO_2; negative values indicate lower measured emissions). We aggregate by (dataset, backbone, distance) and also report cumulative figures across seeds. For language, we also report Speed ($-\Delta$`time_to_90_percent`): higher values denote fewer steps to reach 90% of final performance. Per-backbone FLOP estimates are reported separately (see Appendix O).

To isolate the effect of the *loss geometry* we swap *only* the classifier head, keeping initialization, preprocessing and augmentation, optimizer and schedule, batch size, epochs, early stopping, context length and seeds fixed across heads. For vision settings with repeated seeds, we also compute 95% confidence intervals across seeds and apply paired Wilcoxon signed-rank tests against the Euclidean harmonic head and the cross-entropy baseline on the matched configuration set. These tests serve only as robustness checks on the main trends: the qualitative conclusions reported in the radar plots and aggregate tables are retained only when the direction of the mean effect is consistent across seeds and is not contradicted by the paired nonparametric test. Hyperparameter grids, including head-specific parameters such as p for Minkowski and the covariance form for Mahalanobis, are in Appendix E.

4.2 Vision: Radar Plots

Figure 1 summarizes distance-based harmonic losses across all vision settings, including Marathi Sign (224×224 on ResNet50, 32×32 elsewhere) and TinyImageNet in addition to CIFAR-100; results on MNIST and CIFAR10, and on the MLP and CNN backbones (see Figure 5), are provided in Appendix K. All vision runs share one fixed budget, with no per-architecture tuning, so the comparison between heads is controlled while the absolute level sits below tuned state of the art (see Appendix A).

RQ1: Model Performance (F1, Accuracy). The comparison that isolates the effect of geometry is against *Euclidean* harmonic loss, and there the non-Euclidean heads win. Four of them have positive median differences over Euclidean across all 16 dataset–backbone pairs (see Table 22); and in the nine settings with per-head tables (see Appendix Q), Bray–Curtis (normalized) improves on it in 7 of 8, Mahalanobis (Cholesky) in 6 of 8, and cosine (stable/unstable) in 5 of 6, by $+0.5$ to $+1.5$ accuracy

points in median; only Chebyshev (standard) is consistently worse. Against cross-entropy the picture is split. The best non-Euclidean head wins in 6 of those 9 settings, by $+3.7$ points on CIFAR-100 with ResNet50 (0.736 against 0.698) and $+8.3$ on CIFAR-10 with ResNet50 (0.926 against 0.843), and on Marathi Sign and TinyImageNet cosine (stable) and Bray–Curtis match or beat it. Distance heads appear to be competitive where the backbone separates features well; elsewhere the metric still decides how much is lost. Three variants fail outright, losing 21–65 accuracy points in median, and we exclude them from our recommendations.

RQ2: Interpretability (PC2 EV, PCA 90%). Across Marathi Sign, TinyImageNet and CIFAR-100, Bray–Curtis and Chebyshev (standard) yield the largest PC2 explained variance and the lowest dimensionality for 90% EV, i.e. compact feature spaces. Mahalanobis concentrates variance to an extreme, but that clarity sometimes co-occurs with less stable optimization. We note that these metrics are only meaningful at comparable accuracy: a collapsed representation scores well on both. This consideration is particularly important in our language experiments described next.

RQ3: Sustainability (Duration/Epoch/GFLOPs, Emissions). Cosine is neutral to favorable in emissions. Bray-Curtis incurs modest overhead, and Mahalanobis is the most costly. Even at 224×224 the head is a small fraction of total FLOPs.

Takeaway. Replacing the Euclidean distance is close to free: the four best heads improve on it in most settings. In more detail, Cosine is the best all-around choice; Bray–Curtis and Chebyshev are the most interpretability-forward, reliably concentrating variance and reducing PCA 90% dimensionality, with accuracy effects that are positive but more configuration-dependent; Mahalanobis buys representation clarity at a higher sustainability cost. The effect of the geometry is consistent across benchmarks of both resolutions.

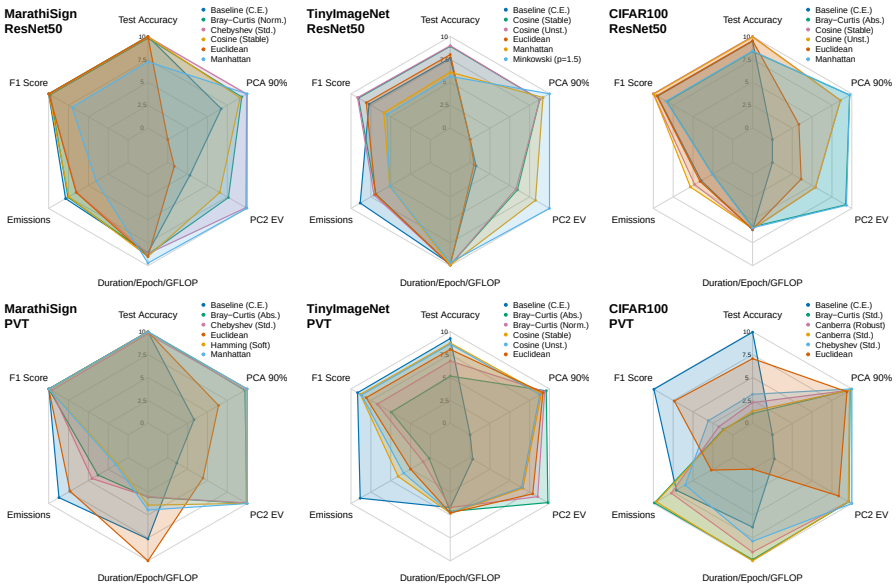

Figure 1: Vision: Radar plots for ResNet50 and PVT: 1) *Model Performance* (F1, Accuracy); 2) *Interpretability* (PC2 EV, PCA 90%), and 3) *Sustainability* (Duration/Epoch/GFLOPs, Emissions). Plots feature Baseline (Cross-Entropy), Euclidean harmonic, and the four top-performing non-Euclidean harmonic losses. The MLP and CNN visualizations are in Figure 5.

4.3 Language

We study distance heads in language models in three settings. *(i) Small models*: GPT, BERT and Qwen2 on OpenWebText, summarized by the radar plots of Figure 2. *(ii) Pretraining*: SmolLM2 pretrained from scratch on MiniPile for 328M tokens (see Appendix H). Cosine heads take the power form $p_v \propto ((1+\cos)/2)^m$; *(iii) A pretrained 7B model*: Qwen2.5-7B (Yang et al., 2024) with low-rank adapters (Hu et al., 2022) and a K-way harmonic head, fine-tuned on SST-2 (Socher et al.,

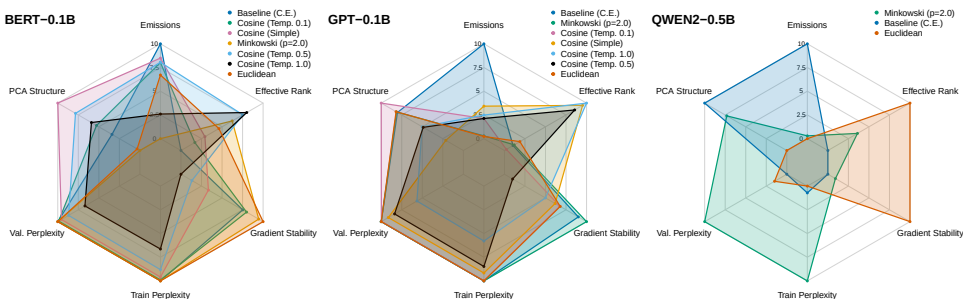

Figure 2: Language: Radar plots: 1) *Model Performance* (Perplexity, Effective Rank, Gradient Stability); 2) *Interpretability* (PCA5 EV), and 3) *Sustainability* (Emissions). Plots feature Baseline (CE), Euclidean harmonic, and the top-performing non-Euclidean harmonic losses.

2013), AG News (Zhang et al., 2015), Banking77 (Casanueva et al., 2020) and CLINC150 (Larson et al., 2019) as described in Appendix I. CLINC150's out-of-scope queries are excluded from training and kept for out-of-distribution detection.

RQ1: Model performance. On the small models, cosine heads pair competitive perplexity with better gradient stability and effective rank than both references; on Qwen no head dominates. In pretraining, Euclidean harmonic loss is the only head that improves on cross-entropy at equal steps, lowering perplexity from 21.82 to 21.44 and ECE by 20%, while cosine ($m{=}\sqrt{d}$) matches it and Hamming and cosine ($m{=}16$) trade perplexity (43.1 and 32.9) for the structure analysed below (see Table 13). Two heads yield unsuccessful training at this width: cosine ($m{=}1$) ends at perplexity 1,380 and the fast Manhattan head at 34,428. Minkowski ($p{=}1.5$) clips on three quarters of its steps. On the pretrained 7B model, accuracy is essentially and remarkably unchanged: pooled over the four tasks and five seeds every harmonic head lies within 0.2 points of cross-entropy and no difference is significant (see Table 3). Per task, the spread is wider: the largest gap is soft Hamming on Banking77 (-2.7 points).

RQ2: Interpretability at matched quality. Representation concentration changes with model quality: along the cross-entropy reference curve, the dimensions holding 90% of the variance rise from 198 at perplexity 119 to 625 at perplexity 21.8. A less-trained model can therefore appear more concentrated simply because it predicts worse. To control for this effect, we train cross-entropy and Euclidean references for 500–5,000 steps and interpolate each reference at the perplexity reached by the head under comparison. Before computing these comparisons, we set the threshold for a gain at $+0.014$ PC2 EV together with 33 fewer PCA@90% dimensions, ten seed standard deviations.

Soft Hamming is the only head that exceeds both thresholds against both references. At matched perplexity, it needs 243 dimensions for 90% of the variance, compared with 495 for cross-entropy and 363 for Euclidean, and reaches PC2 EV 0.202, compared with 0.130 and 0.167 (see Table 2). It also outperforms both cross-entropy runs. Against Euclidean, the dimensionality gains survive this stricter check, while the PC1 and PC2 EV gains do not: the 1,000-step Euclidean model reaches higher values at perplexity 71.

On the pretrained 7B model, where accuracies are statistically indistinguishable, PC2 EV rises for every head and task. The coordinate-wise heads (Hamming, Bray–Curtis, and Manhattan) remove about 220 PCA@90% dimensions, compared with 77 for Euclidean; on CLINC150 the count falls from 583 to 136–143. An independent measure confirms this structure: relative to cross-entropy, the silhouette score rises for every head on every task by $+0.10$ to $+0.33$ (all corrected $p < 0.01$), reaching 0.53 on CLINC150. The tighter class structure does not make the learned prototypes empirical class centres. Cross-entropy's weight vectors align more closely with the class means than the harmonic prototypes do (cosine 0.43–0.56 compared with 0.14–0.38), and each harmonic prototype lies approximately as far from its class mean as the class means lie from one another.

RQ3: Sustainability and cost. The head is lightweight compared with the backbone, so emission differences among the small models come from faster convergence, not per-step FLOPs. During SmolLM2 pretraining at a fixed token budget, energy closely tracks measured step time ($r > 0.99$).

Table 2: Non-Euclidean heads compared with cross-entropy and Euclidean references at matched perplexity: 43.11, 32.86, and 21.87 for the three blocks. Reference values are interpolated to the corresponding head's perplexity. Within each block, the best value in each column is bold. Robustness checks against the reference runs and further metrics are reported in Tables 15 and 16.

Model	Top-1 ↑	ECE ↓	PC1 EV ↑	PC2 EV ↑	PCA@90% ↓	PCA@95% ↓
Hamming (soft)	**.417**	**.036**	**.138**	**.202**	**243**	**488**
Cross-entropy	.406	.050	.080	.130	495	889
Euclidean	.403	.037	.108	.167	363	715
Cosine (m=16)	**.461**	.097	.053	.090	**300**	**520**
Cross-entropy	.437	.052	.076	.127	551	947
Euclidean	.434	**.040**	**.092**	**.146**	430	798
Cosine (m=$\sqrt{d}$)	.479	.052	.073	.126	670	1062
Cross-entropy	**.480**	.053	.081	**.140**	624	1017
Euclidean	.477	**.042**	**.083**	.129	**539**	**920**

Table 3: Harmonic heads minus cross-entropy on a pretrained 7B model, pooled over four tasks and five seeds (AUROC on CLINC150 only); in parentheses, tasks where the harmonic head wins on average. Cross-entropy is used as trained. */**: $p < 0.05$/$p < 0.01$ from a Wilcoxon signed-rank test whose inferential unit is the task–seed pair, pooling four heterogeneous tasks (paired t-test over seeds for AUROC). Holm correction is applied across the six heads *within* each metric. Additional details are in Table 18.

Head	Δ Acc. ↑	Δ ECE ↓	Δ NLL ↓	Δ PC2 EV ↑	Δ PCA@90% ↓	Δ Silh. ↑	Δ AUROC ↑
Euclidean	+.002 (3/4)	-.006** (4/4)	-.090** (4/4)	+.063** (4/4)	-77.1 (3/4)	+.181** (4/4)	+.017**
Cosine	+.002 (2/4)	-.007** (4/4)	-.098** (3/4)	+.072** (4/4)	-11.2 (2/4)	+.148** (4/4)	+.019**
Hamming (soft)	+.000 (2/4)	-.006** (3/4)	-.098** (3/4)	+.105** (4/4)	-223.2** (4/4)	+.239** (4/4)	+.020**
Mahalanobis (diag.)	+.001 (2/4)	-.006** (4/4)	-.090** (3/4)	+.054** (4/4)	-74.2 (2/4)	+.145** (4/4)	+.018**
Bray–Curtis	+.001 (3/4)	-.007** (4/4)	-.097** (4/4)	+.100** (4/4)	-227.1** (4/4)	+.234** (4/4)	+.019**
Manhattan	+.002 (2/4)	-.006** (3/4)	-.098** (4/4)	+.087** (4/4)	-214.5** (3/4)	+.228** (4/4)	+.019**

Cosine, Minkowski (p=2), and Euclidean heads remain within 22% of cross-entropy, less than the 29% variation between two identical cross-entropy runs sharing a node. Soft Hamming has no matrix-multiplication form and requires $6.9\times$ the step time and $8.3\times$ the energy. On the 7B classification tasks, whose heads span only a few hundred classes, measured step times differ by at most 2.2% across heads.

Beyond the three axes. Against cross-entropy as trained, harmonic heads also detect out-of-scope queries better: with maximum softmax probability, AUROC rises from 0.964 to 0.981–0.983, while the false-positive rate at 95% true-positive rate falls from 0.166 to 0.059–0.073 (see Table 19). Because this score depends on logit scale, the comparison does not isolate geometry from post-hoc calibration; we do not evaluate calibrated cross-entropy. Scores based on absolute logits are uninformative for scale-invariant heads. The geometry has to be learned: with a frozen backbone, harmonic heads fall well behind a linear probe (Banking77: 73.0% against 34.6–47.7%). The exponent matters more than the distance, with n=$\sqrt{d}$ achieving the best result in every configuration (see Table 21).

Takeaway. Output-head geometry changes the learned representation in all three language settings, while its performance and computational effects depend on scale. In the 0.1–0.5B models, cosine heads improve gradient stability and effective rank. In vocabulary-scale SmolLM2 pretraining, Euclidean gives the best perplexity and soft Hamming the strongest matched-quality concentration, with substantially higher time and energy for Hamming. In 7B classification, the coordinate-wise heads preserve cross-entropy's accuracy while producing the largest concentration and out-of-distribution gains at comparable step time. Thus, the small-model cosine advantage does not reappear in SmolLM2 pretraining; at 7B scale, the leading concentration gains come from Hamming, Bray–Curtis, and Manhattan.

5 Related Work

Loss functions for classification. Cross-entropy dominates classification for its effectiveness and probabilistic reading, but it only asks that classes be separable, not that the representation be structured or interpretable. Metric-learning losses such as contrastive and triplet loss preserve distances between examples at the cost of sampling strategies and longer training; Boudiaf et al. (2020) unify them with cross-entropy through mutual information, showing that cross-entropy implicitly bounds pairwise objectives. Regularization-based losses instead shape the class geometry directly: *center loss* (Wen et al., 2016) compacts intra-class clusters, AMC-Loss (Choi et al., 2020) adds angular constraints, Orthogonal Projection Loss (Ranasinghe et al., 2021) enforces inter-class orthogonality without sampling, and Class Anchor Clustering (Miller et al., 2021) ties clusters to anchored prototypes for open-set recognition. Empirical comparisons of losses (Cho et al., 2019; Janocha & Czarnecki, 2016) and evolutionary search over them (Gonzalez & Miikkulainen, 2020) report differences in accuracy and convergence speed, but keep the inner product at the output layer; harmonic loss (Baek et al., 2025) replaces it with a distance, which is where we intervene.

Efficiency and Green AI. Green AI treats energy and efficiency as first-class evaluation criteria alongside accuracy (Schwartz et al., 2020), a concern sharpened by the growing footprint of large models (Wang et al., 2023; 2025); recent studies measure the emissions of training choices in practice (Verma et al., 2024; Rafat et al., 2023; Paula et al., 2025). We adopt that perspective for the loss function itself.

Interpretability. Interpretability is usually added after training, whereas *interpretable by design* work argues for building it into architectures and losses (Rudin, 2019; Zhang et al., 2021); harmonic loss belongs to the latter, since its weights are class prototypes. Mechanistic approaches instead read structure out of trained networks (Bereska & Gavves, 2024), and Wen et al. (2025) combine interpretable models with deep networks for time series. Closer to us, several losses are designed for interpretability: sparse coding constraints for word-level attributions (Liu et al., 2022), an interpretative loss for video captioning (Dong et al., 2017), filter-level objectives for CNNs (Zhang et al., 2018), and distance-based penalties that align attributions with user annotations (Hagos et al., 2023).

Distance metrics elsewhere. Outside classification, the choice of metric is known to matter: for anomaly detection under concept drift (Coil et al., 2026), for SVM kernels (Amaya-Tejera et al., 2024), and for k–nearest neighbours (Kalra et al., 2022; Hu et al., 2016), where accuracy and cost both depend on it. Prototype– and distance-based classifiers themselves have a long history: nearest-class-mean classifiers (Mensink et al., 2013), prototypical networks (Snell et al., 2017), convolutional prototype learning (Yang et al., 2018) and cosine-normalized classifiers (Wang et al., 2017; Liu et al., 2017; Gidaris & Komodakis, 2018), several of which already compare Euclidean against cosine geometry in the head. What is new in our work is treating the distance as a design axis of the *training objective* across a catalogue of metrics, and evaluating that choice jointly on accuracy, interpretability and sustainability.

6 Conclusion

We studied *distance-based harmonic losses* as training time alternatives to cross-entropy across five image benchmarks, four backbones, language model pretraining, and a pretrained 7B model, using the distances of Table 1. The results show that geometry matters, but not uniformly: cosine often gives smooth optimization, normalized Bray–Curtis improves most consistently over Euclidean harmonic loss, and some variants collapse or fail at vocabulary width (see Appendix A). Sustainability depends on both distance and architecture; inexpensive heads can match the baseline cost, whereas vocabulary-scale heads without a matrix-multiplication form cost 7–9$\times$ more. Representation concentration must be interpreted at comparable model quality. Under that control, soft Hamming is the clearest pretraining case. Most strikingly, on the pretrained 7B model, changing only the output-head geometry increases class silhouette on every task and improves out-of-distribution detection relative to cross-entropy as trained, with no statistically detectable accuracy loss and at most 2.2% variation in measured step time (see Section 4.3).

USE OF LARGE LANGUAGE MODELS

In this work, we used generative AI tools (an LLM-based coding and writing assistant) for the following tasks. *Drafting sections of the paper:* In particular, the text, tables, and figure of Section 4.2, Section 4.3, Appendix H and Appendix I were first drafted with the assistant, as were the related additions to the abstract, introduction, and conclusion; the authors then reviewed and edited these drafts. *Aid or polish writing:* multiple parts have been polished using generative AI to improve flow and effectively convey information in a more concise way, to make better use of space limitations. *Implementation:* writing and debugging parts of the training, evaluation, and analysis code of the two new studies, including the PCA-based metrics, the matched-perplexity analysis, the classification runner, the job scripts, and the scripts that generate their tables and figures. *Results interpretation:* summarizing seed variability and interpreting the matched-perplexity, out-of-distribution, training-cost, and weight-space analyses. All experiments were run by the authors, and all reported numbers were produced by the released code from the experiment logs. The authors reviewed the AI-assisted code, analyses, and text, and take full responsibility for the content of the paper. We have reviewed all AI-assisted work.

REFERENCES

Charu C. Aggarwal, Alexander Hinneburg, and Daniel A. Keim. On the surprising behavior of distance metrics in high dimensional spaces. In *ICDT*, volume 1973 of *Lecture Notes in Computer Science*, pp. 420–434. Springer, 2001.

Nazhir Amaya-Tejera, Margarita Gamarra, Jorge I Vélez, and Eduardo Zurek. A distance-based kernel for classification via support vector machines. *Frontiers in Artificial Intelligence*, 7:1287875, 2024.

David D. Baek, Ziming Liu, Riya Tyagi, and Max Tegmark. Harmonic loss trains interpretable AI models. *Transactions on Machine Learning Research*, 2025. https://openreview.net/forum?id=ZpSZ7pNoCs.

Adrien Bardes, Jean Ponce, and Yann LeCun. VICReg: Variance-invariance-covariance regularization for self-supervised learning. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022. URL https://openreview.net/forum?id=xm6YD62D1Ub.

Loubna Ben Allal, Anton Lozhkov, Elie Bakouch, Gabriel Martín Blázquez, Guilherme Penedo, Lewis Tunstall, Andrés Marafioti, Hynek Kydlíček, Agustín Piqueres Lajarín, Vaibhav Srivastav, Joshua Lochner, Caleb Fahlgren, Xuan-Son Nguyen, Clémentine Fourrier, Ben Burtenshaw, Hugo Larcher, Haojun Zhao, Cyril Zakka, Mathieu Morlon, Colin Raffel, Leandro von Werra, and Thomas Wolf. SmolLM2: When smol goes big – data-centric training of a small language model. *arXiv preprint arXiv:2502.02737*, 2025.

Magnus Bengtsson. Compressing large language models with PCA without performance loss, 2025. URL https://arxiv.org/abs/2508.04307.

Leonard Bereska and Stratis Gavves. Mechanistic interpretability for AI safety - A review. *Trans. Mach. Learn. Res.*, 2024.

Malik Boudiaf, Jérôme Rony, Imtiaz Masud Ziko, Eric Granger, Marco Pedersoli, Pablo Piantanida, and Ismail Ben Ayed. A unifying mutual information view of metric learning: cross-entropy vs. pairwise losses. In *European conference on computer vision*, pp. 548–564. Springer, 2020.

J. Roger Bray and J. T. Curtis. An ordination of the upland forest communities of southern Wisconsin. *Ecological Monographs*, 27(4):325–349, 1957. doi: 10.2307/1942268.

J. P. Brooks, J. H. Dulá, and E. L. Boone. A pure L_1-norm principal component analysis. *Computational Statistics & Data Analysis*, 61:83–98, 2013. doi: 10.1016/j.csda.2012.11.007.

Iñigo Casanueva, Tadas Temčinas, Daniela Gerz, Matthew Henderson, and Ivan Vulić. Efficient intent detection with dual sentence encoders. In *Proceedings of the 2nd Workshop on Natural Language Processing for Conversational AI*, 2020.

Hanting Chen, Yunhe Wang, Chunjing Xu, Boxin Shi, Chao Xu, Qi Tian, and Chang Xu. AdderNet: Do we really need multiplications in deep learning? In *CVPR*, pp. 1465–1474. Computer Vision Foundation / IEEE, 2020.

Kwantae Cho, Jong-hyuk Roh, Youngsam Kim, and Sangrae Cho. A performance comparison of loss functions. In *2019 International Conference on Information and Communication Technology Convergence (ICTC)*, pp. 1146–1151. IEEE, 2019.

Hongjun Choi, Anirudh Som, and Pavan K. Turaga. AMC-Loss: Angular margin contrastive loss for improved explainability in image classification. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR Workshops 2020, Seattle, WA, USA, June 14-19, 2020*, pp. 3659–3666. Computer Vision Foundation / IEEE, 2020. doi: 10.1109/CVPRW50498.2020.00427. URL https://openaccess.thecvf.com/content_CVPRW_2020/html/w50/Choi_AMC-Loss_Angular_Margin_Contrastive_Loss_for_Improved_Explainability_in_Image_CVPRW_2020_paper.html.

Collin Coil, Kamil Faber, Bartlomiej Sniezynski, and Roberto Corizzo. Distance-based change point detection for novelty detection in concept-agnostic continual anomaly detection. *Journal of Intelligent Information Systems*, 64(1):37–75, 2026. doi: 10.1007/s10844-025-00949-1.

Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. ArcFace: Additive angular margin loss for deep face recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4690–4699, 2019.

Yinpeng Dong, Hang Su, Jun Zhu, and Bo Zhang. Improving interpretability of deep neural networks with semantic information. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4306–4314, 2017.

Nelson Elhage, Tristan Hume, Catherine Olsson, Nicholas Schiefer, Tom Henighan, Shauna Kravec, Zac Hatfield-Dodds, Robert Lasenby, Dawn Drain, Carol Chen, Roger Grosse, Sam McCandlish, Jared Kaplan, Dario Amodei, Martin Wattenberg, and Christopher Olah. Toy models of superposition. *Transformer Circuits Thread*, 2022. URL https://transformer-circuits.pub/2022/toy_model/index.html.

Fartash Faghri, David Duvenaud, David J. Fleet, and Jimmy Ba. A study of gradient variance in deep learning. *CoRR*, abs/2007.04532, 2020. URL https://arxiv.org/abs/2007.04532.

Pedro Freire, ChengCheng Tan, Adam Gleave, Dan Hendrycks, and Scott Emmons. Uncovering latent human wellbeing in language model embeddings. *arXiv preprint arXiv:2402.11777*, 2024.

Alessandro Fuschi, Alessandra Merlotti, and Daniel Remondini. Microbiome data: tell me which metrics and i will tell you which communities. *ISME communications*, 5(1):ycaf125, 2025.

Quentin Garrido, Randall Balestriero, Laurent Najman, and Yann LeCun. RankMe: Assessing the downstream performance of pretrained self-supervised representations by their rank. In *ICML*, volume 202 of *Proceedings of Machine Learning Research*, pp. 10929–10974. PMLR, 2023.

Spyros Gidaris and Nikos Komodakis. Dynamic few-shot visual learning without forgetting. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

Avi Giloni and Manfred Padberg. The finite sample breakdown point of ℓ_1-regression. *SIAM Journal on Optimization*, 14(4):1028–1042, 2004. doi: 10.1137/S1052623403424156.

María José Gómez-Silva, Arturo de la Escalera, and José María Armingol. Back-propagation of the Mahalanobis distance through a deep triplet learning model for person re-identification. *Integrated Computer-Aided Engineering*, 28(3):277–294, 2021.

Santiago Gonzalez and Risto Miikkulainen. Improved training speed, accuracy, and data utilization through loss function optimization. In *2020 IEEE congress on evolutionary computation (CEC)*, pp. 1–8. IEEE, 2020.

Misgina Tsighe Hagos, Niamh Belton, Kathleen M Curran, and Brian Mac Namee. Distance-aware explanation based learning. In *2023 IEEE 35th International Conference on Tools with Artificial Intelligence (ICTAI)*, pp. 279–286. IEEE, 2023.

Yihan He, Yuan Cao, Hong-Yu Chen, Dennis Wu, Jianqing Fan, and Han Liu. Learning spectral methods by transformers. *arXiv preprint arXiv:2501.01312*, 2025.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.

Li-Yu Hu, Min-Wei Huang, Shih-Wen Ke, and Chih-Fong Tsai. The distance function effect on k-nearest neighbor classification for medical datasets. *SpringerPlus*, 5(1):1304, 2016.

Robert Huben, Hoagy Cunningham, Logan Riggs Smith, Aidan Ewart, and Lee Sharkey. Sparse autoencoders find highly interpretable features in language models. In *International Conference on Learning Representations (ICLR), Poster*, 2024. URL https://openreview.net/forum?id=F76bwRSLeK.

Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. In *ICLR (Poster)*. OpenReview.net, 2017.

Katarzyna Janocha and Wojciech Marian Czarnecki. On loss functions for deep neural networks in classification. *Schedae Informaticae*, 25:49–59, 2016. doi: 10.4467/20838476SI.16.004.6185.

Mingyu Jin, Qinkai Yu, Jingyuan Huang, Qingcheng Zeng, Zhenting Wang, Wenyue Hua, Haiyan Zhao, Kai Mei, Yanda Meng, Kaize Ding, Fan Yang, Mengnan Du, and Yongfeng Zhang. Exploring concept depth: How large language models acquire knowledge and concept at different layers? In *Proceedings of the 31st International Conference on Computational Linguistics*, pp. 558–573, Abu Dhabi, UAE, 2025. Association for Computational Linguistics. URL https://aclanthology.org/2025.coling-main.37/.

Li Jing, Pascal Vincent, Yann LeCun, and Yuandong Tian. Understanding dimensional collapse in contrastive self-supervised learning. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022. URL https://openreview.net/forum?id=YevsQ05DEN7.

Ole Jorgensen. Understanding and controlling the activations of language models. MSc dissertation, Imperial College London, 2023. URL https://ojorgensen.github.io/assets/pdfs/Imperial_Dissertation.pdf.

Jean Kaddour. The MiniPile challenge for data-efficient language models. *arXiv preprint arXiv:2304.08442*, 2023.

Vandana Kalra, Indu Kashyap, and Harmeet Kaur. Effect of distance measures on k-nearest neighbour classifier. In *2022 Second International Conference on Computer Science, Engineering and Applications (ICCSEA)*, pp. 1–7. IEEE, 2022.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 conference on empirical methods in natural language processing (EMNLP)*, pp. 6769–6781, 2020.

Stephen L Keeling and Karl Kunisch. Robust ℓ_1 approaches to computing the geometric median and principal and independent components. *Journal of Mathematical Imaging and Vision*, 56(1): 99–124, 2016.

Godfrey N. Lance and William T. Williams. A general theory of classificatory sorting strategies: 1. Hierarchical systems. *The Computer Journal*, 9(4):373–380, 1967.

Stefan Larson, Anish Mahendran, Joseph J. Peper, Christopher Clarke, Andrew Lee, Parker Hill, Jonathan K. Kummerfeld, Kevin Leach, Michael A. Laurenzano, Lingjia Tang, and Jason Mars. An evaluation dataset for intent classification and out-of-scope prediction. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 1311–1316. Association for Computational Linguistics, 2019. doi: 10.18653/v1/D19-1131. URL https://aclanthology.org/D19-1131/.

Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. *Advances in Neural Information Processing Systems*, 31, 2018.

Yunwen Lei and Yiming Ying. Fine-grained analysis of stability and generalization for stochastic gradient descent. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pp. 5809–5819. PMLR, 2020. URL http://proceedings.mlr.press/v119/lei20c.html.

Junhong Liu, Yijie Lin, Liang Jiang, Jia Liu, Zujie Wen, and Xi Peng. Improve interpretability of neural networks via sparse contrastive coding. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pp. 460–470. Association for Computational Linguistics, 2022. doi: 10.18653/V1/2022.FINDINGS-EMNLP.32. URL https://doi.org/10.18653/v1/2022.findings-emnlp.32.

Weiyang Liu, Yandong Wen, Zhiding Yu, Ming Li, Bhiksha Raj, and Le Song. SphereFace: Deep hypersphere embedding for face recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

Chunjie Luo, Jianfeng Zhan, Xiaohe Xue, Lei Wang, Rui Ren, and Qiang Yang. Cosine normalization: Using cosine similarity instead of dot product in neural networks. In *ICANN (1)*, volume 11139 of *Lecture Notes in Computer Science*, pp. 382–391. Springer, 2018.

Chris J. Maddison, Andriy Mnih, and Yee Whye Teh. The concrete distribution: A continuous relaxation of discrete random variables. In *ICLR (Poster)*. OpenReview.net, 2017. URL https://openreview.net/forum?id=S1jE5L5gl.

Thomas Mensink, Jakob Verbeek, Florent Perronnin, and Gabriela Csurka. Distance-based image classification: Generalizing to new classes at near-zero cost. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(11):2624–2637, 2013.

Dimity Miller, Niko Sunderhauf, Michael Milford, and Feras Dayoub. Class anchor clustering: A loss for distance-based open set recognition. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 3570–3578, 2021.

Ibrahim Omara, Ahmed Hagag, Guangzhi Ma, Fathi E. Abd El-Samie, and Enmin Song. A novel approach for ear recognition: learning mahalanobis distance features from deep CNNs. *Mach. Vis. Appl.*, 32(1):38, 2021.

Tianyu Pang, Chao Du, and Jun Zhu. Max-Mahalanobis linear discriminant analysis networks. In *International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 4016–4025. PMLR, 2018.

Eileen Paula, Jayesh Soni, Himanshu Upadhyay, and Leonel Lagos. Comparative analysis of model compression techniques for achieving carbon efficient AI. *Scientific Reports*, 15(1):23461, 2025.

Alethea Power, Yuri Burda, Harri Edwards, Igor Babuschkin, and Vedant Misra. Grokking: Generalization beyond overfitting on small algorithmic datasets. *CoRR*, abs/2201.02177, 2022.

Kazi Rafat, Sadia Islam, Abdullah Al Mahfug, Md Ismail Hossain, Fuad Rahman, Sifat Momen, Shafin Rahman, and Nabeel Mohammed. Mitigating carbon footprint for knowledge distillation based deep learning model compression. *Plos one*, 18(5):e0285668, 2023.

Kanchana Ranasinghe, Muzammal Naseer, Munawar Hayat, Salman H. Khan, and Fahad Shahbaz Khan. Orthogonal projection loss. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*, pp. 12313–12323. IEEE, 2021. doi: 10.1109/ICCV48922.2021.01211. URL https://doi.org/10.1109/ICCV48922.2021.01211.

Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 3982–3992, 2019.

Olivier Roy and Martin Vetterli. The effective rank: A measure of effective dimensionality. In *2007 15th European signal processing conference*, pp. 606–610. IEEE, 2007.

Cynthia Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 1(5):206–215, 2019.

Roy Schwartz, Jesse Dodge, Noah A Smith, and Oren Etzioni. Green AI. *Communications of the ACM*, 63(12):54–63, 2020.

Jake Snell, Kevin Swersky, and Richard S. Zemel. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems*, 2017.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Y. Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, 2013.

Kuncheng Song, Fred A Wright, and Yi-Hui Zhou. Systematic comparisons for composition profiles, taxonomic levels, and machine learning methods for microbiome-based disease prediction. *Frontiers in Molecular Biosciences*, 7:610845, 2020.

Pei-Xia Sun, Hui-Ting Lin, and Tao Luo. Learning discriminative CNN features and similarity metrics for image retrieval. In *2016 IEEE International Conference on Signal Processing, Communications and Computing (ICSPCC)*, pp. 1–5. IEEE, 2016. doi: 10.1109/ICSPCC.2016.7753634.

Alexander Matt Turner, Lisa Thiergart, Gavin Leech, David Udell, Juan J. Vazquez, Ulisse Mini, and Monte MacDiarmid. Steering language models with activation engineering. *arXiv preprint arXiv:2308.10248*, 2023.

Anil Verma, Sumit Kumar Singh, Rupesh Kumar Sah, Rajiv Misra, and TN Singh. Performance comparison of deep learning models for CO2 prediction: Analyzing carbon footprint with advanced trackers. In *2024 IEEE International Conference on Big Data (BigData)*, pp. 4429–4437. IEEE, 2024.

Feng Wang, Xiang Xiang, Jian Cheng, and Alan L. Yuille. NormFace: L2 hypersphere embedding for face verification. In *Proceedings of the 25th ACM International Conference on Multimedia*, 2017.

Hao Wang, Yitong Wang, Zheng Zhou, Xing Ji, Dihong Gong, Jingchao Zhou, Zhifeng Li, and Wei Liu. CosFace: Large margin cosine loss for deep face recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5265–5274, 2018.

Irene Wang, Mostafa Elhoushi, H. Ekin Sumbul, Samuel Hsia, Daniel Jiang, Newsha Ardalani, Divya Mahajan, Carole-Jean Wu, and Bilge Acun. CATransformers: Carbon aware transformers through joint model-hardware optimization. In *Advances in Neural Information Processing Systems*, volume 38, pp. 28305–28337, 2025. doi: 10.52202/085713-0841.

Xiaorong Wang, Clara Na, Emma Strubell, Sorelle Friedler, and Sasha Luccioni. Energy and carbon considerations of fine-tuning BERT. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 9058–9069. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.findings-emnlp.607. URL https://aclanthology.org/2023.findings-emnlp.607/.

Yandong Wen, Kaipeng Zhang, Zhifeng Li, and Yu Qiao. A discriminative feature learning approach for deep face recognition. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 499–515. Springer, 2016. doi: 10.1007/978-3-319-46478-7_31.

Yunshi Wen, Tengfei Ma, Ronny Luss, Debarun Bhattacharjya, Achille Fokoue, and Anak Agung Julius. Shedding light on time series classification using interpretability gated networks. In *ICLR*. OpenReview.net, 2025.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

Hong-Ming Yang, Xu-Yao Zhang, Fei Yin, and Cheng-Lin Liu. Robust classification with convolutional prototype learning. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

Quanshi Zhang, Ying Nian Wu, and Song-Chun Zhu. Interpretable convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 8827–8836, 2018.

Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems*, 2015.

Yu Zhang, Peter Tiňo, Aleš Leonardis, and Ke Tang. A survey on neural network interpretability. *IEEE transactions on emerging topics in computational intelligence*, 5(5):726–742, 2021.

A Limitations

This appendix summarizes the scope of the current evaluation and identifies extensions that would broaden it.

Vision PCA metrics. The vision analysis fits PCA with at most 50 components, so a PCA@90% value of 50.0 means that 50 or more components are required. Cross-entropy reaches this cap in the CIFAR-100 settings; consequently, reductions measured against it are conservative lower bounds. The explained-variance metrics are unaffected because they are computed relative to the total feature variance. The language studies use full-rank PCA, which permits values above 50.

Representation metrics. PCA concentration does not by itself distinguish class structure from low-dimensional representations associated with lower predictive accuracy. For example, the Euclidean head on CIFAR-100 with PVT has PC2 EV 0.84 and PCA@90%$=4$, together with 28.6% accuracy (see Table 25). We therefore interpret geometry metrics only among heads with comparable accuracy, including when reading the radar plots. The silhouette scores in the 7B classification study provide a complementary, non-PCA measure of class separation (see Section 4.3). Future evaluations could incorporate additional measures of class structure and test their relation to downstream generalization.

Distance variants. Mahalanobis (Std.), Bray–Curtis (Std.), and Chebyshev (Smooth) have median accuracy reductions of 21–65 points across dataset–backbone combinations (see Table 22). They are reported for completeness and included in aggregate statistics over all distances, but they are not among the configurations highlighted in our recommendations. Characterizing the optimization and geometric conditions associated with their performance is left for future work.

Training budgets. All vision runs use a common training budget without per-architecture tuning, providing a controlled comparison among heads. This protocol is not intended to optimize absolute accuracy; for example, PVTv2-B0 reaches 39.9% on CIFAR-100 under this budget. Future studies could tune each architecture independently and examine whether the observed ranking of losses is retained.

Energy measurements. Emissions are estimated with CodeCarbon on shared compute nodes, where two identical cross-entropy runs differed by 29% (see Section 4.3). The measurements therefore distinguish large effects, including the 7–9$\times$ step-time cost of heads without a matrix-multiplication form, but have limited resolution for smaller differences between individual runs.

Repeated measurements on dedicated nodes would enable a more precise comparison of those smaller differences.

B Theoretical Properties of Distance-based Probabilistic Layers

Setup. Let $\{(x_i, y_i)\}_{i=1}^n$ be the training set with $y_i \in \{1, ..., K\}$. Each class has a prototype $w_k \in \mathbb{R}^d$ and a nonnegative distance $d(x, w) \geq 0$. Given a decreasing link $\kappa : \mathbb{R}_+ \to \mathbb{R}_+$ we define

$$p_k(x_i) \;=\; \frac{\kappa\big(d(x_i, w_k)\big)}{\sum_{j=1}^K \kappa\big(d(x_i, w_j)\big)}, \qquad \mathcal{L}(\{w_k\}) \;=\; -\sum_{i=1}^n \log p_{y_i}(x_i).$$

harmonic $\kappa(r) = r^{-\omega}$ with $\omega > 0$; while distances include Euclidean/Mahalanobis, L_p, Bregman divergences, and cosine/angle on the sphere.

B.1 Scale invariance

Definition B.1 (Homogeneity)**.** A distance d is 1-*homogeneous* if $d(cx, cw) = |c|\, d(x, w)$ for all $c > 0$, and 0-*homogeneous* if $d(cx, cw) = d(x, w)$.

Proposition B.2 (Joint scale invariance of the harmonic link)**.** *Let $\kappa(r) = r^{-\omega}$ with $\omega > 0$ and let d be 1-homogeneous. Then for every $c > 0$,*

$$p_k(cx_i; cW) \;=\; \frac{\big(c\, d(x_i, w_k)\big)^{-\omega}}{\sum_j \big(c\, d(x_i, w_j)\big)^{-\omega}} \;=\; p_k(x_i; W),$$

so the empirical loss satisfies $\mathcal{L}(cx, cW) = \mathcal{L}(x, W)$. The same holds trivially for 0-homogeneous d such as cosine, Bray–Curtis and Canberra.

Proof. The factor $c^{-\omega}$ cancels between numerator and denominator; summing $-\log p_{y_i}$ over the training set gives the statement. □

What this does and does not give. Proposition B.2 is invariance under the *joint* map $(x, W) \mapsto (cx, cW)$. It does *not* say that $\mathcal{L}(x, cW) = \mathcal{L}(x, W)$ with the data held fixed, and it therefore does not by itself imply that minimizers have finite norm: the loss can keep decreasing as $p_{y_i} \to 1$, and correct classification alone fixes only the arg max. Baek et al. (2025) prove a finite-minimizer result for the Euclidean harmonic loss under their own assumptions (their Sec. G, Thm. 1), and we cite that statement directly: we have not established it for the wider family of distances studied here, and a proof would have to handle 0-homogeneous distances, where prototypes are identified only up to scale, and the singularity of κ at $r = 0$. Empirically we observe no norm growth for any head we train.

Margin-based generalization. Baek et al. (2025) also give a PAC-Bayes margin bound (their Sec. G, Thm. 2) whose complexity term is controlled by a weight-norm capacity, and which is finite because their harmonic solution has finite norm. We do not restate it as a result of ours. Transferring such a bound to a general distance is not immediate: it requires an explicit prior and posterior, perturbation or Lipschitz conditions that depend on the distance, and a capacity measure appropriate to it — prototype norm is not a meaningful capacity for a scale-invariant cosine head. A finite bound is also not automatically an informative one. We therefore make no generalization claim, and we read the margin $\gamma(W) = \min_i[\min_{j \neq y_i} d(x_i, w_j) - d(x_i, w_{y_i})]$ only as a descriptive quantity.

C Integration into Deep Learning Pipelines

The `DistLayer` abstraction isolates the distance-based harmonic head from the rest of the training pipeline. Its `forward` method computes pairwise distances between representations and prototype weights and converts them to the harmonic probabilities of Equation (1). The vision and 7B classification heads use the equivalent log-space logits $-n \log(d_k + \varepsilon)$, whereas the language-model pretraining heads form normalized inverse-distance weights directly with the safeguards described in Appendix H. In both cases, changing the distance requires no change to the surrounding training loop.

Several design choices make the implementation robust. First, the distance functions use batched tensor operations and avoid explicit Python loops; their computational cost nevertheless depends on whether the distance admits a matrix-multiplication implementation and on the size of the output space (see Table 17). Second, numerical safeguards (e.g., ε-offsets, clamping before roots and divisions, regularization of covariance matrices) prevent instability across diverse datasets and architectures. Third, the registry-based design allows new distance functions to be added without disrupting the existing workflow, reinforcing the flexibility of harmonic loss as a general framework.

From a methodological perspective, this implementation highlights one of the key contributions of this work: the ease of replacing cross-entropy with distance-based harmonic loss. Unlike cross-entropy, which relies on unbounded logit growth, the harmonic formulation treats classification as a problem of minimizing distances to learned class prototypes. The common `DistLayer` interface therefore permits alternative geometries (e.g., cosine, Mahalanobis, and Bray–Curtis) to be evaluated with limited changes to the model code, although their computational costs can differ substantially.

D Model Architectures

D.1 Vision

We detail the architectures of the vision models used in our experiments – including a simple MLP, a small CNN, ResNet-50, and PVTv2-B0 – specifying their layers and neuron counts for reproducibility. All models were implemented in PyTorch, and for distance-based variants, the final fully-connected layer is replaced by a specialized distance layer as noted below.

MLP: **Input Layer:** Accepts the flattened image input (e.g., $28 \times 28 = 784$ features for MNIST, $32 \times 32 \times 3 = 3072$ for CIFAR). **Hidden Layer 1:** Fully-connected layer with 512 neurons, followed by ReLU. **Hidden Layer 2:** Fully-connected layer with 256 neurons, followed by ReLU. **Output Layer:** Linear mapping from 256 units to the number of classes (10 for MNIST/CIFAR-10, 100 for CIFAR-100). In `_DIST` variants, this layer is replaced with a distance-based classification head (e.g. Euclidean, cosine) that converts distances between the embedding and class prototypes into the harmonic probabilities of Equation (1).

CNN: **Conv Layer 1:** 2D convolution, 32 filters, kernel size 3×3, padding 1, followed by ReLU, then 2×2 max pooling. **Conv Layer 2:** 2D convolution, 64 filters, kernel size 3×3, padding 1, followed by ReLU, then 2×2 max pooling. **Fully Connected Layer:** Flattened output fed into a 128-unit linear layer with ReLU. **Output Layer:** Linear layer mapping the 128-D representation to the number of classes. In `_DIST` variants, this is replaced by a distance metric layer.

ResNet-50: **Stem:** Standard 7×7 convolution with 64 filters and stride 2, batch norm, ReLU, then 3×3 max pooling. For CIFAR/MNIST, we use a 3×3 conv with stride 1 and remove max pooling. **Stage 1:** 3 bottleneck blocks, output 256 channels. **Stage 2:** 4 bottleneck blocks, output 512 channels. **Stage 3:** 6 bottleneck blocks, output 1024 channels. **Stage 4:** 3 bottleneck blocks, output 2048 channels. **Global Pooling and Output:** Global average pooling yields a 2048-D vector. In the baseline, a linear FC layer maps to logits. In `_DIST` variants, the FC is replaced by a distance layer that produces harmonic class probabilities from prototype distances.

Pyramid Vision Transformer (PVTv2-B0): **Stage 1:** Overlapping patch embedding with a 7×7 conv (stride 4), output 32 channels, followed by 2 Transformer encoder layers (1 attention head). **Stage 2:** 3×3 conv (stride 2), output 64 channels, followed by 2 encoder layers (2 heads). **Stage 3:** 3×3 conv (stride 2), output 160 channels, followed by 2 encoder layers (5 heads). **Stage 4:** 3×3 conv (stride 2), output 256 channels, followed by 2 encoder layers (8 heads). **Global Pooling and Output:** Global average pooling yields a 256-D vector. A linear classifier maps to the number of classes in the baseline, while in `_DIST` variants this is replaced with a distance layer that produces harmonic class probabilities from prototype distances.

Preprocessing Pipelines: **MNIST:** For MLP/CNN, grayscale input normalized to mean 0.5, std 0.5. For ResNet, normalization uses dataset statistics (mean 0.1307, std 0.3081). For PVT, grayscale converted to 3 channels, resized to 32 (PVT), normalized to mean/std 0.5. **CIFAR-10:** Normalization with mean (0.4914, 0.4822, 0.4465) and std (0.2023, 0.1994, 0.2010). ResNet uses data augmentation (random flips, crops, small rotations). **CIFAR-100:** Normalization with mean (0.5071, 0.4867,

0.4408) and std (0.2675, 0.2565, 0.2761). Stronger augmentation (random flips, crops, rotations, color jitter). PVT models use 32×32 resized inputs with normalization.

D.2 LLMs

This section documents the **LLM** configurations used in our experiments for reproducibility. We report data preprocessing, architectural details for **GPT**, **BERT**, and **Qwen2**-style models, how **distance-based heads** are integrated in place of the standard linear classifier, and the training/evaluation/emissions-logging pipeline. All models are implemented in PyTorch and trained with mixed precision when available.

Data and Preprocessing Corpus and Storage. We pre-process a text corpus into contiguous token ID arrays and store them as memory-mapped files:

- `train.bin` and `val.bin`: `np.memmap` arrays of type `uint16` containing token IDs.
- `meta.pkl`: contains metadata including `vocab_size` (used to configure model embeddings).

Let V denote the discovered vocabulary size from `meta.pkl` (fallback $V{=}50304$ if not found).

Batching. For a given `block_size` L, batches are sampled by picking random starting indices and slicing L tokens:

`X = data[i : i+L], Y = data[i+1 : i+1+L]` (causal LM)

All batching is performed on-device with pinned memory. We denote `batch_size` by B.

Masking for MLM (BERT). For BERT runs, we construct masked language modeling (MLM) batches with the standard 15% corruption:

- Select $\approx 15\%$ token positions per sequence to form mask indices $\mathcal{M}$.
- For each $i \in \mathcal{M}$: with 80% probability replace x_i with `[MASK]` (id ≤ 103 or capped by $V{-}1$), with 10% replace by a random token in $[0, V)$, with 10% keep x_i unchanged.
- Labels use the original token at masked positions and -100 (ignore index) elsewhere.

This yields `input_ids`, `attention_mask` (all-ones here), and `labels` containing ground-truth only at masked positions.

Architectures Across models below, the principal hyperparameters are: layers (n_ℓ), heads (n_h), embedding dim (d), context length ($L{=}$`block_size`), vocab size (V). Unless otherwise specified, positional encodings follow each model's default (e.g., learned or rotary).

GPT2 (CAUSAL LM)

Backbone. A standard decoder-only Transformer with n_ℓ blocks. Each block has:

- Multi-Head Causal Self-Attention with n_h heads, hidden size d, and causal mask.
- Position-wise MLP of width typically $\approx 4d$ with nonlinearity (e.g., GELU).
- Pre/post LayerNorm and residual connections as in GPT-style decoders.

Token Embeddings. Learnable token and (implicit) position embeddings of sizes $V \times d$ and $L \times d$ (or rotary embeddings if enabled). **Projection Head (baseline).** A linear layer $W_{\text{lm}} \in \mathbb{R}^{d \times V}$ producing logits over the vocabulary at each position. **Distance Head (_DIST).** The linear projection is replaced by a distance layer that treats the vocabulary columns as prototypes $\{w_v \in \mathbb{R}^d\}_{v=1}^{V}$. Given a hidden state h_t, the layer computes $D(h_t, w_v; \Theta)$ for every token and converts these distances to the normalized harmonic probabilities of Equation (1); the implementation used in these pretraining runs is detailed in Appendix H.

BERT (MASKED LM)

Backbone. An encoder-only Transformer with n_ℓ layers, each with:

- Multi-Head Self-Attention (bidirectional) with n_h heads.
- Position-wise MLP, LayerNorm, residual connections.

Embeddings. Token embeddings $V \times d$, segment/type embeddings (size 2), and positional embeddings of length L. **Head (baseline).** The standard MLM classifier projects $d \to V$ (optionally via an intermediate nonlinearity tied to the embedding matrix). **Distance Head (_DIST).** We replace the MLM classifier with the same prototype-based distance layer used for GPT, applied only at masked positions. For each masked-token representation h_i, the head computes the harmonic token probabilities from $D(h_i, w_v; \Theta)$, and the negative log probability of the ground-truth token is minimized.

QWEN2-STYLE DECODER (CAUSAL LM)

Backbone. A decoder-only Transformer similar to GPT, with model-specific details:

- Rotary Position Embeddings (RoPE) with θ (e.g., $\theta = 10^6$).
- RMSNorm with ϵ (e.g., 10^{-6}) in place of LayerNorm.
- Grouped key/value heads: `num_key_value_heads` may be $< n_h$.
- Intermediate MLP width (`intermediate_size`) configurable.

Vocabulary. By default, we use Qwen's native vocabulary (`vocab_size`=151,936); alternatively, one can adapt to the dataset vocab. **Head (baseline vs. _DIST).** As with GPT, the final projection is either a linear layer to V or a distance-based head over V prototype vectors.

DISTANCE-BASED OUTPUT LAYER

For all three families (GPT, BERT/MLM, Qwen2), the baseline $d \to V$ classifier is replaced in `_DIST` runs by a head that first computes a nonnegative token–prototype distance:

$$d_v(h) \;=\; \begin{cases} \|h - w_v\|_2 & \text{(Euclidean)} \\ \|h - w_v\|_1 & \text{(Manhattan)} \\ \|h - w_v\|_p & \text{(Minkowski, } p \text{ specified)} \\ 1 - \frac{h^\top w_v}{\|h\|_2 \|w_v\|_2} & \text{(Cosine)} \\ D_{\text{Canberra}}(h, w_v) \;\text{ or }\; D_{\text{Bray–Curtis}}(h, w_v) & \text{(variants as defined)} \\ \|h - w_v\|_\infty & \text{(Chebyshev)} \\ \sqrt{(h - w_v)^\top \Sigma^{-1} (h - w_v)} & \text{(Mahalanobis, variants)} \\ D_{\text{Hamming}}(h, w_v) & \text{(soft/Gumbel/hard)} \end{cases}$$

Here w_v is a learned prototype vector. The distance is not used directly as a logit: it is transformed into harmonic probabilities, $p_v \propto (d_v + \varepsilon)^{-n}$, with the distance-specific transformations and numerical safeguards reported in Appendix H. The negative log probability of the target token is then minimized at each causal or masked position.

TRAINING SETUP AND OPTIMIZATION

Device and Precision. We use `bfloat16/float16/float32` (configurable) with automatic mixed precision:

```
torch.autocast(device_type='cuda', dtype=ptdtype).
```

Training can run in single-GPU or **DDP** (`torch.distributed`) multi-GPU mode. In DDP, `LOCAL_RANK` selects the device, and gradients are synchronized across ranks.

Initialization and Checkpointing. Models are initialized *from scratch* using the specified architecture config (layers, heads, width, L, V). For GPT-only runs we optionally support `init_from='gpt2*'`, and for BERT we support `init_from='bert*'` (when provided), with appropriate overrides. Checkpoints store model/optimizer state, `iter_num`, `best_val_loss`, and the configuration.

Optimizer and LR Schedule. We use the model's `configure_optimizers` helper to instantiate an Adam/AdamW-style optimizer with weight decay and (β_1, β_2). Learning rate follows cosine decay with warmup:

$$\mathrm{lr}(t)=\begin{cases}\mathrm{lr}_{\max}\cdot t/\text{warmup} & t<\text{warmup},\\ \mathrm{lr}_{\min}+\frac{1}{2}\Big(1+\cos\frac{\pi(t-\text{warmup})}{T-\text{warmup}}\Big)\big(\mathrm{lr}_{\max}-\mathrm{lr}_{\min}\big) & t\le T,\end{cases}$$

where T is `lr_decay_iters`. We apply gradient accumulation (`gradient_accumulation_steps`), optional gradient clipping (`grad_clip`), and AMP scaling (`GradScaler`).

Objectives.

- **GPT/Qwen2 (causal LM):** next-token cross-entropy over V at each position.
- **BERT (MLM):** cross-entropy computed only at masked positions; non-masked labels set to -100 (ignored).

Accuracy reporting: we compute token-level accuracy for monitoring (on next-token for causal LM, on masked tokens for MLM).

Evaluation and Early Signals. At fixed `eval_interval`, we run `estimate_loss()` over `eval_iters` batches on train/val splits (model in `eval()`), then resume training. Best validation loss checkpoints are saved; optional compile (`torch.compile`) can be enabled.

SUSTAINABILITY TRACKING

We integrate *CodeCarbon* to measure energy and emissions. At each evaluation interval:

1. Stop the tracker and record interval-level metrics: emissions (kg CO_2), duration, estimated CPU/GPU/RAM power and energy.
2. Log cumulative emissions and training metrics (loss, lr) to W&B (if enabled).
3. Restart the tracker for the next interval to avoid long-running file locks and to attribute emissions to training phases cleanly.

At the end of training, we stop the tracker one final time and persist all accumulated records to a CSV (`emissions_*.csv`) alongside model checkpoints.

Key Configurations (Reproducibility) The following knobs are saved in run configs/checkpoints and should be reported alongside results:

$$(n_\ell,\, n_h,\, d,\, L,\, V),\quad \text{distance head type and parameters } (\Theta),\quad \text{batch size } B,\ \text{precision},$$
$$\text{optimizer \& betas},\ \text{lr schedule } (\text{warmup},\, T,\, \mathrm{lr}_{\max},\, \mathrm{lr}_{\min}),\ \text{grad accumulation},$$
$$\text{grad clip},\ \text{DDP world size}.$$

When using `_DIST` variants, we also report which distance (Euclidean, cosine, Manhattan, Minkowski(p), Canberra, Bray–Curtis, Chebyshev, Mahalanobis, Hamming), any normalization/scaling flags, and regularization choices (e.g., Mahalanobis covariance learning/regularization).

In all models, the sole architectural change introduced by harmonic loss is confined to the **output head**: a drop-in replacement of the linear classifier with a **distance-based prototype head** over the vocabulary. This isolates the effect of the loss geometry while keeping the Transformer backbone (and training recipe) unchanged, enabling controlled comparisons across distances on *accuracy*, *interpretability* (e.g., PCA-based analyses), and *sustainability* (emissions and runtime).

E HYPERPARAMETER CONFIGURATIONS

The hyperparameter settings in Tables 4–11 were chosen to balance *comparability*, *training stability*, and *sustainability*. Below we highlight several important considerations.

E.1 LANGUAGE MODELS (OPENWEBTEXT)

Table 4 specifies the core training parameters for GPT, BERT, and Qwen on OpenWebText. The main goal was to maintain a fair comparison across models of varying scale by using effective batch sizes of similar order (76–128). This ensures that any differences observed in performance or emissions are attributable to the *loss formulation*, not simply to batch scaling. The use of AdamW with default β values (0.9, 0.999) follows standard practice for stability.

Table 5 details architecture-specific modifications. BERT includes type embeddings and a masked language modeling (MLM) setup, while GPT and Qwen use causal language modeling (CLM). Qwen, being substantially larger, incorporates more advanced design elements such as grouped query attention (GQA) and rotary position embeddings (RoPE). Table 6 summarizes these differences: GPT and Qwen follow causal objectives, while BERT relies on bidirectional context, which may affect the degree to which distance-based losses interact with their representations.

Table 4: Core configuration for GPT, BERT, Qwen, and GPT-2B on OpenWebText.

Configuration	**GPT**	**BERT**	**Qwen**	**GPT-2B**
n_{layer}	12	12	24	48
n_{head}	12	12	14	20
n_{embd}	768	768	896	1600
Vocab size	50304	50304	151936	50304
Dropout	0.1	0.1	0.0	0.1
Bias	True	True	True	True
Batch size	16	38	6	3
Grad. accum. steps	8	2	10	21
Effective batch size	128	76	60	63
Learning rate	2e-4	1e-4	1e-4	1e-4
Warmup iters	500	1000	1000	1000
Weight decay	0.01	0.01	0.01	0.01
Grad clip	1.0	1.0	1.0	1.0
Min LR	2e-6	1e-6	1e-6	1e-6
Decay LR	True	True	True	True
LR decay iters	10000	10000	10000	10000
Max iters	10000	10000	10000	10000
Dataset	OpenWebText	OpenWebText	OpenWebText	OpenWebText
dtype	bfloat16	bfloat16	bfloat16	bfloat16
Optimizer	AdamW	AdamW	AdamW	AdamW
β_1, β_2	0.9, 0.999	0.9, 0.999	0.9, 0.999	0.9, 0.999
Eval interval	1000	1000	1000	500
Eval iters	100	100	100	25
Log interval	50	50	50	25
Scale attn by inverse layer idx	False	False	False	False

E.2 VISION MODELS

Tables 7–10 provide the vision settings across datasets. As shown in Table 7, optimizer and learning-rate schedules are backbone-specific: Adam for MLPs and CNNs, AdamW for transformers (PVT), and SGD with momentum for ResNet50. This reflects both convention and empirical stability in preliminary experiments. Batch size selection (Table 8) reflects hardware utilization on H100 GPUs. Lightweight backbones (e.g., CNNs) leverage very large batches (up to 8192 for MNIST), while PVT uses dataset-dependent batches of 64–512 to fit memory constraints. These design choices affect emissions profiles: large-batch training can reduce wall-clock time but at the cost of GPU

Table 5: Architecture-specific settings for GPT, BERT, Qwen, and GPT-2B.

Configuration	GPT	BERT	Qwen	GPT-2B
Block size / Seq length	1024	512	1024	512
Type vocab size	–	2	–	–
Pad token id	–	0	–	–
MLM probability	–	0.15	–	–
Intermediate size	–	–	4864	–
# key-value heads	–	–	2	–
RMSNorm ϵ	–	–	1e-6	–
RoPE θ	–	–	1,000,000.0	–

Table 6: Key differences summary (task and position encoding).

Aspect	GPT	BERT	Qwen	GPT-2B
Model size (approx.)	∼124M	∼110M	∼494M	∼2B
Attention	Causal	Bidirectional	Causal (GQA)	Causal
Training task	CLM	MLM	CLM	CLM
Position encoding	Learned	Learned	RoPE	Learned

CLM = Causal Language Modeling; MLM = Masked Language Modeling; GQA = Grouped Query Attention.

memory overhead. Learning-rate schedulers differ across models. For example, PVT employs cosine annealing, which smooths convergence and interacts well with distance-based loss formulations. ResNet50 relies on multi-step decay, ensuring stability across the long 200-epoch training horizon on CIFAR-100.

Distance Layer Parameters. Table 11 summarizes the shared hyperparameters across all distance functions. The exponent n is fixed to 1.0 and $\varepsilon = 10^{-4}$ provides numerical stability. Importantly, distances are not scaled post hoc, ensuring that differences in results are directly attributable to the geometric properties of the chosen distance (Euclidean, Manhattan, Mahalanobis, etc.), with auxiliary tuning held fixed.

Table 7: Core training configuration by backbone and dataset.

Configuration	MLP	CNN	PVT	ResNet50
LR (MNIST)	3e-4	3e-4	1e-3	0.1
LR (CIFAR-10)	3e-4	3e-4	1e-3	0.1
LR (CIFAR-100)	3e-4	3e-4	5e-4	0.1
LR (MarathiSign)	1.5e-4	1.5e-4	5e-4	0.05
LR (TinyImageNet)	3e-4	3e-4	5e-4	0.1
Epochs (MNIST)	40	40	80	100
Epochs (CIFAR-10)	40	40	80	100
Epochs (CIFAR-100)	150	150	150	200
Epochs (MarathiSign)	50	50	100	75
Epochs (TinyImageNet)	100	100	200	150
Optimizer	Adam	Adam	AdamW	SGD
Weight decay	0	0	0.01	1e-4
Momentum	–	–	–	0.9

E.3 Discussion

Language models. GPT and BERT use comparable depth/width with learned positional encodings, while Qwen is larger, adopts RoPE, and GQA. Effective batch sizes (via gradient accumulation) normalize throughput across models for fair comparison on OpenWebText.

Table 8: Batch size configuration on H100 GPU.

Model	MNIST	CIFAR-10	CIFAR-100	MarathiSign	TinyImageNet
MLP	2048	1024	1024	128	256
CNN	8192	4096	512	512	512
PVT	256	512	256	64	128
ResNet50	512	512	256	128	256

Table 9: Learning-rate schedulers by backbone and dataset.

Model	MNIST	CIFAR-10	CIFAR-100	MarathiSign	TinyImageNet
MLP	None	None	StepLR (50, 0.5)	ReduceLR*	StepLR (50, 0.5)
CNN	None	None	StepLR (50, 0.5)	ReduceLR*	StepLR (50, 0.5)
PVT	CosineAnn. (80)	CosineAnn. (80)	CosineAnn. (150)	CosineAnn. (100)	CosineAnn. (200)
ResNet50	StepLR (30, 0.1)	StepLR (30, 0.1)	MultiStep**	StepLR (25, 0.1)	MultiStep***

*ReduceLROnPlateau (mode=max, factor=0.5, patience=5, min_lr=1e-6)
**MultiStepLR (milestones=[60,100,140], γ=0.2)
***MultiStepLR (milestones=[80,120], γ=0.2)

Vision models. Optimizer and scheduler choices follow common practice: Adam/AdamW for MLP/CNN/PVT, SGD with momentum for ResNet50; deeper/longer CIFAR-100 runs employ stepped or cosine schedules. Early-stopping patience scales with dataset difficulty.

DistLayer defaults. A unified setting (n=1.0, ε=10^{-4}, no scaling) ensures distance variants differ only in geometry, not in auxiliary hyperparameters. These settings match the configuration used in our main experiments and figures.

Table 10: Dataset metadata and early-stopping settings (vision).

Parameter	MNIST	CIFAR-10	CIFAR-100	MarathiSign	TinyImageNet
Num classes	10	10	100	43	200
Early stopping patience	15	15	25	10	15
Min improvement (%)	0.01	0.01	0.01	0.01	0.01
Native image size	$28\times28\times1$	$32\times32\times3$	$32\times32\times3$	varies	$64\times64\times3$
Processed size (MLP/CNN/PVT)	$28\times28\times1$	$32\times32\times3$	$32\times32\times3$	$32\times32\times3$	$224\times224\times3$
Processed size (ResNet50)	$28\times28\times1$	$32\times32\times3$	$32\times32\times3$	$224\times224\times3$	$224\times224\times3$

Table 11: Distance-layer shared parameters (all backbones).

Parameter	Value
n	1.0
ε	1e-4
Scale distances	False

F Evaluation Metrics: Definitions

Note on the vision PCA metrics. In the vision pipeline, PCA is fitted with at most 50 components, so PCA@90% (the number of components needed to reach 90% explained variance) is right-censored: a reported value of 50.0 means "≥ 50". Explained-variance metrics are unaffected. The language studies use full-rank PCA. See Appendix A. This appendix gives the definitions of the language model metrics summarized in Section 4.1.

Perplexity (Train / Val). Given a sequence of targets $\{y_t\}_{t=1}^{T}$ and model probabilities $p_\theta(y_t \mid \text{context})$, where θ denotes the model parameters, the average negative log-likelihood is:

$$\mathcal{L}_{\text{NLL}} = -\frac{1}{T}\sum_{t=1}^{T} \log p_\theta\big(y_t \mid \text{context}\big),$$

and the corresponding perplexity is $\text{PPL} = \exp\big(\mathcal{L}_{\text{NLL}}\big)$. Lower perplexity indicates better next-token prediction. [2]

Gradient Stability (GS). To quantify the smoothness of optimization, we measure the variance of the L_2-norm of the gradient across consecutive training steps. Let $\mathcal{L}_t$ denote the training loss at step t; then:

$$\text{GS} = 1 - \frac{\text{Var}\left(\|\nabla_\theta \mathcal{L}_t\|_2\right)}{\text{Var}\left(\|\nabla_\theta \mathcal{L}_t\|_2\right)_{\text{CE}}},$$

where $\text{Var}(\cdot)$ is computed over a fixed evaluation window (*e.g.*, 500 steps) and the denominator corresponds to the variability under cross-entropy (CE). This metric is motivated by gradient variance analyses (Faghri et al., 2020; Lei & Ying, 2020), which demonstrate that gradient variance dynamics correlate with training stability and convergence behavior. Thus, $\text{GS}=0$ indicates equal smoothness as CE, $\text{GS} > 0$ indicates reduced gradient variance (smoother training), and $\text{GS} < 0$ reflects more unstable gradient dynamics (higher variance than CE). This metric is anchored in standard variance-of-gradient analyses used in optimizing large-scale LLMs.

Effective Rank. This metric is inspired by analyses of dimensional collapse (Jing et al., 2022; Bardes et al., 2022), that track the covariance spectrum of learned representations to detect embedding collapse and maintain representation quality in neural networks. Let $\Sigma_h = \text{Cov}(\mathbf{h}_t)$ with eigenvalues

[2] For visualization in the radar plots, we invert perplexity (and all metrics where lower values indicate better performance) and then normalize to the range $[0, 10]$ relative to the harmonic Euclidean baseline. The effect is that the greater the coverage on the plot, the better the relative performance compared to Euclidean. Importantly, the absolute numeric values on the radial axis do not have a direct "good/bad" interpretation in perplexity space; they are meaningful only as normalized, experiment-specific comparisons against the Euclidean harmonic.

$\{\lambda_k\}_{k=1}^{d}$. Define the normalized eigenvalues $p_k = \lambda_k / \sum_j \lambda_j$. The effective rank is:

$$\mathrm{ER}(\Sigma_h) = \exp\left(-\sum_{k=1}^{d} p_k \ln p_k \right).$$

This is the exponential of the Shannon entropy of the eigenvalue distribution (Garrido et al., 2023) (Roy & Vetterli, 2007). Metric values are in the range $[1, d]$, where a value of 1 means total collapse (one dimension dominates), and value of d means perfectly uniform spread. Higher values indicate that the learned weight matrices retain higher intrinsic dimensionality, suggesting the distance metric discourages low-rank collapse in the parameter space and is consistent with more distributed use of the embedding capacity.

G Distance Catalogue: Details

This appendix gives the full description of each distance summarized in Table 1.

Euclidean. Baseline Euclidean distance between feature and prototype:

$$d_{\text{Euclidean}}(\mathbf{h}, \mathbf{w}) = \|\mathbf{h} - \mathbf{w}\|_2.$$

Manhattan (L_1). The L_1 norm emphasizes absolute differences, providing robustness to outliers since large deviations contribute linearly to the total distance (Keeling & Kunisch, 2016; Brooks et al., 2013; Giloni & Padberg, 2004).

$$d_{\text{Manhattan}}(\mathbf{h}, \mathbf{w}) = \|\mathbf{h} - \mathbf{w}\|_1.$$

Computing L_1 distance also requires only additions and subtractions without squaring or square-root operations, making it computationally efficient. Recent work on AdderNets demonstrates that replacing multiplication-based convolutions with L_1-based operations can reduce energy consumption while maintaining competitive accuracy (Chen et al., 2020).

Chebyshev (L_∞).

$$d_{\text{Chebyshev}}(\mathbf{h}, \mathbf{w}) = \|\mathbf{h} - \mathbf{w}\|_\infty.$$

Captures the maximum coordinate deviation, offering a highly interpretable measure of the most discriminative feature dimension. Its simplicity makes it computationally efficient.

Minkowski (L_p). Generalizes both L_1 and L_2, with a tunable exponent p that controls the trade-off between robustness and sensitivity (Aggarwal et al., 2001; Hu et al., 2016):

$$d_{\text{Minkowski}}(\mathbf{h}, \mathbf{w}; p) = \|\mathbf{h} - \mathbf{w}\|_p.$$

Lower values of p (closer to 1) down-weight large individual coordinate differences, yielding behavior more robust to outliers, while higher values (toward 2 or above) increase sensitivity to larger deviations. This flexibility allows the metric to be tailored to dataset characteristics, treating p as a hyperparameter to optimize accuracy. In our experiments, we do not perform an unrestricted search over p. We use p=1.5 and p=3 as representative non-Euclidean points on either side of the Euclidean norm, while p=2 is retained as a sanity-check configuration corresponding to standard L_2 geometry. Preliminary runs with p>3 did not improve the accuracy–structure–emissions trade-off and were therefore excluded from the main analysis.

Cosine. Ignores vector magnitudes and instead measures angular similarity, making it particularly effective for high-dimensional embeddings (Reimers & Gurevych, 2019; Deng et al., 2019; Wang et al., 2018; Sun et al., 2016; Karpukhin et al., 2020).

$$d_{\text{cosine}}(\mathbf{h}, \mathbf{w}) = 1 - \frac{\mathbf{h}^\top \mathbf{w}}{\|\mathbf{h}\|_2 \, \|\mathbf{w}\|_2}.$$

Because the output is bounded, cosine similarity decreases activation variance compared to unbounded dot products, which can improve generalization (Luo et al., 2018). The computational overhead is minimal, only requiring an additional normalization step beyond the standard dot product.

Hamming. Counts mismatches directly, providing a highly interpretable signal: each unit of distance corresponds to exactly one differing feature dimension.

$$d_{\text{Hamming}}(\mathbf{h}, \mathbf{w}) = \frac{1}{d} \sum_{i=1}^{d} \mathbf{1}_{\{h_i \neq w_i\}}.$$

Although inherently discrete, Hamming distance can be integrated into neural network training via continuous relaxations such as the Gumbel-Softmax trick (Jang et al., 2017; Maddison et al., 2017), which provide differentiable approximations to binary sampling. We instantiate three gradient-compatible Hamming variants. The soft variant applies a temperature-scaled sigmoid to features and prototypes before computing the average mismatch. The Gumbel variant adds numerically stabilized Gumbel noise before the sigmoid during training, and falls back to deterministic sigmoid binarization at evaluation time. The hard variant uses a straight-through estimator: the forward pass thresholds features and prototypes into binary codes, while the backward pass routes gradients through a temperature-scaled sigmoid approximation. Binary embeddings offer substantial efficiency gains – comparing binary codes via Hamming distance can be significantly faster and more memory-efficient than comparing full-precision vectors.

Canberra. Normalizes each coordinate difference by the sum of the magnitudes, yielding heightened sensitivity when both values are near zero (Lance & Williams, 1967).

$$d_{\text{Canberra}}(\mathbf{h}, \mathbf{w}) = \sum_{i=1}^{d} \frac{|h_i - w_i|}{|h_i| + |w_i| + \varepsilon}.$$

Since each term is bounded in $[0, 1]$, the metric emphasizes proportional differences. This property can benefit fine-grained recognition tasks in which small variations in low-magnitude features carry important discriminative information.

Bray-Curtis. A normalized variant of the L_1 distance that measures proportional differences across all features (Fuschi et al., 2025; Bray & Curtis, 1957; Song et al., 2020).

$$d_{\text{Bray-Curtis}}(\mathbf{h}, \mathbf{w}) = \frac{\sum_{i=1}^{d} |h_i - w_i|}{\sum_{i=1}^{d} (|h_i| + |w_i|) + \varepsilon}.$$

Widely used in ecology for comparing species abundance profiles, Bray–Curtis produces values in $[0, 1]$ that are easy to interpret: 0 indicates identical composition, 1 indicates no overlap. Because it does not require estimating a covariance matrix, it offers a computationally lighter alternative to Mahalanobis distance while still accounting for relative differences. We report three Bray–Curtis variants consistently with the implementation. The displayed equation corresponds to the *Absolute* variant, which uses the absolute-sum denominator $\sum_i (|h_i| + |w_i|) + \varepsilon$ and is the safest form for signed neural features. The *Standard* variant keeps raw embeddings and uses the signed denominator $\sum_i (h_i + w_i)$ before numerical absolute-value clamping. The *Normalized* variant first maps features and prototypes to positive normalized composition vectors and then applies the same proportional discrepancy, isolating compositional differences from vector scale.

Mahalanobis. Incorporates feature correlations by weighting distances according to the inverse covariance matrix Σ^{-1}, effectively measuring distances in a whitened feature space (Pang et al., 2018; Lee et al., 2018; Gómez-Silva et al., 2021; Omara et al., 2021).

$$d_{\text{Mahalanobis}}(\mathbf{h}, \mathbf{w}; \Sigma) = \sqrt{(\mathbf{h} - \mathbf{w})^\top \Sigma^{-1} (\mathbf{h} - \mathbf{w})}.$$

This can improve discrimination when features have different scales or are correlated; for example, Lee et al. (2018) achieve state-of-the-art out-of-distribution detection in deep networks using Mahalanobis distance on CNN features. The computational trade-off is that estimating and inverting the covariance matrix incurs overhead, particularly in high dimensions. However, the resulting distance is scale-invariant and unitless, facilitating the interpretability of class boundaries as ellipsoids in feature space.

In our framework, we generalize the harmonic loss by replacing the Euclidean distance in Equation (1) with one of the distances defined above. This substitution is applied only at the final classification layer, replacing the standard softmax cross-entropy head. The feature extraction backbone remains unchanged; we do not introduce additional normalization layers or architectural modifications beyond the classification head.

Compared to standard cross-entropy, distance-based harmonic losses offer several potential benefits: the scale-invariant formulation avoids unbounded logit growth, the geometric interpretation provides transparency (class decisions correspond to proximity to prototype vectors), and certain distance choices (*e.g.*, L_1, cosine) can reduce computational cost.

A formal treatment of our distance-based probabilistic layer is provided in Appendix B. There, we record the scale invariance of the harmonic link for the distance families studied here, and explain why the finite-minimizer and PAC-Bayes results of Baek et al. (2025) are not restated as results of ours. These results clarify when geometry choices are well-posed and why the resulting classifiers admit standard generalization guarantees.

H SmolLM2 Study: Setup and Additional Results

This appendix provides the setup and additional results for the SmolLM2 study of Section 4.3. The scripts in the repository's `llms/analysis` folder generate all reported values from the evaluation logs and CodeCarbon records.

H.1 Setup

Model and data. We use the SmolLM2 architecture (Ben Allal et al., 2025), a Llama-style decoder, trained from scratch (Table 12). The corpus is MiniPile (Kaddour, 2023), tokenized with the GPT-2 BPE vocabulary used by our other language models; a 5,000-step run consumes less than one epoch. The output head is the only architectural component varied across runs. The backbone is initialized before the output head, so runs that share a seed begin with identical backbone weights; the seed also determines the order in which training windows are sampled.

Heads. This study uses $n{=}d$ throughout, following Baek et al. (2025). Every comparison fixes the exponent across heads, so the reported differences isolate head type under this common exponent. We do not claim that head rankings are invariant to n; the exponent ablation in the pretrained 7B study instead favors $n{=}\sqrt{d}$ for output spaces with a few hundred classes (see Table 21). Two details of the optimized objective matter for reproduction: the training loss applies label smoothing of 0.1 and is clamped to $[0, 20]$, so its exponential is *not* perplexity. Every perplexity and NLL we report is recomputed from the same logits without smoothing or clamping, on a fixed set of validation windows shared by all heads. For the Euclidean, Minkowski, and Hamming heads, probabilities follow Equation (1) with exponent $n{=}d{=}2048$, after dividing each token's distances by d and by their minimum over the vocabulary. The Euclidean head applies the exponent to the *squared* distance, following Baek et al. (2025), whereas the Minkowski head with $p{=}2$ applies it to the unsquared distance. In the reported runs, their perplexities differ by 0.1 and their PCA@90% values by 91 dimensions (see Table 13). Cosine heads use $p_v \propto \big((1 + \cos(\mathbf{h}, \mathbf{w}_v))/2\big)^m$ with $m{=}\sqrt{d} \approx 45$ or $m{=}16$. With $m{=}1$, a token orthogonal to the hidden state receives half the weight of a perfectly aligned token, leaving the distribution too diffuse over 50,304 vocabulary items; this run ended at perplexity 1,380 (see Table 13). Soft Hamming maps features and prototypes through a sigmoid and uses the mean absolute difference, $d_{\text{Hamming}}(\mathbf{h}, \mathbf{w}) = \frac{1}{d}\sum_i |\sigma(h_i) - \sigma(w_i)|$. Unlike Euclidean and cosine distances, this has no matrix-multiplication form, so every token–prototype pair requires a pass over all d coordinates. Minkowski distances with $p \neq 2$ likewise require coordinate-wise computation. At $p{=}1.5$, the measured step time is $9\times$ that of cross-entropy (see Table 17). We therefore use a separate 1,000-step run for the matched-perplexity comparison instead of extending it to 5,000 steps (see Table 16).

H.2 Evaluation Protocol

Each final checkpoint is evaluated on the same 819,200 validation tokens (200 batches of $4 \times 1{,}024$ windows drawn with a fixed evaluation seed), yielding perplexity, top-1 accuracy, and ECE with 15 equal-width confidence bins. PCA metrics use 16,384 final hidden states (the output of the last RMSNorm over 16 validation sequences), standardized per feature, and the full 2,048-dimensional spectrum $\lambda_1 \geq \cdots \geq \lambda_d$. We report PC1 EV and PC2 EV (the variance fraction of the first and of the first two components), PCA@90% and PCA@95% (components needed to reach that fraction), the participation ratio $(\sum_k \lambda_k)^2 / \sum_k \lambda_k^2$, and the effective rank of Section 4.1 computed on the same spectrum. For all but PC1 and PC2 EV, lower values mean more concentrated representations. Table 13 lists every run.

Table 12: SmolLM2 training configuration, identical for every head.

Setting	**Value**
Layers / attention heads / KV heads	24 / 32 / 32
Hidden size d / MLP width (SwiGLU)	2048 / 8192
Normalization / positions	RMSNorm (ϵ=10^{-5}) / RoPE (θ=1.3×10^5)
Biases / dropout	none / 0.0
Non-embedding parameters	1.61B
Vocabulary	GPT-2 BPE, 50,304 (padded)
Context length	1,024
Batch	4 sequences $\times$ 16 accumulation steps = 65,536 tokens
Steps	5,000 (328M tokens); references also 500, 1,000, 2,000
Optimizer	AdamW, β=$(0.9, 0.999)$, weight decay 0.01
Learning rate	10^{-4} peak, 10^{-6} final; linear warmup over the first 20% of steps, cosine decay to the last step
Gradient clipping	1.0
Precision / attention	bfloat16 autocast / PyTorch SDPA, `torch.compile`
Seeds	5000 and 1 (cosine m=$\sqrt{d}$: also 2)
Hardware / software	one NVIDIA H100 per run / PyTorch 2.6, CUDA 12.4, CodeCarbon 3.3

Table 13: Every full-length SmolLM2 run after 5,000 steps. PR: participation ratio. Cosine (m=1) and Manhattan (fast) remain at perplexities $1{,}380$ and $34{,}428$, respectively, and are omitted from aggregate comparisons.

Head	Seed	PPL	Top-1	ECE	PC1 EV	PC2 EV	PCA@90%	PCA@95%	PR	kWh
Cross-entropy	5000	21.79	.480	.0527	.081	.139	626	1019	63.2	2.96
Cross-entropy	1	21.85	.480	.0526	.081	.141	623	1015	62.5	3.38
Euclidean	5000	21.43	.480	.0425	.083	.129	541	923	61.0	3.54
Euclidean	1	21.45	.480	.0421	.082	.127	548	929	62.8	3.61
Minkowski (p=2)	5000	21.54	.480	.0475	.076	.131	636	1031	67.7	3.60
Cosine (m=$\sqrt{d}$)	5000	21.85	.479	.0512	.073	.126	670	1062	74.2	3.06
Cosine (m=$\sqrt{d}$)	1	21.88	.479	.0521	.074	.128	667	1059	73.1	3.00
Cosine (m=$\sqrt{d}$)	2	21.88	.479	.0516	.073	.126	672	1064	74.3	3.00
Cosine (m=16)	5000	32.86	.461	.0967	.053	.090	300	520	83.4	2.99
Hamming (soft)	5000	42.98	.417	.0356	.136	.200	244	488	32.7	26.55
Hamming (soft)	1	43.24	.417	.0360	.140	.205	242	487	31.4	26.08
Cosine (m=1)	5000	1380.31	.030	.0112	.310	.534	6	8	5.3	–
Manhattan (fast)	5000	34428.07	.014	.0144	.062	.114	320	649	57.1	3.07

H.3 Matched-Perplexity Comparison

Protocol. The cross-entropy and Euclidean references are trained for 500 (cross-entropy only), 1,000, 2,000, and 5,000 steps. Each short run warms up over its first 20% of steps and decays its learning rate to the final value at its last step, so each reference completes its schedule within its budget. For each reference and metric, we average seeds at each budget, giving the curves in Table 14 and Figure 3. A head at perplexity P is compared with the reference value interpolated linearly in $\log P$ between the two budgets whose perplexities bracket P; heads outside a reference's perplexity range are not compared with it. As a robustness check, we mark a difference with † when the head also outperforms both bracketing reference runs, in which case the comparison does not depend on interpolation. The pre-specified threshold ($+0.014$ PC2 EV and 33 fewer PCA@90% dimensions) was set before any matched comparison was computed, at ten times the pooled seed standard deviations (0.0014 and 3.3).

Results. Only Hamming (soft) exceeds both pre-specified thresholds against both references (see Table 16). Against cross-entropy, Hamming also outperforms both bracketing checkpoints on every concentration metric and on ECE (see Table 15). Against Euclidean, the bracketing-checkpoint comparison holds for PCA@90%, PCA@95%, and the effective rank of the unstandardized hidden

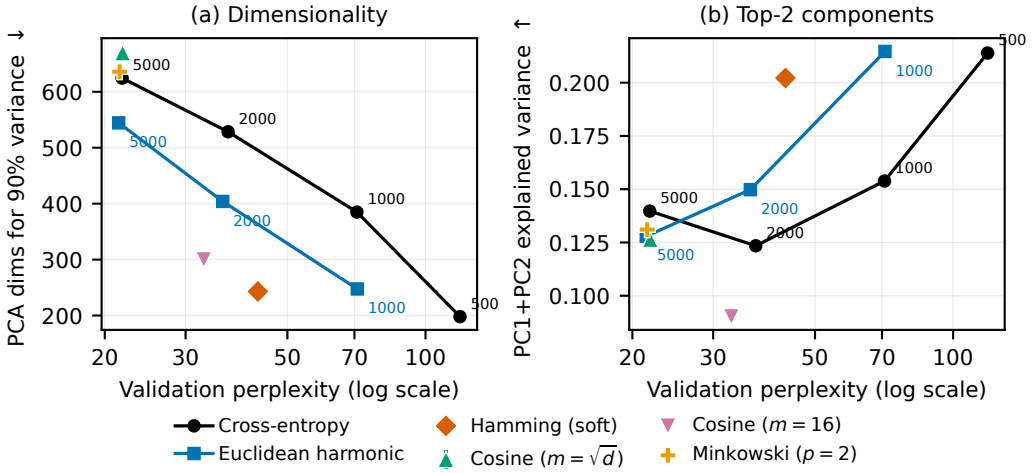

Figure 3: SmolLM2: representation concentration against perplexity. Lines show cross-entropy and Euclidean reference checkpoints trained for 500–5,000 steps (labels), each with a completed learning-rate schedule; markers show the other heads after 5,000 steps. Along both reference curves, checkpoints with higher perplexity have more concentrated representations. At matched perplexity, a head is more concentrated when it lies below the reference curve in (a) and above it in (b); only Hamming (soft) satisfies both conditions against both references.

Table 14: Reference curves: cross-entropy and Euclidean harmonic loss at each step budget, mean over n seeds.

Reference	Steps	n	PPL	Top-1	ECE	PC1 EV	PC2 EV	PCA@90%	PCA@95%
Cross-entropy	500	1	118.90	.305	.036	.140	.214	198	501
Cross-entropy	1,000	2	70.87	.349	.041	.097	.154	385	763
Cross-entropy	2,000	2	37.14	.424	.052	.075	.123	529	927
Cross-entropy	5,000	2	21.82	.480	.053	.081	.140	625	1017
Euclidean	1,000	2	71.03	.345	.031	.146	.215	248	561
Euclidean	2,000	2	36.14	.423	.040	.094	.150	404	769
Euclidean	5,000	2	21.44	.480	.042	.082	.128	545	926

states. It does not hold for PC1 or PC2 EV: the 1,000-step Euclidean checkpoint (perplexity 71) has higher PC1 and PC2 EV than Hamming and nearly the same PCA@90% (248 vs. 243). Across this interval, each Euclidean metric changes monotonically over the measured checkpoints, so the interpolated reference lies between the two bracketing values. Outperforming both checkpoints is therefore a stricter sufficient condition than outperforming the interpolated reference. At matched perplexity, Euclidean harmonic loss requires 85 fewer PCA@90% dimensions than cross-entropy but has 0.011 lower PC2 EV. Its clearest improvements over cross-entropy are therefore in perplexity and calibration.

H.4 Gradient Statistics and Training Cost

Table 17 reports gradient statistics and cost. Clip % is the fraction of logged steps whose gradient norm exceeds the clipping threshold, and $\mathrm{CV}_{\mathrm{late}}$ is the coefficient of variation of the gradient norm over the second half of training, a scale-free counterpart of the gradient stability (GS) score of Section 4.1. $\mathrm{CV}_{\mathrm{late}}$ varies by up to a factor of two between seeds of the same head (cross-entropy: 0.052 and 0.104), comparable to its variation between heads. We therefore do not attribute the observed differences in gradient stability to head type at this scale. Clipping, in contrast, is reproducible: the Euclidean head clips on 24% of logged steps against 11% for cross-entropy. Runs shared a node with other jobs, so step time varied between otherwise identical runs (cross-entropy: 4.2 vs. 5.5 s per

Table 15: Hamming (soft), perplexity 43.11 (mean of two seeds), against each reference interpolated to the same perplexity. †: Hamming also outperforms both bracketing reference runs (1,000 and 2,000 steps).

Metric	Hamming	Cross-entropy	Δ	Euclidean	Δ
Top-1 accuracy ↑	.417	.406	+.010	.403	+.014
ECE ↓	.0358	.0499	−.0141†	.0375	−.0017
PC1 EV ↑	.138	.080	+.058†	.108	+.031
PC2 EV ↑	.202	.130	+.072†	.167	+.035
PCA@90% ↓	243	495	−252†	363	−120†
PCA@95% ↓	488	889	−401†	715	−227†
Effective rank ↓	133.4	216.5	−83.1†	152.3	−19.0
Participation ratio ↓	32.0	56.6	−24.5†	39.8	−7.8
Effective rank, raw states ↓	610.0	914.6	−304.6†	742.3	−132.3†

Table 16: Matched-perplexity comparisons using the pre-specified thresholds of $+0.014$ PC2 EV and at least 33 fewer PCA@90% dimensions. A negative PCA@90% reduction means that the head uses more dimensions than its reference; † is defined in Table 15. Minkowski (p=2) lies below the cross-entropy interpolation range and is therefore compared only with Euclidean. The cross-entropy row reports its comparison with the Euclidean reference. Minkowski (p=1.5) is a separate 1,000-step run outside the Euclidean range.

Head	PPL	Reference	PC2 EV gain	PCA@90% reduction	Meets threshold
Hamming (soft)	43.11	Cross-entropy	+0.072†	252†	yes
Hamming (soft)	43.11	Euclidean	+0.035	120†	yes
Cosine (m=16)	32.86	Cross-entropy	−0.037	251†	no
Cosine (m=16)	32.86	Euclidean	−0.056	130†	no
Cosine ($m=\sqrt{d}$)	21.87	Cross-entropy	−0.013	−46	no
Cosine ($m=\sqrt{d}$)	21.87	Euclidean	−0.002	−130	no
Cross-entropy	21.82	Euclidean	+0.011	−85	no
Minkowski (p=2)	21.54	Euclidean	+0.003	−93	no
Minkowski (p=1.5)	75.22	Cross-entropy	+0.007	49	no

step). Energy closely followed runtime, with an average draw of 0.44–0.48 kW for all heads except Hamming, so total energy primarily reflects measured runtime under these shared-node conditions. The original Minkowski (p=1.5) run provides gradient and cost records through 800 steps; a separate quality-control run completed 1,000 steps and provides the matched-perplexity metrics in Table 16.

Table 17: Gradient statistics and training cost for the full-length runs, mean ± s.d. over seeds. Minkowski (p=1.5) is summarized from the 800 steps available in its original run; its separate 1,000-step matched-perplexity run is reported in Table 16.

Head	n	Clip %	CV_{late}	s/step	kWh	kg CO_2
Cross-entropy	2	10.6 ± 0.3	0.078 ± 0.037	4.86 ± 0.87	3.17 ± 0.30	0.69 ± 0.07
Euclidean	2	23.7 ± 0.7	0.082 ± 0.012	5.82 ± 0.15	3.57 ± 0.05	0.78 ± 0.01
Minkowski (p=2)	1	11.4	0.066	5.94	3.60	0.79
Cosine ($m=\sqrt{d}$)	3	10.1 ± 0.1	0.073 ± 0.034	4.59 ± 0.10	3.02 ± 0.04	0.66 ± 0.01
Cosine (m=16)	1	1.8	0.106	4.50	2.99	0.65
Hamming (soft)	2	15.4 ± 0.0	0.050 ± 0.012	33.44 ± 0.24	26.31 ± 0.33	5.75 ± 0.07
Minkowski (p=1.5)	1	72.8	0.162	44.43	–	–

H.5 WEIGHT-SPACE PCA

We also apply PCA to the backbone weights of the final checkpoints, using every fourth attention and MLP weight matrix and standardizing each parameter across runs. The first component primarily reflects the initialization seed: runs of the same head with different seeds lie 2–4 times farther apart

than runs with different heads and a shared seed. The distances between heads are consistent across the two replicated seeds. Measured in units of 10^3 in the standardized weight space, the distance from the cross-entropy backbone to the cosine ($m{=}\sqrt{d}$) backbone is 10.0 for both seed 5000 and seed 1; the corresponding distances to the Euclidean backbone are 15.5 and 15.7. By comparison, the two cross-entropy backbones initialized with different seeds are 38.9 units apart. We therefore restrict weight-space comparisons between losses to runs with a shared initialization. Among the seed-5000 runs, the backbones trained with Hamming (soft) and cosine ($m{=}16$) have the largest distances from cross-entropy (17.5 and 17.7), while Minkowski ($p{=}2$) and cosine ($m{=}\sqrt{d}$) have the smallest (9.1 and 10.0).

I Classification with Harmonic Heads on a Pretrained 7B Model

This appendix documents the study of Section 4.3. Every number is produced by the runner and the export script in the `llms` folder of our repository.

Setup. The backbone is Qwen2.5-7B (Yang et al., 2024) (28 layers, hidden size 3584), kept in bfloat16. Low-rank adapters (Hu et al., 2022) of rank 16 ($\alpha{=}32$, dropout 0.05) are placed on every linear layer of the backbone and trained together with the head; the pretrained backbone weights remain fixed. An example is represented by the final hidden state at its last token, standardized per dimension with statistics computed from 2,048 training examples before training starts; every head sees the same features. We train with AdamW (head 10^{-3}, adapters 10^{-4}; 6% linear warmup, then linear decay), batch size 16, at most 128 tokens per example, and gradient clipping 1.0, for two epochs on SST-2 and AG News (20,000 training examples each) and five on Banking77 and CLINC150 (full training sets). Where a benchmark has no validation split we hold out 10% of its training data; the split is fixed across heads and seeds, and the epoch with the best validation accuracy is evaluated on the test set.

Heads. Harmonic heads produce logits $-n \log d_k$ with $n{=}\sqrt{d}$, which is exactly the harmonic probability $p_k \propto d_k^{-n}$ of Equation (1) computed in log space; the cosine head uses $m \log\big((1{+}\cos)/2\big)$ with $m{=}\sqrt{d}$. The Mahalanobis head learns one precision per dimension. The cross-entropy baseline is a linear layer with bias. Prototypes start from a random initialization, as do the baseline's weights.

Metrics. ECE uses 15 equal-width confidence bins. The PCA metrics are computed by the same function as the pretraining study, on the test features that enter the head, and the silhouette score uses those features and the true labels. Proto. cos is the cosine between each class's mean test feature and that class's head vector, both centred over classes; it asks whether head vectors can be read as class centres. For out-of-distribution detection, the 1,000 out-of-scope CLINC150 test queries are scored against the in-scope test set with three standard confidence scores: maximum softmax probability, maximum logit, and energy ($\log \sum_k e^{z_k}$).

Ablations. The frozen-backbone ablation (Table 20) computes the features once and trains only the head, for 30 epochs and three seeds. The exponent ablation (Table 21) repeats the Euclidean and Hamming heads with $n{=}1$ and $n{=}d$ on the two intent tasks, three seeds each.

Compute. A run takes 5–12 minutes on one H100 depending on the task, and the study comprises 248 runs.

J Statistical Comparisons with Euclidean Harmonic Loss

J.1 Wilcoxon Signed-rank Tests

To quantify whether non-Euclidean harmonic losses differ systematically from the Euclidean reference, we ran paired Wilcoxon signed-rank tests over all dataset–backbone combinations ($N{=}16$ pairs per distance). The resulting median score improvements and p–values are reported in Tables 22–24. Because the same 17 distances are tested on several metrics, we also report Holm–Bonferroni adjusted p–values (p_{adj}) over the full family of distance $\times$ metric tests behind these three tables. After correction *no* comparison reaches $p_{\text{adj}} < 0.05$, so the tests below should be read as descriptive effect sizes only. The confirmatory tests in this paper are those of Section 4.3, which are corrected within their own, much smaller family.

Table 18: Classification with a pretrained 7B model and LoRA adapters: mean ± sd over five seeds. PCA metrics and silhouette are computed on the test features that enter the head; Proto. cos is the cosine between each class's mean feature and its head vector, both centred over classes.

Task	Head	Acc. ↑	ECE ↓	NLL ↓	PC2 EV ↑	PCA@90% ↓	Silh. ↑	Proto. cos
SST-2	Cross-entropy	.967±.003	.023±.002	.133±.013	.544±.031	88±22	.481±.018	.432±.046
	Euclidean	.968±.004	.022±.003	.128±.011	.625±.074	86±52	.587±.070	.254±.039
	Cosine	.966±.002	.023±.004	.135±.012	.698±.041	35±19	.641±.028	.136±.022
	Hamming (soft)	.964±.001	.024±.002	.135±.004	.663±.072	60±33	.621±.058	.203±.047
	Mahalanobis (diag.)	.966±.002	.023±.002	.136±.008	.595±.053	88±36	.580±.046	.259±.038
	Bray–Curtis	.965±.005	.022±.003	.131±.013	.657±.057	64±36	.619±.026	.311±.033
	Manhattan	.966±.002	.023±.001	.130±.002	.630±.034	77±35	.610±.025	.303±.029
AG News	Cross-entropy	.944±.001	.032±.001	.202±.004	.342±.029	174±47	.227±.032	.507±.017
	Euclidean	.944±.001	.028±.001	.191±.003	.450±.058	205±127	.393±.058	.205±.033
	Cosine	.943±.001	.029±.001	.191±.001	.441±.052	210±123	.369±.029	.172±.009
	Hamming (soft)	.943±.001	.029±.002	.194±.003	.475±.053	155±92	.420±.041	.241±.021
	Mahalanobis (diag.)	.943±.001	.029±.002	.191±.002	.443±.041	222±107	.388±.045	.203±.016
	Bray–Curtis	.944±.001	.028±.002	.189±.003	.478±.023	147±46	.438±.022	.281±.009
	Manhattan	.943±.001	.029±.002	.193±.006	.458±.022	180±30	.419±.017	.292±.012
Banking77	Cross-entropy	.936±.004	.049±.004	.459±.018	.066±.003	560±38	.082±.004	.555±.005
	Euclidean	.942±.003	.037±.003	.303±.019	.103±.005	442±37	.229±.014	.340±.010
	Cosine	.942±.004	.033±.003	.274±.013	.083±.002	654±22	.201±.008	.351±.009
	Hamming (soft)	.941±.003	.032±.003	.273±.015	.165±.009	162±28	.347±.028	.337±.007
	Mahalanobis (diag.)	.941±.006	.036±.003	.297±.019	.105±.009	447±53	.224±.015	.337±.018
	Bray–Curtis	.940±.005	.034±.004	.279±.017	.160±.010	145±28	.355±.021	.363±.014
	Manhattan	.941±.004	.034±.005	.281±.015	.157±.013	147±19	.357±.014	.365±.015
CLINC150	Cross-entropy	.975±.001	.022±.001	.329±.029	.053±.001	583±57	.211±.018	.543±.009
	Euclidean	.977±.001	.015±.001	.145±.007	.076±.006	363±63	.385±.019	.321±.006
	Cosine	.977±.002	.014±.002	.132±.010	.071±.003	462±39	.383±.008	.347±.004
	Hamming (soft)	.976±.001	.014±.001	.132±.004	.120±.004	136±3	.537±.006	.355±.002
	Mahalanobis (diag.)	.977±.001	.014±.001	.139±.007	.079±.005	352±44	.390±.011	.318±.007
	Bray–Curtis	.977±.002	.014±.001	.135±.009	.108±.008	141±4	.526±.010	.383±.003
	Manhattan	.978±.001	.013±.002	.128±.013	.107±.006	143±7	.526±.010	.379±.004

Table 19: CLINC150 out-of-scope detection (LoRA; mean ± sd over five seeds). The out-of-scope queries are excluded from training and used only here. MSP: maximum softmax probability. A harmonic head's logits are $-n \log d_k$, so scaling every distance by a constant leaves the probabilities unchanged but shifts the logits: the maximum-logit and energy scores are therefore not meaningful for the scale-invariant distance heads, while MSP is.

Head	AUROC (MSP) ↑	AUROC (max logit) ↑	AUROC (energy) ↑	FPR95 (MSP) ↓	FPR95 (max logit) ↓
Cross-entropy	.964±.002	.968±.003	.968±.004	.166±.010	.143±.019
Euclidean	.981±.001	.476±.051	.444±.052	.073±.004	.998±.002
Cosine	.982±.002	.986±.002	.986±.002	.065±.008	.047±.007
Hamming (soft)	.983±.002	.795±.026	.718±.034	.066±.009	.849±.019
Mahalanobis (diag.)	.982±.002	.461±.030	.428±.030	.072±.006	.998±.001
Bray–Curtis	.983±.002	.984±.002	.984±.003	.065±.004	.058±.007
Manhattan	.983±.002	.708±.023	.618±.025	.059±.005	.979±.009

On *model performance* (Table 22), the non-Euclidean distances do not achieve a statistically significant *positive* median improvement over the Euclidean harmonic loss. Several metrics (e.g., Mahalanobis (Std.), Bray–Curtis (Std.), Canberra variants, Manhattan, Minkowski, Hamming) show *negative* medians at uncorrected $p < 0.05$ (none significant after correction), indicating that when a difference is present it tends to favor the Euclidean reference in raw accuracy. This is consistent with our main results, where non-Euclidean geometries aim at interpretability and sustainability. Three variants show particularly large accuracy reductions: Mahalanobis (Std.), Bray–Curtis (Std.), and Chebyshev (Smooth) have median differences of -64.8, -34.4, and -21.4 percentage points, respectively, reflecting divergence or representation collapse in several dataset–backbone combinations. We retain them for completeness but exclude them from our recommendations; aggregate statistics that include them should be interpreted accordingly.

Table 20: Frozen-backbone ablation: the pretrained model is not adapted and only the head trains, on features computed once (mean ± sd over three seeds). Every head then sees identical features, so the PCA metrics and silhouette are identical within a task and are omitted.

Task	Head	Acc. ↑	ECE ↓	NLL ↓
SST-2	Cross-entropy	.891±.002	.030±.000	.274±.004
	Euclidean	.861±.011	.106±.007	.391±.001
	Cosine	.860±.006	.095±.009	.384±.005
	Hamming (soft)	.837±.003	.101±.004	.420±.003
	Mahalanobis (diag.)	.865±.005	.082±.005	.366±.001
	Bray–Curtis	.842±.004	.101±.003	.413±.002
	Manhattan	.839±.005	.109±.009	.419±.004
AG News	Cross-entropy	.891±.000	.046±.000	.363±.001
	Euclidean	.855±.001	.042±.003	.431±.002
	Cosine	.861±.001	.041±.002	.427±.002
	Hamming (soft)	.836±.002	.063±.001	.487±.001
	Mahalanobis (diag.)	.862±.002	.038±.001	.412±.005
	Bray–Curtis	.841±.001	.060±.003	.474±.002
	Manhattan	.836±.001	.065±.002	.488±.004
Banking77	Cross-entropy	.730±.001	.117±.002	1.322±.011
	Euclidean	.441±.005	.193±.002	2.385±.011
	Cosine	.477±.008	.219±.007	2.272±.010
	Hamming (soft)	.355±.007	.186±.006	2.812±.007
	Mahalanobis (diag.)	.457±.005	.187±.003	2.301±.011
	Bray–Curtis	.371±.005	.193±.003	2.741±.006
	Manhattan	.346±.005	.183±.004	2.843±.007
CLINC150	Cross-entropy	.834±.004	.123±.002	1.408±.041
	Euclidean	.703±.001	.172±.003	1.305±.009
	Cosine	.729±.001	.207±.002	1.243±.003
	Hamming (soft)	.622±.002	.251±.002	1.817±.003
	Mahalanobis (diag.)	.721±.001	.146±.002	1.200±.002
	Bray–Curtis	.642±.002	.234±.002	1.697±.002
	Manhattan	.617±.003	.254±.002	1.851±.001

Table 21: Sensitivity to the harmonic exponent n (LoRA; mean test values over three seeds for $n \in \{1, d\}$ and five for the default $n=\sqrt{d}$; best per block in bold). The default $n=\sqrt{d}$ gives the best accuracy and calibration in every case; $n=1$ cannot produce a peaked distribution and is badly calibrated.

Task	Head	n	Acc. ↑	ECE ↓	PCA@90% ↓
Banking77	Euclidean	1	0.911	0.225	63
		d	0.920	0.065	334
		$\sqrt{d}$	**0.942**	**0.037**	442
	Hamming (soft)	1	0.852	0.689	55
		d	0.914	0.061	176
		$\sqrt{d}$	**0.941**	**0.032**	162
CLINC150	Euclidean	1	0.967	0.360	125
		d	0.967	0.028	539
		$\sqrt{d}$	**0.977**	**0.015**	363
	Hamming (soft)	1	0.967	0.907	120
		d	0.951	0.036	186
		$\sqrt{d}$	**0.976**	**0.014**	136

For *interpretability* (Table 23), we observe the opposite pattern. Distances such as Bray–Curtis (Norm.), Canberra (Robust/Std.), Chebyshev (Std.), Manhattan, and both cosine variants exhibit shifts (uncorrected $p < 0.05$) in the number of principal components needed to explain 90% of the variance. The median differences are large in magnitude (e.g., $+12.8$ for Bray–Curtis (Norm.),

Table 22: Paired Wilcoxon signed-rank comparisons with Euclidean harmonic loss for final test accuracy (N=16 dataset–backbone pairs per head). "Median improvement" is the median percentage-point difference (non-Euclidean minus Euclidean), so positive values favor the non-Euclidean head. p_{adj} is Holm–Bonferroni adjusted over the 51 distance–metric tests in the three tables.

Comparison	**N Pairs**	**Median Impr.**	**p-value**	**p_{adj}**
Bray–Curtis (Norm.)	16	0.5317	0.17060	1.0000
Cosine (Stable)	16	0.1598	0.45339	1.0000
Cosine (Unst.)	16	0.0965	0.55208	1.0000
Mahalanobis (Chol.)	16	0.0400	0.58717	1.0000
Bray–Curtis (Abs.)	16	-1.1967	0.10335	1.0000
Mahalanobis (Diag.)	16	-1.4633	0.08323	1.0000
Chebyshev (Std.)	16	-1.7232	< 0.001	0.0736
Minkowski (p=3.0)	16	-1.8183	0.00567	0.6234
Canberra (Weighted)	16	-2.8167	0.00295	0.3565
Manhattan	16	-4.9467	0.00249	0.3083
Minkowski (p=1.5)	16	-5.8117	0.00176	0.2232
Hamming (Soft)	16	-6.3183	0.00348	0.4110
Canberra (Robust)	16	-16.5883	< 0.001	0.0736
Canberra (Std.)	16	-18.3767	< 0.001	0.0736
Chebyshev (Smooth)	16	-21.3817	< 0.001	0.0736
Bray–Curtis (Std.)	16	-34.4379	< 0.001	0.0975
Mahalanobis (Std.)	16	-64.8082	< 0.001	0.0736

$+9.8$ for Canberra (Robust)), providing descriptive evidence that changing the distance can alter representation concentration across datasets and backbones.

For *sustainability* (Table 24), four distances reach uncorrected $p < 0.05$: Mahalanobis (Std.) and Bray–Curtis (Std.) emit less than Euclidean, Canberra (Weighted) and Mahalanobis (Chol.) more. The measured differences are generally modest and model-dependent: some geometries have positive deltas and others negative ones, and none remains significant after correction.

Overall, these nonparametric tests are consistent with our main claims: non-Euclidean harmonic losses do not uniformly dominate Euclidean in accuracy, while several of them shift representation structure substantially, with only mild and mixed effects on emissions. Given the family size, however, we treat the vision comparisons against Euclidean as exploratory: the effect sizes are large, but with N=16 paired settings per distance they do not survive correction.

J.2 Accuracy with Confidence Intervals

To complement the aggregate tables and Wilcoxon tests, Figure 4 reports accuracy curves for the top-performing losses on each dataset/backbone pair. For every setting we re-train each candidate with three random seeds and plot the mean trajectory together with a shaded 95% confidence interval (n=3).

Across datasets and architectures, two consistent patterns emerge. First, the ranking suggested by our radar plots and summary tables is preserved under multi-seed training: distance-based harmonic losses that previously appeared as strong contenders (e.g., cosine, Bray–Curtis, Minkowski) continue to track at least as well as, and often above, the cross-entropy and Euclidean baselines throughout training. In several regimes (CIFAR-10/CIFAR-100 with ResNet50 and MNIST with ResNet50/PVT), the confidence bands of the leading non-Euclidean harmonic loss lie above those of the baselines during later epochs. These plots show variation across the three runs but are not used as separate significance tests.

Second, the width of the confidence intervals is often comparable or smaller for harmonic losses than for standard baselines. On datasets where optimization is more fragile (e.g., CIFAR-100 with PVT), cross-entropy and some regularized baselines (Focal, Center Loss) display visibly wider bands and occasional late-epoch fluctuations, whereas several harmonic distances have smoother trajectories with tighter intervals. We do not observe a reversal in the mean ranking of a harmonic loss that outperforms Euclidean in the aggregate tables.

Table 23: Paired Wilcoxon signed-rank comparisons with Euclidean harmonic loss for PCA@90% (N=16 dataset–backbone pairs per head). "Median improvement" is the median paired reduction (Euclidean minus non-Euclidean), so positive values indicate that the non-Euclidean head needs fewer components. p_{adj} uses the correction defined in Table 22; per-setting values appear in Appendix Q.

Comparison	N Pairs	Median Impr.	p-value	p_{adj}
Bray–Curtis (Norm.)	16	12.8333	< 0.001	0.1142
Canberra (Robust)	16	9.8333	0.00162	0.2086
Canberra (Std.)	16	8.1667	< 0.001	0.0988
Chebyshev (Std.)	16	7.7083	0.01864	1.0000
Manhattan	16	7.6667	0.00412	0.4764
Cosine (Unst.)	16	5.8333	0.01043	1.0000
Canberra (Weighted)	16	5.2083	0.26768	1.0000
Cosine (Stable)	16	4.3333	0.01127	1.0000
Mahalanobis (Std.)	16	4.0417	0.12716	1.0000
Bray–Curtis (Std.)	16	2.1667	0.34869	1.0000
Bray–Curtis (Abs.)	16	1.0000	0.66019	1.0000
Hamming (Soft)	16	0.7083	0.77730	1.0000
Minkowski (p=1.5)	16	-0.0000	0.80665	1.0000
Minkowski (p=3.0)	16	-0.0000	0.75554	1.0000
Mahalanobis (Diag.)	16	-0.0000	0.30656	1.0000
Chebyshev (Smooth)	16	-0.2083	0.77638	1.0000
Mahalanobis (Chol.)	16	-0.3333	0.85062	1.0000

Table 24: Paired Wilcoxon signed-rank comparisons with Euclidean harmonic loss for measured emissions in gCO_2eq (N=16 dataset–backbone pairs per head). "Median improvement" is the median paired reduction (Euclidean minus non-Euclidean), so positive values indicate lower measured emissions. p_{adj} uses the correction defined in Table 22.

Comparison	N Pairs	Median Impr.	p-value	p_{adj}
Mahalanobis (Std.)	16	1.8037	< 0.001	0.114
Bray–Curtis (Std.)	16	1.1880	0.01620	1.000
Cosine (Unst.)	16	0.2341	0.05249	1.000
Canberra (Std.)	16	0.0351	0.77611	1.000
Chebyshev (Smooth)	16	0.0208	0.73679	1.000
Canberra (Robust)	16	-0.0206	0.73679	1.000
Cosine (Stable)	16	-0.0239	0.97937	1.000
Minkowski (p=1.5)	16	-0.0492	0.36552	1.000
Hamming (Soft)	16	-0.0909	0.26625	1.000
Bray–Curtis (Abs.)	16	-0.1066	0.20520	1.000
Chebyshev (Std.)	16	-0.1421	0.14056	1.000
Bray–Curtis (Norm.)	16	-0.1625	0.11477	1.000
Manhattan	16	-0.1976	0.12716	1.000
Minkowski (p=3.0)	16	-0.4158	0.05249	1.000
Mahalanobis (Diag.)	16	-0.4587	0.05249	1.000
Canberra (Weighted)	16	-0.7687	0.03188	1.000
Mahalanobis (Chol.)	16	-0.9431	0.00411	0.476

Overall, these multi-seed curves supplement the aggregate vision results by showing both mean training trajectories and between-seed variation. They support the descriptive comparisons while leaving the corrected Wilcoxon tests above as the basis for inferential interpretation.

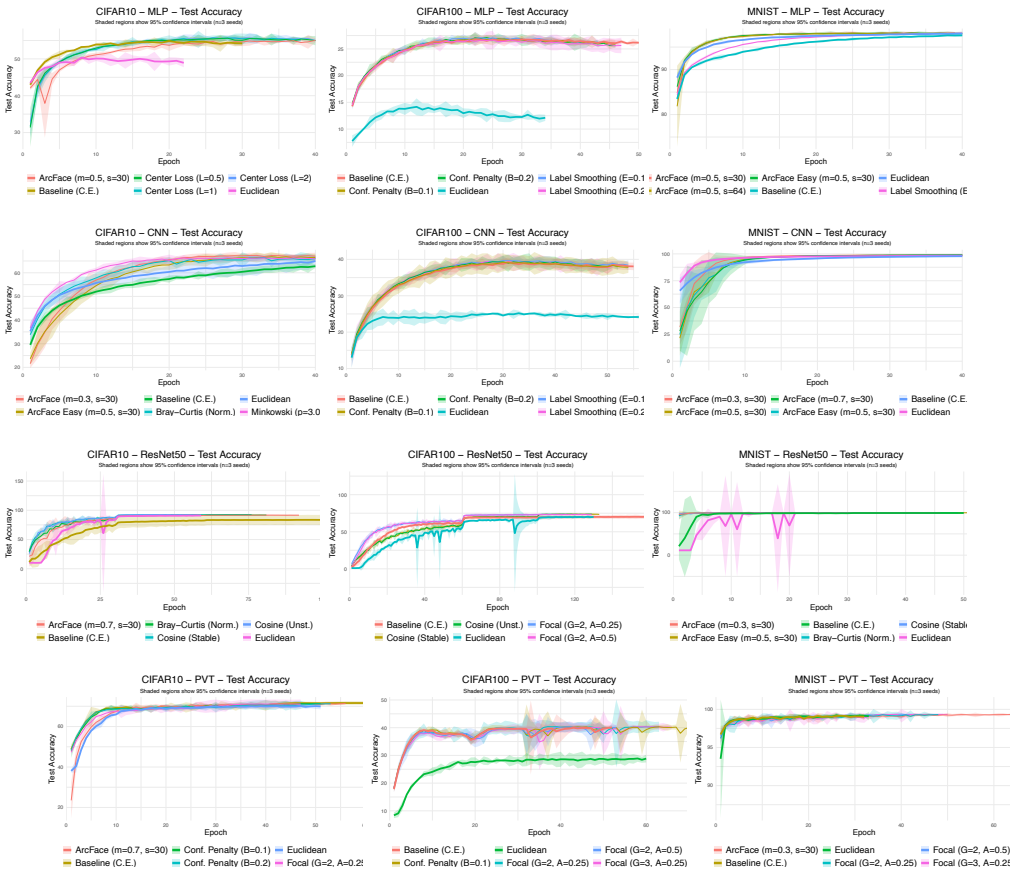

Figure 4: Vision accuracy over training for the selected losses. Lines show the mean over three seeds; shaded regions show 95% confidence intervals.

K VISION: RADAR PLOTS: ADDITIONAL DATASETS (MNIST, CIFAR10)

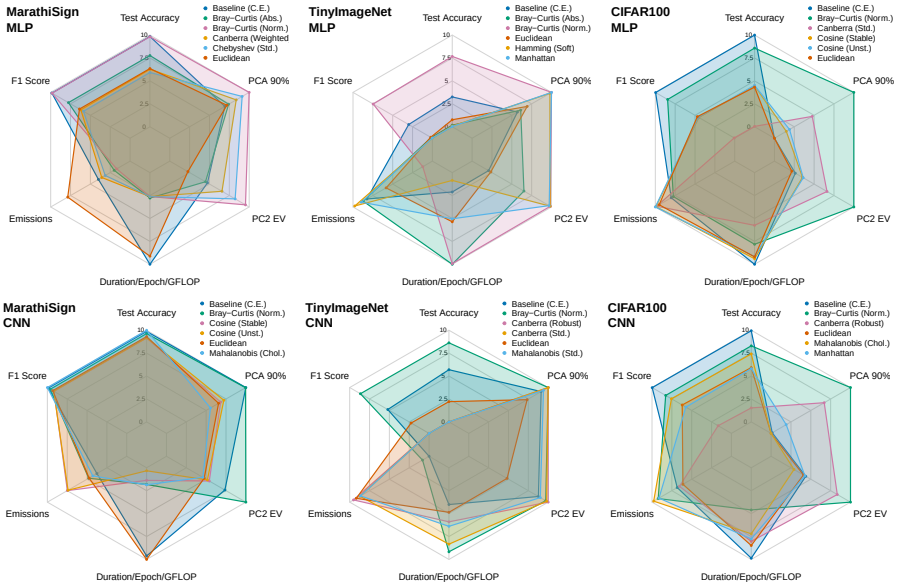

Figure 5: Vision: Radar plots for the MLP and CNN backbones, on the same axes as Figure 1, from which they were omitted for space.

Figure 6 reports results for distance-based harmonic losses on MNIST and CIFAR10 across all four backbones (MLP, CNN, ResNet50, PVT). The MNIST and CIFAR-10 radar plots provide additional descriptive comparisons across performance, representation concentration, and measured emissions. Several non-Euclidean heads, particularly cosine, Bray–Curtis, and Chebyshev variants, compare favorably with Euclidean harmonic loss and cross-entropy on at least one of these axes.

K.1 MNIST

RQ1: Model Performance (F1, Accuracy). Across all MNIST backbones, accuracy and F1 are saturated for several distances, but cosine-based harmonic losses (stable/unstable) and Bray–Curtis (normalized) remain among the most reliable high-performers. On MLP and CNN, these distances match or slightly exceed both Euclidean harmonic loss and the cross-entropy baseline. On ResNet50 and PVT, where capacity is ample, almost all distances reach nearly perfect accuracy; within these runs, changing the distance does not materially change performance.

RQ2: Interpretability (PC2 EV, PCA 90%). Even on this simple dataset, non-Euclidean distances already reshape the embedding geometry. Bray–Curtis (normalized) and Chebyshev (standard) produce noticeably higher PC2 explained variance and reduce the number of components needed to reach 90% EV, indicating compact representations. Cosine harmonic losses also improve EV relative to Euclidean while maintaining top accuracy. Mahalanobis and Minkowski variants (on ResNet50) further concentrate variance, but their interpretability advantage is less pronounced on MNIST because the task is almost linearly separable.

RQ3: Sustainability (Duration/Epoch/GFLOPs, Emissions). For MNIST, the harmonic head constitutes a tiny fraction of the overall compute, so all distances exhibit similar Duration/Epoch/GFLOPs and emissions. Cosine and Bray–Curtis are essentially neutral relative to Euclidean and cross-entropy; small differences arise mainly from minor variations in convergence speed, at comparable per-step cost. The key takeaway from MNIST is therefore that non-Euclidean harmonic losses can improve representation structure without sacrificing accuracy or sustainability.

K.2 CIFAR10

RQ1: Model Performance (F1, Accuracy). On CIFAR-10, cosine harmonic losses become clearly advantageous. For MLP and CNN, cosine (stable/unstable) and Bray–Curtis (normalized) consistently occupy the highest or near-highest F1 and accuracy, outperforming Euclidean harmonic loss and the cross-entropy baseline. On ResNet50 and PVT, cosine again delivers strong accuracy while remaining competitive with the best non-Euclidean alternatives (e.g., Minkowski p=3.0). Overall, cosine is the most robust choice across architectures once the task requires nontrivial feature extraction.

RQ2: Interpretability (PC2 EV, PCA 90%). CIFAR-10 further highlights the interpretability benefits of non-Euclidean geometry. Bray–Curtis and Chebyshev systematically increase PC2 EV and reduce PCA 90% dimensionality on MLP, CNN, and ResNet50, yielding sharper, more compact embeddings than Euclidean or cross-entropy. Cosine harmonic losses also improve EV over Euclidean, providing a favorable accuracy/interpretability compromise. On PVT, Canberra-weighted and Bray–Curtis variants similarly enhance variance concentration while preserving strong performance, reinforcing the observation that prototype-friendly distances induce more structured feature spaces.

RQ3: Sustainability (Duration/Epoch/GFLOPs, Emissions). On CIFAR-10, sustainability trends mirror those seen on larger datasets. Cosine harmonic loss is typically neutral to slightly favorable in Duration/Epoch/GFLOPs and emissions relative to Euclidean and cross-entropy, especially on CNN and ResNet50 where convergence is faster. Bray–Curtis and Canberra variants introduce modest overhead, reflecting their more complex computations, but remain within the same qualitative efficiency regime. In all cases, the harmonic head is lightweight compared to the backbone, so the main sustainability differences arise from reduced steps-to-high accuracy, at comparable per-step cost.

L Results with Alternative Loss Functions

The experiments in this appendix were conducted separately from those in the per-dataset tables of Appendix Q. Here, the baselines and additional losses were retrained together, so absolute values for the same dataset–backbone setting can differ slightly between the two campaigns. For example, cross-entropy on CIFAR-100 with ResNet50 reaches 0.7006 here and 0.6983 in Appendix Q, a difference within the observed run-to-run variation. All percentage changes are computed within the corresponding table. To contextualize our distance-based harmonic losses, we benchmark against four widely used alternatives motivated by calibration, low-data robustness, or representational compactness. We summarize each objective and its relevance to effectiveness, computational cost, and representation structure. Unless otherwise noted, these baselines use a conventional linear head and softmax.

Focal Loss (calibration, anti-grokking, class imbalance). Focal Loss reweights examples by their difficulty:

$$\mathcal{L}_{\text{focal}}(\mathbf{z}, y) = -\,\alpha\,(1-p_y)^{\gamma}\,\log p_y, \quad p_y = \frac{e^{z_y}}{\sum_j e^{z_j}},$$

with focusing parameter $\gamma \geq 0$ and class weight $\alpha \in (0, 1]$. By down-weighting well-classified (overconfident) samples, it yields smoother gradient signals, often improving calibration and mitigating late-stage overfitting behaviors akin to grokking. In our grids we consider $\gamma \in \{2, 3\}$ and $\alpha \in \{0.25, 0.5\}$. *Sustainability:* modest compute overhead (same forward/backward shape as CE), but potentially fewer effective updates on easy samples; net carbon effect is typically neutral to slightly higher than CE, depending on convergence behavior.

Label Smoothing (reduced overconfidence, low-data stability). Label Smoothing replaces the one-hot target with

$$\tilde{\mathbf{y}} = (1-\varepsilon)\,\mathbf{e}_y + \frac{\varepsilon}{K}\,\mathbf{1}, \quad \mathcal{L}_{\text{LS}}(\mathbf{z}, y) = -\sum_{c=1}^{K} \tilde{y}_c \log p_c,$$

where $\varepsilon \in [0, 1)$ controls smoothing (we use $\varepsilon \in \{0.1, 0.2\}$). The softened targets reduce overconfidence and improve generalization in scarce-label settings; they also stabilize optimization

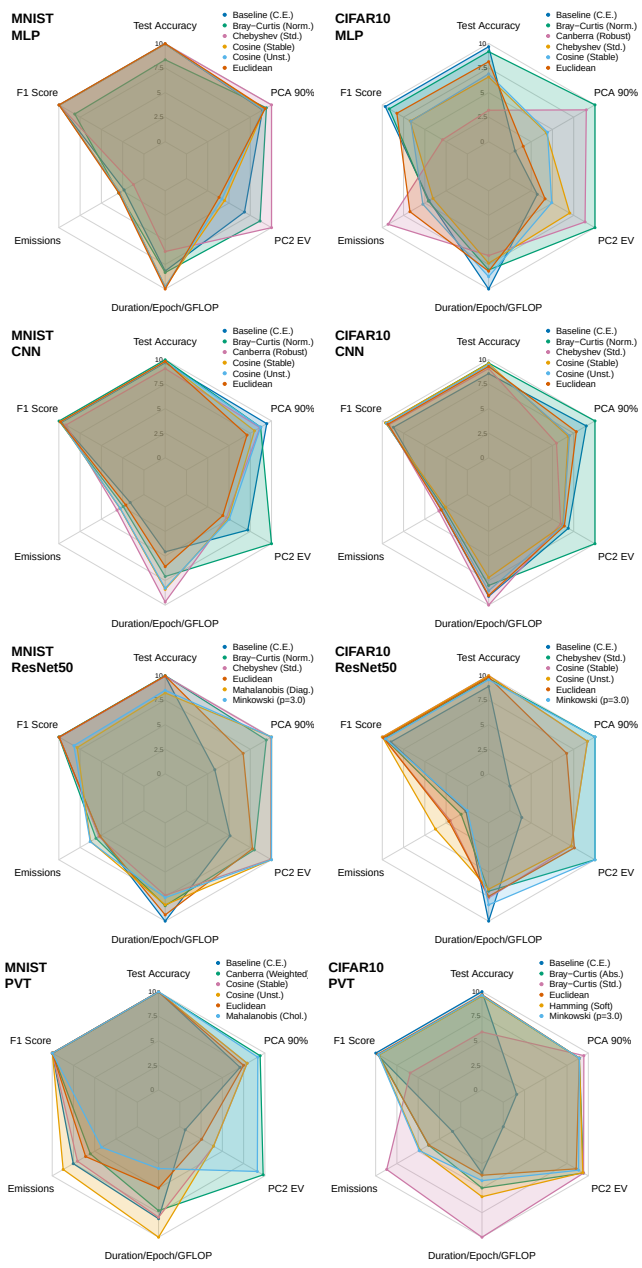

Figure 6: Vision: Radar plots: 1) *Model Performance* (F1, Accuracy); 2) *Interpretability* (PC2 EV, PCA 90%), and 3) *Sustainability* (Duration/Epoch/GFLOPs, Emissions). Plots feature Baseline (Cross-Entropy), Euclidean harmonic, and the four top-performing non-Euclidean harmonic losses.

by shrinking logit magnitudes. *Interpretability:* mild regularization can yield more isotropic features; *sustainability:* training cost matches CE, with potential reductions in steps-to-target when overconfidence previously harmed convergence.

Center Loss (prototype compactness, cluster interpretability). Center Loss explicitly penalizes the distance to a class prototype:

$$\mathcal{L}_{\text{center}}(\mathbf{h}, y) = \frac{1}{2} \left\| \mathbf{h} - \mathbf{c}_y \right\|_2^2, \qquad \mathcal{L} = \mathcal{L}_{\text{CE}} + \lambda \mathcal{L}_{\text{center}},$$

with learnable class centers $\{\mathbf{c}_k\}$ and trade-off $\lambda > 0$ (we test $\lambda \in \{0.5, 1, 2\}$). This encourages intra-class compactness and inter-class separability—properties that make feature clusters and decision prototypes easier to inspect. *Sustainability:* small extra memory and updates for $\mathbf{c}_k$; the overhead is minor relative to the backbone but measurable in long runs.

Confidence Penalty (entropy regularization, anti-grokking). Confidence Penalty adds a negative-entropy term to discourage over-peaked posteriors:

$$\mathcal{L}_{\text{CP}}(\mathbf{z}, y) = \mathcal{L}_{\text{CE}}(\mathbf{z}, y) \; - \; \beta\, \mathcal{H}(\mathbf{p}), \quad \mathcal{H}(\mathbf{p}) = -\sum_{c=1}^{K} p_c \log p_c,$$

with $\beta > 0$ (we use $\beta \in \{0.1, 0.2\}$). By explicitly rewarding higher predictive entropy when appropriate, it reduces brittle overconfidence and can temper delayed generalization (grokking-like) effects. *Sustainability:* essentially identical compute to CE; any carbon changes stem from altered convergence trajectories.

ArcFace (angular margins, maximized class separation). ArcFace (Deng et al., 2019) introduces an *additive angular margin* that enlarges the decision boundary between classes on the unit hypersphere. Given normalized features $\mathbf{h}$ and normalized class weights $\mathbf{w}_c$, the cosine similarity $\cos\theta_c = \langle \mathbf{h}, \mathbf{w}_c \rangle$ is modified for the target class y by adding a fixed angular margin m:

$$\cos(\theta_y + m) = \cos\theta_y \cos m - \sin\theta_y \sin m.$$

ArcFace replaces the final linear classifier with a scaled angular softmax:

$$\mathcal{L}_{\text{arcface}}(\mathbf{h}, y) = -\log \frac{\exp\left(s\, \cos(\theta_y + m)\right)}{\exp\left(s\, \cos(\theta_y + m)\right) + \sum_{c \neq y} \exp\left(s\, \cos\theta_c\right)},$$

where s is a feature-scale parameter (typically $s = 30$–64). By manipulating angles, ArcFace enforces tighter class clustering and larger inter-class separation, and is widely regarded as a *strong margin-based baseline* in metric learning. This makes it a particularly relevant comparator for harmonic losses: both approaches normalize features to a hypersphere and control geometry around class prototypes, but ArcFace explicitly pushes angular margins, whereas harmonic losses reshape the whole distance function. *Sustainability:* ArcFace is lightweight (same complexity as cosine classifiers), but the angular margin can slightly increase optimization stiffness, occasionally raising per-step compute or slowing convergence. In our experiments we consider $m \in \{0.3, 0.5\}$ and $s \in \{30, 64\}$.

Each baseline targets a related objective through a different inductive bias: Focal Loss and Label Smoothing modify confidence and example weighting, Center Loss encourages prototype compactness, and Confidence Penalty discourages overconfident predictions. They therefore provide relevant comparisons across predictive performance, representation structure, and measured resource use.

L.1 Vision: Fine-grained results

A cross-backbone inspection in Tables 25–28 shows that the additional baselines (Focal, Label Smoothing, Confidence Penalty, Center Loss, ArcFace) are competitive on accuracy, but do not displace the non-Euclidean harmonic losses as the most balanced options.

On **CIFAR-10 and CIFAR-100 with ResNet-50**, cosine-based harmonic heads remain among the strongest configurations: they deliver the largest accuracy and F1 gains over cross-entropy (up to $\approx 9.9\%$ relative on CIFAR-10 and 5% on CIFAR-100), while simultaneously reducing emissions by 10–30% and sharply concentrating the representation (PC90% dropping from 50 to 5–8 components). Focal and ArcFace variants sometimes match top accuracy, but typically exhibit weaker EV/PC90% improvements or higher emissions, so they do not dominate the multi-criteria trade-off.

For **PVT backbones**, where capacity and input resolution are higher, the picture is more nuanced. On CIFAR-10 PVT, calibration-oriented losses (Label Smoothing, Focal) offer slight accuracy improvements and sizable emissions reductions, yet Euclidean harmonic achieves the most compact geometry (PC90% from 17 to 3) at only a small performance cost. On CIFAR-100 PVT, focal and confidence-penalty losses are best in accuracy, but Euclidean harmonic again produces the most

concentrated feature spaces (PC90% 4 vs. 50), highlighting an interpretability advantage even when it is not the accuracy winner.

On the **Marathi Sign dataset** (224×224 on ResNet50, 32×32 on the other backbones), nearly all methods saturate accuracy (≥ 0.999), so the comparison is driven by structure and sustainability. Here, cosine-based harmonic losses for ResNet-50 (and to a lesser extent for PVT) achieve substantial EV/PC90% gains. e.g., Cosine (Unst.) reduces PC90% from 15.5 to 6.5 while slightly lowering emissions, demonstrating that our non-Euclidean harmonic heads remain competitive even in settings where strong baselines like ArcFace and Center Loss are present.

Table 25: Results for CIFAR100 PVT (top-8 losses) and % changes w.r.t. Baseline (CE).

Method	**Acc**	**F1**	**gCO$_2$eq**	**EV**	**PC90%**
Baseline	0.3994	0.3970	3.67	0.0973	50.0
Focal (γ=2, α=0.25)	0.4017 (0.59%)	0.3996 (0.65%)	4.9609 (-35.35%)	0.1257 (29.22%)	50.0 (0%)
Focal (γ=3, α=0.25)	0.4015 (0.53%)	0.3999 (0.72%)	4.9567 (-35.24%)	0.1364 (40.21%)	50.0 (0%)
Focal (γ=2, α=0.5)	0.4010 (0.40%)	0.3997 (0.66%)	3.9182 (-6.90%)	0.1271 (30.64%)	50.0 (0%)
Conf. Penalty (β=0.1)	0.3999 (0.13%)	0.3977 (0.17%)	5.2574 (-43.44%)	0.0963 (-0.96%)	50.0 (0%)
Label Smoothing (ε=0.1)	0.3894 (-2.50%)	0.3888 (-2.08%)	2.6626 (27.36%)	0.0818 (-15.92%)	50.0 (0%)
Conf. Penalty (β=0.2)	0.3859 (-3.38%)	0.3834 (-3.45%)	2.1718 (40.74%)	0.0749 (-23.00%)	50.0 (0%)
Label Smoothing (ε=0.2)	0.3847 (-3.67%)	0.3851 (-3.00%)	2.2097 (39.71%)	0.0742 (-23.74%)	50.0 (0%)
ArcFace (m=0.5, s=30)	0.3728 (-6.65%)	0.3772 (-5.00%)	2.6171 (28.60%)	0.1259 (29.41%)	50.0 (0%)
Euclidean	0.2864 (-28.29%)	0.2945 (-25.83%)	6.4329 (-75.51%)	0.8414 (765.15%)	4.0 (92.00%)

Table 26: Results for CIFAR100 ResNet50 (top-8 losses) and % changes w.r.t. Baseline (CE).

Method	**Acc**	**F1**	**gCO$_2$eq**	**EV**	**PC90%**
Baseline	0.7006	0.6993	89.64	0.1069	50.0
Cosine (Stable)	0.7381 (5.35%)	0.7384 (5.59%)	79.2831 (11.55%)	0.5915 (453.51%)	8.0 (84.00%)
Focal (γ=2, α=0.25)	0.7349 (4.90%)	0.7342 (5.00%)	72.1795 (19.48%)	0.1468 (37.32%)	50.0 (0%)
Focal (γ=2, α=0.5)	0.7341 (4.79%)	0.7332 (4.85%)	78.7234 (12.18%)	0.1224 (14.52%)	50.0 (0%)
Cosine (Unst.)	0.7340 (4.77%)	0.7349 (5.09%)	72.5413 (19.07%)	0.5891 (451.21%)	8.0 (84.00%)
Focal (γ=3, α=0.25)	0.7311 (4.36%)	0.7308 (4.51%)	81.5875 (8.98%)	0.1554 (45.41%)	50.0 (0%)
Label Smoothing (ε=0.1)	0.7261 (3.64%)	0.7248 (3.65%)	79.8223 (10.95%)	0.1469 (37.44%)	50.0 (0%)
Label Smoothing (ε=0.2)	0.7221 (3.08%)	0.7206 (3.04%)	81.1883 (9.43%)	0.1524 (42.59%)	50.0 (0%)
ArcFace (m=0.7, s=30)	0.7166 (2.29%)	0.7150 (2.25%)	70.5590 (21.29%)	0.6059 (466.93%)	48.33 (3.33%)
Euclidean	0.7047 (0.59%)	0.7055 (0.89%)	87.7280 (2.13%)	0.4301 (302.49%)	33.67 (32.67%)

Table 27: Results for MarathiSign PVT (top-8 losses) and % changes w.r.t. Baseline (CE).

Method	**Acc**	**F1**	**gCO$_2$eq**	**EV**	**PC90%**
Baseline	0.9965	0.9964	2.85	0.2135	24.75
ArcFace (m=0.7, s=30)	0.9997 (0.33%)	0.9997 (0.33%)	4.2108 (-47.88%)	0.1132 (-47.00%)	36.0 (-45.45%)
Focal (γ=3, α=0.25)	0.9997 (0.32%)	0.9996 (0.32%)	2.7240 (4.34%)	0.2749 (28.74%)	19.33 (21.89%)
Center Loss (λ=1)	0.9995 (0.31%)	0.9995 (0.31%)	5.1746 (-81.72%)	0.1824 (-14.56%)	29.67 (-19.87%)
Conf. Penalty (β=0.2)	0.9995 (0.30%)	0.9994 (0.30%)	3.6912 (-29.63%)	0.1504 (-29.58%)	33.67 (-36.03%)
Focal (γ=2, α=0.25)	0.9993 (0.29%)	0.9993 (0.28%)	2.5648 (9.93%)	0.2853 (33.63%)	19.33 (21.89%)
ArcFace (m=0.3, s=30)	0.9992 (0.28%)	0.9992 (0.28%)	7.5158 (-163.94%)	0.1756 (-17.76%)	28.0 (-13.13%)
Label Smoothing (ε=0.2)	0.9992 (0.28%)	0.9991 (0.27%)	2.6977 (5.26%)	0.1190 (-44.26%)	35.0 (-41.41%)
Cosine (Unst.)	0.9991 (0.27%)	0.9991 (0.26%)	7.3447 (-157.93%)	0.5552 (160.04%)	7.0 (71.72%)
Euclidean	0.9994 (0.30%)	0.9994 (0.30%)	4.3621 (-53.19%)	0.5035 (135.83%)	14.25 (42.42%)

Table 28: Results for MarathiSign ResNet50 (top-8 losses) and % changes w.r.t. Baseline (CE).

Method	Acc	F1	gCO_2eq	EV	PC90%
Baseline	0.9998	0.9998	35.41	0.4507	15.5
Conf. Penalty (β=0.1)	0.9999 (0.01%)	0.9999 (0.01%)	29.6596 (16.24%)	0.4393 (-2.55%)	17.5 (-12.90%)
Cosine (Stable)	0.9999 (0.01%)	0.9999 (0.01%)	54.8139 (-54.80%)	0.7321 (62.42%)	5.5 (64.52%)
Conf. Penalty (β=0.2)	0.9998 (0.00%)	0.9998 (0.00%)	30.3697 (14.23%)	0.3288 (-27.05%)	33.0 (-112.90%)
Focal (γ=2, α=0.25)	0.9998 (0.00%)	0.9998 (0.00%)	24.95 (29.54%)	0.4016 (-10.89%)	17.0 (-9.68%)
Focal (γ=3, α=0.25)	0.9998 (0.00%)	0.9998 (0.00%)	28.1206 (20.58%)	0.4286 (-4.91%)	16.5 (-6.45%)
Cosine (Unst.)	0.9997 (-0.01%)	0.9997 (-0.01%)	33.8287 (4.46%)	0.6427 (42.58%)	6.5 (58.06%)
Label Smoothing (ε=0.1)	0.9997 (-0.01%)	0.9997 (-0.01%)	33.5266 (5.31%)	0.1758 (-60.99%)	36.0 (-132.26%)
Label Smoothing (ε=0.2)	0.9997 (-0.01%)	0.9997 (-0.01%)	38.7093 (-9.32%)	0.1641 (-63.60%)	36.0 (-132.26%)
Euclidean	0.9984 (-0.14%)	0.9983 (-0.15%)	50.2050 (-41.79%)	0.2787 (-38.18%)	50.0 (-222.58%)

L.2 Vision: Radar Plots with Additional Losses (MNIST, CIFAR10, CIFAR100)

Figure 7 presents an expanded multi-criteria analysis across MNIST, CIFAR-10, and CIFAR-100 using MLP, CNN, ResNet50, and PVT backbones. This comparison places the harmonic losses alongside widely adopted alternatives for regularization, representation compactness, and robustness.

RQ1: Model Performance (F1, Test Accuracy). On the deeper backbones (ResNet50, and PVT outside CIFAR-100), cosine- and Bray–Curtis-based harmonic variants are often among the strongest observed performers. Focal Loss and Label Smoothing are competitive in several settings, and the relative ranking varies across datasets and backbones. As noted in the main text, the vision comparisons are descriptive because none of the paired accuracy differences survives Holm correction.

RQ2: Interpretability (PC2 EV, PCA 90%). Non-Euclidean harmonic variants, especially Bray–Curtis and Chebyshev, often concentrate more variance in fewer principal components. Center Loss and the regularized cross-entropy variants also change these metrics in some settings. Because concentration can reward collapse, these observations concern representation structure rather than human interpretability and should be compared only at similar accuracy.

RQ3: Sustainability (Duration/Epoch, Emissions). Measured emissions vary with architecture and run conditions. Several harmonic variants are competitive with cross-entropy and the added baselines, but differences near the observed 29% run-to-run variation cannot be resolved by these measurements (Appendix A).

Overall, the comparison indicates architecture-dependent trade-offs rather than general dominance by one loss family: cosine and Bray–Curtis are frequently competitive, while the regularized baselines remain preferable in some settings.

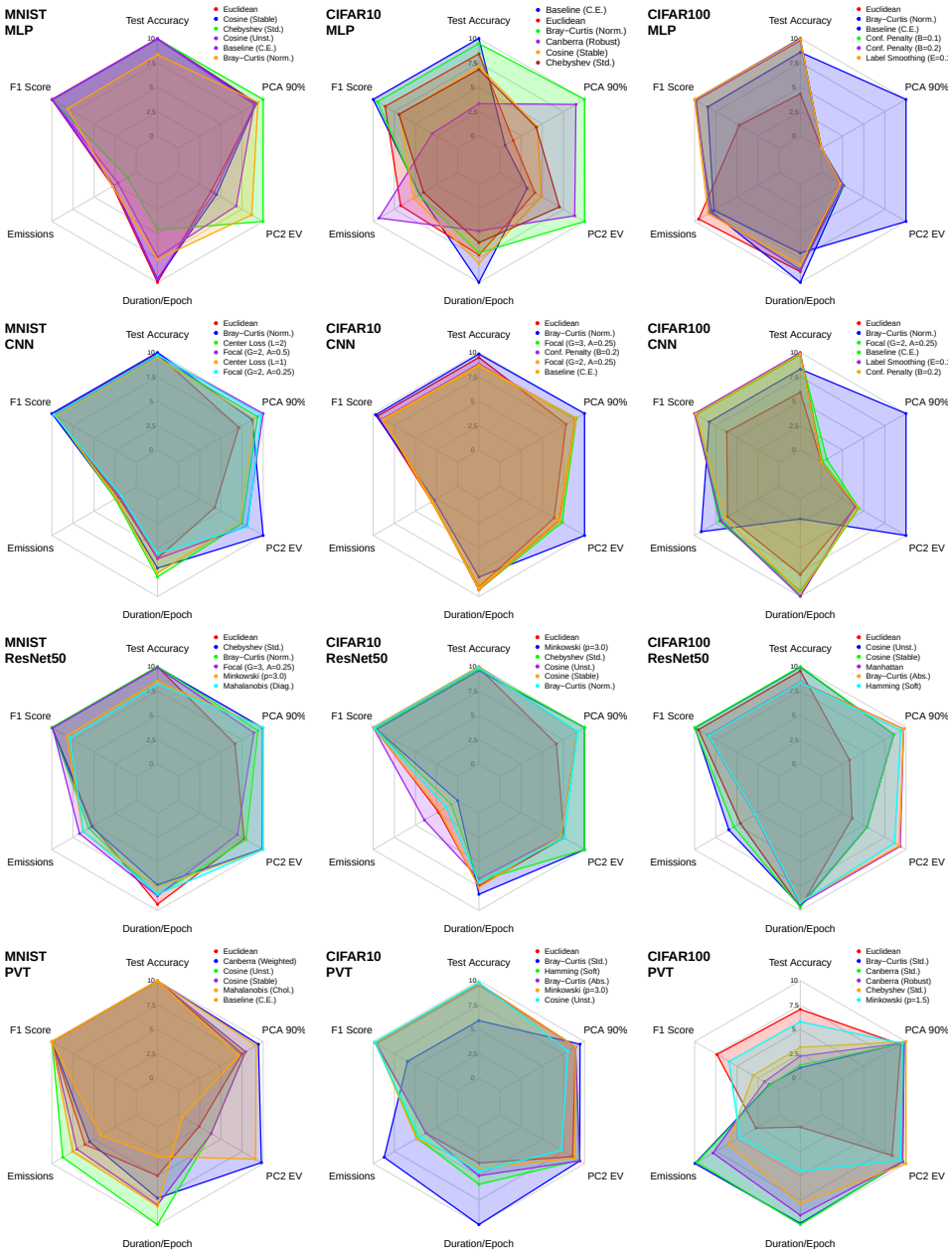

Figure 7: Vision: Radar plots – MNIST, CIFAR10, CIFAR100: 1) *Model Performance* (F1, Accuracy); 2) *Interpretability* (PC2 EV, PCA 90%), and 3) *Sustainability* (Duration/Epoch, Emissions). Plots feature Baseline (Cross-Entropy), Euclidean harmonic, and the four top-performing losses.

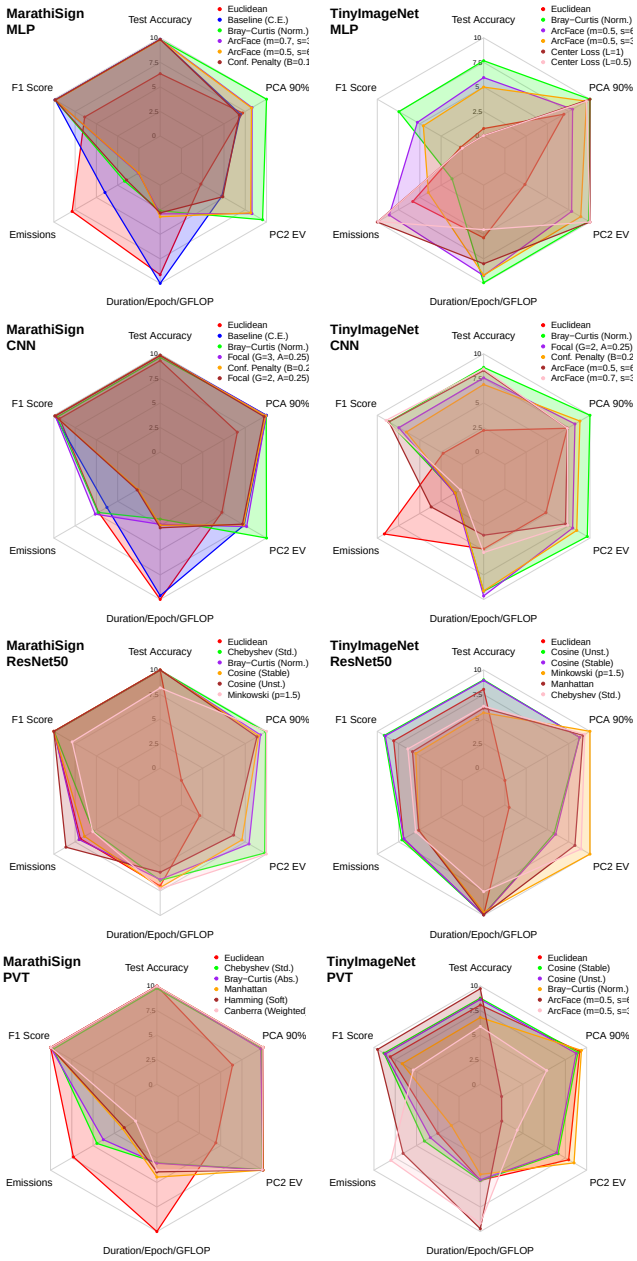

Figure 8: Vision: Radar plots – Marathi Sign Language, TinyImageNet: 1) *Model Performance* (F1, Accuracy); 2) *Interpretability* (PC2 EV, PCA 90%), and 3) *Sustainability* (Duration/Epoch, Emissions). Plots feature Baseline (Cross-Entropy), Euclidean harmonic, and the four top-performing losses.

L.3 VISION: RADAR PLOTS WITH ADDITIONAL LOSSES (MARATHI SIGN LANGUAGE, TINYIMAGENET)

Figure 8 extends the radar analysis to higher-resolution benchmarks (Marathi Sign Language, TinyImageNet) and augments the comparison set with strong loss baselines such as Focal Loss, ArcFace, Center Loss, and Confidence Penalty.

RQ1: Model Performance (Accuracy, F1). On Marathi Sign, the added losses make MLP and CNN particularly competitive: ArcFace and Focal occasionally obtain strong accuracy/F1, yet non-Euclidean harmonic losses remain among the best methods. For MLP, **Bray–Curtis (Normalized)** tracks or exceeds both cross entropy and ArcFace while preserving a smooth performance profile. For CNN, **Bray–Curtis (Normalized)** and **cosine (stable/unstable)** consistently occupy the top accuracy/F1 slices; Focal and Confidence Penalty are competitive but never clearly dominate. On deeper backbones, the picture is even clearer: for **ResNet50** and **PVT** on Marathi Sign, all top-performing methods are harmonic losses, indicating that distance-based harmonic heads outperform alternative losses outright in this regime.

On TinyImageNet, a harder and more fine-grained benchmark, a similar pattern emerges. For MLP, ArcFace and Center Loss join **Bray–Curtis (Normalized)** and Euclidean in the top-performing losses, but Bray–Curtis remains competitive in accuracy while providing different geometric and sustainability properties. For CNN, the strongest Focal and ArcFace variants reach high F1, yet **Bray–Curtis (Normalized)** again sits near the performance frontier. On **ResNet50**, the top-performing losses are *entirely harmonic* (cosine, Minkowski, Chebyshev, Euclidean), and on **PVT** TinyImageNet the leaders are dominated by cosine and Bray–Curtis, with ArcFace appearing only as an alternative angular baseline. Overall, even in the presence of sophisticated angular-margin and confidence-shaping losses, non-Euclidean harmonic heads remain on or very near the performance Pareto frontier.

RQ2: Interpretability (PC2 EV, PCA 90%). The higher-resolution datasets accentuate differences in representation geometry. On Marathi Sign, harmonic distances such as **Bray–Curtis (Normalized/Absolute)**, **Chebyshev (Standard)**, and **Canberra/Hamming** for PVT yield the strongest PCA structure: they maximize PC2 explained variance and minimize the number of components required to reach 90% EV, indicating compact embeddings. ArcFace and Focal improve angular separation but generally do not achieve the same variance concentration as the best harmonic distances.

TinyImageNet confirms this trend. On MLP and CNN, Bray–Curtis and Chebyshev produce markedly higher PC2 EV and lower PCA 90% dimensionality than Euclidean and most additional baselines, including Center Loss and Focal. For ResNet50 and PVT, **cosine** and **Bray–Curtis** continue to enlarge the PCA wedges relative to Euclidean, whereas ArcFace's contribution is mainly on performance, with little change in variance concentration. Thus, across both Marathi Sign and TinyImageNet, the most interpretable geometries are consistently induced by non-Euclidean harmonic losses, and not by the newly added baselines.

RQ3: Sustainability (Duration/Epoch/GFLOPs, Emissions). The sustainability axes show that richer loss design does not necessarily translate into greener training. On Marathi Sign MLP/CNN, several harmonic distances (e.g., **Bray–Curtis (Normalized)**, **Chebyshev**) attain *equal or lower* normalized Duration/Epoch/GFLOPs and emissions than cross-entropy and the added baselines; Focal and ArcFace occasionally incur slightly higher emissions due to their sharper gradients and additional computations. For ResNet50 and PVT, where backbone FLOPs dominate, all harmonic variants remain sustainability-competitive.

On TinyImageNet, the pattern persists. ArcFace and Focal may match harmonic losses in accuracy, but they usually do so with similar or higher emissions. Cosine and Bray–Curtis heads on ResNet50 and PVT often achieve comparable or better emissions than Euclidean, while still improving representation structure. Center Loss introduces modest overhead but does not surpass harmonic distances in overall sustainability.

Across Marathi Sign and TinyImageNet, adding strong baselines such as ArcFace, Focal, Center Loss, and Confidence Penalty *does not displace* the non-Euclidean harmonic losses from the top tier. Whenever these baselines are competitive in accuracy, harmonic distances typically offer superior interpretability and comparable or lower emissions. This reinforces our central claim that distance-tailored harmonic heads provide a robust, geometry-aware alternative to contemporary loss

designs, remaining competitive or superior across performance, structure, and sustainability, even on challenging high-resolution vision benchmarks.

L.4 Vision: Aggregated Emissions

Figure 9 reports the cumulative CO_2 emissions for all vision experiments (MNIST, CIFAR-10/100, Marathi Sign, and TinyImageNet), expressed as the difference in grams of CO_2 relative to the cross-entropy baseline (total CE emissions $= 650.49\,\mathrm{gCO_2eq}$ over 680 runs). Negative values indicate lower measured cumulative emissions and positive values indicate higher measured cumulative emissions.

Observed cumulative differences. Mahalanobis (Std.), Center Loss (λ=2), Bray–Curtis (Std.), and Canberra (Std.) have the largest negative measured totals. Mahalanobis (Chol.), Bray–Curtis (Norm./Absolute), and Canberra (Weighted) have the largest positive totals. The remaining methods span intermediate values, including several configurations close to the cross-entropy baseline.

Behavior of additional baselines. These cumulative totals combine measurements from shared compute nodes and do not control for model quality or the number and duration of runs represented by each configuration. They are therefore descriptive accounting summaries, not estimates of an architecture-independent effect. The paired comparisons in Table 24 and the limitations in Appendix A provide the appropriate inferential context.

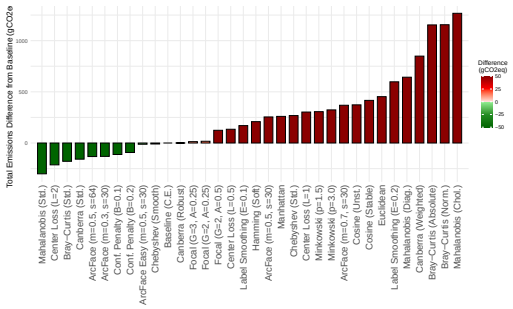

Figure 9: Cumulative measured emission differences across 680 vision runs. Values are summed relative to the corresponding cross-entropy baselines; negative values indicate lower measured emissions and positive values indicate higher measured emissions.

M Convergence Analysis

Vision: Figure 10 reports the training and validation loss trajectories for PVT and ResNet50 across all datasets and all non-Euclidean harmonic losses. A practical concern is whether changing the distance introduces visible optimization instability. Across the displayed MNIST, CIFAR-10, CIFAR-100, and Marathi Sign runs, the training objectives decrease and the validation curves remain bounded. Convergence rates differ among heads, but the plots do not show sustained divergence for the reported configurations. These curves provide a descriptive optimization check; they do not identify loss-surface curvature or establish equivalent convergence rates.

Language: Figure 11 reports training/validation loss, training accuracy, and (for GPT-2B) training and validation perplexity for cross-entropy, Euclidean harmonic, and Minkowski (p=2) heads across BERT-0.1B, GPT-0.1B, GPT-2B, and Qwen2–0.5B. Across the displayed runs, losses decrease and accuracy approaches a plateau without sustained divergence.

For BERT-0.1B, Euclidean and Minkowski harmonic losses reduce both training and validation loss more quickly than cross-entropy and converge to a lower plateau, while achieving higher final training accuracy. GPT-0.1B shows a similar pattern: all three heads converge, and the harmonic variants reach several intermediate accuracy levels earlier. For GPT-2B and Qwen2–0.5B, the three heads

track each other closely in both loss and accuracy. The validation loss curves mirror the training behaviour: no divergence or late-stage degradation is observed for any harmonic configuration.

The GPT-2B perplexity panel further corroborates this picture. Training and validation perplexity decrease and stabilize at comparable levels for all heads; the harmonic variants sometimes have slightly faster early reductions. Overall, the plotted runs show that the evaluated distance heads can be optimized under the reported settings without visible divergence.

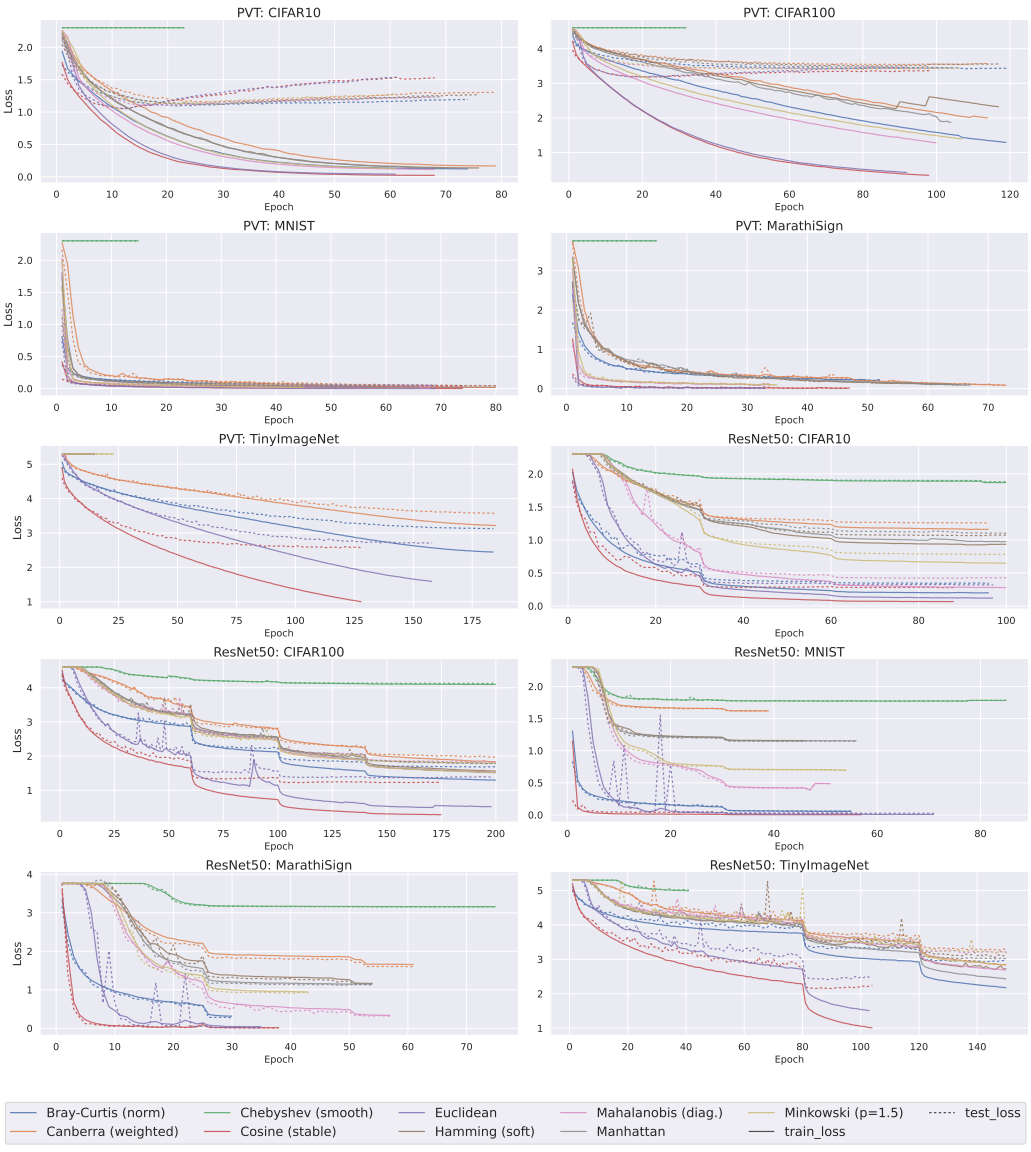

Figure 10: Training and validation loss for PVT and ResNet50 across the evaluated vision datasets and non-Euclidean harmonic heads.

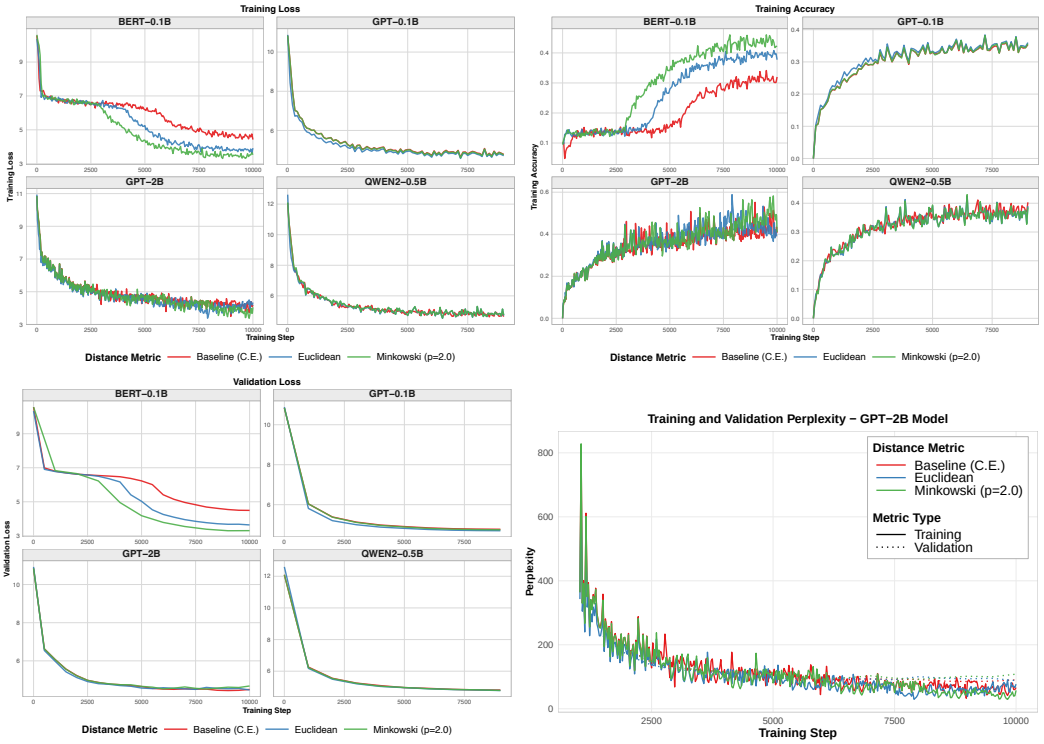

Figure 11: Training and validation trajectories for BERT-0.1B, GPT-0.1B, Qwen2-0.5B, and GPT-2B.

N GROKKING ANALYSIS: MODULO ADDITION

Figure 12 summarizes the behavior of standard MLPs and H-MLPs trained on the synthetic modulo-addition task, a setting known to exhibit pronounced grokking effects under cross-entropy. The top two rows illustrate training curves and corresponding 2D embeddings for baseline cross-entropy models (first two columns) and Euclidean harmonic loss (third and fourth columns). The remaining rows extend this comparison to alternative non-Euclidean harmonic losses.

Cross-entropy exhibits clear grokking. For both the standard MLP and its lightly regularized variant, cross-entropy produces the characteristic grokking pattern: training accuracy rapidly converges while test accuracy improves only after a long delay. This decoupling is consistent with prior observations in algorithmic tasks, where cross-entropy tends to overfit memorization pathways before discovering the true modular arithmetic structure. The PCA plots confirm this: the learned embeddings under cross-entropy have diffuse, irregular geometry, and the first two principal components explain only a small fraction of the variance (EV $\approx$ 20–30%).

Euclidean harmonic loss shows no grokking delay in this experiment. In contrast, the Euclidean harmonic model reaches high train *and* test accuracy simultaneously. No grokking delay is observed. In the PCA projection, the latent representation forms an approximately two-dimensional circle, and the first two principal components explain nearly all variance (EV $\approx 100\%$). This is consistent with the cyclic structure of the task, although the visualization alone does not establish an isometry.

Other distance-based harmonic losses show similar structure in the displayed runs. The bottom rows show that this desirable behavior is *not* unique to Euclidean distance. Cosine, stable cosine, Manhattan (1–norm), several Canberra variants, Hamming losses, Minkowski $p = 3$, Chebyshev, and others all produce the same qualitative outcome:

- **Immediate or near-immediate generalization** in the displayed runs, with no observed grokking phase.
- **Highly structured 2D embeddings**, often forming a near-perfect circle.
- **Explained variance approaching 100%**, indicating strong alignment to a low-dimensional manifold reflecting the algebraic symmetry of the task.

Some distances (e.g., Hamming and Chebyshev) produce slightly rotated or warped circles, but the displayed runs retain the same qualitative structure and variance concentration.

Harmonic losses reduce grokking compared to cross-entropy. Across the non-Euclidean distances displayed, the runs show two differences from cross-entropy:

1. **Reduced or absent delayed generalization**. Training and test accuracy rise together in these runs.
2. **A more concentrated geometric structure**. The learned representations form an approximately circular low-dimensional manifold with EV close to 1.0.

These results show that harmonic losses can produce different low-dimensional representations and learning dynamics on this modular-addition case study. Explained variance near 100% indicates concentration in the first two principal components; it does not by itself establish prototype alignment or human interpretability.

O COMPUTATIONAL COMPLEXITY: FLOPS

Table 29 reports the approximate floating-point operations per forward pass for each backbone-dataset. On 32×32 inputs (MNIST resized, CIFAR-10/100, MarathiSign), the FLOP hierarchy is consistent: MLP is cheapest (<0.004 GFLOPs), CNN roughly $3\times$ more expensive (≈ 0.012 GFLOPs), PVT adds another $\sim3\times$ (≈ 0.038 GFLOPs), and ResNet50 is about $2\times$ PVT (≈ 0.08 GFLOPs). Moving to high-resolution inputs (224×224 for TinyImageNet and our high-resolution MarathiSign runs) increases

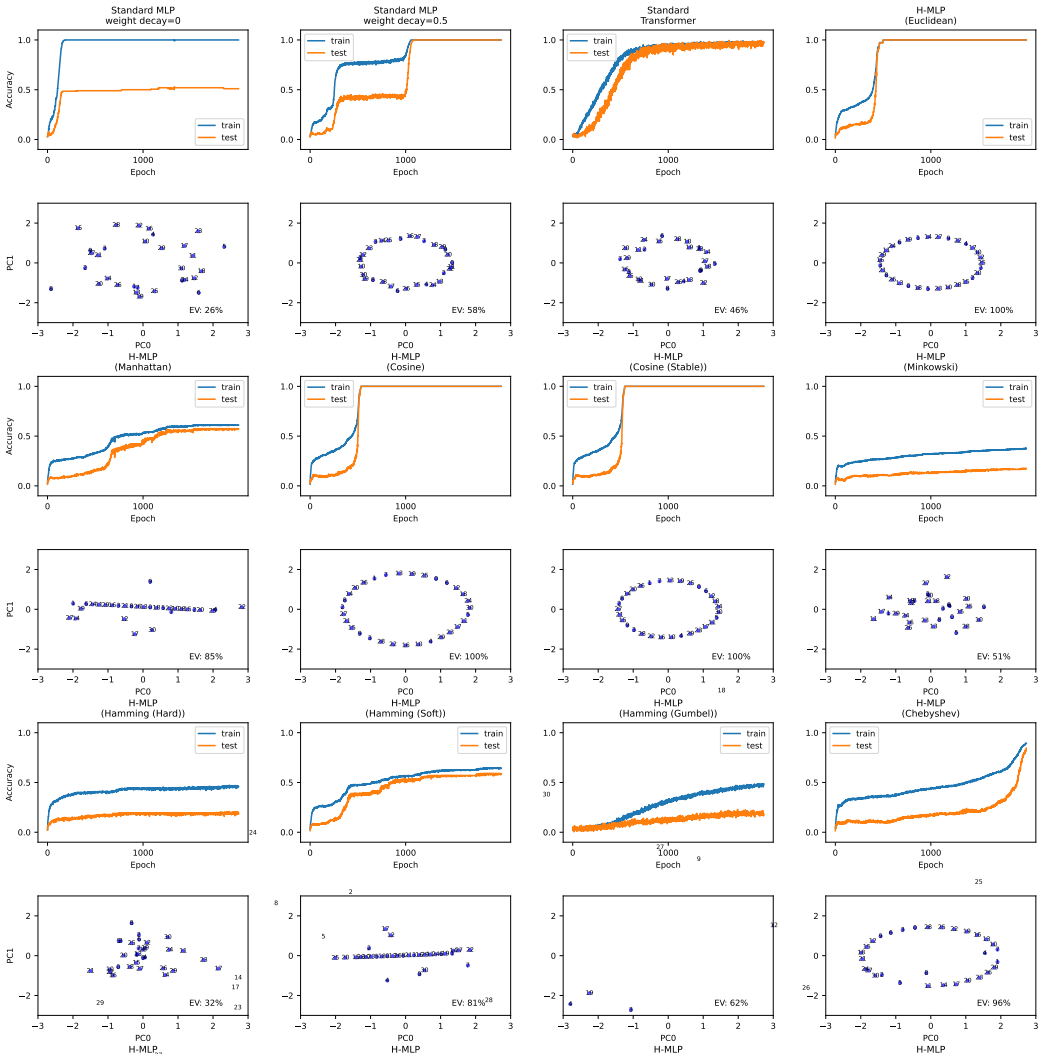

Figure 12: Results on standard MLP trained for modular addition. In the displayed run, the harmonic model trained for modular addition generalizes quickly without an observed grokking delay, and its embedding forms an approximately two-dimensional circle. EV reports the explained variance of the first two principal components.

cost by two orders of magnitude: PVT rises to $\sim$1.9 GFLOPs and ResNet50 to $\sim$4.1 GFLOPs per forward pass. These numbers highlight that i) sustainability differences across *architectures* are dominated by backbone FLOPs, while ii) swapping Euclidean harmonic loss for alternative distances or baselines changes only the final classifier head, adding an $O(Cd)$ cost that is negligible compared to the convolutional / transformer body. Consequently, the per-step FLOP budget is effectively distance-invariant, and our sustainability comparisons across losses can be interpreted as differences in optimization dynamics (steps-to-target, stability), at comparable arithmetic cost.

Table 29: Per-sample FLOPs, GFLOPs, and parameter counts for each backbone and dataset.

Model	Dataset	In Ch.	H	W	#Cls	FLOPs	Params	GFLOPs
CNN	CIFAR10	3	32	32	10	12307072	545098	0.0123
CNN	CIFAR100	3	32	32	100	12330112	556708	0.0123
CNN	MarathiSign	3	32	32	43	12315520	549355	0.0123
CNN	MNIST	1	28	28	10	8520064	421642	0.0085
CNN	TinyImageNet	3	224	224	200	602966144	25735432	0.6029
MLP	CIFAR10	3	32	32	10	3413760	1707274	0.0034
MLP	CIFAR100	3	32	32	100	3459840	1730404	0.0034
MLP	MarathiSign	3	32	32	43	3430656	1715755	0.0034
MLP	MNIST	1	28	28	10	1070848	535818	0.0010
MLP	TinyImageNet	3	224	224	200	309536256	154769096	0.3095
PVT	CIFAR10	3	32	32	10	38268630	12746560	0.0382
PVT	CIFAR100	3	32	32	100	38314710	12769600	0.0383
PVT	MarathiSign	3	32	32	43	38285526	12755008	0.0382
PVT	MNIST	3	32	32	10	38268630	12746560	0.0382
PVT	TinyImageNet	3	224	224	200	1899590400	12795200	1.8995
ResNet50	CIFAR10	3	32	32	10	79618429	23472480	0.0796
ResNet50	CIFAR100	3	32	32	100	79987069	23656800	0.0799
ResNet50	MarathiSign	3	224	224	43	4096080128	23540064	4.0960
ResNet50	MNIST	1	28	28	10	60007460	23472480	0.0600
ResNet50	TinyImageNet	3	224	224	200	4096723200	23861600	4.0967

P GEOMETRIC INSIGHTS

What the metric implies about the prototypes. Write the harmonic loss as $\mathcal{L} = \sum_i \left[\omega \log d(x_i, w_{y_i}) + \log \sum_j d(x_i, w_j)^{-\omega}\right]$. The first term attracts each prototype toward representations from its class, while the normalizing term also depends on its distances to other classes. If the normalizing term is omitted, stationarity of the attraction term for prototype w_k requires

$$\sum_{i:\, y_i = k} \frac{\nabla_w d(x_i, w_k)}{d(x_i, w_k)} \;=\; 0,$$

which is a prototype condition weighted by inverse distance. For $d = \|x - w\|_2^2$, this gives $w_k = \left(\sum_i r_i^{-2} x_i\right) / \left(\sum_i r_i^{-2}\right)$ with $r_i = \|x_i - w_k\|_2$, an inverse-squared-distance weighted mean. For $d = \|x - w\|_1$, it gives $\sum_i \operatorname{sign}(w_k - x_i)/r_i = 0$, the condition for a weighted coordinate-wise median. With a fixed Mahalanobis factor A, the corresponding condition holds in the whitened coordinates Ax. These statements describe only the attraction term. Because the full loss includes the normalizer, its stationary prototypes need not equal empirical class centers, consistent with the measurements in Section 4.3.

To illustrate how harmonic distances are associated with embedding geometry, we visualize two-dimensional PCA projections of the final ResNet50 representations on MNIST (see Figure 13) and CIFAR-10 (see Figure 14), with class prototypes overlaid as markers. Because PCA discards information outside the displayed plane, these plots are qualitative views of the learned geometry. For the Euclidean harmonic head, several class clusters appear approximately compact in the projection, and the displayed decision regions reflect radial distance to the prototypes.

Under cosine harmonic loss, the projected clusters are arranged more strongly by direction. This is consistent with a head whose scores depend on the angles between normalized features and prototypes. The two-dimensional projection alone does not establish hyperspherical concentration or optimization dynamics.

Mahalanobis harmonic loss instead permits anisotropic geometry. With a fixed precision matrix, Euclidean contours in whitened coordinates correspond to ellipsoidal contours in the original feature space. The elongated clusters visible in some projections are consistent with this directional scaling, although the plots do not by themselves identify its effect on optimization.

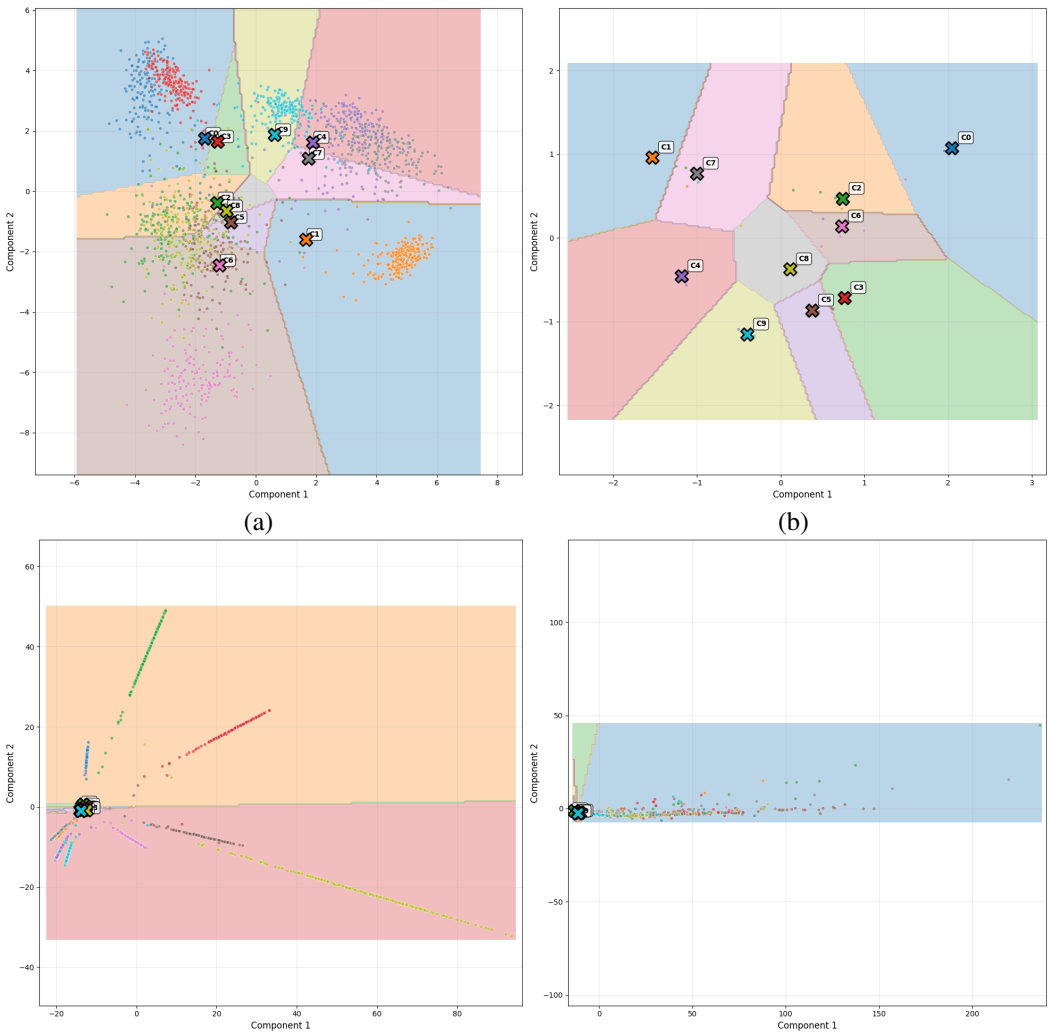

Figure 13: Geometric effect of distance-based harmonic losses on ResNet50 embeddings (MNIST). From top to bottom: Baseline (a), Euclidean harmonic loss (b), cosine harmonic loss (c), and Mahalanobis harmonic loss (d).

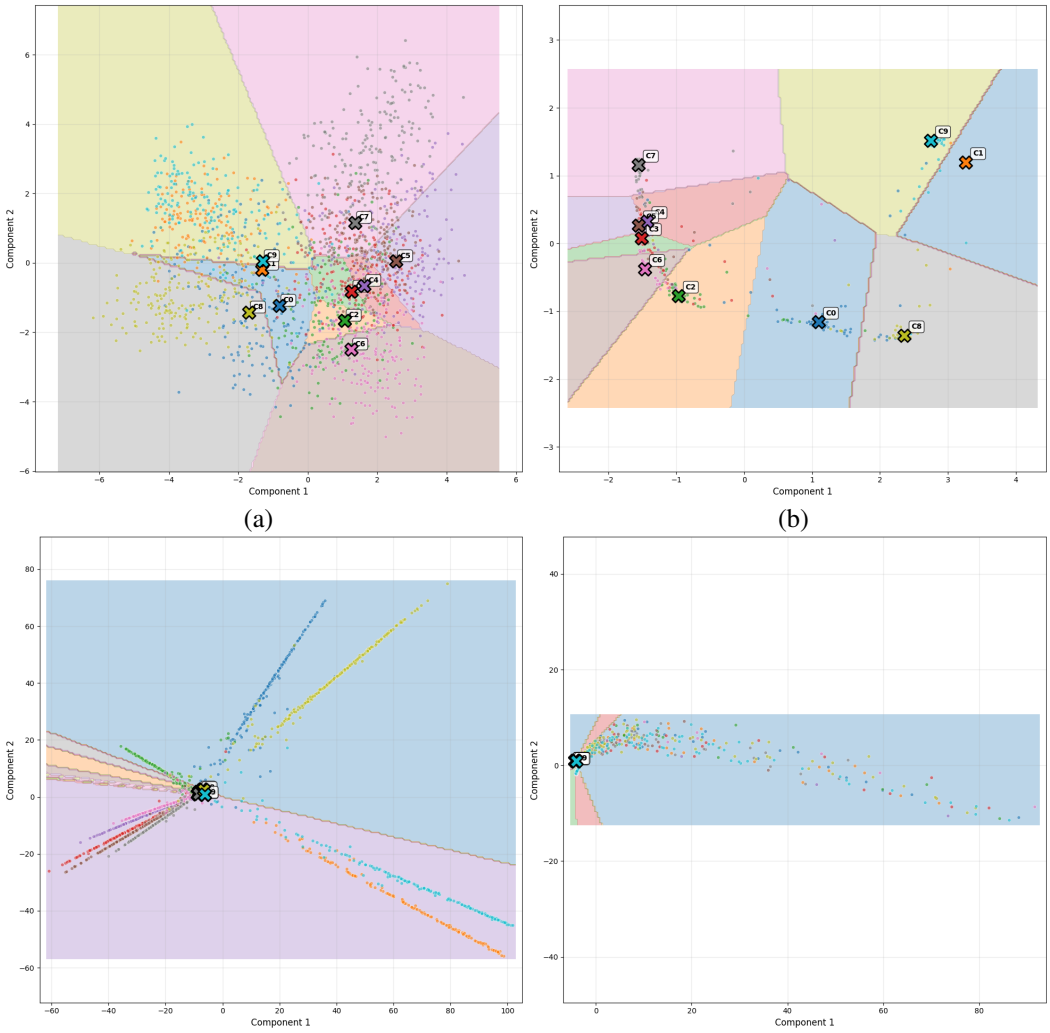

Figure 14: Geometric effect of distance-based harmonic losses on ResNet50 embeddings (CIFAR10). From top to bottom: Baseline (a), Euclidean harmonic loss (b), cosine harmonic loss (c), and Mahalanobis harmonic loss (d).

Table 30: Results for CIFAR100 CNN. Parentheses: % changes w.r.t. Baseline (Cross-Entropy).

Method	Acc	F1	gCO_2eq	EV	PC90%
Baseline	0.3795	0.3795	1.18	0.459295	49.3333
Bray-Curtis (Norm.)	0.3229 (-14.91%)	0.3182 (-16.16%)	0.8132 (30.94%)	0.9094 (98%)	2.6667 (94.59%)
Mahalanobis (Chol.)	0.2927 (-22.86%)	0.2921 (-23.04%)	0.727 (38.25%)	0.341 (-25.75%)	50 (-1.35%)
Cosine (Unst.)	0.2602 (-31.44%)	0.2667 (-29.73%)	2.1156 (-79.68%)	0.5306 (15.52%)	45 (8.78%)
Cosine (Stable)	0.2501 (-34.09%)	0.2516 (-33.71%)	1.4263 (-21.14%)	0.5216 (13.57%)	45 (8.78%)
Euclidean	0.2413 (-36.4%)	0.2431 (-35.95%)	1.2866 (-9.28%)	0.4362 (-5.02%)	50 (-1.35%)

Table 31: Results for CIFAR100 MLP. Parentheses: % changes w.r.t. Baseline (Cross-Entropy).

Method	Acc	F1	gCO_2eq	EV	PC90%
Baseline	0.2617	0.2582	0.85	0.285203	50.0
Bray-Curtis (Norm.)	0.226 (-13.64%)	0.2191 (-15.15%)	0.8938 (-5.35%)	0.9843 (245.11%)	1 (98%)
Mahalanobis (Chol.)	0.1833 (-29.96%)	0.1811 (-29.85%)	1.0889 (-28.34%)	0.0354 (-87.57%)	50 (-0%)
Bray-Curtis (Abs.)	0.1444 (-44.81%)	0.1392 (-46.07%)	2.1255 (-150.51%)	0.5317 (86.43%)	47.6667 (4.67%)
Cosine (Unst.)	0.1237 (-52.74%)	0.1186 (-54.06%)	0.5064 (40.31%)	0.3799 (33.21%)	40.6667 (18.67%)
Euclidean	0.119 (-54.53%)	0.1222 (-52.69%)	0.589 (30.58%)	0.2437 (-14.55%)	50 (-0%)

Q Additional results

Q.1 Vision: Tables

The empirical evaluation of non-Euclidean harmonic losses across MNIST, CIFAR-10, and CIFAR-100 with MLP, CNN, and ResNet50 backbones reveals several descriptive patterns.

Model Performance. Cosine is frequently among the strongest harmonic variants, particularly on deeper models, while Bray–Curtis and Mahalanobis are more configuration-dependent. These rankings are descriptive: the paired vision comparisons do not survive Holm correction.

Interpretability. Distances reshape the geometry of the learned representations. Cosine and Bray–Curtis often increase explained variance, and Mahalanobis sometimes produces extreme concentration. PCA@90% counts principal components—not prototypes—and lower values indicate concentration rather than human interpretability; the comparison is meaningful only at similar predictive accuracy.

Sustainability. Measured emissions are mixed and architecture-dependent. Some cosine runs are lower than Euclidean, while Mahalanobis is often costlier; differences within the observed run-to-run variation should not be interpreted as resolved effects.

Trade-offs. Taken together, the results show that distance choice is not neutral. Cosine is often competitive, Bray–Curtis occupies a middle ground, and Mahalanobis trades strong concentration in some settings for higher computational cost; no metric is uniformly best.

Conclusion. This study establishes non-Euclidean harmonic losses as a flexible design space whose accuracy, representation structure, and computational cost vary by metric and architecture.

Table 32: Results for CIFAR100 ResNet50. Parentheses: % changes w.r.t. Baseline (Cross-Entropy).

Method	Acc	F1	gCO_2eq	EV	PC90%
Baseline	0.6983	0.6969	87.77	0.107216	50.0
Cosine (Stable)	0.7357 (5.35%)	0.736 (5.61%)	72.9745 (16.85%)	0.5979 (457.66%)	8 (84%)
Cosine (Unst.)	0.7323 (4.87%)	0.7332 (5.21%)	71.7592 (18.24%)	0.5857 (446.27%)	8 (84%)
Bray-Curtis (Norm.)	0.655 (-6.19%)	0.6513 (-6.54%)	106.4049 (-21.24%)	0.7131 (565.08%)	6 (88%)
Mahalanobis (Chol.)	0.6274 (-10.15%)	0.6239 (-10.47%)	138.9317 (-58.3%)	0.7353 (585.81%)	17.5 (65%)
Euclidean	0.7055 (1.03%)	0.7062 (1.33%)	97.432 (-11.01%)	0.5679 (429.66%)	25.5 (49%)

Table 33: Results for CIFAR10 CNN. Parentheses: % changes w.r.t. Baseline (Cross-Entropy).

Method	Acc	F1	gCO_2eq	EV	PC90%
Baseline	0.6278	0.6269	1.12	0.688081	9.0
Mahalanobis (Chol.)	0.6644 (5.82%)	0.6642 (5.95%)	1.1139 (0.68%)	0.4752 (-30.93%)	50 (-455.56%)
Bray-Curtis (Norm.)	0.6597 (5.08%)	0.6551 (4.5%)	1.1489 (-2.45%)	0.8913 (29.54%)	4.3333 (51.85%)
Minkowski (p=3.0)	0.6589 (4.95%)	0.6593 (5.17%)	1.1598 (-3.42%)	0.5425 (-21.15%)	50 (-455.56%)
Cosine (Stable)	0.6584 (4.87%)	0.6566 (4.74%)	1.1663 (-3.99%)	0.647 (-5.97%)	18.6667 (-107.41%)
Euclidean	0.6495 (3.45%)	0.6476 (3.31%)	1.1228 (-0.12%)	0.6582 (-4.34%)	14.3333 (-59.26%)

Table 34: Results for CIFAR10 MLP. Parentheses: % changes w.r.t. Baseline (Cross-Entropy).

Method	Acc	F1	gCO_2eq	EV	PC90%
Baseline	0.5397	0.5385	0.53	0.346504	47.0
Bray-Curtis (Norm.)	0.5224 (-3.21%)	0.5201 (-3.41%)	0.5264 (0.81%)	0.967 (179.07%)	1 (97.87%)
Mahalanobis (Chol.)	0.5087 (-5.75%)	0.5088 (-5.51%)	0.458 (13.7%)	0.0522 (-84.94%)	50 (-6.38%)
Bray-Curtis (Abs.)	0.4934 (-8.59%)	0.4924 (-8.55%)	0.6313 (-18.96%)	0.2434 (-29.76%)	50 (-6.38%)
Bray-Curtis (Std.)	0.4931 (-8.64%)	0.4935 (-8.35%)	0.6435 (-21.25%)	0.2906 (-16.14%)	50 (-6.38%)
Euclidean	0.4871 (-9.74%)	0.4852 (-9.9%)	0.4303 (18.92%)	0.4303 (24.19%)	42.3333 (9.93%)

Table 35: Results for CIFAR10 ResNet50. Parentheses: % changes w.r.t. Baseline (Cross-Entropy).

Method	Acc	F1	gCO_2eq	EV	PC90%
Baseline	0.843	0.8431	48.65	0.257211	50.0
Cosine (Stable)	0.9262 (9.87%)	0.9262 (9.86%)	40.6776 (16.39%)	0.7559 (193.9%)	5 (90%)
Cosine (Unst.)	0.9234 (9.54%)	0.9234 (9.53%)	29.3968 (39.58%)	0.761 (195.86%)	5 (90%)
Bray-Curtis (Norm.)	0.9193 (9.05%)	0.9192 (9.02%)	45.6222 (6.23%)	0.7883 (206.49%)	5 (90%)
Chebyshev (Std.)	0.905 (7.36%)	0.905 (7.34%)	48.5505 (0.21%)	0.9995 (288.59%)	1 (98%)
Euclidean	0.9185 (8.96%)	0.9185 (8.94%)	45.8759 (5.71%)	0.683 (165.56%)	25.5 (49%)

Table 36: Results for MNIST CNN. Parentheses: % changes w.r.t. Baseline (Cross-Entropy).

Method	Acc	F1	gCO_2eq	EV	PC90%
Baseline	0.9782	0.9782	1.19	0.585633	10.6667
Bray-Curtis (Norm.)	0.9889 (1.09%)	0.9888 (1.09%)	1.1348 (4.42%)	0.7225 (23.38%)	13.6667 (-28.12%)
Mahalanobis (Chol.)	0.9879 (1%)	0.9879 (0.99%)	1.0639 (10.39%)	0.4673 (-20.2%)	36.3333 (-240.63%)
Minkowski (p=3.0)	0.9877 (0.97%)	0.9876 (0.96%)	1.1154 (6.06%)	0.4195 (-28.37%)	49.3333 (-362.5%)
Hamming (Soft)	0.9833 (0.52%)	0.9832 (0.51%)	1.1815 (0.49%)	0.3089 (-47.26%)	50 (-368.75%)
Euclidean	0.9831 (0.5%)	0.9831 (0.5%)	1.1543 (2.78%)	0.4413 (-24.65%)	20.3333 (-90.62%)

Table 37: Results for MNIST MLP. Parentheses: % changes w.r.t. Baseline (Cross-Entropy).

Method	Acc	F1	gCO_2eq	EV	PC90%
Baseline	0.976	0.9758	0.55	0.565723	10.3333
Cosine (Unst.)	0.978 (0.2%)	0.9778 (0.2%)	0.5264 (3.58%)	0.382 (-32.48%)	10 (3.23%)
Mahalanobis (Chol.)	0.9774 (0.14%)	0.9771 (0.14%)	0.5611 (-2.78%)	0.092 (-83.74%)	50 (-383.87%)
Cosine (Stable)	0.9766 (0.06%)	0.9764 (0.06%)	0.5266 (3.54%)	0.4033 (-28.71%)	9.3333 (9.68%)
Chebyshev (Std.)	0.9756 (-0.04%)	0.9754 (-0.04%)	0.5881 (-7.73%)	0.7865 (39.03%)	5.6667 (45.16%)
Euclidean	0.9799 (0.4%)	0.9798 (0.41%)	0.5221 (4.35%)	0.358 (-36.72%)	9 (12.9%)

Table 38: Results for MNIST ResNet50. Parentheses: % changes w.r.t. Baseline (Cross-Entropy).

Method	Acc	F1	gCO_2eq	EV	PC90%
Baseline	0.9909	0.9909	29.36	0.420353	50.0
Bray-Curtis (Norm.)	0.9962 (0.52%)	0.9961 (0.53%)	25.2889 (13.86%)	0.8453 (101.09%)	4 (92%)
Cosine (Unst.)	0.996 (0.51%)	0.996 (0.52%)	26.1851 (10.8%)	0.6888 (63.87%)	6 (88%)
Cosine (Stable)	0.9953 (0.44%)	0.9953 (0.45%)	26.4064 (10.05%)	0.6974 (65.91%)	6 (88%)
Mahalanobis (Chol.)	0.9938 (0.29%)	0.9938 (0.3%)	31.9246 (-8.75%)	0.9966 (137.09%)	1 (98%)
Euclidean	0.9934 (0.25%)	0.9934 (0.25%)	24.457 (16.69%)	0.9998 (137.84%)	1 (98%)

Q.2 VISION: SUSTAINABILITY

Q.2.1 MNIST

Figure 15 summarizes the *carbon deltas* (gCO_2eq relative to cross-entropy) when swapping the training objective for harmonic-loss variants on MNIST across four backbones.

MLP. Most distances have lower measured per-step emissions than cross-entropy, while a few show small increases. Euclidean and Bray–Curtis are among the variants with modest reductions. Because these measurements were collected on shared nodes, small differences should be interpreted descriptively (see Appendix A).

CNN. A broad set of distances have lower measured emissions than the baseline. Standardized Mahalanobis and Chebyshev are among the lowest values in this setting, while Bray–Curtis and Euclidean also remain below cross-entropy.

ResNet50. This backbone shows the largest negative measured deltas: many distances fall below cross-entropy, while a few Chebyshev and Canberra parameterizations have small positive deltas.

PVT (vision transformer). In contrast to the CNN family, most distances have higher measured per-step emissions than the baseline; only a few standardized or normalized variants show small reductions.

Takeaways. On MNIST, many distance heads have lower measured per-step emissions for MLP, CNN, and ResNet50, whereas most PVT variants have higher values. These observations do not establish end-to-end energy savings; that comparison also requires matched-quality training budgets and repeated measurements on dedicated nodes.

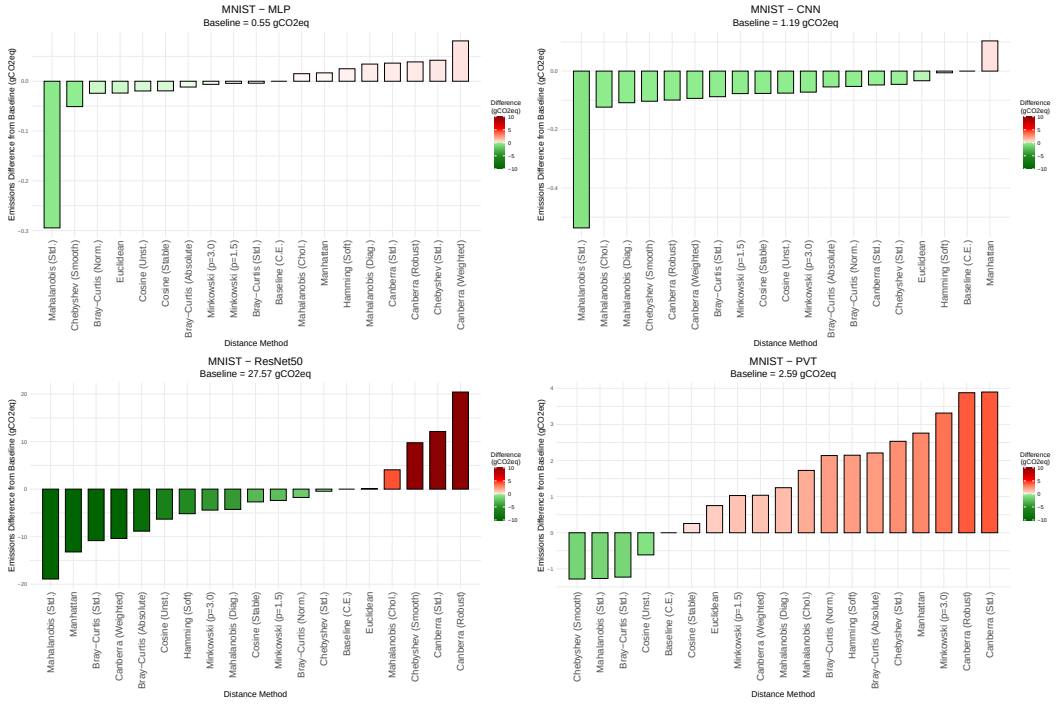

Figure 15: Measured emission differences for MNIST across four backbones (MLP, CNN, ResNet50, PVT). Bars report gCO_2eq relative to cross-entropy; positive values indicate higher measured emissions and negative values indicate lower measured emissions.

Q.2.2 CIFAR-10

Figure 16 reports carbon deltas in gCO_2eq relative to cross-entropy when training with harmonic-loss distances on CIFAR-10.

MLP. Most distances have lower measured per-step emissions than the baseline, while a few have small positive deltas.

CNN. A broad set of distances have lower measured per-step emissions than cross-entropy; only a few variants are near zero or slightly positive.

ResNet50. All evaluated distances fall below the measured baseline, and this backbone has the largest negative deltas in the figure.

PVT (vision transformer). Most distances have lower measured emissions than cross-entropy, although the spread is narrower than for ResNet50 and a few variants are near parity or slightly higher.

Takeaways. On CIFAR-10, most distance heads have lower measured per-step emissions for CNN, ResNet50, and PVT, while the MLP results are more mixed. End-to-end comparisons require energy-to-target-quality measurements under controlled hardware conditions.

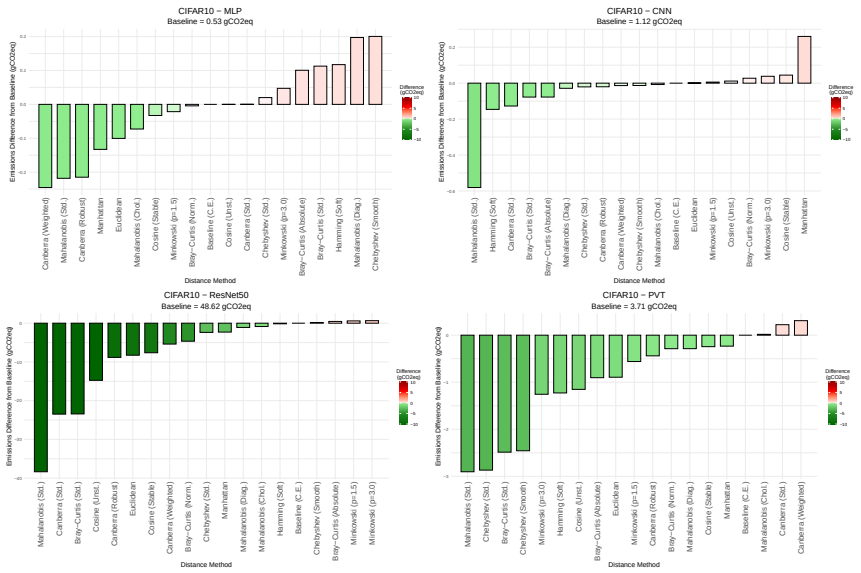

Figure 16: Measured emission differences for CIFAR-10 across four backbones (MLP, CNN, ResNet50, PVT). Bars report gCO_2eq relative to cross-entropy; positive values indicate higher measured emissions and negative values indicate lower measured emissions.

Q.2.3 CIFAR-100

Figure 17 shows the carbon *delta* (gCO_2eq vs. cross-entropy) when training with harmonic-loss distances on CIFAR-100.

MLP. The measured differences are modest and geometry-dependent. Cosine, Euclidean, and some Minkowski and Canberra settings fall below cross-entropy, while several covariance-based Mahalanobis variants have positive deltas.

CNN. Many distances fall below the 1.18 gCO_2eq baseline; several Mahalanobis and Bray–Curtis settings have the largest negative measured deltas. A few cosine, Canberra, and Minkowski configurations are near parity or slightly higher.

ResNet50. The measured deltas have a wide spread: many distances fall below cross-entropy, while others are clearly above it. Cosine variants are among the lower measured values, whereas some Chebyshev, Minkowski, and absolute Bray–Curtis settings are higher.

PVT (vision transformer). Most distances have higher measured emissions than the 3.67 gCO_2eq baseline, with only a few standardized or smoothed variants slightly below it.

Takeaways. On CIFAR-100, the measured deltas depend strongly on both distance and backbone: many CNN and some ResNet50 configurations fall below cross-entropy, whereas most PVT configurations lie above it. Because accuracy converges differently across distances, these measurements should be paired with energy-to-target-quality before drawing end-to-end conclusions.

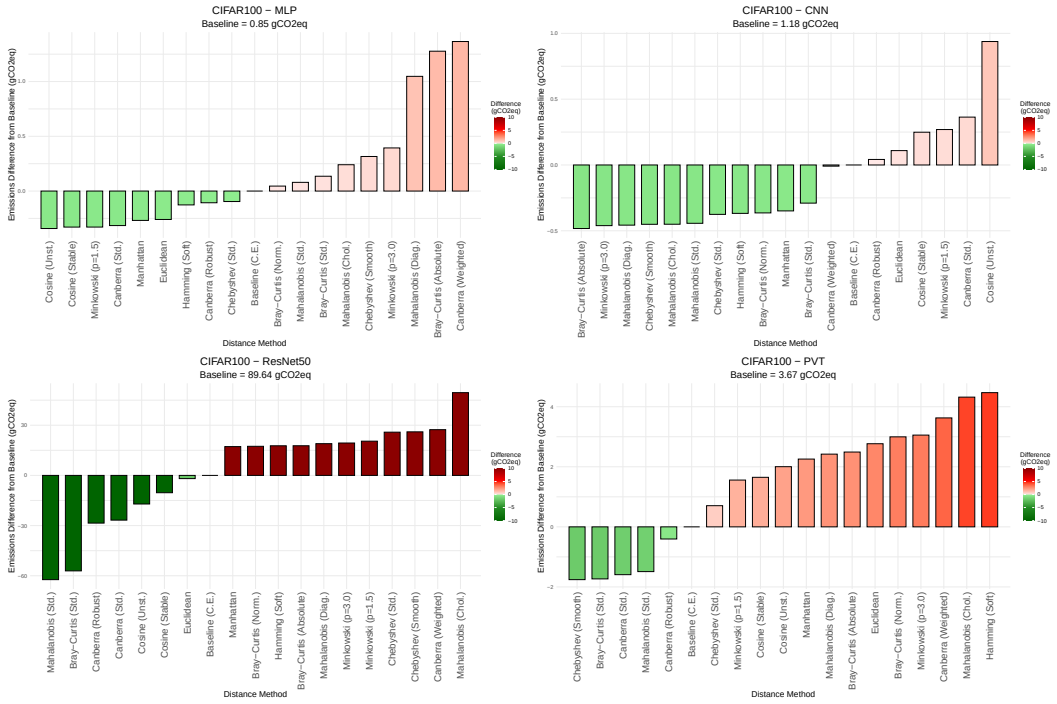

Figure 17: Measured emission differences for CIFAR-100 across four backbones (MLP, CNN, ResNet50, PVT). Bars report gCO_2eq relative to cross-entropy; positive values indicate higher measured emissions and negative values indicate lower measured emissions.

Insights across datasets. Across these measurements, PVT often has positive deltas, particularly on CIFAR-100 (see Figure 17), whereas CNN and ResNet50 more often have negative deltas (see Figures 15 to 17). The direction and magnitude nevertheless vary by dataset and distance.

Cross-architecture summary. MLP deltas generally remain close to the baseline. Several CNN and ResNet50 configurations have lower measured emissions, while PVT configurations more often have higher values, especially on CIFAR-100. Given the measurement variability described in Appendix A, these patterns are descriptive and do not establish an architecture-level causal effect.

Overall, measured energy and emissions vary with the distance, backbone, and dataset. Controlled repeated measurements and matched-quality budgets are needed to distinguish small differences reliably.

Q.3 Language: Sustainability

Figure 18 reports *per-1k-step* carbon differences (gCO_2eq) when replacing cross-entropy with distance-based harmonic losses for BERT, GPT, and Qwen. Positive bars indicate higher emissions than the cross-entropy baseline (annotated atop each subplot).

Overall. Across all three backbones, most distance-based losses have higher measured per-1k-step emissions than cross-entropy. Cosine variants have the smallest deltas, while Mahalanobis and Minkowski have the largest.

BERT. Cosine variants have small positive deltas relative to the 7.87 gCO_2eq baseline. Euclidean and Bray–Curtis lie between cosine and the larger measured deltas of the Mahalanobis and Minkowski variants.

GPT. All evaluated distances have higher measured emissions than the 60.36 gCO_2eq baseline. Cosine has the smallest delta, Euclidean and Manhattan are intermediate, and the Mahalanobis and Minkowski variants have the largest.

Qwen. For this model (baseline 75.29 gCO_2eq), the two evaluated alternatives, Minkowski/L2 and Euclidean, both have positive measured deltas, with the larger value for Minkowski/L2. The limited set of heads does not support a broader comparison among distance families.

Implications. Among the evaluated alternatives, cosine has the smallest measured per-step overhead, while Mahalanobis and Minkowski have the largest. Because the reported values are per 1,000 steps, end-to-end comparisons also require steps-to-target-quality under controlled hardware conditions.

Summary. The measured per-step energy cost differs across heads. Energy-normalized quality metrics and repeated measurements on dedicated nodes would provide a stronger basis for comparing end-to-end efficiency.

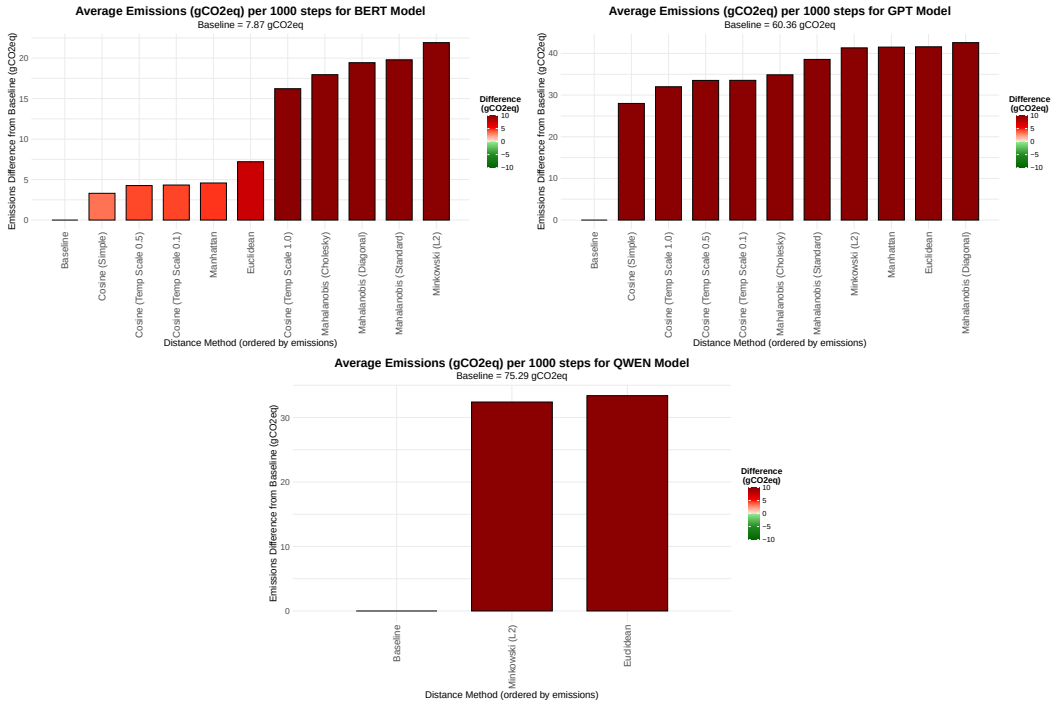

Figure 18: Measured emission differences for language-model pretraining on OpenWebText (BERT, GPT-2, Qwen). Bars report gCO_2eq per 1,000 steps relative to cross-entropy; positive values indicate higher measured emissions and negative values indicate lower measured emissions.

Q.4 Language: Interpretability

Mechanistic and representation-level interpretability of large language models (LLMs) increasingly leverages the hypothesis that internal activations admit *approximately linear* structure: many features behave like directions in an activation space, and linear operations can steer or probe them (Elhage et al., 2022; Huben et al., 2024; Turner et al., 2023). Within this line of work, Principal Component Analysis (PCA) is a simple, well-understood lens for: i) summarizing dominant sources of variance in activations; ii) stabilizing analyses by denoising; and (iii) producing human-auditable axes that can be inspected, correlated with concepts, and tracked over time.

Given a layer ℓ with residual-stream activations $H_\ell \in \mathbb{R}^{N\times d}$ collected across N tokens (or prompts), PCA factorizes H_ℓ via SVD to yield orthogonal directions $\{u_k\}_{k=1}^{d}$ ordered by explained variance. In practice this supports:

1. **Concept probing and visualization.** Projections onto top PCs often align with semantically meaningful contrasts; e.g., the first PC of GPT-style embeddings correlated with human well-being judgments in zero-shot tests (Freire et al., 2024), and per-layer PCA can reconstruct or predict response modes in GPT-2 (Jorgensen, 2023).
2. **Diagnosing and localizing phenomena.** Layer-wise or head-wise PCA reveals where variance concentrates, helping localize depth at which concepts emerge or consolidate (complementary to linear probing) (Jin et al., 2025). Tracking *subspace distance* across checkpoints detects representational drift during fine-tuning or domain shift.
3. **Sanity checks and baselines.** With growing interest in sparse autoencoders (SAEs) for monosemantic features (Huben et al., 2024), PCA is a transparent baseline decomposition: if SAEs meaningfully improve sparsity/faithfulness over PCA while matching reconstruction, that strengthens the interpretability claim (Huben et al., 2024).

PCA is most compelling under: a) approximately linear feature superposition and b) high signal-to-noise in dominant directions. Toy and empirical studies argue that Transformers often encode many features as *directions* (superposition) (Elhage et al., 2022), and even simple linear additions to activations can steer model behavior (Turner et al., 2023). PCA then becomes an appropriate first-pass tool to:

- extract high-variance axes that frequently correlate with coherent features or tasks,
- reduce dimensionality before causal tests (e.g., ablate/project-out a PC and re-evaluate behavior),
- build compact surrogates (e.g., PCA embeddings for downstream analyses or compression) (Bengtsson, 2025; He et al., 2025).

Under widely observed linear-structure assumptions in Transformer activations, PCA offers an interpretable, testable starting point: it surfaces dominant directions, supports hypothesis generation, and provides quantitative targets for more advanced decompositions.