PROTOSEAM: LIFTING CLASSIFIER TRAINING WITH LATENT GAUSSIAN MIXTURE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

We propose a lifted reformulation of supervised classification that improves the final accuracy of standard classifiers without changing the architecture at inference time. A network $N = N_2 \circ N_1$ is split at a single semantic interface and one learnable prototype per class is inserted there. Training combines a quadratic consensus penalty that pulls $N_1(x)$ toward the prototype of its class with a classification loss of N_2 evaluated on samples drawn around the prototypes, whereat no gradient crosses the interface. At inference the prototypes are discarded and the unmodified network $N_2 \circ N_1$ is used. Across CIFAR-10, CIFAR-100, and TinyImageNet with ResNet and vision transformer backbones, lifted training improves test accuracy by up to five percentage points over variants without lifting under a shared tuning protocol. Moreover, we provide theoretical justification of those results.

1 INTRODUCTION

Deep networks for supervised classification are usually trained end-to-end. A single loss is back-propagated through the entire network N in order to maximize test accuracy. We instead view the network as a composition $N = N_2 \circ N_1$, which introduces no additional parameters at inference time. During end-to-end training, the two subnetworks are coupled in two ways. First, by the chain rule, the gradient signal reaching the early subnetwork N_1 must pass through the Jacobian of N_2. Whenever this Jacobian is ill-conditioned or rank-deficient, this can lead to a distorted gradient signal for N_1 is degraded. Second, the classifier head N_2 is trained exclusively on the embeddings $N_1(x)$ of the training samples. Nothing in the objective constrains its behavior in a neighborhood of these embeddings, even though small perturbations of the representation are precisely what the head encounters as N_1 continues to evolve during training.

Prior work addressed the first coupling through decoupled and local training, using per-layer auxiliary variables, local error signals, or synthetic gradients (see Section 2). However, these methods target parallelism or reduced memory, and at best match the accuracy of end-to-end training. Prototype- and margin-based losses, in contrast, shape the embedding geometry but retain the end-to-end gradient path and thus leave the second coupling in place.

Here we propose a lifted reformulation of classifier training that removes both couplings at once, at a single, semantically meaningful interface (the "seam"). The construction transfers Bock's direct multiple shooting method Bock & Plitt (1984) from control to classification. There, additional variables are introduced together with matching conditions, leading to the same solution upon convergence. Among other advantages, this formulation leads to better conditioning and convergence gains (Lampel & Sager, 2025; 2026). Here, we insert n learnable lifted variables $s_1, \ldots, s_n$, one prototype per class, at the interface between N_1 and N_2 (Figure 2). Assuming that the output of N_1 for class i is normally distributed around s_i, the training loss has two decoupled terms:

(i) a consensus penalty that drives $N_1(x)$ toward the prototype of its class, and

(ii) a classification loss that trains N_2 on samples drawn from the class-conditional distribution around each prototype.

An inter-class repulsion penalty keeps the prototypes separated. No gradient crosses the interface, and by discarding the lifted variables and reconnecting both networks we recover the unmodified composition $N_2 \circ N_1$.

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.

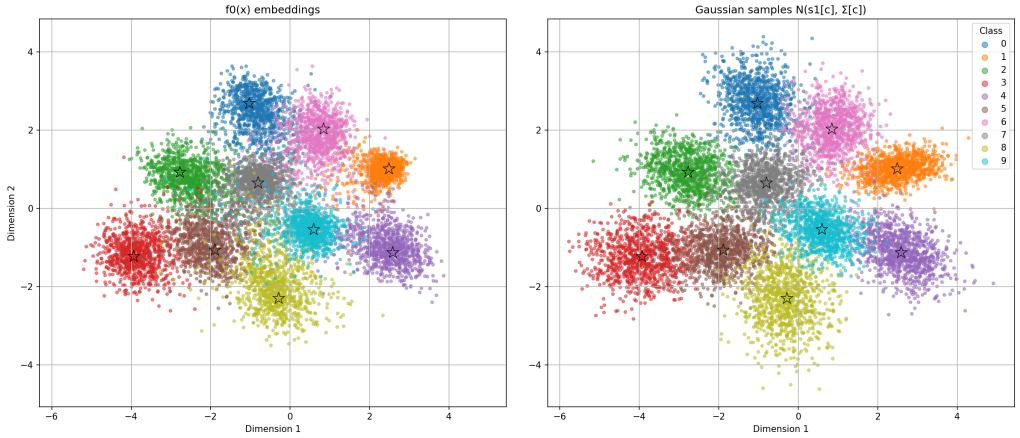


Figure 1: Output of the trained first network part N_1 (left) and the sampled input for the second part N_2 (right) for the MNIST dataset. The stars denote the position of the lifted variables, i.e., the class means of the sampled normal distribution. Here, N_1 consists of two convolutional layers, mapping to the embedding dimension $\mathbb{R}^2$. N_2 consists of one fully connected layer from $\mathbb{R}^2$ to $\mathbb{R}^{10}$.

The reformulation delivers improved final test accuracy over both a matched end-to-end control and a standard baseline across six dataset-architecture pairs (CIFAR-10, CIFAR-100, and TinyImageNet, each with a ResNet and a custom small vision transformer (ViT-S) backbone), under a shared hyperparameter-tuning protocol (Table 1). We also account for the gain theoretically in Section B).

Contributions.

1. We formulate lifting at a single semantic interface of a classifier, with a Gaussian class-prototype structure at the lifting point and no gradient across it (Section 3, Algorithm 1). The deployed architecture is unchanged.
2. We demonstrate consistent accuracy improvements over both a matched end-to-end control and an unsplit baseline, up to five percentage points across six dataset-architecture pairs, together with a $k, \rho_{\max}$ sweep on CIFAR-100 for the ViT-S backbones (Section 4).
3. We give a theoretical account of the gain (Section B). An interface risk-transfer bound shows that the sampled classification loss is the deployed risk up to mismatch terms the objective itself controls (Theorem B.3). A second-order analysis identifies sampling as an explicit sensitivity regularizer on the head (Theorem B.5). Finally, we consider an exactly solvable one-dimensional model in which the selection effect is closed-form (Theorem B.6).

The Gaussian distributions we impose per class is empirically well-motivated by the neural collapse phenomenon (Papyan et al., 2020). Deep classifiers trained with cross-entropy converge to tight class-conditional clusters with means arranged as a Simplex Equiangular Tight Frame during the terminal phase of training. Our lifted loss promotes a similar geometry from initialization rather than waiting for it to emerge asymptotically (Figure 1). During this process the within-class variance degenerates towards zero, conflicting with our approach of sampling from the empirical covariance. We address this explicitly in Section 3.

The term *lifting* has been used for a multitude of ideas across various fields. This work is conceptually distinct from the "lifted neural networks" of Askari et al. (2018), which introduce auxiliary activation variables at every layer to enable block-coordinate descent without imposing any distributional structure. It also differs from metric-learning methods that use the word "lifted" in the sense of lifting pairwise distances to the full-batch distance matrix (Song et al., 2016). Our approach introduces a single set of class prototypes at one semantically meaningful location, grounded both in the multiple-shooting tradition and in the distributional assumptions of variational latent-variable models.

Notation. Throughout, n is the number of classes, m the number of training samples, d the input dimension and k the embedding (lifting) dimension. We write $[n] = \{1, \dots, n\}$. The network is split

as $N = N_2 \circ N_1$ with parameters θ_1, θ_2; $N_1 : \mathbb{R}^d \to \mathbb{R}^k$ and $N_2 : \mathbb{R}^k \to \mathbb{R}^n$. Class prototypes are $s_1, \ldots, s_n \in \mathbb{R}^k$, collected column-wise in $S \in \mathbb{R}^{k \times n}$. For a symmetric positive semi-definite A we write $\kappa(A) = \lambda_{\max}(A)/\lambda^+_{\min}(A)$, with $\lambda^+_{\min}$ the smallest positive eigenvalue, and vec denotes column-stacking vectorisation, so that $\mathrm{vec}(AB) = (B^\top \otimes I)\,\mathrm{vec}(A)$.

2 Related Work

Lifting and decoupled training. For the numerical solution of optimal control problems direct multiple shooting (Bock & Plitt, 1984) has become a standard approach. The decomposition into smaller subproblems together with matching constraints has various advantages. Among those are superior stability, reduced nonlinearity, and improved convergence speed. The latter aspect has been investigated for boundary value and optimal control problems, including neural ODEs (Lampel & Sager, 2025; 2026; Chen et al., 2018). For multiple shooting the matching constraints hold exactly at the solution (Nocedal & Wright, 2006), whereas we only enforce it using a penalty term in our final objective. The analogy is therefore structural rather than one of constraint satisfaction.

The same idea appears in deep learning as per-layer auxiliary variables: lifted neural networks (Askari et al., 2018) and their Fenchel relaxation (Gu et al., 2020), the Method of Auxiliary Coordinates (Carreira-Perpiñán & Wang, 2014), and ADMM training (Taylor et al., 2016). Related work removes the end-to-end path using local error signals (Nøkland & Eidnes, 2019), greedy layer-wise training (Belilovsky et al., 2019), or predicted gradients (Jaderberg et al., 2017). Techniques from differential equations were transferred to deep residual neural networks to enable layer-parallel training (Gunther et al., 2020). All these decouple every layer and none impose a structure on the latent representation.

Class prototypes and margin losses. Nearest-class-mean classifiers (Mensink et al., 2013) and Prototypical Networks (Snell et al., 2017) evaluate an input based on its distance to a class centroid, which is computed either post hoc or from an episode support set. In constrast, our s_i are being optimized with the network and enter the losses of both subnetworks. The closest related idea is center loss (Wen et al., 2016), which jointly learns a center c_y per class and penalizes $\|f(x) - c_y\|_2^2$ alongside softmax. The consensus term of (5) coincides with this penalty. What is new in our approach is the removal of the gradient path from the class loss back to the parameters of N_1 and the inter-class repulsion $\mathcal{P}$. Finally, angular-margin objectives (Liu et al., 2016; 2017; Deng et al., 2019) constrain the classifier weights, while remaining end-to-end.

Gaussian embeddings and neural collapse. Wan et al. (2018) proposed the Large-Margin Gaussian Mixture (L-GM) loss, which models deep features as class-conditional Gaussians and adds a likelihood term to the cross-entropy objective. This is the closest existing work in terms of the assumed distributional model assumed. Both L-GM and our method impose per-class Gaussianity in the penultimate feature space. The distinction is that L-GM trains end-to-end, while our lifted loss decouples N_1 from N_2. Lee et al. (2018) confirm empirically that class-conditional Gaussians fit penultimate-layer features well, using a Mahalanobis-distance detector for out-of-distribution inputs. The *neural collapse* phenomenon of Papyan et al. (2020) further establishes that deep classifiers naturally develop class-conditional clusters with means on a Simplex ETF at the end of training. Zhu et al. (2021) proved the global optimality of this structure under an unconstrained features model. Closest in spirit to our approach of enforcing structure is Yang et al. (2022), who *fix* the classifier to a Simplex ETF and train only the backbone against it. That work fixes the geometry a priori while we learn the prototypes and additionally cut the gradient path between the two subnetworks.

Latent-variable models. We borrow the reparameterization trick from the variational auto-encoders (VAE) (Kingma & Welling, 2013), and the per-class Gaussian structure is shared with deep latent Gaussian mixtures (Nalisnick et al., 2016; Kingma et al., 2014) and with FlowGMM (Izmailov et al., 2020). The latter one enforces class-conditional Gaussianity in a normalizing-flow latent space (Rezende & Mohamed, 2015). There are, however, two important differences. First, flows require invertible architectures, incompatible with a standard ResNet (He et al., 2016), whereas our Gaussian structure is a soft penalty applicable to any differentiable model. Second, unlike a VAE, our reparameterization is purely a sampling device for training N_2.

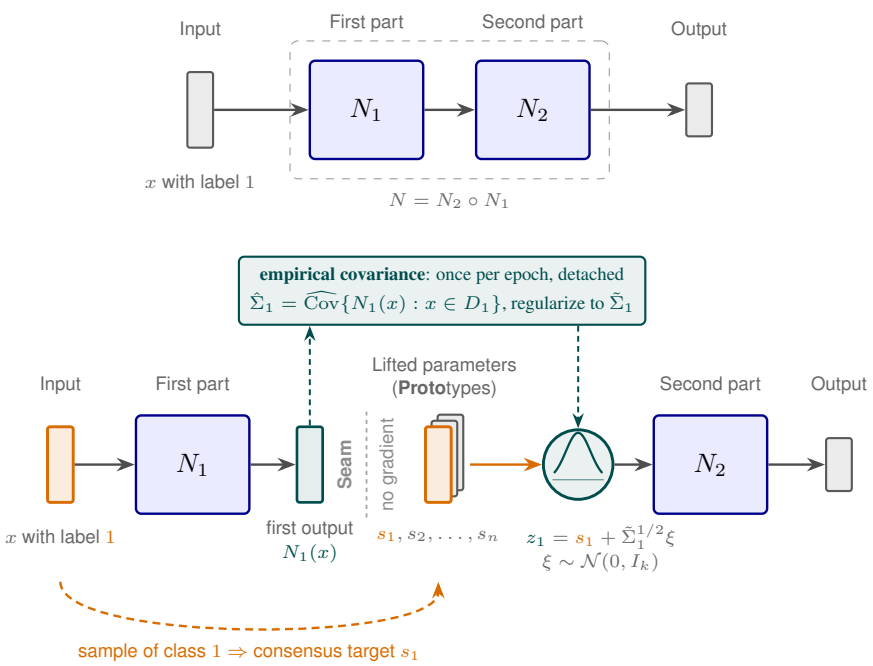


Figure 2: **The lifting procedure.** The original network (top) is split into two parts and one lifted variable is introduced per class (bottom). The part N_1 is trained against s_i by the consensus term (orange), while N_2 is trained on samples z_i drawn around s_i, so no gradient crosses the interface (seam). The class-conditional covariance (teal) is estimated once per epoch from the embeddings $N_1(x)$ of the whole class, regularized by (3), and then used as the scale of the sampling step (6). At inference lifted variables and sampling step are discarded and the original path $N_2 \circ N_1$ is restored.

3 Method

Let $D = \{(x^{(j)}, i^{(j)})\}_{j=1}^{m}$ be a labeled dataset with samples $x^{(j)} \in \mathbb{R}^d$ and labels $i^{(j)} \in [n]$, and let $D_i = \{(x, l) \in D \mid l = i\}$ denote the samples of class i. Our goal is to train a network that assigns each sample its corresponding class.

Let N be a neural network which we split into two parts N_1 and N_2 such that $N(x) = N_2(N_1(x))$. Following the reparameterization trick that underlies variational autoencoders (Kingma & Welling, 2013), we make the central assumption that the output of N_1 is normally distributed within each class in the embedding space $\mathbb{R}^k$. Writing (X, Y) for a random labeled sample,

$$N_1(X) \mid Y = i \;\sim\; \mathcal{N}(s_i, \Sigma_i), \qquad \forall\, i \in [n], \tag{1}$$

where $s_i \in \mathbb{R}^k$ and $\Sigma_i \in \mathbb{R}^{k \times k}$ denote the class-specific mean and covariance. Equation (1) is a modeling assumption and does not hold for any arbitrary trained network. The consensus penalty below is what makes it approximately true.

The training objective is twofold. On the one hand, the prototypes $s_1, \ldots, s_n$ must be positioned such that N_2 separates the confidence regions of the distributions $\mathcal{N}(s_i, \Sigma_i)$. On the other hand, the output of N_1 has to actually follow (1).

Covariance estimation Rather than learning means and covariances jointly, we treat only the means $s_1, \ldots, s_n$ as learnable parameters and re-estimate the covariances empirically once per epoch from the class-conditional embeddings,

$$\hat{\Sigma}_i = \frac{1}{|D_i|} \sum_{(x,\,\cdot\,) \in D_i} \big(N_1(x) - \hat{\mu}_i\big)\big(N_1(x) - \hat{\mu}_i\big)^\top, \qquad \hat{\mu}_i = \frac{1}{|D_i|} \sum_{(x,\,\cdot\,) \in D_i} N_1(x), \tag{2}$$

so that each estimate is formed exclusively from the embeddings of the corresponding class. Note that (2) centers on the empirical mean $\hat{\mu}_i$ rather than on the prototype s_i. Centering on s_i instead would fold the consensus residual into the covariance and inflate it, so we retain $\hat{\mu}_i$.

In practice, $\hat{\Sigma}_i$ is singular whenever $k \geq |D_i|$ and potentially ill-conditioned well before that (Section 3.1). Moreover, as the consensus penalty tightens, $\hat{\Sigma}_i \to 0$, so the samples fed to N_2 degenerate to the n points $s_1, \ldots, s_n$. In that limit N_2 is trained on a dataset of n distinct inputs and its behavior between prototypes is unconstrained, while at inference it receives genuine embeddings $N_1(x) \neq s_i$. To account for this effect and potential numerical inaccuracies, we always ensure that the estimate remains positive definite, adding a scaled identity matrix if necessary, i.e.,

$$\tilde{\Sigma}_i = \hat{\Sigma}_i + \sigma_0^2 I_k, \qquad \sigma_0 > 0, \tag{3}$$

which then admits a stable Cholesky factorization. The σ_0 can equivalently be interpreted as how much of the embedding space around each prototype N_2 is required to classify correctly.

Inter-class repulsion As an additional form of regularization, we require the class clusters to be mutually well separated. We therefore introduce the penalty

$$\mathcal{P}(s_1, \ldots, s_n) = \sum_{\substack{i,j \in [n] \\ i<j}} \exp\bigl(-\alpha \|s_i - s_j\|_2\bigr), \qquad \alpha > 0, \tag{4}$$

which decays rapidly once the means are far apart and thus penalizes only those pairs that remain close. For all benchmarks we chose the fixed value of $\alpha = 2$. The main purpose of this penalty is to separate the class clusters at the beginning of the training process. Without this penalty all clusters usually remain very close to each other and are never really separated.

The lifted objective Let $\mathcal{L}_{\text{cls}}$ denote the chosen classification loss. We define the lifted training objective as

$$\mathcal{L}(\theta_1, \theta_2, S) = \underbrace{\frac{\rho}{2m} \sum_{(x,i) \in D} \bigl\| N_1(x) - s_i \bigr\|_2^2}_{\text{consensus, trains } \theta_1 \text{ and } S} + \underbrace{\frac{1}{n} \sum_{i=1}^{n} \mathbb{E}_\xi \bigl[\mathcal{L}_{\text{cls}}\bigl(N_2(z_i),\, i\bigr) \bigr]}_{\text{classification, trains } \theta_2 \text{ and } S} + \rho\, \mathcal{P}(s_1, \ldots, s_n), \tag{5}$$

where the embedding fed to N_2 is sampled via the reparameterization

$$z_i = s_i + \tilde{\Sigma}_i^{1/2} \xi, \qquad \xi \sim \mathcal{N}(0, I_k), \tag{6}$$

with $\tilde{\Sigma}_i^{1/2}$ the Cholesky factor of (3), treated as a constant with respect to θ_1 within an epoch. The first term pulls the embedding of each sample toward the prototype of its class; the second trains N_2 on samples drawn from the assumed class distribution itself.

Three points deserve emphasis.

(i) The repulsion weight is shared. In our implementation the repulsion strength is given by $\rho\mathcal{P}(\cdots)$, tied to the penalty ρ of the least squares term. Theoretically, these two terms are independent and introducing a separate scale for $\mathcal{P}$ might be worthwhile. However, since we perform a large ablation study over ρ, we believe our formulation to be more practical.

(ii) The classification term does not depend on x. Because z_i is a function of $(s_i, \tilde{\Sigma}_i, \xi)$ only, the second term of (5) depends on the data solely through the class index. This is the precise sense in which the two subnetworks are decoupled: the θ_2-subproblem is an n-component Gaussian classification problem whose cost is independent of m, and it may be optimized with as many Monte-Carlo draws per class as desired, independently of the minibatch composition. We use the uniform class weighting $\frac{1}{n}$ in (5); weighting by empirical frequency $|D_i|/m$ is the alternative appropriate for imbalanced data.

(iii) ρ has two readings. On the one side, it acts as a conventional penalty weight. On the other side, $\frac{\rho}{2}\|N_1(x) - s_i\|_2^2$ is the negative log-density of an isotropic Gaussian with variance ρ^{-1} (up to an additive constant). Hence, its reciprocal quantifies the tolerated spread around each prototype.

Algorithm 1 Lifted training of $N = N_2 \circ N_1$

Require: dataset D, epochs T, schedule $\rho(\cdot)$, floor σ_0

initialize θ_1, θ_2; initialize S (Section 3.1); set $\tilde{\Sigma}_i \leftarrow \sigma_0^2 I_k$
for $t = 1, \dots, T$ **do**
  for each minibatch $B \subseteq D$ **do**
    $\mathcal{L}_{\text{con}} \leftarrow \frac{\rho(t)}{2|B|} \sum_{(x,i)\in B} \|N_1(x) - s_i\|_2^2$
    draw $\xi_i \sim \mathcal{N}(0, I_k)$ and set $z_i \leftarrow s_i + \tilde{\Sigma}_i^{1/2} \xi_i$ for $i \in [n]$ ▷ $\tilde{\Sigma}_i$ detached
    $\mathcal{L}_{\text{cls}} \leftarrow \frac{1}{n} \sum_{i=1}^{n} \mathcal{L}_{\text{cls}}(N_2(z_i), i)$
    update θ_1, θ_2, S on $\mathcal{L}_{\text{con}} + \mathcal{L}_{\text{cls}} + \rho\, \mathcal{P}(S)$
  end for
  estimate $\hat{\Sigma}_i$ over D and regularize $\tilde{\Sigma}_i$ by (3) if necessary
end for
return $N_2 \circ N_1$ ▷ S, $\tilde{\Sigma}$ discarded

This lifting procedure is applied during training only. For validation and at inference time the lifted variables are discarded and the two subnetworks are reconnected, recovering the original architecture $N = N_2 \circ N_1$.

The per-epoch empirical covariance computation in Algorithm 1 costs one additional forward pass over the training set, i.e. roughly a $1/3$ overhead per epoch relative to a standard forward–backward pass, plus $O(nk^2)$ memory for the factors.

There are alternative and considerably cheaper approaches to compute the covariance. For example, we could compute and add up the dyadic products of $N_1(x) - s_i$ during the training process to obtain running estimates of the class covariances for the next epoch. In our experiments, this produced almost identical results but occasionally led to numerical difficulties whenever the output distribution of N_1 changed rapidly during one epoch. An even cheaper approach, with practically no additional cost, would be to estimate the covariance using a scaled identity matrix. For the right scaling of the identity matrices this once again produced similar results to the empirical covariance computation. Based on the interpretation of ρ, choosing a scaling of ρ^{-1} would be the obvious choice. A reasonable middle ground would be to start with the separate covariance computation and then switch to the faster variant once the learning rate has decayed enough. For our benchmarks, the main quantity of interest is the final accuracy. Therefore, and to avoid further hyperparameter dependencies through scaling, we use an additional pass to compute the empirical covariance after every epoch.

3.1 Design Choices

The formulation above leaves several choices open which we discuss in the following and investigated with numerical sweeps reported in Section 4.

Lifting location. The lifting point should not immediately follow a ReLU since the output of N_1 would then be confined to the non-negative orthant. In this case, a cluster whose prototype lies on the boundary of that region cannot be symmetrically surrounded by the Gaussian of (1). Hence, the model is misspecified exactly where the consensus penalty is tightest. We therefore always lift before such an activation or before the final classification layer. Given our assumption that every class is normally distributed around one mean, the network N_2 need not be very complex to separate these clusters. In fact, we only use a single fully connected layer for N_2. Using more complex modeling assumptions, such as a Gaussian mixture for every class, might necessitate larger N_2. We address the detailed construction in Section 4.

Lifting dimension. Our approach assumes that the output of N_1 is normally distributed within each class and estimates the corresponding covariance from data. A non-singular sample covariance in dimension k requires at least $k + 1$ samples per class, so $\min_i |D_i| > k$ is a hard upper bound on the embedding dimension. Regularization might rectify this issue, but accuracy requires considerably more. For sub-Gaussian class-conditional embeddings, the sample covariance $\hat{\Sigma}_{m'}$ formed from m'

samples satisfies (Vershynin, 2018, Thm. 4.7.1 and Rem. 4.7.2)

$$\mathbb{E}\big\|\hat{\Sigma}_{m'} - \Sigma\big\| \;\le\; C\Big(\sqrt{\tfrac{k}{m'}} + \tfrac{k}{m'}\Big)\,\|\Sigma\|, \tag{7}$$

with an absolute constant C depending only on the sub-Gaussian norm, so a relative error ε requires $m' \gtrsim \varepsilon^{-2}k$. This bound is very restrictive, applying it to CIFAR-100 with $m' = 500$ training images per class results for $\varepsilon = 0.1$ in $k \lesssim 5$. In our experiments we observe better performance for way higher dimensions than this. We conjecture that this is a combined effect of the additional repulsion term and the covariances tending towards zero, rendering the relative error bound (7) unimportant.

Initialization of class means. The lifted variables admit custom initialization. The natural choices are the empirical class means $s_i = \mathbb{E}_{X|Y=i}[N_1(X)]$ computed from the initial network. Following Lampel & Sager (2025; 2026), we refer to this approach as Forward Sweep Initialization (FSInit).

Using FSInit is motivated by the observation that semantically similar classes are also close to each other in the embedding space, compare Figure 1. Therefore, we expect the FSInit outcome to be an approximation of this final positioning, thus avoiding having to reorder the class clusters. However, the benefits clearly depend on the initialization of the network weights θ_1.

Alternative initializations are based on random or user-guided values. Also unit vectors can be used if the embedding dimension k is equal to the number n of classes. We only observed a minor impact of initialization on the final accuracy, with FSInit performing best.

Penalty annealing. Rather than holding ρ fixed, we increase it depending on epoch t as

$$\rho(t) = \rho_{\min} + (\rho_{\max} - \rho_{\min}) \sin\left(\frac{\pi}{2} \cdot \frac{t-1}{T-1}\right), \qquad t = 1, \dots, T. \tag{8}$$

This increases the penalty fastest towards the beginning of training, then slowly approaches $\rho_{\max}$ for larger t. Using smaller values of $\rho(t)$ in the early training phase is motivated by allowing the clusters to separate rather than collapse immediately. For later training epochs t the enforcement of consensus becomes a higher priority.

4 Experiments

4.1 Protocol

For a given dataset $\mathcal{D}$ and architecture family M, we train and compare three variants under a common recipe, so that any accuracy gap can be attributed to the lifting mechanism rather than to confounding architectural or optimization differences.

Baseline. An independent, non-split reference network is trained end-to-end in the standard way, providing an external calibration point against a conventional model of the same family.

ProtoSeam. The architecture is partitioned into a feature extractor N_1 and a classifier head N_2, coupled through the per-class prototype matrix $S \in \mathbb{R}^{k\times n}$. Parameters θ_1, θ_2, S are optimized under (5) with penalty weight ρ. We lift just before the classification layer, meaning that the last fully connected layer of the original architecture (the baseline) is replaced by one mapping to $\mathbb{R}^k$ to form N_1. Consequently, N_2 consists of only one fully connected layer mapping from $\mathbb{R}^k$ to $\mathbb{R}^n$.

Unlifted. The ProtoSeam architecture is trained end-to-end by ordinary backpropagation, without consensus penalty, the repulsion term and the auxiliary variable S. This separates the effect of the lifting mechanism from those of the minor architecture change between Baseline and ProtoSeam (remember that the final fully connected layer in Baseline mapping to $\mathbb{R}^n$ is replaced by two fully connected layers; the first maps to $\mathbb{R}^k$ and the second from $\mathbb{R}^k$ to $\mathbb{R}^n$, with no activation function in-between).

Sweep. We perform an exemplary sweep across both the embedding dimension k and the penalty $\rho_{\max}$ for the ViT-S on CIFAR-100 in Figure 3. Tuning both these values for every model type and dataset would increase the lifted accuracy further, but also entail a further expensive hyperparameter search. Therefore, we chose an embedding dimension of $k = 32$ and a penalty of $\rho_{\max} = 16$ for all further benchmarks in Section 4.3.

Common training recipe. All variants use the same splits, epoch budget T, and optimizer family (SGD with Nesterov momentum, weight decay, short linear warm-up, cosine-annealed decay).

Remark 4.1 (Hyperparameter tuning). To ensure that the better accuracy is not merely an artifact of different hyperparameters, we ran a large grid search to determine the best learning rate and weight decay for both the lifted and the baseline variant, for every dataset and model. For the unlifted variant we used the same hyperparameters as for the baseline, since both train end-to-end. The determined optimal parameters are summarized in Table 2.

4.2 Datasets and architectures

We consider a ResNet (He et al., 2016) and a custom small Vision Transformer (ViT-S) (Dosovitskiy et al., 2021) backbone on CIFAR-10, CIFAR-100 (Krizhevsky et al., 2009), and TinyImageNet (Le & Yang, 2015). On the one hand these models and datasets are complex enough so that there are still meaningful gains in accuracy to be made. On the other hand, they are small enough to make an exhaustive comparison across a wide range of hyperparameters.

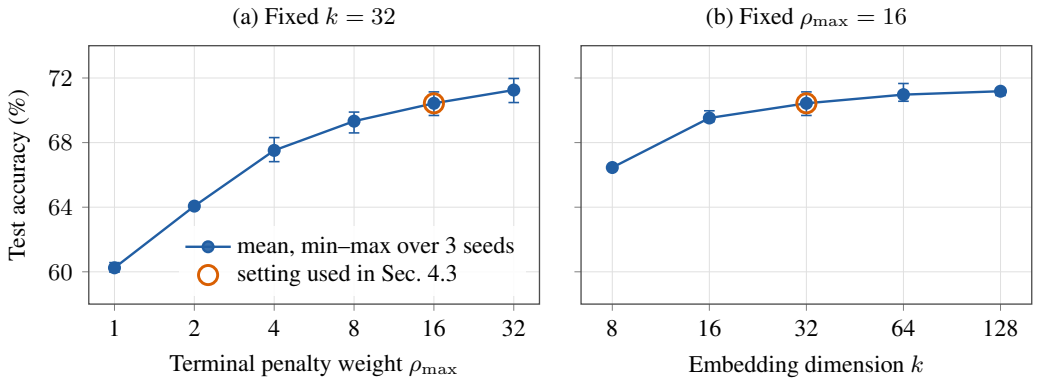


Figure 3: ViT-S on CIFAR-100, using the empirically computed covariance. On the left hand side we compare the test accuracy for different values of $\rho_{\max}$ and a fixed embedding dimension of $k = 32$. The right plot shows the test accuracy for different embedding dimensions, this time for a fixed penalty of $\rho_{\max} = 16$. For both ablations we observe a monotonic climb which first increases steeply and then stabilizes. There is no meaningful difference between the test and validation accuracies. We use $\rho_{\max} = 16$ and $k = 32$ in the following, which we found to work best across all datasets.

4.3 Results

We observe a consistent trend across both architectures. Our method always beats both the baseline and the unlifted variant. Therefore, we can rule out the possibility that the advantage comes from the altered architecture. For TinyImageNet, the accuracy improves by more than five percentage points compared to the baseline for ViT-S.

We show an exemplary training run in Figure 4, which presents a typical pattern for all training runs. At first, our method trails behind, while the output of N_1 is still noisy and the class prototypes are being positioned. After a while, our lifted formulation then overtakes both other variants and maintains its lead.

All technical details are described in Section A.

5 Conclusion

We introduced a new technique for training neural networks for classification problems. By splitting a network $N = N_2 \circ N_1$ at a single interface and inserting one Gaussian prototype per class, we formulated an objective in which N_1 is trained toward the prototype of its class and N_2 on samples drawn around it, with no gradient crossing the seam. At inference the prototypes are discarded, and the unmodified network is deployed at no additional cost. Across CIFAR-10, CIFAR-100 and

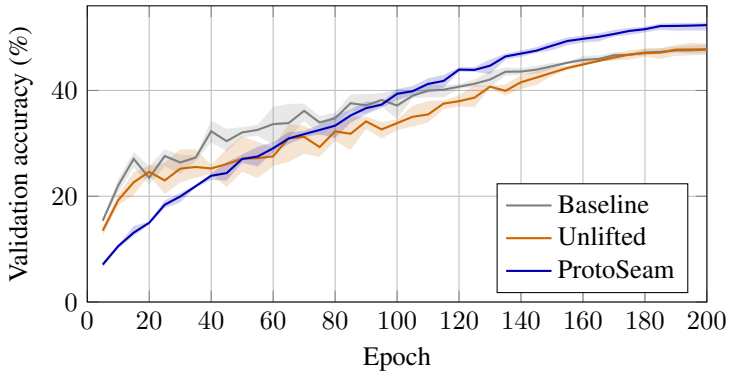


Figure 4: ViT-S on TinyImageNet: validation accuracy over the epochs for baseline, unlifted, and lifted variant. We show the mean over 3 seeds $(42, 43, 44)$, sampled every 5 epochs. The shaded bands show the full seed range (min to max) at each sampled epoch.

Dataset	Model	Variant	seed 42	seed 43	seed 44	Mean	Spread
CIFAR-10	ResNet-8	Baseline	94.88	95.09	94.77	94.91	0.32
		Unlifted	95.09	95.38	95.00	95.16	0.38
		ProtoSeam	95.27	95.54	95.57	**95.46**	0.30
	ViT-S	Baseline	89.45	89.50	89.34	89.43	0.16
		Unlifted	90.55	89.94	89.66	90.05	0.89
		ProtoSeam	90.64	90.34	90.17	**90.38**	0.47
CIFAR-100	ResNet-8	Baseline	76.76	76.96	76.55	76.76	0.41
		Unlifted	74.12	74.04	74.55	74.24	0.51
		ProtoSeam	78.18	78.21	78.63	**78.34**	0.45
	ViT-S	Baseline	66.11	65.63	65.99	65.91	0.48
		Unlifted	65.49	65.38	64.04	64.97	1.45
		ProtoSeam	70.48	71.00	70.12	**70.53**	0.88
TinyImageNet	ResNet-8	Baseline	63.17	63.13	63.25	63.18	0.12
		Unlifted	57.70	58.51	58.75	58.32	1.05
		ProtoSeam	65.64	65.48	65.69	**65.60**	0.21
	ViT-S	Baseline	46.42	47.45	47.01	46.96	1.03
		Unlifted	48.54	47.48	47.76	47.93	1.06
		ProtoSeam	51.71	51.88	52.78	**52.12**	1.07

Table 1: Multi-seed confirmation (seeds 42/43/44): test accuracy (%) for baseline, unlifted, and lifted across both architectures and all three datasets. **Bold** = lifted's mean, which beats both comparators at every individual seed on every combination.

TinyImageNet with ResNet-8 and ViT-S backbones, lifted training improved test accuracy over both a baseline model and the matched end-to-end unlifted control on every dataset–architecture pair and at every seed, by up to 5.2 and 7.3 percentage points, respectively (Table 1). The unlifted control rules out the modified architecture as the source. On the theoretical side, we addressed the question of why lifting improves the final accuracy in Section B. Moreover, we constructed a one-dimensional model which exhibits the resulting effect in closed form (Section B.3). Together, these results identify a mechanism by which lifting selects a different, locally more robust local optimum, rather than optimizing the same objective faster.

Reproducibility statement

The software used to produce all results in this work will be made publicly available in a Git repository upon acceptance of this paper.

AI use statement

In this work, we used generative AI tools to formulate and prove the mathematical claims in the appendix, for feedback on research methodology or experiments, and interpretation of results. We have not used generative AI tools for the generation of synthetic datasets, developing theoretical models, assistance with translation, reformatting datasets, or qualitative and thematic data analysis. Additionally, we used generative AI tools for refining the language, grammar, and readability of this paper, for formatting and the creation of figures, to discover relevant literature, and to assist in the writing of the software code. We have reviewed all AI-assisted work. We checked LLM-generated proofs and theorems for correctness. The LLM-generated code was verified and tested for correctness. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

Ethics statement

This is a methodological contribution to supervised image classification. It involves no human subjects, no personal or otherwise sensitive data, and no collection of new data. All experiments use established public benchmarks (CIFAR-10, CIFAR-100 and TinyImageNet) in their standard form and within their intended research use. We are therefore not aware of any application risk it introduces beyond those already associated with image classifiers in general.

References

Armin Askari, Geoffrey Negiar, Rajiv Sambharya, and Laurent El Ghaoui. Lifted neural networks. *arXiv preprint arXiv:1805.01532*, 2018.

Eugene Belilovsky, Michael Eickenberg, and Edouard Oyallon. Greedy layerwise learning can scale to ImageNet. In *International Conference on Machine Learning (ICML)*, pp. 583–593, 2019.

Rajendra Bhatia, Tanvi Jain, and Yongdo Lim. On the Bures–Wasserstein distance between positive definite matrices. *Expositiones Mathematicae*, 37(2):165–191, 2019. arXiv:1712.01504.

Christopher M. Bishop. Training with noise is equivalent to Tikhonov regularization. *Neural Computation*, 7(1):108–116, 1995. doi: 10.1162/neco.1995.7.1.108.

Hans Georg Bock and K. J. Plitt. A multiple shooting algorithm for direct solution of optimal control problems. In *IFAC Proceedings Volumes*, volume 17, pp. 1603–1608. Elsevier, 1984.

Miguel Á. Carreira-Perpiñán and Weiran Wang. Distributed optimization of deeply nested systems. In *International Conference on Artificial Intelligence and Statistics*, pp. 10–19. PMLR, 2014.

Olivier Chapelle, Jason Weston, Léon Bottou, and Vladimir Vapnik. Vicinal risk minimization. In *Advances in Neural Information Processing Systems 13*, pp. 416–422, 2000.

Ricky T. Q. Chen, Yulia Rubanova, Jesse Bettencourt, and David K. Duvenaud. Neural ordinary differential equations. In *Advances in Neural Information Processing Systems*, pp. 6572–6583, 2018.

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.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations (ICLR)*, 2021.

Clark R. Givens and Rae Michael Shortt. A class of Wasserstein metrics for probability distributions. *Michigan Mathematical Journal*, 31(2):231–240, 1984. doi: 10.1307/mmj/1029003026.

Fangda Gu, Armin Askari, and Laurent El Ghaoui. Fenchel lifted networks: A lagrange relaxation of neural network training. In *International Conference on Artificial Intelligence and Statistics*, pp. 3362–3371. PMLR, 2020.

Stefanie Gunther, Lars Ruthotto, Jacob B Schroder, Eric C Cyr, and Nicolas R Gauger. Layer-parallel training of deep residual neural networks. *SIAM Journal on Mathematics of Data Science*, 2(1): 1–23, 2020.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Pavel Izmailov, Polina Kirichenko, Marc Finzi, and Andrew Gordon Wilson. Semi-supervised learning with normalizing flows. In *International conference on machine learning*, pp. 4615–4630. PMLR, 2020.

Max Jaderberg, Wojciech Marian Czarnecki, Simon Osindero, Oriol Vinyals, Alex Graves, David Silver, and Koray Kavukcuoglu. Decoupled neural interfaces using synthetic gradients. In *International Conference on Machine Learning (ICML)*, pp. 1627–1635, 2017.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Diederik P. Kingma, Danilo J. Rezende, Shakir Mohamed, and Max Welling. Semi-supervised learning with deep generative models. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 3581–3589, 2014.

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

Robert Lampel and Sebastian Sager. On liftings that improve convergence properties of newton's method for boundary value optimization problems. 2025.

Robert Lampel and Sebastian Sager. An adaptive multiple shooting strategy for optimal control. *Optimal Control Applications and Methods*, 2026.

Ya Le and Xuan Yang. Tiny ImageNet visual recognition challenge. Technical report, Stanford University, CS231N, 2015.

Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. In *Advances in Neural Information Processing Systems*, pp. 7167–7177, 2018.

Weiyang Liu, Yandong Wen, Zhiding Yu, and Meng Yang. Large-margin softmax loss for convolutional neural networks. In *International Conference on Machine Learning*, pp. 507–516. PMLR, 2016.

Weiyang Liu, Yandong Wen, Zhiding Yu, Ming Li, Bhiksha Raj, and Le Song. SphereFace: Deep hypersphere embedding for face recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 212–220, 2017.

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.

Eric Nalisnick, Lars Hertel, and Padhraic Smyth. Approximate inference for deep latent gaussian mixtures. In *NIPS Workshop on Bayesian Deep Learning*, volume 2, pp. 131, 2016.

Jorge Nocedal and Stephen J. Wright. *Numerical Optimization*. Springer, 2nd edition, 2006.

Arild Nøkland and Lars Hiller Eidnes. Training neural networks with local error signals. In *International Conference on Machine Learning*, pp. 4839–4850. PMLR, 2019.

Vardan Papyan, X. Y. Han, and David L. Donoho. Prevalence of neural collapse during the terminal phase of deep learning training. *Proceedings of the National Academy of Sciences*, 117(40): 24652–24663, 2020.

Danilo Jimenez Rezende and Shakir Mohamed. Variational inference with normalizing flows. In *International Conference on Machine Learning*, pp. 1530–1538. PMLR, 2015.

Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems*, pp. 4077–4087, 2017.

Hyun Oh Song, Yu Xiang, Stefanie Jegelka, and Silvio Savarese. Deep metric learning via lifted structured feature embedding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4004–4012, 2016.

Gavin Taylor, Ryan Burmeister, Zheng Xu, Bharat Singh, Ankit Patel, and Tom Goldstein. Training neural networks without gradients: A scalable ADMM approach. In *International Conference on Machine Learning*, pp. 2722–2731. PMLR, 2016.

Roman Vershynin. *High-Dimensional Probability: An Introduction with Applications in Data Science*. Cambridge University Press, 2018.

Cédric Villani. *Optimal Transport: Old and New*, volume 338 of *Grundlehren der mathematischen Wissenschaften*. Springer, Berlin, Heidelberg, 2009. doi: 10.1007/978-3-540-71050-9.

Weitao Wan, Yuanyi Zhong, Tianpeng Li, and Jiansheng Chen. Rethinking feature distribution for loss functions in image classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 9117–9126, 2018.

Yandong Wen, Kaipeng Zhang, Zhifeng Li, and Yu Qiao. A discriminative feature learning approach for deep face recognition. In *European Conference on Computer Vision*, pp. 499–515. Springer, 2016.

Yibo Yang, Shixiang Chen, Xiangtai Li, Liang Xie, Zhouchen Lin, and Dacheng Tao. Inducing neural collapse in imbalanced learning: Do we really need a learnable classifier at the end of deep neural network? In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 37991–38002, 2022.

Zhihui Zhu, Tianyu Ding, Jinxin Zhou, Xiao Li, Chong You, Jeremias Sulam, and Qing Qu. A geometric analysis of neural collapse with unconstrained features. In *Advances in Neural Information Processing Systems*, pp. 29820–29834, 2021.

A TECHNICAL DETAILS

For both the ResNet model and the ViT and every dataset we performed a hyperparameter grid search across the learning rates $\{0.01, 0.02, 0.05, 0.1\}$ and weight decays $\{10^{-4}, 2 \times 10^{-4}, 5 \times 10^{-4}, 10^{-3}, 2 \times 10^{-3}\}$. We always picked the combination with the highest validation accuracy after a full run of 200 epochs. The hyperparameters for the *unlifted* variant were chosen as the ones from the baseline. All determined hyperparameter combinations are summarized in Table 2.

Dataset	Model	Variant	η	λ
CIFAR10	ResNet8	Baseline	0.02	10^{-3}
		ProtoSeam	0.05	$5{\times}10^{-4}$
	ViT	Baseline	0.02	10^{-4}
		ProtoSeam	0.01	$5{\times}10^{-4}$
CIFAR100	ResNet8	Baseline	0.05	$2{\times}10^{-3}$
		ProtoSeam	0.02	$2{\times}10^{-3}$
	ViT	Baseline	0.01	10^{-4}
		ProtoSeam	0.01	$5{\times}10^{-4}$
TinyImageNet	ResNet8	Baseline	0.05	10^{-3}
		ProtoSeam	0.02	$2{\times}10^{-3}$
	ViT	Baseline	0.1	$2{\times}10^{-4}$
		ProtoSeam	0.01	$5{\times}10^{-4}$

Table 2: Final, confirmed best hyperparameters for every (dataset, model, variant) combination determined by the grid search.

All three datasets (CIFAR-10, CIFAR-100, TinyImageNet) use the same training-time augmentation pipeline: a random crop with 4-pixel zero-padding back to the native resolution (32×32 for CIFAR-10/100, 64×64 for TinyImageNet), a random horizontal flip, RandAugment (default policy), normalization to each dataset's own per-channel mean/std, and finally random erasing (probability 0.5, area fraction 0.02–0.33). The held-out validation and test sets receive no stochastic augmentation, only normalization. Hence, accuracy on those splits is never inflated by test-time randomness.

B WHY LIFTING IMPROVES ACCURACY

The gains of Table 1 call for an explanation. Two mechanisms could be responsible. The first is optimization: the lifted objective may simply be easier to minimize. We do not rule this mechanism out, but it is not the one analysed here. The second mechanism is statistical: the lifted objective may select a *different* minimizer. This section develops the statistical account in three steps. First, the sampled classification term is not an arbitrary surrogate: it equals the risk of the reconnected network up to an explicit interface-mismatch term, and every component of that term is bounded by a quantity that the method controls (Theorems B.3 and B.4). Second, relative to the loss of the head at the prototypes, sampling adds a non-negative smoothing penalty, which to second order is a curvature penalty on the head (Theorem B.5): in the small-perturbation regime, among heads with equal loss at the prototypes, the sampled objective prefers the one whose loss curves least around them. Third, Section B.3 exhibits a small instance in which this selection effect can be computed in closed form.

B.1 THE SAMPLED LOSS IS THE DEPLOYED RISK UP TO INTERFACE MISMATCH

At inference the prototypes are discarded, and the deployed classifier is the reconnected network $N_2 \circ N_1$. Writing

$$\ell_i(z) := \mathcal{L}_{\mathrm{cls}}\big(N_2(z),\, i\big), \qquad z \in \mathbb{R}^k, \tag{9}$$

for the loss of the head on an embedding of class i, the population risk of the deployed classifier under balanced classes is

$$R(\theta_1, \theta_2) := \mathbb{E}\big[\ell_Y\big(N_1(X)\big)\big] = \frac{1}{n}\sum_{i=1}^{n} \mathbb{E}_{z\sim P_i}\, \ell_i(z), \tag{10}$$

where P_i denotes the law of $N_1(X)$ conditioned on $Y = i$ (an italic P, not to be confused with the repulsion penalty $\mathcal{P}$), with mean μ_i and covariance Σ_i. The classification term of (5) is the same average taken over the sampling distributions instead,

$$L_{\mathrm{smp}} := \frac{1}{n}\sum_{i=1}^{n} \mathbb{E}_{z\sim Q_i}\, \ell_i(z), \qquad Q_i := \mathcal{N}\big(s_i, \tilde{\Sigma}_i\big). \tag{11}$$

Training N_2 on (11) is vicinal risk minimization Chapelle et al. (2000) at the lifting interface: the empirical embeddings are replaced by Gaussian vicinities around the prototypes. The question is how far L_{smp} can be from the risk R that actually matters at deployment.

We measure the discrepancy between the embedding law and the sampling law in the 1-Wasserstein distance W_1 Villani (2009) and, between covariance matrices, in the Bures distance

$$B_W(A, B) := \Big(\operatorname{tr} A + \operatorname{tr} B - 2\operatorname{tr}\big[\big(B^{1/2} A B^{1/2}\big)^{1/2}\big]\Big)^{1/2}, \tag{12}$$

which is a metric on positive semi-definite matrices (Bhatia et al., 2019) and is exactly the covariance part of the 2-Wasserstein distance between Gaussians (Givens & Shortt, 1984).

Assumption B.1. *Each head loss $\ell_i : \mathbb{R}^k \to \mathbb{R}$ is G-Lipschitz: $|\ell_i(z) - \ell_i(z')| \le G\|z - z'\|_2$ for all $z, z' \in \mathbb{R}^k$ and all $i \in [n]$.*

Remark B.2 (The assumption is exact for our architecture). All experiments in Section 4 use a single linear head, $N_2(z) = W_2 z + b$, with the cross-entropy loss. Then $\nabla_z \ell_i(z) = W_2^\top\big(p(z) - e_i\big)$ with $p(z) = \mathrm{softmax}(W_2 z + b)$, and since $\|p - e_i\|_2 \le \sqrt{2}$ for any probability vector p, Theorem B.1 holds globally with $G = \sqrt{2}\,\|W_2\|_2$. The weight decay applied to N_2 therefore directly controls the constant in the bound below.

Theorem B.3 (Interface risk transfer)**.** *Under Assumption B.1 and balanced classes,*

$$\big|R - L_{\mathrm{smp}}\big| \;\le\; \frac{G}{n}\sum_{i=1}^{n} W_1\big(P_i, Q_i\big), \tag{13}$$

and for every class i, if P_i has finite second moment,

$$W_1\big(P_i, Q_i\big) \;\le\; \underbrace{W_1\big(P_i, \mathcal{N}(\mu_i, \Sigma_i)\big)}_{\text{non-Gaussianity } \Delta_i} + \Big(\underbrace{\|\mu_i - s_i\|_2^2}_{\text{mean mismatch}} + \underbrace{B_W^2\big(\Sigma_i, \tilde{\Sigma}_i\big)}_{\text{covariance mismatch}}\Big)^{1/2}. \tag{14}$$

Proof. For (13), fix i and let γ be any coupling of (P_i, Q_i), i.e. a joint law of (z, z') with marginals P_i and Q_i. Then

$$\big|\mathbb{E}_{P_i}\ell_i - \mathbb{E}_{Q_i}\ell_i\big| = \big|\mathbb{E}_{(z,z')\sim\gamma}\big[\ell_i(z) - \ell_i(z')\big]\big| \le G\,\mathbb{E}_\gamma\|z - z'\|_2,$$

by Theorem B.1. Taking the infimum over couplings gives $|\mathbb{E}_{P_i}\ell_i - \mathbb{E}_{Q_i}\ell_i| \le G\,W_1(P_i, Q_i)$ (Villani, 2009). Averaging over i and using the triangle inequality for the average yields (13).

For (14), insert the moment-matched Gaussian $\Gamma_i := \mathcal{N}(\mu_i, \Sigma_i)$ and use that W_1 is a metric: $W_1(P_i, Q_i) \le W_1(P_i, \Gamma_i) + W_1(\Gamma_i, Q_i)$. By Jensen's inequality $W_1 \le W_2$, and for Gaussian measures the W_2 distance is available in closed form (Givens & Shortt, 1984):

$$W_2^2\big(\mathcal{N}(\mu_i, \Sigma_i), \mathcal{N}(s_i, \tilde{\Sigma}_i)\big) = \|\mu_i - s_i\|_2^2 + B_W^2\big(\Sigma_i, \tilde{\Sigma}_i\big),$$

with B_W as in (12). Combining the three displays proves the claim. □

The argument is elementary. Its point is not technical difficulty but accounting: each term of (14) is bounded by a quantity that the lifted objective controls, as the next corollary makes precise.

Corollary B.4 (The objective controls its own transfer gap)**.** *Let* $L_{\mathrm{con}} := \frac{\rho}{2}\,\mathbb{E}\|N_1(X) - s_Y\|_2^2$ *denote the population consensus term of* (5)*, let* $\tilde{\Sigma}_i = \hat{\Sigma}_i + \sigma_0^2 I_k$ *as in* (3)*, and let* $\bar{\Delta} := \frac{1}{n}\sum_i \Delta_i$ *be the average non-Gaussianity. Then, under the assumptions of Theorem B.3,*

$$R \;\le\; L_{\mathrm{smp}} + G\Big[\;\bar{\Delta} \;+\; \sqrt{\tfrac{2}{\rho}L_{\mathrm{con}}} \;+\; \sqrt{k}\,\sigma_0 \;+\; \tfrac{1}{2\sigma_0}\cdot\tfrac{1}{n}\textstyle\sum_i \big\|\hat{\Sigma}_i - \Sigma_i\big\|_F\;\Big]. \tag{15}$$

Moreover, the non-Gaussianity is itself controlled by the consensus term, $\bar{\Delta} \le 2\sqrt{\frac{2}{\rho}L_{\mathrm{con}}}$*, and*

$$R \;\le\; L_{\mathrm{smp}} + G\Big[\;\sqrt{5}\,\sqrt{\tfrac{2}{\rho}L_{\mathrm{con}}} \;+\; \sqrt{k}\,\sigma_0 \;+\; \tfrac{1}{2\sigma_0}\cdot\tfrac{1}{n}\textstyle\sum_i \big\|\hat{\Sigma}_i - \Sigma_i\big\|_F\;\Big]. \tag{16}$$

Proof. Apply (13), (14) and $\sqrt{a^2+b^2} \le a+b$ (valid for $a, b \ge 0$) to each class:

$$R - L_{\mathrm{smp}} \;\le\; \frac{G}{n}\sum_{i=1}^{n}\Big[\Delta_i + \|\mu_i - s_i\|_2 + B_W\big(\Sigma_i, \tilde{\Sigma}_i\big)\Big].$$

Mean term. By Jensen's inequality applied twice,

$$\|\mu_i - s_i\|_2 = \big\|\mathbb{E}_{P_i}[z] - s_i\big\|_2 \le \big(\mathbb{E}_{P_i}\|z - s_i\|_2^2\big)^{1/2},$$
$$\frac{1}{n}\sum_i \big(\mathbb{E}_{P_i}\|z - s_i\|_2^2\big)^{1/2} \le \Big(\frac{1}{n}\sum_i \mathbb{E}_{P_i}\|z - s_i\|_2^2\Big)^{1/2},$$

and under balanced classes the bracketed average equals $\mathbb{E}\|N_1(X) - s_Y\|_2^2 = \frac{2}{\rho}L_{\mathrm{con}}$.

Non-Gaussianity term. Inserting the point mass δ_{μ_i}, and using that the only coupling with a point mass is the product coupling,

$$\Delta_i \le W_1(P_i, \delta_{\mu_i}) + W_1(\delta_{\mu_i}, \Gamma_i) = \mathbb{E}_{P_i}\|z - \mu_i\|_2 + \mathbb{E}_{\Gamma_i}\|z - \mu_i\|_2 \le 2\,(\operatorname{tr}\Sigma_i)^{1/2},$$

by Jensen's inequality. Since $\mathbb{E}_{P_i}\|z - s_i\|_2^2 = \|\mu_i - s_i\|_2^2 + \operatorname{tr}\Sigma_i$, the Cauchy–Schwarz inequality in $\mathbb{R}^2$ gives

$$\|\mu_i - s_i\|_2 + \Delta_i \le \|\mu_i - s_i\|_2 + 2(\operatorname{tr}\Sigma_i)^{1/2} \le \sqrt{5}\,\big(\mathbb{E}_{P_i}\|z - s_i\|_2^2\big)^{1/2},$$

and also $\Delta_i \le 2(\mathbb{E}_{P_i}\|z - s_i\|_2^2)^{1/2}$. Averaging over i exactly as for the mean term yields $\bar{\Delta} \le 2\sqrt{2L_{\mathrm{con}}/\rho}$ and the $\sqrt{5}$ term of (16).

Covariance term. Since B_W is a metric on positive semi-definite matrices (Bhatia et al., 2019),

$$B_W\big(\Sigma_i, \tilde{\Sigma}_i\big) \le B_W\big(\Sigma_i, \Sigma_i + \sigma_0^2 I_k\big) + B_W\big(\Sigma_i + \sigma_0^2 I_k, \hat{\Sigma}_i + \sigma_0^2 I_k\big).$$

For the first summand, the variational characterization $B_W(A,B) = \min_{U\ \mathrm{unitary}} \|A^{1/2} - B^{1/2}U\|_F$ (Bhatia et al., 2019) with $U = I$ gives $B_W(A,B) \le \|A^{1/2} - B^{1/2}\|_F$; because Σ_i and $\Sigma_i + \sigma_0^2 I_k$ commute, this Frobenius norm equals $\big(\sum_{j=1}^{k}(\sqrt{\lambda_j + \sigma_0^2} - \sqrt{\lambda_j})^2\big)^{1/2} \le \sqrt{k}\,\sigma_0$, where $\lambda_j \ge 0$ are the eigenvalues of Σ_i and we used $\sqrt{\lambda + \sigma_0^2} - \sqrt{\lambda} \le \sigma_0$.

For the second summand, let $A := \Sigma_i + \sigma_0^2 I_k$ and $B := \hat{\Sigma}_i + \sigma_0^2 I_k$; both dominate $\sigma_0^2 I_k$, since $\hat{\Sigma}_i$ is a sample covariance and hence positive semi-definite. Diagonalize $A = U\operatorname{diag}(\alpha)U^\top$ and $B = V\operatorname{diag}(\beta)V^\top$ with U, V orthogonal and $\alpha_j, \beta_l \ge \sigma_0^2$, and set $C := U^\top V$. Then $U^\top(A-B)V$ and $U^\top(A^{1/2} - B^{1/2})V$ have entries $(\alpha_j - \beta_l)C_{jl}$ and $(\sqrt{\alpha_j} - \sqrt{\beta_l})C_{jl}$, respectively. Since $\sqrt{\alpha} - \sqrt{\beta} = (\alpha - \beta)/(\sqrt{\alpha} + \sqrt{\beta})$ with $\sqrt{\alpha} + \sqrt{\beta} \ge 2\sigma_0$, and the Frobenius norm is invariant under orthogonal transformations,

$$\|A^{1/2} - B^{1/2}\|_F^2 = \sum_{j,l}\big(\sqrt{\alpha_j} - \sqrt{\beta_l}\big)^2 C_{jl}^2 \le \frac{1}{4\sigma_0^2}\sum_{j,l}(\alpha_j - \beta_l)^2 C_{jl}^2 = \frac{1}{4\sigma_0^2}\|A - B\|_F^2.$$

With $A - B = \Sigma_i - \hat{\Sigma}_i$ and $B_W \le \|A^{1/2} - B^{1/2}\|_F$ as above, the second summand is at most $\frac{1}{2\sigma_0}\|\hat{\Sigma}_i - \Sigma_i\|_F$.

Collecting the mean, non-Gaussianity and covariance terms yields (15); replacing the first two by their joint $\sqrt{5}$ bound yields (16). □

Every term on the right of (15) is either part of the lifted objective or explicitly managed by the algorithm:

- the *sampled loss* L_{smp} is the classification term of (5), minimized directly;
- the *mean mismatch* $\sqrt{2L_{\mathrm{con}}/\rho}$ is the root-mean-square distance of the embeddings from their prototypes. It does not depend on ρ directly, but it is exactly what the consensus term penalizes, and the annealing schedule (8), which raises ρ over training, increases the weight the objective places on it as training proceeds;
- the *non-Gaussianity* $\bar{\Delta}$ is the modeling residual of assumption (1). It is not optimized directly, but it is bounded by the same root-mean-square distance, $\bar{\Delta} \le 2\sqrt{2L_{\mathrm{con}}/\rho}$, so concentrating the embeddings around their prototypes shrinks it whatever their shape; this is what (16) records. The bound (15) is the sharper of the two when the class-conditional laws are close to Gaussian, as the success of Gaussian models of penultimate features suggests (Lee et al., 2018), and within-class variability is known to collapse late in training (Papyan et al., 2020), which drives $\bar{\Delta}$ to zero;
- the *covariance estimation error* $\|\hat{\Sigma}_i - \Sigma_i\|_F$ is the error of the per-epoch covariance refresh of Algorithm 1. Covariance concentration bounds such as (7) control it when the refreshed embeddings are treated as independent draws from P_i (a bound stated in operator norm converts via $\|\cdot\|_F \le \sqrt{k}\,\|\cdot\|_2$). Because N_1 is fitted to the same data, this is an idealization rather than a guarantee, but it ties the sample-size discussion of Section 3.1 to a term of the risk bound;
- the *floor bias* $\sqrt{k}\,\sigma_0$ is the price paid for the variance floor — the sampling law deliberately over-disperses relative to the embedding law. It trades off against the estimation term, which the floor shrinks: with $\bar{\varepsilon} := \frac{1}{n}\sum_i \|\hat{\Sigma}_i - \Sigma_i\|_F$, the sum $\sqrt{k}\,\sigma_0 + \bar{\varepsilon}/(2\sigma_0)$ is minimized at $\sigma_0^\star = \big(\bar{\varepsilon}/(2\sqrt{k})\big)^{1/2}$, where it equals $\sqrt{2}\,k^{1/4}\bar{\varepsilon}^{1/2}$. The bound itself thus favors an intermediate floor.

Two caveats apply. The bound controls the cross-entropy risk, which bounds the misclassification probability only up to the factor $1/\log 2$, so it speaks to accuracy indirectly. And L_{con} and Σ_i are population quantities: the bound holds deterministically for any parameters, and the question of generalization is moved into the gap between the population and training values of the consensus term. With these caveats, discarding the prototypes at inference is sound because the objective that trained N_2 was, up to these audited terms, the deployed risk itself.

B.2 Sampling is an Explicit Curvature Regularizer

Corollary B.4 explains why reconnection does not degrade the deployed risk. However, it does not explain why lifted training can *gain* accuracy over end-to-end training. The gain has a separate source: the sampled loss evaluates the head on entire neighborhoods of the prototypes rather than at isolated points. For our head this can only increase the loss relative to the prototypes themselves, and to second order the increase is an explicit, non-negative curvature penalty.

Proposition B.5 (Second-order form of the sampled loss)**.** *Let $\ell : \mathbb{R}^k \to \mathbb{R}$ be three times continuously differentiable with $\big|\nabla^3\ell(z)[u,u,u]\big| \le M\|u\|_2^3$ for all $z, u \in \mathbb{R}^k$. Then for $Q = \mathcal{N}(s,\Sigma)$,*

$$\Big|\,\mathbb{E}_{z\sim Q}\,\ell(z) - \ell(s) - \tfrac{1}{2}\operatorname{tr}\big(\Sigma\,\nabla^2\ell(s)\big)\Big| \;\le\; \tfrac{M}{6}\,3^{3/4}\,(\operatorname{tr}\Sigma)^{3/2}. \tag{17}$$

For the linear head with cross-entropy (Theorem B.2), each ℓ_i is convex, its Hessian is $\nabla^2\ell_i(z) = W_2^\top H(z) W_2$ with $H(z) = \operatorname{diag}\big(p(z)\big) - p(z)p(z)^\top \succeq 0$ for every z, and the hypothesis above holds with $M = \|W_2\|_2^3/\sqrt{2}$. Consequently

$$\mathbb{E}_{z\sim Q_i}\,\ell_i(z) \;\ge\; \ell_i(s_i) \tag{18}$$

exactly, and

$$\mathbb{E}_{z\sim Q_i}\,\ell_i(z) \;=\; \ell_i(s_i) \;+\; \underbrace{\tfrac{1}{2}\operatorname{tr}\big(\tilde{\Sigma}_i\,W_2^\top H(s_i) W_2\big)}_{\ge 0} \;+\; r_i, \qquad |r_i| \le \tfrac{3^{3/4}}{6\sqrt{2}}\,\|W_2\|_2^3\,(\operatorname{tr}\tilde{\Sigma}_i)^{3/2}. \tag{19}$$

Proof. Write $z = s + \zeta$ with $\zeta \sim \mathcal{N}(0, \Sigma)$. Third-order Taylor expansion of $t \mapsto \ell(s + t\zeta)$ with Lagrange remainder gives, for every ζ,

$$\left|\ell(s+\zeta) - \ell(s) - \nabla\ell(s)^\top\zeta - \tfrac{1}{2}\zeta^\top\nabla^2\ell(s)\zeta\right| \le \tfrac{M}{6}\|\zeta\|_2^3.$$

Taking expectations, $\mathbb{E}[\nabla\ell(s)^\top\zeta] = 0$ and $\mathbb{E}[\frac{1}{2}\zeta^\top\nabla^2\ell(s)\zeta] = \frac{1}{2}\operatorname{tr}(\Sigma\nabla^2\ell(s))$. For the remainder, Hölder's inequality and the Gaussian fourth moment $\mathbb{E}\|\zeta\|_2^4 = (\operatorname{tr}\Sigma)^2 + 2\operatorname{tr}(\Sigma^2) \le 3(\operatorname{tr}\Sigma)^2$ give $\mathbb{E}\|\zeta\|_2^3 \le (\mathbb{E}\|\zeta\|_2^4)^{3/4} \le 3^{3/4}(\operatorname{tr}\Sigma)^{3/2}$, proving (17).

For the cross-entropy linear head, $\ell_i(z) = -\log p_i(z) = \mathrm{LSE}(W_2 z + b) - (W_2 z + b)_i$ with $p(z) = \mathrm{softmax}(W_2 z + b)$ and LSE the log-sum-exp function. As a convex function of an affine map minus an affine map, ℓ_i is convex, and since $\mathbb{E}_{Q_i}[z] = s_i$, Jensen's inequality gives (18). The chain rule gives $\nabla\ell_i(z) = W_2^\top(p(z) - e_i)$ and $\nabla^2\ell_i(z) = W_2^\top\big(\operatorname{diag}(p(z)) - p(z)p(z)^\top\big)W_2$. The middle factor $H(z)$ is positive semi-definite: for any $v \in \mathbb{R}^n$, $v^\top H(z)v = \sum_j p_j v_j^2 - (\sum_j p_j v_j)^2 \ge 0$ by Jensen's inequality, hence $\operatorname{tr}(\tilde{\Sigma}_i W_2^\top H(s_i) W_2) \ge 0$ since the trace of a product of two positive semi-definite matrices is non-negative.

For the third derivative, fix z, u and let $w := W_2 u$. Up to an affine function of t, $t \mapsto \ell_i(z + tu)$ equals $\log\sum_j p_j(z)e^{tw_j}$, the cumulant generating function of $V := w_J$ with $J \sim p(z)$. Hence $\nabla^3\ell_i(z)[u,u,u] = \mathbb{E}[(V - \mathbb{E}V)^3]$. With $r := \max_j w_j - \min_j w_j$ we have $|V - \mathbb{E}V| \le r$ and, by Popoviciu's inequality, $\operatorname{Var} V \le r^2/4$, so $|\mathbb{E}[(V - \mathbb{E}V)^3]| \le r^3/4$. Finally $r \le \sqrt{2}\,\|w\|_2 \le \sqrt{2}\,\|W_2\|_2\|u\|_2$, which gives $M = \|W_2\|_2^3/\sqrt{2}$. Equation (19) follows from (17) applied to ℓ_i at $s = s_i$, $\Sigma = \tilde{\Sigma}_i$. □

Equation (19) is the class-prototype analogue of the classical result that training with input noise is, to second order, equivalent to Tikhonov regularization (Bishop, 1995): the sampled loss equals the prototype loss plus a penalty on the $\tilde{\Sigma}_i$-weighted curvature of the head at the prototype. Fitting the prototypes constrains $\ell_i(s_i)$ only; among heads that fit them equally well, the sampled objective prefers the one whose loss curves least around them, in the directions of $\tilde{\Sigma}_i$. Those are the directions in which the head's inputs at inference, $N_1(x) = s_i + \delta$ with a nonzero reconnection residual δ, deviate from the prototype.

Two qualifications apply to this reading. First, it is a second-order statement. The remainder constant in (19) grows like $\|W_2\|_2^3$, so ranking heads by the curvature term alone is justified only when the logit perturbation scale $\|W_2\|_2(\operatorname{tr}\tilde{\Sigma}_i)^{1/2}$ is small. Second, curvature at the prototype is not a global sensitivity measure. Writing $\omega_j^\top$ for the rows of W_2 and $\bar{\omega} := \sum_j p_j(s_i)\,\omega_j$,

$$\operatorname{tr}\big(W_2^\top H(s_i) W_2\big) = \sum_{j=1}^{n} p_j(s_i)\,\|\omega_j - \bar{\omega}\|_2^2, \tag{20}$$

the $p(s_i)$-weighted spread of the class weight vectors. A head that is saturated at the prototype, $p(s_i) \approx e_i$, makes this spread small even when $\|W_2\|_2$, and with it the Lipschitz constant G, is large. Scaling up W_2 in fact drives the curvature penalty to zero exponentially fast. That regime lies outside the validity of the expansion. What keeps the head out of it is the weight decay on N_2, which also controls G in (15).

Relation to end-to-end training. Theorem B.5 compares the sampled loss with the loss at the prototypes, not with the end-to-end loss $\mathbb{E}_{P_i}\ell_i$ on the embeddings themselves. The latter comparison holds exactly in the idealized case $s_i = \mu_i$, $\hat{\Sigma}_i = \Sigma_i$. Then Q_i is the convolution of $\Gamma_i = \mathcal{N}(\mu_i, \Sigma_i)$ with $\mathcal{N}(0, \sigma_0^2 I_k)$, and convexity of ℓ_i gives

$$\mathbb{E}_{Q_i}\ell_i = \mathbb{E}_{z\sim\Gamma_i}\,\mathbb{E}_{\xi\sim\mathcal{N}(0,\sigma_0^2 I_k)}\,\ell_i(z+\xi) \;\ge\; \mathbb{E}_{\Gamma_i}\ell_i \;\ge\; \mathbb{E}_{P_i}\ell_i - G\,\Delta_i,$$

where the last step is the coupling argument of Theorem B.3. Applying (17) to both Gaussians shows that, to second order, the excess of the sampled loss over $\mathbb{E}_{\Gamma_i}\ell_i$ is $\frac{\sigma_0^2}{2}\operatorname{tr}(W_2^\top H(s_i)W_2)$. Relative to end-to-end training, the additional regularization therefore comes from the floor part of $\tilde{\Sigma}_i$, not from the whole of $\tilde{\Sigma}_i$. Outside the idealized case, the mean and covariance mismatches of Theorem B.4 enter as well.

It is worth mentioning that the variance regularization via σ_0 keeps the smoothing alive under collapse. Since Q_i is also the convolution of $\mathcal{N}(s_i, \sigma_0^2 I_k)$ with $\mathcal{N}(0, \hat{\Sigma}_i)$, the same convexity argument gives

$$\mathbb{E}_{Q_i} \ell_i \;\geq\; \mathbb{E}_{\xi \sim \mathcal{N}(0, \sigma_0^2 I_k)}\, \ell_i(s_i + \xi), \tag{21}$$

and, to second order, since $\tilde{\Sigma}_i \succeq \sigma_0^2 I_k$ and $W_2^\top H(s_i) W_2 \succeq 0$,

$$\tfrac{1}{2}\,\mathrm{tr}\big(\tilde{\Sigma}_i W_2^\top H(s_i) W_2\big) \;\geq\; \tfrac{\sigma_0^2}{2}\;\mathrm{tr}\big(W_2^\top H(s_i) W_2\big). \tag{22}$$

Even in the neural-collapse limit $\hat{\Sigma}_i \to 0$, in which the estimated covariance degenerates, the sampled loss therefore remains an isotropically smoothed loss at scale σ_0 rather than reducing to the point loss $\ell_i(s_i)$. This resolves the tension noted in Section 1: within-class collapse makes the covariance *estimate* degenerate, but the floor converts the degenerate limit into fixed isotropic smoothing rather than none. The floor alone does not keep the penalty bounded away from zero: by (20), the right side of (22) vanishes as the head saturates. The two safeguards act together: the floor fixes the smoothing scale, and weight decay on N_2 bounds $\|W_2\|_2$ and hence the degree of saturation.

B.3 An Exactly Solvable One-Dimensional Instance

The results of Sections B.1 and B.2 are stated as bounds with remainder terms. This subsection exhibits the selection effect of Theorem B.5 exactly, in the smallest setting in which it occurs: one embedding dimension, two classes, and a one-parameter family of heads.

The example holds the first subnetwork fixed and looks only at the classification problem that the second subnetwork has to solve. It is consequently not a claim that conditioning by itself improves generalization. What it does show, in closed form, is the following. Fitting the class prototypes leaves one degree of freedom of the head completely undetermined, sampling in a neighborhood of the prototypes removes that ambiguity, and the solution it picks out is the one whose output is least sensitive to a perturbation of its input.

B.3.1 Setup

Consider balanced binary classification with labels $Y \in \{-1, +1\}$. Take the lifting dimension to be $k = 1$ and assume that the output of the fixed first subnetwork is

$$N_1(X) = Y + \varepsilon, \qquad \varepsilon \sim \mathcal{N}(0, \sigma^2), \qquad \varepsilon \text{ independent of } Y, \tag{23}$$

so that the two class prototypes are $s_- = -1$ and $s_+ = +1$. This is the one-dimensional, homoscedastic instance of the class-conditional model (1); the variance floor of (3) can be absorbed into σ^2. We use the squared classification loss $\mathcal{L}_{\mathrm{cls}}(y, i) = \frac{1}{2}\|y - e_i\|_2^2$ and replace the linear head by the nonlinear one-parameter family

$$N_2(z; a) = g_a(z) := z + a\,(z^3 - z), \qquad a \in \mathbb{R}. \tag{24}$$

The point of this family is that the factor $z^3 - z$ vanishes at both prototypes. Every member therefore reproduces the prototypes exactly,

$$g_a(1) = 1, \qquad g_a(-1) = -1 \qquad \text{for all } a \in \mathbb{R}, \tag{25}$$

while the members differ in how they behave *near* the prototypes, since

$$g_a'(z) = 1 + a\,(3z^2 - 1), \qquad \text{so} \qquad g_a'(\pm 1) = 1 + 2a. \tag{26}$$

The parameter a is thus invisible to any criterion that only inspects the prototypes, and it is exactly the local sensitivity of the head.

To separate "fitting the prototypes" from "fitting their neighborhoods", define

$$L_{\mathrm{point}}(a) := \mathbb{E}_Y\big[(g_a(Y) - Y)^2\big], \tag{27}$$

$$L_{\mathrm{sample}}(a) := \mathbb{E}_{Y,\varepsilon}\big[(g_a(Y + \varepsilon) - Y)^2\big]. \tag{28}$$

The first is the zero-variance limit $\sigma \downarrow 0$, in which N_2 sees only the two prototypes. The second is the population version of the classification term in (5)–(6).

B.3.2 THE SELECTION RESULT

Proposition B.6 (Sampling determines the local sensitivity)**.** *Assume* (23)–(24)*. Then*

(i) $L_{\mathrm{point}}(a) = 0$ *for every* $a \in \mathbb{R}$*; the prototype loss does not identify* a*.*

(ii) For every $\sigma > 0$ *the sampled loss is the strictly convex quadratic*

$$L_{\mathrm{sample}}(a) = \sigma^2 + 2a\,(2\sigma^2 + 3\sigma^4) + a^2(4\sigma^2 + 39\sigma^4 + 15\sigma^6), \tag{29}$$

with the unique minimizer

$$a^\star(\sigma) = -\frac{2 + 3\sigma^2}{4 + 39\sigma^2 + 15\sigma^4}. \tag{30}$$

(iii) The sensitivity of the selected head at the prototypes is

$$g'_{a^\star}(\pm 1) = \frac{33\sigma^2 + 15\sigma^4}{4 + 39\sigma^2 + 15\sigma^4} \in (0, 1), \qquad g'_{a^\star}(\pm 1) = \tfrac{33}{4}\sigma^2 + O(\sigma^4) \longrightarrow 0 \quad \text{as } \sigma \downarrow 0. \tag{31}$$

In words: fitting the prototypes leaves the sensitivity of the head undetermined, whereas sampling any neighborhood of them fixes it, and in the small-noise limit the value selected is the one that makes the head stationary at both prototypes.

Proof. Part (i) is immediate from (25).

For (ii), put $q := \sigma^2$ and $Z := Y + \varepsilon$. Expanding the cube and using $Y^2 = 1$, hence $Y^3 = Y$,

$$\begin{aligned} Z^3 - Z &= (Y + \varepsilon)^3 - (Y + \varepsilon) \\ &= Y^3 + 3Y^2\varepsilon + 3Y\varepsilon^2 + \varepsilon^3 - Y - \varepsilon \\ &= 2\varepsilon + 3Y\varepsilon^2 + \varepsilon^3, \end{aligned} \tag{32}$$

so that the residual of the head is

$$g_a(Z) - Y = \varepsilon + a\,U, \qquad U := 2\varepsilon + 3Y\varepsilon^2 + \varepsilon^3. \tag{33}$$

Because ε is centred Gaussian and independent of Y, and because $\mathbb{E}[Y] = 0$ by balance, every monomial containing an odd power of ε or an odd power of Y has zero expectation. With the Gaussian moments

$$\mathbb{E}[\varepsilon^2] = q, \qquad \mathbb{E}[\varepsilon^4] = 3q^2, \qquad \mathbb{E}[\varepsilon^6] = 15q^3, \tag{34}$$

the two remaining expectations are

$$\mathbb{E}[\varepsilon U] = \mathbb{E}\big[2\varepsilon^2 + 3Y\varepsilon^3 + \varepsilon^4\big] = 2q + 3q^2, \tag{35}$$

$$\mathbb{E}[U^2] = \mathbb{E}\big[4\varepsilon^2 + 12Y\varepsilon^3 + (9Y^2 + 4)\varepsilon^4 + 6Y\varepsilon^5 + \varepsilon^6\big] = 4q + 39q^2 + 15q^3. \tag{36}$$

Squaring (33) and taking expectations gives $L_{\mathrm{sample}}(a) = q + 2a(2q + 3q^2) + a^2(4q + 39q^2 + 15q^3)$, which is (29). The coefficient of a^2 is $\mathbb{E}[U^2] > 0$ for every $q > 0$, so the quadratic is strictly convex and its stationary point is its unique minimizer:

$$a^\star = -\frac{\mathbb{E}[\varepsilon U]}{\mathbb{E}[U^2]} = -\frac{2q + 3q^2}{4q + 39q^2 + 15q^3} = -\frac{2 + 3q}{4 + 39q + 15q^2}, \tag{37}$$

which is (30). Part (iii) follows from (26):

$$g'_{a^\star}(\pm 1) = 1 + 2a^\star = \frac{(4 + 39q + 15q^2) - (4 + 6q)}{4 + 39q + 15q^2} = \frac{33q + 15q^2}{4 + 39q + 15q^2}, \tag{38}$$

which lies in $(0, 1)$ for every $q > 0$ and equals $\frac{33}{4}q + O(q^2)$ as $q \downarrow 0$. Substituting $q = \sigma^2$ completes the proof. □

B.3.3 INTERPRETATION

The proposition separates two things that the full algorithm does at once. Identity (25) says that classifying the prototypes perfectly puts no constraint at all on how N_2 responds to the *actual* output of N_1, which is a noisy version of a prototype. The expectation in (28) supplies exactly the missing constraint.

As $\sigma \downarrow 0$ the minimizer tends to $a^\star = -1/2$, that is, to

$$g_{-1/2}(z) = \tfrac{3}{2}z - \tfrac{1}{2}z^3, \qquad g'_{-1/2}(\pm 1) = 0. \tag{39}$$

This is Theorem B.5 made exact: expanding (29) to leading order gives $L_{\mathrm{sample}}(a) = \sigma^2(1+2a)^2 + O(\sigma^4) = \sigma^2\, g'_a(\pm 1)^2 + O(\sigma^4)$, which is precisely the sensitivity penalty $\frac{1}{2}\operatorname{tr}(\Sigma \nabla^2 \ell(s))$ of (17) in this model (the prototype loss vanishes by (25)), and its minimizer is the flattest head. The higher-order terms in (29) are the remainder of (17), here available in closed form. The reason this matters is visible in the response to a small deterministic reconnection error δ. Setting $\varepsilon = \delta$ in (33),

$$g_a(Y+\delta) - Y = (1+2a)\,\delta + 3aY\,\delta^2 + a\,\delta^3. \tag{40}$$

The identity head $a = 0$ makes an error of order δ; the limiting sampled solution $a = -1/2$ cancels the linear term and makes an error of order δ^2, hence a squared loss of order δ^4 rather than δ^2. This is an exact toy instance of feature-space noise acting as a sensitivity regularizer for the second subnetwork.

Remark B.7 (The selected head is matched to the noise level). Sampling does not always favour a flat head. By (30), $a^\star(\sigma) \to 0$ as $\sigma \to \infty$, so at large noise the sampled objective returns to the identity head, and $g'_{a^\star}(\pm 1)$ increases monotonically from 0 to 1 as σ grows. The mechanism selects the head matched to the noise level, and flatness is the small-noise end of that family rather than the target of the objective.

Remark B.8 (Scope of the conclusion). The conclusion is deliberately local, for two reasons.

First, it is not a statement about clean accuracy. The head g_a is odd, so whenever it is increasing on the interval that carries the data, its decision boundary is $z = 0$ and its accuracy equals that of the identity head. By (26), however, g_a with $a < 0$ is increasing only on $|z| \le \sqrt{(1-a)/(-3a)}$ and changes sign again at $|z| = \sqrt{1 - 1/a}$. For the limiting head $a = -1/2$ these radii are 1 and $\sqrt{3}$: a sample with $|Z| > \sqrt{3}$ is misclassified even though it is on the correct side of the origin. Since Z is Gaussian, this happens with positive probability, and at moderate noise it is not negligible — Section B.3.4 measures a loss of roughly seven accuracy points at $\sigma = 0.5$ for $a = -1/2$. The σ-matched head $a^\star(\sigma)$ is far milder, because $|a^\star|$ shrinks as σ grows (Theorem B.7); its accuracy stays within 0.15 points of the identity for $\sigma \le 0.5$ and within 0.6 points over the whole range tested.

Second, the cubic should not be extrapolated to large $|z|$, where its behavior is not representative of a practical classifier.

What Theorem B.6 does establish is a concrete mechanism by which the *changed* lifted objective — rather than faster optimization of the same objective — selects a different and locally more robust solution. The experiments below test whether this mechanism mediates the accuracy gains of Section 4.3.

B.3.4 ILLUSTRATION AND NUMERICAL VERIFICATION

The proposition can be checked without training a network. On a grid of noise levels σ we draw $n = 2 \cdot 10^5$ samples (Y, Z) from (23) and minimize the empirical version of (28). Because that empirical loss is again quadratic in a, its minimizer is available in closed form,

$$\hat{a} = -\frac{\sum_i (Z_i - Y_i)\,(Z_i^3 - Z_i)}{\sum_i (Z_i^3 - Z_i)^2}, \tag{41}$$

which we cross-check against a derivative-free line search. Each configuration is repeated over 25 seeds; the reported error bars are standard deviations across seeds. A held-out sample of the same size is used for the reported losses and accuracies.

Across $\sigma \in [0.02, 1]$ the estimate $\hat{a}$ agrees with (30) and the measured sensitivity $|g'_{\hat{a}}(1)|$ agrees with (31) to within one seed standard deviation; the empirical loss agrees with the closed form (29) to within Monte-Carlo error. Figure 5 shows both comparisons, and Figure 6 shows the response to a bounded reconnection error.

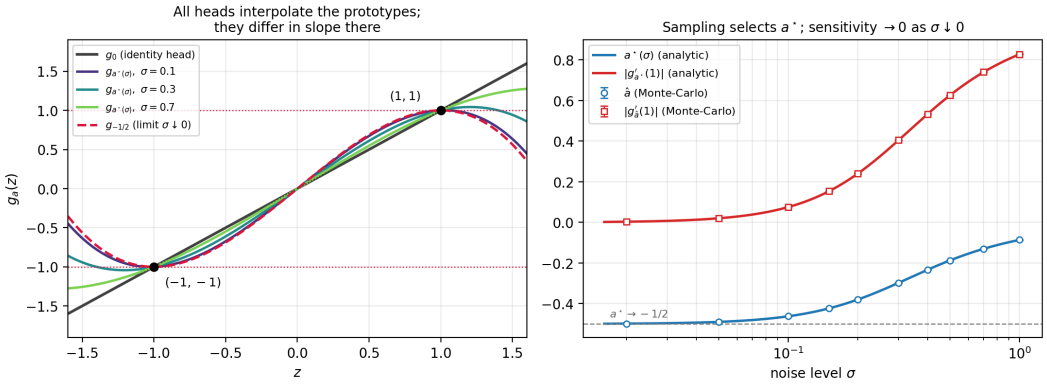


Figure 5: **Left:** the identity head g_0, the sampled optima $g_{a^\star(\sigma)}$ for $\sigma \in \{0.1, 0.3, 0.7\}$, and the small-noise limit $g_{-1/2}$. All curves pass through $(-1, -1)$ and $(1, 1)$; the dotted lines are the tangents there, which are the only thing that distinguishes the family. **Right:** Monte-Carlo estimates of $a^\star(\sigma)$ and of $|g'_{a^\star}(1)|$ against the closed forms (30) and (31). Markers are means over 25 seeds and error bars are one standard deviation; they are smaller than the markers at most noise levels.

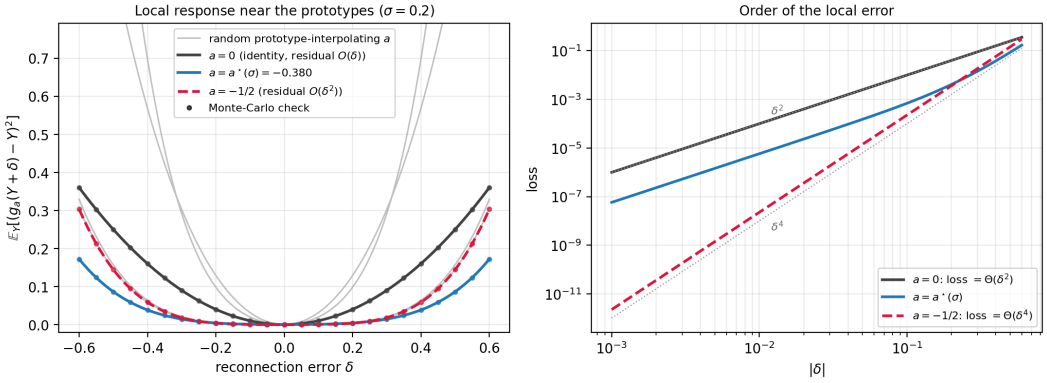


Figure 6: Loss $\mathbb{E}_Y[(g_a(Y + \delta) - Y)^2]$ under a bounded deterministic reconnection error δ, restricted to the neighborhood of the prototypes for which the local model is intended, at $\sigma = 0.2$. Grey curves are randomly drawn prototype-interpolating values of a, which illustrate that interpolation alone constrains nothing. **Right:** the same curves on log–log axes, showing the $\Theta(\delta^2)$ behavior of the identity head against the $\Theta(\delta^4)$ behavior of $g_{-1/2}$ predicted by (40).