Simplex Deep Linear Discriminant Analysis

Arman Bolatov[1,4], Maxat Tezekbayev[2], and Zhenisbek Assylbekov[3,4]

[1]Mohamed bin Zayed University of Artificial Intelligence
[2]Nazarbayev University
[3]Purdue University Fort Wayne
[4]National Center of Space Research and Technology, Almaty, Kazakhstan
arman.bolatov@mbzuai.ac.ae, maxat.tezekbayev@alumni.nu.edu.kz, zassylbe@pfw.edu

We revisit Deep Linear Discriminant Analysis (Deep LDA) from a likelihood-based perspective. While classical LDA is a simple Gaussian model with linear decision boundaries, attaching an LDA head to a neural encoder raises the question of how to train the resulting deep classifier by maximum likelihood estimation (MLE). We first show that end-to-end MLE training of an unconstrained Deep LDA model ignores discrimination: when both the LDA parameters and the encoder parameters are learned jointly, the likelihood admits a degenerate solution in which some of the class clusters may heavily overlap or even collapse, and classification performance deteriorates. Batchwise moment re-estimation of the LDA parameters does not remove this failure mode. We then propose a constrained Deep LDA formulation that fixes the class means to the vertices of a regular simplex in the latent space and restricts the shared covariance to be spherical, leaving only the priors and a single variance parameter to be learned along with the encoder. Under these geometric constraints, MLE becomes stable and yields well-separated class clusters in the latent space. On images (Fashion-MNIST, CIFAR-10, CIFAR-100) and texts (AG News, CLINC150), the resulting Deep LDA models achieve accuracy competitive with softmax baselines while offering a simple, interpretable latent geometry that is clearly visible in two-dimensional projections.

1. Introduction

Linear Discriminant Analysis (LDA) traces its roots to the classical works of Fisher [1] and Rao [2]. It remains one of the most widely used classification methods and is valued for its interpretability. LDA assumes that the class label $Y \in \{1, \dots, C\}$ follows a categorical prior with probabilities $\pi = (\pi_1, \dots, \pi_C)$ on the probability simplex:

$$\Pr(Y = c) = \pi_c, \qquad \pi_c \geq 0, \ \sum_{c=1}^{C} \pi_c = 1. \tag{1}$$

Conditional on the class, the feature vector $X \in \mathbb{R}^d$ is assumed Gaussian with a *shared* covariance matrix $\Sigma \in \mathbb{R}^{d \times d}$:

$$X \mid Y = c \ \sim \ \mathcal{N}(\mu_c, \ \Sigma), \qquad c = 1, \dots, C, \tag{2}$$

so that each class has its own mean $\mu_c \in \mathbb{R}^d$ while the covariance Σ is common to all classes.

By specifying the marginal law of Y and the class-conditional law of $X \mid Y$, LDA defines a joint probabilistic model over (X, Y). As a generative model, its parameters $\big(\pi, \{\mu_c\}_{c=1}^{C}, \Sigma\big)$ can be estimated by maximum likelihood. Once fitted, prediction for a new sample x proceeds via the posterior class probabilities $\Pr(Y = c \mid X = x)$, obtained from the prior $\Pr(Y = c)$ and the class-conditionals $p(x \mid Y = c)$ by Bayes' rule.

Because LDA posits an explicit data-generating story, it offers clear interpretability. A well-known limitation, however, is that the resulting decision boundaries are *linear* between any pair of classes (Figure 1), which can be too restrictive when classes are not linearly separable in the input space.

Third Conference on Parsimony and Learning (CPAL 2026).

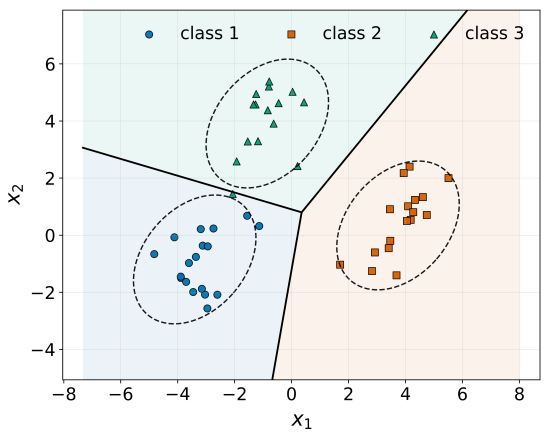


Figure 1: Three-class LDA example with shared covariance. Ellipses that contain 90% of the probability for each of the three classes are shown. Faint colors show decision regions, and solid lines mark decision boundaries.

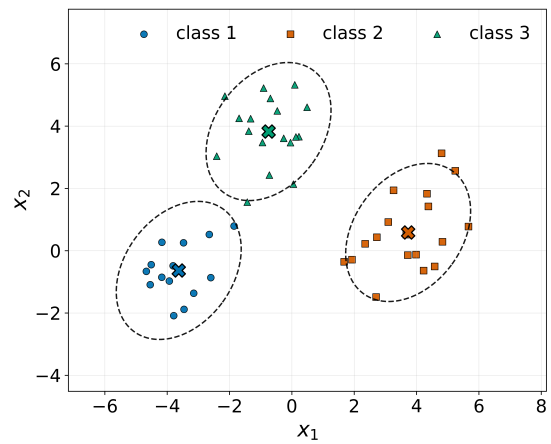


Figure 2: Gradient-based ML fit of a three-class LDA model ($n = 50$). Dashed ellipses show the 90% contours of the learned Gaussian classes, and filled "X" markers indicate the estimated class centers. The recovered parameters nearly coincide with the closed-form MLE.

To address this, *Deep LDA* [3] couples a learnable representation with an LDA head. Concretely, one first maps inputs into a latent space via a neural encoder $Z = f_\psi(X)$ parameterized by ψ. In that latent space, the pair (Z, Y) is modeled according to the LDA assumptions in (1)–(2) with class means $\{\mu_c\}$ and a shared covariance Σ. In effect, Deep LDA replaces the conventional softmax layer with an LDA head, while letting the encoder learn a representation in which the LDA model is appropriate.

In contrast to a maximum-likelihood treatment, Deep LDA [3] optimizes Fisher's generalized eigenvalue objective in the latent space. A straightforward maximization, however, turned out to be unstable: as noted by Dorfer et al. [3], the network tends to inflate a single largest eigenvalue and essentially ignore poorly separated classes, which leads to trivial solutions. To mitigate this, they introduce a heuristic variant of the criterion that focuses only on the eigen-directions with small discriminative power (see their Eq. (9)), departing from the classical Fisher objective.

However, LDA is a classical parametric probabilistic model, and such models are typically trained with maximum likelihood estimation (MLE). This naturally raises the question:

> *Why not train Deep LDA by maximum likelihood, using standard gradient-based methods end-to-end?*

However, as we show in Section 2, when both the class means and the shared covariance are learned jointly with a flexible encoder, the likelihood admits a maximizer that ignores discrimination: the encoder not only collapses each sample onto its corresponding class mean, but also may drive some of the class means toward each other so that the resulting clusters become arbitrarily tight and heavily overlapping. This yields an arbitrarily high training likelihood but destroys the geometric structure required for successful discrimination.

The failure is a matter of structural degeneracy of the model. Even replacing gradient updates of the LDA parameters with batchwise moment estimates does not prevent collapse: the encoder can always shrink between-class variation, and nothing in the free LDA parameterization opposes this.

These observations motivate a different standpoint. Instead of allowing the LDA head to adapt freely to the encoder, we constrain its geometry so that maximum likelihood becomes well-behaved in the deep setting. In Section 3 we introduce a simple and effective modification: fix the class means to the vertices of a regular simplex and restrict the shared covariance to be spherical. Under these

constraints, the likelihood no longer admits degenerate solutions, and end-to-end MLE becomes stable. The encoder learns to map inputs to well-separated clusters around fixed prototypes, preserving the probabilistic structure of LDA while avoiding the collapse seen in the unconstrained model.

The remainder of the paper develops this perspective and demonstrates its practical viability on both synthetic and real datasets. Our code is available at `https://github.com/zh3nis/simplex-lda`.

2. Maximum-Likelihood Training: A First Look

As a warm-up, we first verify that the classical (non-deep) LDA admits accurate MLE via gradient-based optimization on synthetic data.

2.1. Warm-up on Classical LDA

We consider the data generation model in (1) and (2). Let $\theta \coloneqq \big(\pi, \{\mu_c\}_{c=1}^{C}, \Sigma\big)$ denote the parameters with π on the probability simplex and $\Sigma \succ 0$. Given a sample $\{(x_1, y_1), \ldots, (x_n, y_n)\}$ drawn from (1)–(2), the log-likelihood is

$$\mathcal{L}(\theta) = \frac{1}{n} \sum_{i=1}^{n} \log\Big(\pi_{y_i}\, \phi(x_i;\, \mu_{y_i}, \Sigma)\Big), \tag{3}$$

where, for $x \in \mathbb{R}^d$, $\phi(x; \mu, \Sigma) = (2\pi)^{-d/2} \, (\det \Sigma)^{-1/2} \exp\big\{-\frac{1}{2}(x-\mu)^\top \Sigma^{-1}(x-\mu)\big\}$ is the $\mathcal{N}(\mu, \Sigma)$ density.

Optimization setup. We maximize the log-likelihood $\mathcal{L}(\theta)$ with Adam, treating its negative $-\mathcal{L}(\theta)$ as a loss. In practice it is convenient to use unconstrained reparameterizations: (i) logits $\alpha \in \mathbb{R}^C$ with $\pi = \mathrm{softmax}(\alpha)$, and (ii) a Cholesky factorization $\Sigma = LL^\top$ with L lower triangular and positive diagonal. This enforces $\pi \in \Delta^{C-1}$ and $\Sigma \succ 0$ while enabling standard auto-differentiation.

Result. On synthetic data generated from (1)–(2), gradient descent with Adam reliably converges to the maximum-likelihood solution; see Figure 2. This confirms that, at least for the classical LDA model, direct gradient-based likelihood maximization is effective and accurate.

Relation to Fisher's criterion. Classical LDA can also be derived by maximizing Fisher's generalized eigenvalue objective

$$\max_{W \in \mathbb{R}^{d \times (C-1)}} J(W) \coloneqq \mathrm{tr}\big((W^\top S_W W)^{-1} W^\top S_B W\big), \tag{4}$$

where S_W and S_B are the within-class and between-class scatter matrices. For Gaussian class-conditional models with shared covariance, maximizing (4) yields the same discriminant subspace—and hence the same decision boundaries—as the maximum-likelihood solution of the generative LDA model.

To verify this equivalence in our setting, we repeated the experiment above but trained the encoder by optimizing Fisher's criterion instead of the log-likelihood. As expected, the learned decision boundaries, and recovered parameters matched those obtained from maximum-likelihood training, confirming that both objectives recover the same classifier in the shared-covariance Gaussian regime.

2.2. A Cautionary Experiment: Likelihood Training of Deep LDA

For the Deep LDA experiment, we again generate a dataset $\{(x_1, y_1), \ldots, (x_n, y_n)\}$ from the model in (1)–(2). The encoder $f_\psi(\cdot)$ is a two-layer feed-forward network with ReLU activations and 32 hidden units. The resulting log-likelihood is

$$\mathcal{L}(\theta) = \frac{1}{n} \sum_{i=1}^{n} \log(\pi_{y_i}\, \phi(f_\psi(x_i);\, \mu_{y_i},\, \Sigma))\,, \tag{5}$$

where $\theta := (\psi, \pi, \{\mu_c\}_{c=1}^{C}, \Sigma)$.

We generate 20,000 training and 4,000 test samples. Training is run for 100 epochs using Adam with PyTorch's default hyperparameters and a minibatch size of 256 (batch size 1024 at evaluation). The learned embeddings $f_\psi(x_i)$ are shown in Figure 3.

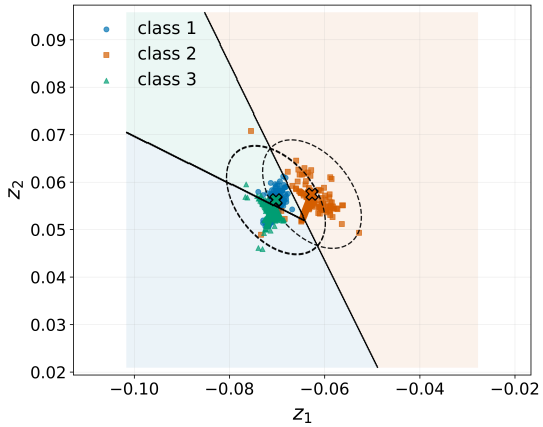


Figure 3: Deep LDA embeddings $z_i = f_\psi(x_i)$ after likelihood training. Two classes collapse; samples lie near their class centroids $\{\mu_c\}$ with a nearly singular shared covariance Σ ($|\Sigma| \approx 3 \cdot 10^{-10}$). Training accuracy: 67.1%; test accuracy: 66.6%.

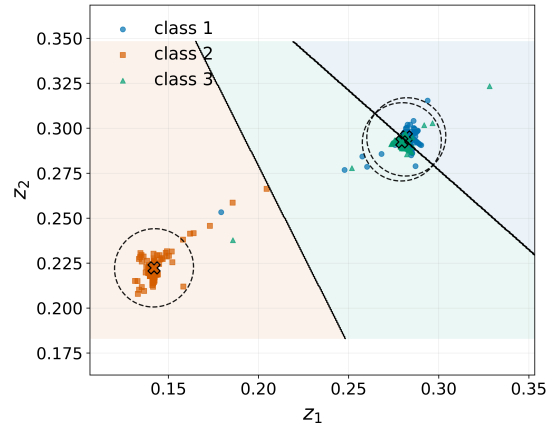


Figure 4: Deep LDA embeddings $f_\psi(x_i)$ after likelihood training with batchwise LDA re-estimation. Two class clusters collapse; embeddings concentrate near their centroids $\{\mu_c\}$ while the shared covariance becomes ill-conditioned ($|\Sigma| \approx 10^{-8}$). Training accuracy: 67.8%; test accuracy: 66.6%.

The pathology is immediate: the optimizer does not enforce separation among all classes—two of the class clusters collapse into one. Moreover, for each labeled pair (x, y), the encoder maps x extremely close to its class centroid μ_y, driving the squared Mahalanobis distance

$$(f_\psi(x) - \mu_y)^\top \Sigma^{-1} (f_\psi(x) - \mu_y)$$

toward zero, while Σ simultaneously shrinks toward near-singularity. Both effects inflate the Gaussian term $\phi(f_\psi(x); \mu_y, \Sigma)$, yielding a high training likelihood.[1]

This behavior exposes a fundamental mismatch between the maximum-likelihood objective and the goal of accurate classification: the training and test accuracies are only 67.1% and 66.6%, respectively.

2.3. Batchwise LDA Re-estimation

In an attempt to mitigate the collapse observed under end-to-end likelihood training, we stopped updating the LDA-head parameters $(\pi, \{\mu_c\}_{c=1}^{C}, \Sigma)$ by gradient descent and instead *estimated* them on mini-batches, while training only the encoder f_ψ. Let $\{(x_1, y_1), \ldots, (x_m, y_m)\}$ be the current batch and set $z_i = f_\psi(x_i)$. Define the batch index sets $I_c = \{i \in \{1, \ldots, m\} : y_i = c\}$ and $n_c = |I_c|$. The plug-in LDA estimates on this batch are

$$\widehat{\pi}_c^{(t)} = \frac{n_c}{m}, \qquad \widehat{\mu}_c^{(t)} = \frac{1}{n_c} \sum_{i \in I_c} z_i, \qquad c = 1, \ldots, C,$$

$$\widehat{\Sigma}^{(t)} = \frac{1}{m} \sum_{c=1}^{C} \sum_{i \in I_c} \big(z_i - \widehat{\mu}_c^{(t)}\big)\big(z_i - \widehat{\mu}_c^{(t)}\big)^\top.$$

[1]We note that replacing the log-likelihood with Fisher's criterion does not alleviate this issue: in our experiments, optimizing the Fisher objective produced the same collapse of embeddings and class means as the maximum-likelihood formulation. A similar observation was made by Dorfer et al. [3].

Because successive batches can vary substantially, we smooth these estimates with an exponential moving average (EMA). Writing tildes for the smoothed parameters, we use

$$\begin{aligned}
\widetilde{\pi}_c^{(t)} &= \beta\,\widetilde{\pi}_c^{(t-1)} + (1-\beta)\,\widehat{\pi}_c^{(t)}, \qquad c = 1,\dots,C,\\
\widetilde{\mu}_c^{(t)} &= \beta\,\widetilde{\mu}_c^{(t-1)} + (1-\beta)\,\widehat{\mu}_c^{(t)}, \qquad c = 1,\dots,C,\\
\widetilde{\Sigma}^{(t)} &= \beta\,\widetilde{\Sigma}^{(t-1)} + (1-\beta)\,\widehat{\Sigma}^{(t)},
\end{aligned}$$

with $\beta \in [0,1)$ (e.g., $\beta = 0.9$). At iteration t, the likelihood in (5) retains the same form, but now only the encoder parameters are optimized, i.e., $\theta := \psi$, while $\{\pi_c, \mu_c\}_{c=1}^C, \Sigma$ are the EMA-smoothed re-estimates $\{\widetilde{\pi}_c^{(t)}, \widetilde{\mu}_c^{(t)}\}_{c=1}^C, \widetilde{\Sigma}^{(t)}$.

Result. We run this modified Deep LDA training procedure on data generated from (1)–(2). Unfortunately, the batchwise re-estimation with EMA does not resolve the collapse; see Figure 4. Two of the three classes still collapse. For each labeled pair (x, y), the encoder continues to map x to an embedding $z = f_\psi(x)$ extremely close to its class mean μ_y, while the pooled covariance Σ contracts toward near-singularity ($|\Sigma| \approx 10^{-8}$). Both effects again inflate the Gaussian likelihood term, resulting in a large training objective but poor classification performance.[2]

We hypothesize that the issue is not with gradient-based MLE itself—which we still wish to retain—but with the unconstrained Deep LDA parameterization. This leads to the central question: *how should Deep LDA be constrained so that MLE learns an accurate classifier?*

3. Successful Training of Deep LDA with MLE

The failures observed in Section 2 stem from the excessive flexibility of the deep model: all the LDA parameters (π, μ_c and Σ) are free to adapt to the encoder f_ψ, allowing not only the embeddings to collapse tightly around their class centroids but also the class centroids themselves to drift closer together, producing overlapping clusters. In this section we show that likelihood-based training can lead to discrimination by *constraining* the geometry of the LDA head.

Simplex-Constrained Means and Spherical Covariance. We keep the encoder $Z = f_\psi(X)$ as before, but replace the fully free LDA head with a structured variant:

- The class means $\{\mu_c\}_{c=1}^C$ are fixed to the vertices of a regular simplex in $\mathbb{R}^{C-1}$, centered at the origin (an example for $C = 4$ is given in Figure 5).
- The shared covariance is restricted to be spherical: $\Sigma = \sigma^2 I_d$ with a *single* trainable variance parameter σ^2.
- The class priors $\{\pi_c\}_{c=1}^C$ remain learnable, parameterized via unconstrained logits.

Concretely, for C classes we first construct a regular simplex in $\mathbb{R}^{C-1}$ whose vertices have unit norm and zero mean. We rescale the simplex so that the Euclidean distance between any pair of class means is a fixed constant $s > 0$ (we use $s = 6$ so that the classes are well-separated at initialization when we set $\sigma = 1$).

The output of the encoder is made $(C-1)$-dimensional, since for C classes LDA needs at most $d = C - 1$ dimensions. For an embedding $Z = f_\psi(X)$, the class-conditional model is

$$Z \mid Y = c \sim \mathcal{N}(\mu_c,\ \sigma^2 I_d), \qquad c = 1,\dots,C.$$

The squared Mahalanobis distance reduces to a scaled squared Euclidean distance

$$\|z - \mu_c\|_{\Sigma^{-1}}^2 \;=\; (z-\mu_c)^\top(\sigma^{-2} I_d)(z-\mu_c) \;=\; \frac{1}{\sigma^2}\,\|z-\mu_c\|^2.$$

A closely related geometric structure has been observed empirically in deep classifiers trained with cross-entropy. In particular, Papyan et al. [4] document the *neural collapse* phenomenon: in the

[2] A similar experiment using Fisher's criterion in place of the log-likelihood showed no improvement: the same degeneracy persisted despite the batchwise moment updates.

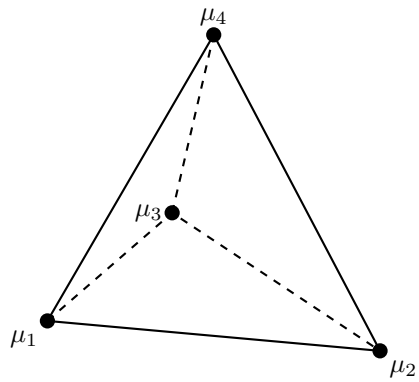


Figure 5: Regular simplex in $\mathbb{R}^3$ for $C = 4$. The four fixed class means μ_c form a tetrahedron with edges of equal length; dashed lines indicate hidden edges in the 2D projection.

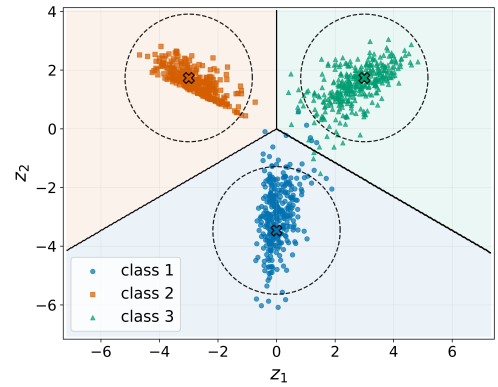


Figure 6: Embeddings $f_\psi(x)$ after training Deep LDA with simplex-constrained means and spherical covariance via MLE. Class-conditional clouds are well separated around the fixed simplex vertices (marked by crosses). Training set accuracy is 99.2%, test set accuracy is 99.4%.

terminal phase of training, the last-layer features collapse to their class means, these means themselves converge to the vertices of a simplex equiangular tight frame, and the classifier weights align with those means so that prediction effectively reduces to a nearest-class-center rule. Our simplex-constrained Deep LDA head can be viewed as an explicit generative realization of a similar latent geometry: the simplex structure is imposed *a priori* through the class-conditional model, and the encoder is trained by maximum likelihood to populate these fixed prototypes.

Discriminant Functions and Likelihood. For our simplex-based LDA head, we can write the unnormalized log density of (Y, Z) as

$$\delta_c(z) \;:=\; \log \pi_c - \frac{1}{2}\left(\frac{\|z-\mu_c\|_2^2}{\sigma^2} + d\log\sigma^2\right), \qquad c = 1, \ldots, C. \tag{6}$$

Up to an additive constant that does not depend on c or on the parameters, $\delta_c(z)$ equals

$$\log \Pr(Y = c) + \log \phi(z; \mu_c, \sigma^2 I_d).$$

Thus, for labeled data $\{(x_i, y_i)\}_{i=1}^n$ and embeddings $z_i = f_\psi(x_i)$, the log-likelihood is (up to an additive constant)

$$\mathcal{L}(\theta) = \frac{1}{n}\sum_{i=1}^{n} \delta_{y_i}(z_i), \tag{7}$$

where $\theta := \{\psi, \pi, \sigma^2\}$, and $\delta_c(z)$ is given by (6).

Unlike the fully free case in Section 2, the class means cannot collapse: their locations and pairwise distances are fixed by the simplex construction. In our implementation, the simplex means $\{\mu_c\}$ are stored as a non-trainable buffer, the log-variance $\log\sigma^2$ and the prior logits are free parameters, and the forward pass returns the $\delta_c(z)$ values in (6) for each sample and class.

Why maximum likelihood. A central goal of this work is to train the Deep LDA head by maximum likelihood (equivalently, minimizing negative log-likelihood). Prior "Deep LDA" style approaches typically optimize modified Fisher-type criteria rather than the LDA likelihood directly [3, 5, 6], or use discriminative objectives defined over Gaussian/Mahalanobis distances [7]. We make this distinction explicit because our contribution is precisely to obtain a well-posed likelihood-based formulation under simple geometric constraints.

Training follows standard end-to-end maximum-likelihood optimization: we fix the simplex means $\{\mu_c\}$, and jointly update the encoder parameters ψ and head parameters (π, σ^2) by minimizing the negative log-likelihood (7) with Adam. Unlike the batchwise re-estimation scheme in Section 2.3, there is no moment-matching: all learnable parameters are updated by backpropagation. Our implementation is available at `https://github.com/zh3nis/simplex-lda`.

Results. On synthetic data generated from (1)–(2), we apply early stopping, as the training accuracy reaches 99.2% after only three epochs. The resulting Deep LDA model yields separated class clusters in the latent space and attains 99.4% test accuracy; see Figure 6.[3]

Having validated the approach on synthetic data, we now proceed to real-data evaluation.

4. Experiments on Real Data

We now evaluate the proposed Deep LDA model of Section 3 on image classification tasks. Across all experiments, the LDA head uses the same structure as described in Section 3. The encoder f_ψ is trained jointly with (π, σ^2) by maximizing the log-likelihood (7). We compare this model to an identical encoder equipped with a conventional linear softmax layer trained via cross-entropy. To visualize the learned geometry, we additionally project the encoder output for a sample of trained embeddings to 2D via PCA.

Datasets. We evaluate on three standard image classification benchmarks. **Fashion-MNIST** [8] contains 70 000 grayscale images of size 28×28 from 10 classes, split into 60 000 training and 10 000 test samples. **CIFAR-10** and **CIFAR-100** [9] consist of 32×32 color images with 10 and 100 classes respectively, each with 50 000 training and 10 000 test images. Following common practice, we apply standard data augmentation to CIFAR-10/100 (random horizontal flips and random crops with padding) and use only per-pixel normalization on Fashion-MNIST.

Encoder architecture. All experiments use the same convolutional encoder architecture, differing only in the number of input channels: one channel for Fashion-MNIST and three channels for CIFAR. The encoder consists of three convolutional blocks with channel widths $\{64, 128, 256\}$. Each block applies two 3×3 $\mathrm{Conv-BN-ReLU}$ layers, with the first two blocks followed by 2×2 max pooling. The final block is followed by global spatial aggregation via adaptive average pooling, producing a 256-dimensional vector. A linear layer projects this vector to the embedding dimension d:

$$\begin{aligned}&\mathrm{Conv}(C_{\mathrm{in}} \to 64) \to \mathrm{Conv}(64 \to 64) \to \mathrm{Pool} \to \mathrm{Conv}(64 \to 128) \to \mathrm{Conv}(128 \to 128)\\ &\quad \to \mathrm{Pool} \to \mathrm{Conv}(128 \to 256) \to \mathrm{Conv}(256 \to 256) \to \mathrm{AdaptiveAvgPool} \to \mathrm{Linear}(256 \to d),\end{aligned}$$

where $C_{\mathrm{in}} = 1$ for Fashion-MNIST and $C_{\mathrm{in}} = 3$ for CIFAR-10/100. This encoder is used unchanged across all experiments, with the output dimension set to $d = C - 1$ for Deep LDA and matched to the classifier input dimension for the softmax baseline.

Heads and training. For each dataset we compare two models built on the same encoder. The first uses a standard linear softmax layer $\mathbb{R}^d \to \mathbb{R}^C$ trained with cross-entropy. The second replaces the softmax with our structurally constrained LDA head trained with MLE and Fisher's objective. Both models are optimized with Adam, batch size 256, for 100 epochs on all three datasets.

Results. Table 1 shows test accuracies averaged over three seeds. On Fashion-MNIST, LDA matches softmax almost exactly. On CIFAR-10, LDA slightly outperforms softmax. On CIFAR-100, the two heads achieve essentially identical accuracy. Overall, our structured LDA is competitive with softmax across datasets while producing substantially cleaner and more structured latent representations.

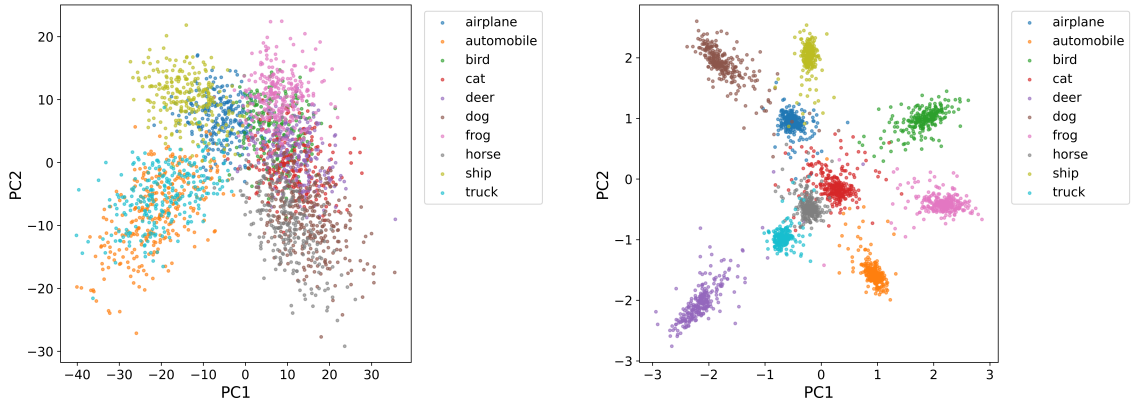
Embedding Geometry. Figure 7 compares PCA projections of the learned embeddings for the softmax and LDA heads. Under the softmax head, the projected embeddings form clusters that partially

[3] In an experiment replacing the log-likelihood with Fisher's generalized eigenvalue objective in the simplex-constrained Deep LDA model produced similar embeddings and nearly identical train & test accuracies.

Table 1: Image classification: Test accuracies (mean $\pm$ std over 3 runs).

Head	Objective	Fashion-MNIST	CIFAR-10	CIFAR-100
Softmax	Cross-entropy loss	$93.61 \pm .38$	$89.07 \pm .20$	$64.67 \pm .30$
Simplex LDA	Log-likelihood	$93.40 \pm .32$	$90.41 \pm .04$	$64.61 \pm .12$
Simplex LDA	Fisher's objective	$92.94 \pm .07$	$90.97 \pm .37$	$64.71 \pm .15$

overlap. In contrast, the LDA head yields very tight class clusters that remain separated even after projection on 2D. This illustrates that LDA imposes a strong geometric prior: maximizing between-class separation while shrinking within-class variance, leading to a much more structured and discriminative representation.



(a) Softmax. (b) Simplex LDA.

Figure 7: PCA projections of CIFAR-10 embeddings learned with two classification heads.

Effect of latent dimension. In the main experiments we set the latent dimension to $d = C - 1$, which is the smallest dimension that can accommodate a $(C-1)$-dimensional simplex. To check the effect of the latent dimension, we ran an additional experiment on CIFAR-100 by varying $d \in \{150, 200, \ldots, 500\}$. Figure 8 reports test accuracy as a function of d. Softmax accuracy is almost insensitive to the choice of d, whereas Deep LDA exhibits a mild decreasing trend as d increases: enlarging the latent space offers no benefit and can slightly hurt performance. Overall, both heads remain within a narrow accuracy band for all tested dimensions, and Deep LDA is competitive with softmax for moderate d. This suggests that the generative head actually prefers compact latent spaces and supports our use of the minimal dimension $d = C - 1$ as a parsimonious and empirically favorable setting.

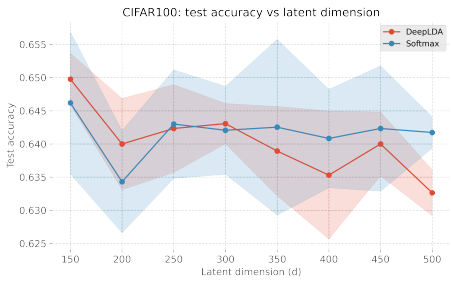


Figure 8: CIFAR-100 test accuracy as a function of latent dimension d for the Softmax and LDA heads (mean ± 2 standard deviations over three runs).

Covariance structure ablation. Our default head uses a spherical shared covariance, $\Sigma = \sigma^2 I$, to keep the head lightweight and comparable in parameter count to a linear softmax layer at the same embedding dimension. To test whether additional covariance flexibility helps, we train simplex-based Deep LDA with spherical, diagonal, and full covariances on FashionMNIST, CIFAR-10, and CIFAR-100 (3 runs per setting). Table 2 reports mean $\pm$ std test accuracy. Neither diagonal nor full covariance improves performance; in fact, full covariance significantly degrades accuracy on CIFAR-100. This suggests that the encoder can adapt to a spheri-

cal covariance structure, while additional covariance degrees of freedom may introduce optimization/estimation difficulties.

Table 2: Ablation of the shared covariance parameterization in simplex-based Deep LDA (mean $\pm$ std over 3 runs).

Covariance	FashionMNIST	CIFAR-10	CIFAR-100
Spherical	93.38 ± 0.06	90.49 ± 0.26	65.20 ± 0.22
Diagonal	93.32 ± 0.07	90.31 ± 0.26	65.02 ± 0.09
Full	93.31 ± 0.11	89.91 ± 0.31	59.64 ± 0.71

Stronger off-the-shelf backbone (ResNet-50). We additionally evaluate our approach with a stronger, widely used backbone. Specifically, we fine-tune an ImageNet-pretrained ResNet-50 encoder and compare (i) our Simplex LDA likelihood-trained head against (ii) a standard Softmax head trained with cross-entropy. To ensure a controlled comparison, we use the *same* encoder, the *same* embedding dimension for both heads, setting $D = C - 1$. Table 3 reports results over 3 runs (mean $\pm$ std). Overall, with a substantially stronger backbone, the simplex-constrained likelihood-trained head remains competitive with the conventional softmax baseline across datasets.

Table 3: Fine-tuning a pretrained ResNet-50 backbone (mean $\pm$ std over 3 runs).

Head	FashionMNIST	CIFAR-10	CIFAR-100
Softmax	93.99 ± 0.17	96.30 ± 0.14	80.85 ± 0.19
Simplex LDA	94.22 ± 0.08	96.35 ± 0.19	80.45 ± 0.22

Beyond images: text classification with a pretrained Transformer. To demonstrate that the proposed head generalizes beyond image benchmarks, we fine-tune a pretrained Transformer (BERT-base-uncased [10]) for text classification on AG News (4 classes; 120k train / 7.6k test)[4] and CLINC150 (151 classes; 15k train / 5.5k test) [11]. We compare (i) a standard linear Softmax head to (ii) our SimplexLDA head, using an architecture-matched setup with an embedding projection to $d = C - 1$. Both models are fine-tuned with AdamW (learning rate 2×10^{-5}, batch size 32) for 3 epochs, and results are averaged over 3 random seeds. Table 4 reports test accuracy (mean $\pm$ std). Overall, SimplexLDA remains competitive with Softmax on AG News and improves performance on CLINC150, supporting the generality and scalability of the proposed likelihood-trained head.

Table 4: Text classification: test accuracy (mean $\pm$ std over 3 runs).

Head	AG News	CLINC150
Softmax	94.40 ± 0.07	87.11 ± 0.56
SimplexLDA	94.37 ± 0.37	88.12 ± 0.34

5. Related Work

Classical Linear Discriminant Analysis dates back to Fisher [1] and Rao [2], who modeled each class as a Gaussian with a shared covariance matrix, yielding linear decision boundaries. Adapting LDA to deep networks has inspired several research directions, most of which modify the classical objective rather than preserve its generative form.

Deep Fisher-style objectives. DeepLDA [3] optimizes Fisher-style generalized eigenvalue criterion, with related extensions for person re-identification [5] and regularized scatter objectives [6].

[4] `http://groups.di.unipi.it/~gulli/AG_corpus_of_news_articles.html`

In its naive form, maximizing the mean eigenvalue of the LDA problem causes the network to favor trivial solutions that magnify a single dominant eigenvalue while neglecting poorly separated classes; this behavior was explicitly observed by Dorfer et al. [3], who therefore replace the original objective with a heuristic that maximizes only the k smallest eigenvalues (their Eqs. (8)–(9)). As a result, these methods promote between/within-class separation but no longer correspond to maximum likelihood under a generative LDA model, in contrast to our constrained Deep LDA formulation.

Fixed classifiers and simplex-structured heads. A complementary line of work studies *fixing* the final classifier layer to impose a prescribed geometry. Hoffer et al. [12] argue that the last linear layer can be fixed (up to scale) with little loss in accuracy. Building on this idea, Pernici et al. [13] fix class vectors to the vertices of regular polytopes (including the simplex) to obtain maximally separated and stationary embeddings. Motivated by the neural collapse phenomenon [4], several works explicitly use a regular simplex (simplex equiangular tight frame) structure for the classifier and/or class centers, either by analyzing the induced optimization landscape [14], by demonstrating that fixing a simplex ETF classifier can induce neural collapse even under class imbalance [15], or by proposing to optimize *towards* the nearest simplex ETF rather than hard-fixing one for improved stability and convergence [16]. Importantly, in these works the simplex (or polytope) structure is primarily leveraged to improve *discriminative* training with losses such as cross-entropy or squared error.

Our contribution. Prior deep LDA variants either modify Fisher's criterion or use Gaussian distances in a discriminative loss, and fixed/simplex classifier works use simplex structure mainly as an inductive bias for discriminative training. In contrast, we retain the generative LDA model and train it by maximum likelihood. In our setting, the regular simplex plays a different role: it acts as a natural constraint that makes end-to-end likelihood training of Deep LDA well-posed, preventing the degeneracies that arise under an unconstrained model (Section 2). Together with a spherical shared covariance, these constraints yield an interpretable, well-behaved generative classifier whose MLE training produces separated class clusters without collapse.

6. Conclusion

We revisited Deep Linear Discriminant Analysis from a likelihood perspective and showed that the unconstrained deep model is ill-posed under maximum-likelihood training: the encoder collapses embeddings, some of the class means drift toward each other, and the shared covariance shrinks toward singularity, yielding high likelihood but poor discrimination. A simple geometric constraint—fixing the class means to a regular simplex and enforcing a spherical shared covariance—removes the pathological tendency of the class clusters to collapse or heavily overlap, and improves the classification performance. On synthetic data and image benchmarks, the resulting Deep LDA classifier yields clean, interpretable latent spaces with well-separated clusters and accuracy competitive with softmax, illustrating that mild geometric structure can restore the viability of likelihood-based training for deep discriminant analysis. We additionally verified that the head transfers beyond images by fine-tuning a pretrained Transformer on text classification (AG News, CLINC150), where it remains competitive with Softmax.

Acknowledgements

This work was supported by the Science Committee of the Ministry of Science and Higher Education of the Republic of Kazakhstan (Grant No. AP26198325).

Reproducibility. Code to reproduce our experiments is available at `https://github.com/zh3nis/simplex-lda`.

AI Assistance. We used an AI-based tool for editorial polishing of the text; the authors take full responsibility for the content.

References

[1] Ronald A Fisher. The use of multiple measurements in taxonomic problems. *Annals of eugenics*, 7(2):179–188, 1936.

[2] C. Radhakrishna Rao. The utilization of multiple measurements in problems of biological classification. *Journal of the Royal Statistical Society. Series B (Methodological)*, 10(2):159–203, 1948. ISSN 00359246.

[3] Matthias Dorfer, Rainer Kelz, and Gerhard Widmer. Deep linear discriminant analysis. In Yoshua Bengio and Yann LeCun, editors, *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, 2016.

[4] Vardan Papyan, XY 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.

[5] Lin Wu, Chunhua Shen, and Anton van den Hengel. Deep linear discriminant analysis on fisher networks: A hybrid architecture for person re-identification. *Pattern Recognit.*, 65:238–250, 2017.

[6] Hongwei Chen and Wen Lu. Regularized deep linear discriminant analysis. *CoRR*, abs/2105.07129, 2021.

[7] Tianyu Pang, Chao Du, and Jun Zhu. Max-mahalanobis linear discriminant analysis networks. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, 2018.

[8] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

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

[10] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pages 4171–4186, 2019.

[11] 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)*, 2019.

[12] Elad Hoffer, Itay Hubara, and Daniel Soudry. Fix your classifier: The marginal value of training the last weight layer. In *International Conference on Learning Representations (ICLR)*, 2018.

[13] Federico Pernici, Matteo Bruni, Claudio Baecchi, and Alberto Del Bimbo. Regular polytope networks. *IEEE Transactions on Neural Networks and Learning Systems*, 33(9):4373–4387, 2022.

[14] 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*, 2021.

[15] 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*, 2022.

[16] Evan Markou, Thalaiyasingam Ajanthan, and Stephen Gould. Guiding neural collapse: Optimising towards the nearest simplex equiangular tight frame. In *Advances in Neural Information Processing Systems*, 2024.