HYPERSPHERICAL FILTERING FOR ONLINE CLASSIFICATION UNDER DRIFT

David Boekestijn
UvA-Bosch Delta Lab
University of Amsterdam
Amsterdam, The Netherlands
david.boekestijn@student.uva.nl

Mona Schirmer
UvA-Bosch Delta Lab
University of Amsterdam
Amsterdam, The Netherlands
m.c.schirmer@uva.nl

ABSTRACT

In online learning, a model processes a nonstationary data stream by alternating between training and prediction steps. Recent work has employed a Gaussian Kalman filter with learnable forgetting coefficient to adapt last-layer classifier weights under sudden distribution shift. Gaussian models assume Euclidean geometry, while softmax heads (especially with normalized features) are primarily directional. We investigate this limitation by modeling each class weight on the hypersphere with a von Mises–Fisher (vMF) posterior. On various drift tasks with pretrained backbones, the vMF filter consistently improves negative log-likelihood, expected calibration error, and Brier score compared to Gaussian Kalman filtering, at the cost of a small reduction in average online accuracy.

1 INTRODUCTION

Deployed models often face distribution shift, yet in many scenarios, only lightweight adaptation is feasible due to cost constraints. A common approach is to keep a backbone feature extractor $\phi(\,\cdot\,)$ fixed and update only the final layer online. In this regime, Titsias et al. (2023) proposed a Bayesian approach that treats the readout weights as latent states and applied Kalman filtering with a learnable forgetting coefficient γ_n to handle abrupt nonstationarity.

A key modeling choice in this Gaussian head filter is Euclidean geometry; each class weight evolves in $\mathbb{R}^d$ with a Gaussian posterior. However, the decision rule of a softmax head is primarily *directional* when features are normalized (as in many modern pipelines). In this setting, uncertainty in $\|w\|$ is largely irrelevant to classification, yet a Gaussian posterior necessarily allocates probability mass to norms, while being able to express uncertainty by shrinking means toward the origin.

We propose a hyperpsherical analogue of supervised online Kalman head learning.[1] We model each class weight on the unit hypersphere $\mathbb{S}^{d-1}$ and represent uncertainty with a von Mises–Fisher (vMF) posterior. We retain the learnable forgetting mechanism of the Gaussian filter by letting γ_n control the strength of spherical diffusion in the predict step. Since the softmax likelihood is nonconjugate to the vMF prior, we approximate the update by a conjugate surrogate, which yields a closed-form vMF update and is empirically more stable than moment matching the true posterior when the predictive softmax is sharp.

We evaluate on sudden drift tasks using pretrained backbones (ResNet-18, CLIP) and test if and under what types of drift the vMF filter might improve over the Gaussian. We use Brier score and average online accuracy as primary metrics and observe that the vMF filter consistently improves probabilistic performance at the cost of accuracy. Our contributions are as follows:

- A vMF state-space model and Kalman-style filtering recursions for online last-layer adaptation on $\mathbb{S}^{d-1}$, with a learnable forgetting coefficient γ_n.
- Practical closed-form approximations for the nonconjugate predict and update steps.
- An empirical study on sudden drift tasks, comparing against Gaussian Kalman head filtering using Brier, ECE, prequential NLL, and online accuracy.

[1] Code: https://github.com/dboekestijn/hyperspherical-filtering

2 RELATED WORK

Titsias et al. (2023) model last-layer weights with a state-space model (SSM) and perform efficient Kalman-style recursions, learning a forgetting coefficient to handle nonstationarity. We follow the same "frozen backbone + adapted head" paradigm and replace the Euclidean Gaussian posterior with a directional posterior on $\mathbb{S}^{d-1}$. Schirmer et al. (2024) propose STAD, an SSM approach to *unsupervised* test-time adaptation that tracks time-evolving class prototypes and includes Gaussian and vMF instantiations. Our setting differs in that we (i) use supervised online updates (labels available), and (ii) adapt discriminative head weights for a fixed representation rather than unlabeled prototypes. vMF-based filtering has been studied for spherical states in signal processing and robotics (e.g., Kurz et al. (2016); Tronarp et al. (2018)). Separately, hyperspherical classification objectives with normalized features/weights (e.g., ArcFace (Deng et al., 2019)) motivate directional modeling when decisions depend on angular similarity. We connect these ideas by applying directional filtering to supervised online adaptation of neural classifier heads under drift.

3 PROBLEM SETTING

We consider online multi-class classification on a stream (x_n, y_n), with input data $x_n \in \mathcal{X}$ and labels $y_n \in \{0,1\}^K$, $\sum_{k=1}^K y_{n,k} = 1$, with K the number of classes. At each step n, the learner (i) makes a prequential prediction for y_n given past observations and the current input x_n, then (ii) updates its state after observing the true label. Following the setup by Titsias et al. (2023), we use a two-part model: (i) a fixed backbone $\phi(\,\cdot\,;\theta)$ that produces a features $\phi_n \coloneqq \phi(x_n;\theta) \in \mathbb{R}^d$, and (ii) a time-varying linear softmax head with weights $W_n = (w_{n,1}, \ldots, w_{n,K})$. We focus on the "frozen backbone + adapted head" regime and keep θ fixed throughout.

4 BACKGROUND: GAUSSIAN ONLINE LEARNER

Since the comparison in our paper is between Euclidean and hyperspherical state models, our primary setting uses ℓ_2-normalized features $\tilde{\phi}_n \coloneqq \phi_n/\|\phi_n\| \in \mathbb{S}^{d-1}$. Given W_n and $\tilde{\phi}_n$, logits are $z_{n,k} = w_{n,k}^\intercal \tilde{\phi}_n$ and the label is modeled by the softmax likelihood

$$p(y_{n,k} = 1 \mid W_n) = \sigma_k(W_n^\intercal \tilde{\phi}_n) \coloneqq \frac{\exp(w_{n,k}^\intercal \tilde{\phi}_n)}{\sum_{j=1}^K \exp(w_{n,j}^\intercal \tilde{\phi}_n)}.$$

All Bayesian methods predict via the Bayesian predictive density

$$p(y_{n,k} = 1 \mid y_{1:n-1}) = \int p(y_{n,k} = 1 \mid W_n)\, p(W_n \mid y_{1:n-1})\, dW_n,$$

which is intractable for the models considered here. Following Titsias et al. (2023), we therefore approximate the density via Monte Carlo sampling:

$$p(y_{n,k} = 1 \mid y_{1:n-1}) \approx \frac{1}{S} \sum_{s=1}^S \sigma_k((W_n^{(s)})^\intercal \tilde{\phi}_n), \qquad W_n^{(s)} \sim p(W_n \mid y_{1:n-1}),$$

and predict the label $k_n^* = \arg\max_k p(y_{n,k} = 1 \mid y_{1:n-1})$. The posterior over W_n is updated after observing y_n.

We maintain posteriors over W_n and update them with Bayesian filtering recursions. As in Titsias et al. (2023), a key mechanism is a learnable forgetting rate $\gamma_n \in (0,1]$ in the transition step, which enables explicit, gradient-based control of adaptation speed under nonstationary. Following Titsias et al. (2023), each class weight $w_{n,k}$ evolves in $\mathbb{R}^d$ via a first-order Gaussian Markov model:

$$\begin{aligned} p(w_{0,k}) &= \mathcal{N}(0, \sigma_w^2 I), \\ p(w_{n,k} \mid w_{n-1,k}) &= \mathcal{N}(\gamma_n w_{n-1,k}, (1-\gamma_n^2)\sigma_w^2 I), \quad n \geq 1. \end{aligned}$$

The predictive and updated posteriors have class-specific means and a shared covariance matrix:

$$p(W_n \mid y_{1:n-1}) = \prod_{k=1}^K \mathcal{N}(m_{n,k}^-, A_n^-), \quad p(W_n \mid y_{1:n}) = \prod_{k=1}^K \mathcal{N}(m_{n,k}, A_n),$$

where $m_{n,k}^{(-)}$ and $A_n^{(-)}$ follow standard Kalman filtering update recursions; we refer to Titsias et al. (2023) for the exact expressions. To achieve this, since the softmax likelihood is nonconjugate to a Gaussian posterior, Titsias et al. (2023) use a Gaussian surrogate on the one-hot targets: $q(y_n \mid W_n) = \prod_{k=1}^{K} \mathcal{N}(y_{n,k} \mid w_{n,k}^{\intercal}\phi_n, \sigma^2)$, yielding a closed-form Kalman update (a standard trick also used in Gaussian process classification (Rasmussen & Williams, 2005)). As in Titsias et al. (2023), we learn γ_n online via an empirical Bayes SGD update on the log predictive density:

$$\gamma_n \leftarrow \gamma_n + \rho_n \nabla_{\gamma_n} \log p(y_n \mid y_{1:n-1}),$$

using the Monte Carlo predictive estimate. We enforce $\gamma_n \in (0, 1]$ via the reparameterization $\gamma_n = \exp(-\frac{1}{2}\delta_n)$ with $\delta_n \geq 0$ enforced by clipping (Titsias et al., 2023).

5 Method: Hyperspherical Online Learner

We replace the Euclidean SSM with a directional model on $\mathbb{S}^{d-1}$. We assume $\tilde{\phi}_n \in \mathbb{S}^{d-1}$ and $w_{n,k} \in \mathbb{S}^{d-1}$. Uncertainty over directions is represented by a von Mises–Fisher (vMF) distribution (Mardia & Jupp, 2009):

$$\mathrm{vMF}(x \mid \mu, \kappa) = C_d(\kappa) \exp(\kappa \mu^{\intercal} x), \qquad \mu \in \mathbb{S}^{d-1},\ \kappa \geq 0,$$

with first moment

$$\mathbb{E}[x] = A_d(\kappa)\mu, \qquad A_d(\kappa) \coloneqq \frac{I_{d/2}(\kappa)}{I_{d/2-1}(\kappa)}.$$

We place a hyperspherical Markov model on each class weight:

$$\begin{aligned} p(w_{0,k}) &= \mathrm{vMF}(\mu_0, \kappa_0), \\ p(w_{n,k} \mid w_{n-1,k}) &= \mathrm{vMF}(w_{n-1,k}, \kappa_{\gamma_n,k}), \qquad n \geq 1, \end{aligned} \tag{1}$$

where $\kappa_{\gamma_n,k}$ controls diffusion on the sphere. We use assumed density filtering (ADF) with a factorized vMF predictive posterior

$$p(W_n \mid y_{1:n-1}) = \prod_{k=1}^{K} \mathrm{vMF}(\mu_{n,k}^{-}, \kappa_{n,k}^{-}),$$

matched by a forward-KL projection. Choosing $\kappa_{\gamma_n,k}$ such that $A_d(\kappa_{\gamma_n,k}) = \gamma_n$ gives the desired moment contraction as in the Gaussian case, $\mathbb{E}[W_n] = \gamma_n \mathbb{E}[W_{n-1}]$, so that the resulting ADF predict step becomes

$$\mu_{n,k}^{-} = \mu_{n-1,k}, \qquad \kappa_{n,k}^{-} = A_d^{-1}(\gamma_n A_d(\kappa_{n-1,k})).$$

We allow class-specific concentrations $\kappa_{n,k}$ so that the filter can represent class-dependent uncertainty: frequently updated classes become more certain, while inactive classes gradually diffuse. This remains a fair comparison to the Gaussian baseline, where class-specific means $m_{n,k}$ already allow each class weight and its effective precision (the norm) to evolve independently; the vMF model captures the same effect by scaling the concentrations. This yields $\mathcal{O}(2dK)$ parameters (directions plus concentration), versus $\mathcal{O}(d^2 + dK)$ when tracking full covariances. In high-dimensional feature spaces the $\mathcal{O}(d^2)$ term can dominate, making the hyperspherical formulation substantially more computationally efficient.

The exact posterior $p(W_n \mid y_{1:n}) \propto p(y_{n,k} = 1 \mid W_n)\, q(W_n \mid y_{1:n-1})$ is nonconjugate due to the softmax likelihood. We considered (i) self-normalized importance sampling (SNIS) moment matching and (ii) conjugate likelihood surrogates. While SNIS is principled, it can be numerically unstable when the predictive softmax is sharp. We therefore investigate the use of conjugate surrogates, analogous in spirit to Titsias et al. (2023). The simplest of these is an improper one that replaces the softmax likelihood with a per-class exponential-family term linear in $w_{n,k}^{\top}\tilde{\phi}_n$:

$$p_k(y_n \mid w_{n,k}) = \exp(\mathbb{I}\{y_n = k\}\phi^{\intercal} w_{n,k}),$$

with $\mathbb{I}\{\cdot\}$ the indicator function. Then, the likelihood term is simply added to the vMF natural parameter:

$$\eta_{n,k} = \eta_{n,k}^{-} + \mathbb{I}\{y_n = k\}\phi_n, \qquad q(w_{n,k} \mid y_{1:n}) = \mathrm{vMF}(\mu_{n,k}, \kappa_{n,k}),$$

with $\mu_{n,k} = \eta_{n,k}/\|\eta_{n,k}\|$, $\kappa_{n,k} = \|\eta_{n,k}\|$. Others are presented in App. A.2.

6 EXPERIMENTS

Setup We evaluate supervised online head adaptation under distribution shift. Our main experiment follows the Split-CIFAR-100 protocol of Titsias et al. (2023) using an ImageNet-pretrained ResNet-18 backbone. In addition, we evaluate on the Yearbook dataset (Ginosar et al., 2015), which exhibits gradual temporal drift. To test robustness across feature representations, we repeat all experiments with a CLIP backbone (Radford et al., 2021).

In all cases, the backbone is kept fixed and only the final-layer head is adapted online. To isolate the effect of the state-space geometry, we use ℓ_2-normalized features for *both* methods. For Split-CIFAR-100, the stream is divided into ten periods using CIFAR-100 coarse labels so that each period contains 6000 images covering ten disjoint classes (Lee et al., 2020; Titsias et al., 2023). For the Yearbook dataset, the images are shuffled within-year and streamed in ascending year-order.

We compare the Gaussian Kalman filter of Titsias et al. (2023) with our vMF filter. Both methods learn the forgetting coefficient γ_n online as described in Sec. 4 and use the same Monte Carlo predictive approximation. For the vMF filter, we find the Laplace projection surrogate likelihood to give the best results (see App. A.2. We report average online accuracy and Brier score as primary metrics.

Results Fig. 1 shows streaming performance on Split-CIFAR-100. The vMF filter consistently improves probabilistic performance, achieving a lower Brier score than the Gaussian baseline throughout the stream. This indicates better-calibrated predictive uncertainty under abrupt distribution changes. Intuitively, uncertainty in the vMF model is expressed by reduced concentration (approaching a uniform distribution on the sphere) rather than shrinking the weight vector toward the origin as in Euclidean space.

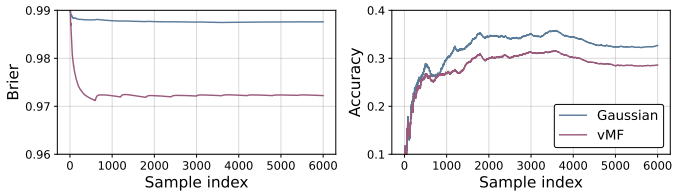

Figure 1: Comparison of Gaussian and vMF filters on Split-CIFAR-100 with an ImageNet-pretrained ResNet-18 backbone.

At the same time, we observe a drop in online accuracy after the first period. This behavior is expected in this adversarial Split-CIFAR-100 protocol, where each period introduces disjoint classes, requiring rapid allocation of probability mass to previously unseen labels. In the Gaussian Kalman filter, resetting via mean shrinkage can accelerate this reallocation, whereas on the sphere the predictive mean direction is preserved and uncertainty is expressed only through concentration. This explanation is supported by the observation that average online accuracy is comparable between the methods during the first period.

Table 1: Comparison of Gaussian and vMF filters on multiple dataset-backbone combinations across three seeds. Due to consistency, standard deviations only reported for the first dataset (accuracy).

		ResNet-18		CLIP	
Dataset	Method	Acc.	Brier	Acc.	Brier
Split-CIFAR-100	Gaussian	0.32 ($\pm$0.01)	0.99	0.76 ($\pm$0.01)	0.99
	vMF	0.25 ($\pm$0.0)	0.97	0.56 ($\pm$0.01)	0.97
Yearbook	Gaussian	0.71	0.49	0.98	0.47
	vMF	0.54	0.5	0.81	0.47

Table 1 shows that this pattern persists across datasets and backbones. The vMF filter consistently improves the Brier score while the Gaussian filter achieves higher online accuracy.

7 DISCUSSION AND CONCLUSION

We replace Gaussian Kalman head filtering with a von Mises–Fisher posterior on $\mathbb{S}^{d-1}$ for online adaptation. On various dataset and backbone combinations testing disjoint label shift and gradual temporal drift, this geometry-matched filter improves calibration with a fixed backbone. However, in our experiments, the vMF filter is unable to improve on accuracy, where the Euclidean shrinkage seems to handle various types of drift better.

8 ACKNOWLEDGMENT

We thank Hany Abdulsamad for helpful discussions. This project was generously supported by the Bosch Center for Artificial Intelligence.

REFERENCES

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.

Shiry Ginosar, Kate Rakelly, Sarah Sachs, Brian Yin, and Alexei A Efros. A century of portraits: A visual historical record of american high school yearbooks. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pp. 1–7, 2015.

Gerhard Kurz, Igor Gilitschenski, and Uwe D Hanebeck. Unscented von mises–fisher filtering. *IEEE Signal Processing Letters*, 23(4):463–467, 2016.

Soochan Lee, Junsoo Ha, Dongsu Zhang, and Gunhee Kim. A neural dirichlet process mixture model for task-free continual learning. *arXiv preprint arXiv:2001.00689*, 2020.

Kanti V Mardia and Peter E Jupp. *Directional statistics*. John Wiley & Sons, 2009.

Christian Naesseth, Francisco Ruiz, Scott Linderman, and David Blei. Reparameterization Gradients through Acceptance-Rejection Sampling Algorithms. In Aarti Singh and Jerry Zhu (eds.), *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, volume 54 of *Proceedings of Machine Learning Research*, pp. 489–498. PMLR, 20–22 Apr 2017. URL https://proceedings.mlr.press/v54/naesseth17a.html.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PmLR, 2021.

Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning*. The MIT Press, 11 2005. ISBN 9780262256834. doi: 10.7551/mitpress/3206.001.0001. URL https://doi.org/10.7551/mitpress/3206.001.0001.

Mona Schirmer, Dan Zhang, and Eric Nalisnick. Test-time adaptation with state-space models. In *ICML 2024 Workshop on Structured Probabilistic Inference {\&} Generative Modeling*, 2024.

Javier Segura. Bounds for ratios of modified bessel functions and associated turán-type inequalities. *Journal of Mathematical Analysis and Applications*, 374(2):516–528, 2011. ISSN 0022-247X. doi: https://doi.org/10.1016/j.jmaa.2010.09.030. URL https://www.sciencedirect.com/science/article/pii/S0022247X10007742.

Michalis K Titsias, Alexandre Galashov, Amal Rannen-Triki, Razvan Pascanu, Yee Whye Teh, and Jorg Bornschein. Kalman filter for online classification of non-stationary data. *arXiv preprint arXiv:2306.08448*, 2023.

Filip Tronarp, Roland Hostettler, and Simo Särkkä. Continuous-discrete von mises-fisher filtering on s 2 for reference vector tracking. In *2018 21st International Conference on Information Fusion (FUSION)*, pp. 1345–1352. IEEE, 2018.

William J Wilkinson, Simo Särkkä, and Arno Solin. Bayes-newton methods for approximate bayesian inference with psd guarantees. *Journal of Machine Learning Research*, 24(83):1–50, 2023.

Andrew TA Wood. The simulation of spherical distributions in the fisher-bingham family. *Communications in Statistics-Simulation and Computation*, 16(3):885–898, 1987.

A Derivations for von Mises–Fisher model

A.1 Predict step

Let the joint prior over the weights $W_{n-1} \in \mathbb{S}^{(m-1)\times K}$ be given as a product of von Mises–Fisher (vMF) densities:

$$p(W_{n-1} \mid y_{1:n-1}) = \prod_{k=1}^{K} \underbrace{\mathrm{vMF}(\mu_{n-1,k}, \kappa_{n-1})}_{p(w_{n-1,k}|y_{1:n-1})}. \tag{2}$$

For completeness' sake, the below derivation is for shared concentrations and thus omits the class-index k for the concentrations κ_n. The derivation is simpler (can be factorized) and carries out in largely the same way for class-specific concentrations $\kappa_{n,k}$.

The predictive posterior is obtained by multiplying the prior with the transition density (Eq. (1)) and integrating out the prior's random variables:

$$\begin{aligned}
p \coloneqq p(W_n \mid y_{1:n-1}) &= \int_{\mathbb{S}^{(d-1\times K)}} p(W_n \mid W_{n-1})\, p(W_{n-1} \mid y_{1:n-1})\, dW_{n-1} \\
&= \int_{\mathbb{S}^{(d-1\times K)}} \left[\prod_{k=1}^{K} \mathrm{vMF}(w_{n,k} \mid w_{n-1,k}, \kappa_{\gamma_n})\, \mathrm{vMF}(w_{n-1,k} \mid \mu_{n-1,k}, \kappa_{n-1})\right] dW_{n-1} \\
&\propto \prod_{k=1}^{K} \int_{\mathbb{S}^{d-1}} \exp(\kappa_{\gamma_n} w_{n-1,k}^{\intercal} w_{n,k}) \exp(\kappa_{n-1} \mu_{n-1,k}^{\intercal} w_{n-1,k})\, dw_{n-1,k} \\
&= \prod_{k=1}^{K} \int_{\mathbb{S}^{d-1}} \underbrace{\exp((\kappa_{\gamma_n} w_{n,k} + \kappa_{n-1}\mu_{n-1,k})^{\intercal} w_{n-1,k})}_{\propto \mathrm{vMF}(w_{n-1,k}|m_{n,k},k_{n,k})}\, dw_{n-1,k} \\
&= \prod_{k=1}^{K} \underbrace{\frac{1}{C_d(k_{n,k})}}_{\propto p(w_{n,k}|y_{1:n-1})},
\end{aligned}$$

where, using $R_{n,k} = \kappa_{\gamma_n} w_{n,k} + \kappa_{n-1}\mu_{n-1}$, the integrand is proportional to a vMF with mean and concentration parameters given by $m_{n,k} = R_{n,k}/\|R_{n,k}\|$ and $k_{n,k} = \|R_{n,k}\|$, respectively.

The last equation follows immediately from the fact that the integrand is an unnormalized vMF, yielding exactly the reciprocal of the vMF normalizing constant $C_d(k_{n,k})$ after integration. However, the resulting density is intractable, as no closed-form distributions satisfy this highly nonlinear form.

To keep the filtering efficient, we fit a proper factorized vMF

$$q \coloneqq q(W_n \mid y_{1:n-1}) = \prod_{k=1}^{K} \mathrm{vMF}(\mu_{n,k}^{-}, \kappa_n^{-})$$

on the obtained predictive posterior (p) by minimizing the forward-KL divergence of p from q:[2]

$$\begin{aligned}
\mathrm{KL}(p \parallel q) &\coloneqq \mathrm{KL}(p(W_n \mid y_{1:n-1}) \parallel q(W_n \mid y_{1:n-1})) \\
&= \int \log\left(\frac{p(W_n \mid y_{1:n-1})}{q(W_n \mid y_{1:n-1})}\right) p(W_n \mid y_{1:n-1})\, dW_n \\
&\propto \int \left[\sum_{k=1}^{K} -\log C_d(\|R_{n,k}^{-}\|) - \log C_d(\kappa_n^{-}) - \kappa_n^{-} (\mu_{n,k}^{-})^{\intercal} w_{n,k}\right] p(W_n \mid y_{1:n-1})\, dW_n.
\end{aligned}$$

[2]For notational brevity, we hereafter omit the domains of integration; they are implied by the variable of integration.

This forward-KL divergence can be minimized analytically as follows. First, since $\mu^-_{n,k}$ must lie on the unit hypersphere—i.e., must satisfy $\|\mu^-_{n,k}\| = 1$—we add a Lagrange term $\frac{\lambda}{2}(1 - (\mu^-_{n,k})^\intercal \mu_{n,k})$ (with Lagrange multiplier λ) to the forward-KL divergence, to get the Lagrangian:

$$\mathcal{L}(\mu^-_{n,k}, \kappa^-_n, \lambda; W_n) \coloneqq \mathrm{KL}(p \parallel q) + \frac{\lambda}{2}(1 - (\mu^-_{n,k})^\intercal \mu^-_{n,k}).$$

Then, we minimize this Lagrangian by setting all its partial derivatives to zero.

For λ:

$$\begin{aligned} & \frac{\partial}{\partial \lambda} \mathcal{L}(\mu^-_{n,k}, \kappa^-_n, \lambda; W_n) = 0 \\ \implies \quad & 1 - (\mu^-_{n,k})^\intercal \mu^-_{n,k} = 0, \end{aligned}$$

which gives us back the constraint we specified. This means λ is a free multiplier.

For $\mu^-_{n,k}$:

$$\begin{aligned} & \nabla_{\mu^-_{n,k}} \mathcal{L}(\mu^-_{n,k}, \kappa^-_n, \lambda; W_n) = 0 \\ \implies & \int [\kappa^-_n w_{n,k}]\, p(w_{n,k} \mid y_{1:n-1})\, dw_{n,k} - \lambda \mu^-_{n,k} = 0 \\ \implies & \kappa^-_n\, \mathbb{E}_p[w_{n,k}] - \lambda \mu^-_{n,k} = 0 \\ \implies & \mu^-_{n,k} = \frac{\kappa^-_n}{\lambda}\, \mathbb{E}_p[w_{n,k}], \end{aligned} \tag{3}$$

with

$$\begin{aligned} \mathbb{E}_p[w_{n,k}] \coloneqq \mathbb{E}_{p(w_{n,k}|y_{1:n-1})}[w_{n,k}] &\overset{\text{(LIE}^3\text{)}}{=} \mathbb{E}_{p(w_{n-1,k}|y_{1:n-1})}\left[\mathbb{E}_{p(w_{n,k}|w_{n-1,k})}[w_{n-1,k}]\right] \\ &\overset{\text{(i)}}{=} \mathbb{E}_{p(w_{n-1,k}|y_{1:n-1})}\left[A_d(\kappa_{\gamma_n}) w_{n-1,k}\right] \\ &= A_d(\kappa_{\gamma_n})\, \mathbb{E}_{p(w_{n-1,k}|y_{1:n-1})}[w_{n-1,k}] \\ &\overset{\text{(ii)}}{=} A_d(\kappa_{\gamma_n}) A_d(\kappa_{n-1}) \mu_{n-1,k}, \end{aligned} \tag{4}$$

where (i) $p(w_{n,k} \mid w_{n-1,k})$ is defined in Eq. (1), and (ii) $p(w_{n-1,k} \mid y_{1:n-1})$ in Eq. (2). Combining Eq. (3) with the fact that λ is a free Lagrange multiplier, unit norm of $\mu^-_{n,k}$ is satisfied when $\frac{\lambda}{\kappa^-_n} = \|\mathbb{E}_p[w_{n,k}]\|$. So:

$$\begin{aligned} \mu^-_{n,k} &= \frac{\mathbb{E}_p[w_{n,k}]}{\|\mathbb{E}_p[w_{n,k}]\|} \\ \mu^-_{n,k} &= \frac{A_d(\kappa_{\gamma_n}) A_d(\kappa_{n-1}) \mu_{n-1,k}}{\|A_d(\kappa_{\gamma_n}) A_d(\kappa_{n-1}) \mu_{n-1,k}\|} \\ &\overset{\|\mu_{n-1,k}\|=1}{=} \mu_{n-1,k}. \end{aligned}$$

In other words, the forward-KL-minimizing mean for the predictive posterior simply equals the prior mean.

[3] Law of Iterated Expectation.

Last, for κ_n^-:

$$\nabla_{\kappa_n^-} \mathrm{KL}(p \parallel q) = \int \sum_{k=1}^{K} \left[A_d(\kappa_n^-) - (\mu_{n,k}^-)^\intercal w_{n,k} \right] p(W_n \mid y_{1:n-1})\, dW_n = 0$$

$$\implies \qquad A_d(\kappa_n^-) = \frac{1}{K}\, \mathbb{E}_{p(W_n \mid y_{1:n-1})} \left[\sum_{k=1}^{K} (\mu_{n,k}^-)^\intercal w_{n,k} \right]$$

$$\implies \qquad A_d(\kappa_n^-) = \frac{1}{K} \sum_{k=1}^{K} \mu_{n-1,k}^\intercal\, \mathbb{E}_{p(w_{n,k} \mid y_{1:n-1})}[w_{n,k}]$$

$$\implies \qquad A_d(\kappa_n^-) \overset{\text{(Eq. (4))}}{=} A_d(\kappa_{\gamma_n})\, A_d(\kappa_{n-1}) \frac{1}{K} \sum_{k=1}^{K} \|\mu_{n-1,k}\|^2$$

$$\implies \qquad A_d(\kappa_n^-) \overset{\|\mu_{n-1,k}\|=1}{=} A_d(\kappa_{\gamma_n})\, A_d(\kappa_{n-1}).$$

As mentioned in Sec. 5, the Gaussian model's mean-shrinkage behavior is matched when $\mathbb{E}[W_n] = \gamma_n\, \mathbb{E}[W_{n-1}]$, or, in terms of vMF first moments, by equating $A_d(\kappa_n^-)\mu_{n,k}^- = \gamma_n\, A_d(\kappa_{n-1})\mu_{n-1,k}$ for all classes k. Equivalently:

$$A_d(\kappa_{\gamma_n})\, A_d(\kappa_{n-1}) = \gamma_n\, A_d(\kappa_{n-1}),$$

so that

$$\kappa_{\gamma_n} = A_d^{-1}(\gamma_n) \tag{5}$$

leads to the desired mean-shrinkage behavior via the transition density of Eq. (1). Finally, this gives the predictive posterior concentration as:

$$\kappa_n^- = A_d^{-1}\big(\gamma_n\, A_d(\kappa_{n-1})\big).$$

Class-specific concentrations If one tracks class-specific concentrations $\kappa_{n,k}$ (with additional class-index k), the derivations can be carried out in largely the same way. Noting that the enforced mean-shrinkage behavior requires that κ_{γ_n} satisfy Eq. (5) for any k still, the derivations lead to the intuitive result:

$$\kappa_{n,k}^- = A_d^{-1}\big(\gamma_n\, A_d(\kappa_{n-1,k})\big).$$

A.2 Update step

When using a softmax likelihood

$$p(y_n = l \mid W_n; \phi_n) = \frac{\exp(w_{n,l}^\intercal \phi_n)}{\sum_{j=1}^{K} \exp(w_{n,j}^\intercal \phi_n)},$$

all class weights in $W_n = (w_{n,1}, \ldots, w_{n,K}) \in \mathbb{S}^{(p-1)\times K}$ are coupled, and thus the update step should be carried out jointly. With factorized predictive posterior

$$q(W_n \mid y_{1:n-1}) = \prod_{k=1}^{K} \mathrm{vMF}(w_{n,k}; \mu_{n,k}^-, \kappa_n^-),$$

the update step after observing label l is given by Bayes' rule:

$$\begin{aligned} q(W_n \mid y_{1:n}) &\propto p(y_{n,l} = 1 \mid W_n)\, q(W_n \mid y_{1:n-1}) \\ &= \frac{\exp(w_{n,l}^\intercal \phi_n)}{\sum_{j=1}^{K} \exp(w_{n,j}^\intercal \phi_n)} \prod_{k=1}^{K} \mathrm{vMF}(w_{n,k}; \mu_{n,k}^-, \kappa_n^-), \end{aligned}$$

which yields a posterior that is again intractable and also does not factorize over k anymore.

Titsias et al. (2023) circumvent this problem by linearizing the softmax likelihood into a product of Gaussian distributions. This makes the likelihood conjugate to the Gaussian posteriors they maintain so that the update step uses again closed-form Kalman filter operations, trading bias (inexact likelihood) for variance (closed-form updates). Several options mirror this design choice with varying degrees of faithfulness to the true observation model (softmax likelihood). In the following exposition, we refer to the natural parameter of the vMF distribution, $\eta = \kappa\mu$.

Improper conjugate likelihood Since the vMF is an angular distribution, it should track the correct mean direction $\mu_{n,k}$ over time. This mean direction must align well with those features ϕ_n that correspond to the observed labels for class k. Therefore, a very simple, but improper, conjugate likelihood may be defined as:

$$p_k(y_n \mid w_{n,k}) = \exp(\mathbb{I}\{y_n = k\}\phi^\intercal w_{n,k}),$$

with $\mathbb{I}\{\cdot\}$ the indicator function. Then, the likelihood term is added to the vMF natural parameter:

$$\eta_{n,k} = \eta_{n,k}^- + \mathbb{I}\{y_n = k\}\phi_n, \qquad q(w_{n,k} \mid y_{1:n}) = \mathrm{vMF}(\mu_{n,k}, \kappa_{n,k}),$$

with $\mu_{n,k} = \eta_{n,k}/\|\eta_{n,k}\|$, $\kappa_{n,k} = \|\eta_{n,k}\|$. Over time, these updates nudge the mean direction $\mu_{n,k}$ in the direction of ϕ_n whenever it corresponds to class k.

Laplace projection A more principled conjugate ADF update uses a local first-order expansion of the softmax likelihood and projects it onto the vMF natural-parameter space (for a general treatment, see, e.g., Wilkinson et al. (2023)). For class k, define the log-likelihood term

$$\ell_{n,k}(w_{n,k}) = \mathbb{I}\{y_n = k\} w_{n,k}^\intercal \phi_n - \log\left(\sum_{j=1}^{K} \exp(w_{n,j}^\intercal \phi_n)\right).$$

A first-order Laplace (ADF) projection evaluates the gradient of this log-likelihood term at the predictive mean direction:

$$\begin{aligned} g_{n,k} &= \nabla_{w_{n,k}} \ell_{n,k}(w_{n,k})\big|_{w_{n,k}=\mu_{n,k}^-} \\ &= \phi_n(\mathbb{I}\{y_n = k\} - p_{n,k}^-), \end{aligned}$$

where

$$p_{n,k}^- = \frac{\exp((\mu_{n,k}^-)^\intercal \phi_n)}{\sum_{j=1}^{K} \exp((\mu_{n,j}^-)^\intercal \phi_n)}.$$

The vMF natural parameter is then updated additively:

$$\eta_{n,k} = \eta_{n,k}^- + g_{n,k}, \qquad q(w_{n,k} \mid y_{1:n}) = \mathrm{vMF}(\mu_{n,k}, \kappa_{n,k}),$$

with $\mu_{n,k} = \eta_{n,k}/\|\eta_{n,k}\|$, $\kappa_{n,k} = \|\eta_{n,k}\|$.

This yields an approximate conjugate update for the vMF density driven by the local softmax gradient.

Moment matching True ADF minimizes the forward-KL divergence of the assumed posterior from the true posterior. For exponential-family distributions in general, forward-KL minimization equates to moment matching. For the vMF distribution specifically, it suffices to match the first theoretical moment $\mathbb{E}_{q(w_{n,k}|y_{1:n-1})}[w_{n,k}]$ to the empirical moment $\bar{R}_{n,k} = \frac{1}{S}\sum_{s=1}^{S} w_{n,k}^{(s)}$, $w_{n,k}^{(s)} \sim p(w_{n,k} \mid y_{1:n-1})$.

However, samples $w_{n,k}^{(s)}$ cannot be drawn from $p(w_{n,k} \mid y_{1:n-1})$ directly. Instead, from importance sampling, one has that the weighted first moment

$$\tilde{R}_{n,k} = \sum_{s=1}^{S} \alpha_n^{(s)} w_{n,k}^{(s)},$$

with

$$\begin{aligned} \alpha_n^{(s)} &= \frac{u_n^{(s)}}{\sum_{z=1}^{S} u_n^{(z)}}, \\ u_n^{(s)} &= p(y_{n,k} = 1 \mid W_n^{(s)}), \end{aligned}$$

and

$$W_n^{(s)} \sim p(W_n \mid y_{1:n-1}),$$

tends to $\mathbb{E}[w_{n,k} \mid y_{1:n}]$ as $S \to \infty$. Then, the assumed posterior parameters are recovered by equating:

$$A_d(\kappa_{n,k})\mu_{n,k} = \tilde{R}_{n,k},$$

i.e.,

$$\mu_{n,k} = \tilde{R}_{n,k}/\|\tilde{R}_{n,k}\|, \qquad \kappa_{n,k} = A_d^{-1}(\|\tilde{R}_{n,k}\|).$$

B Extended results

B.1 Full Split-CIFAR-100 metrics

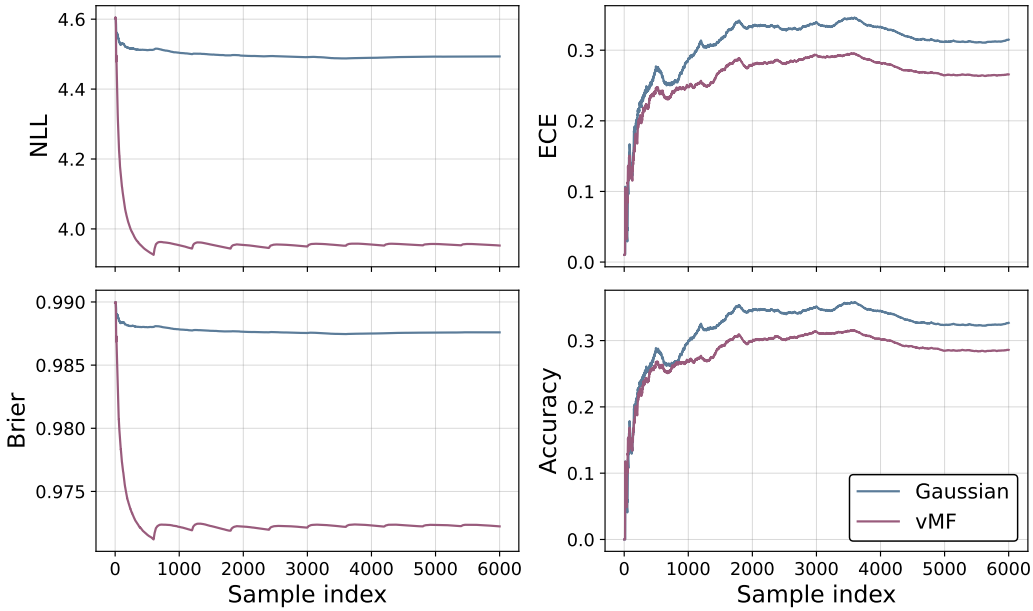

Figure 2: Comparison of Gaussian and vMF filters on Split-CIFAR-100 with an ImageNet-pretrained ResNet-18 backbone.

C Implementation notes

C.1 Approximations of von Mises–Fisher identities

The vMF 'resultant length' is given by (Mardia & Jupp, 2009):

$$A_d(\kappa) \coloneqq \frac{I_{d/2}(\kappa)}{I_{d/2-1}(\kappa)} = -\frac{d}{d\kappa} \log C_d(\kappa).$$

For this paper, it is sufficient to focus on nonnegative, integer d, as d is the dimensionality of the weight vectors.

For large d (~ 100), the upper-bound by Segura (2011) is sufficiently accurate for our purposes (see Fig. 3):

$$A_d(\kappa) = \frac{\kappa}{v + \sqrt{v^2 + \kappa^2}},$$

$v = d/2 - 1$, with inverse

$$A_d^{-1}(z) = \frac{2vz}{1 + z^2}.$$

Their derivatives are given by:

$$A'_d(\kappa) = \frac{u(\kappa)(v + u(\kappa)) - \kappa^2}{u(\kappa)(v + u(\kappa))^2},$$

$u(\kappa) = \sqrt{v^2 + \kappa^2}$, and

$$A_d^{-1\prime}(z) = \frac{2v(1 + z^2)}{(1 - z^2)^2}.$$

Finally, the indefinite integral is given by:

$$\int A_d(\kappa)\, d\kappa = u(\kappa) - v \log(v + u(\kappa)) + C,$$

with which the log-normalizer can be efficiently computed:

$$\begin{aligned}\log C_d(\kappa) &= \log C_d(0) - \left[\log C_d(0) - \log C_d(\kappa)\right] \\ &= \log C_d(0) - \int_0^\kappa A_d(t)\, dt.\end{aligned}$$

These approximations get rid of the computational complexity and small-/large-κ (especially relative to d) numerical inaccuracies, while providing closed forms and constant-time computational complexities. In our experiments, we therefore entirely replace the exact values of A_d and related functions with these approximations.

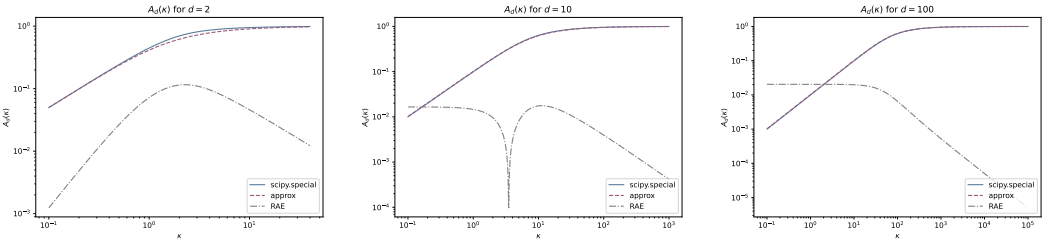

Figure 3: Relative absolute errors $\frac{|\tilde{A}_d - A_d|}{A_d}$ of the $\tilde{A}_d$ approximation due to Segura (2011), vs. the true values of A_d computed using the `scipy.special` module (using `scipy.special.ive` functions). The approximation error decreases in both κ and d.

C.2 Naesseth gradient corrections and GPU-friendly sampling from von Mises–Fisher distributions

von Mises–Fisher sampling Using the established Wood (1987) algorithm, we implement a fast, stable, accurate (see Fig. 4), and GPU-friendly von Mises–Fisher sampler. The sampler supports gradient-based learning via an `rsample` routine in PyTorch (in contrast to non-gradient-safe sampling, and in reference to the eponymous PyTorch implementations of reparameterized sampling routines) that returns alongside the drawn samples the acceptance log-probabilities of the rejection sampling scheme by Wood (1987). Via Naesseth et al. (2017), these log-probabilities allow gradient corrections of, e.g., loss quantities through which gradients should flow to learnable parameters.

Gradient corrections More formally, let $x \sim q_\theta(x)$ be drawn by a rejection sampler with proposal $r_\theta(x)$, acceptance probability $a_\theta(x, u)$, and let $\log a_\theta(x, u)$ be returned by the sampler. Naesseth et al. (2017) give the unbiased score-correction

$$\nabla_\theta \mathbb{E}_{q_\theta}[f(x)] = \mathbb{E}_{q_\theta}\left[\nabla_\theta f(x) + f(x)\, \nabla_\theta \log a_\theta(x, u)\right].$$

In practice, we implement this by adding, in the loss,

$$\left\langle \nabla_\theta \log p(y \mid \phi, x) + \log a_\theta(x, u)\,,\, \theta - \mathrm{stopgrad}(\theta) \right\rangle.$$

This inner product term is zero in the forward pass, while in the backward pass, this allows the sampler's stored log acceptance probabilities $\log a_\theta$ to provide exactly the Naesseth correction term needed for gradient flow through the rejection sampling step.

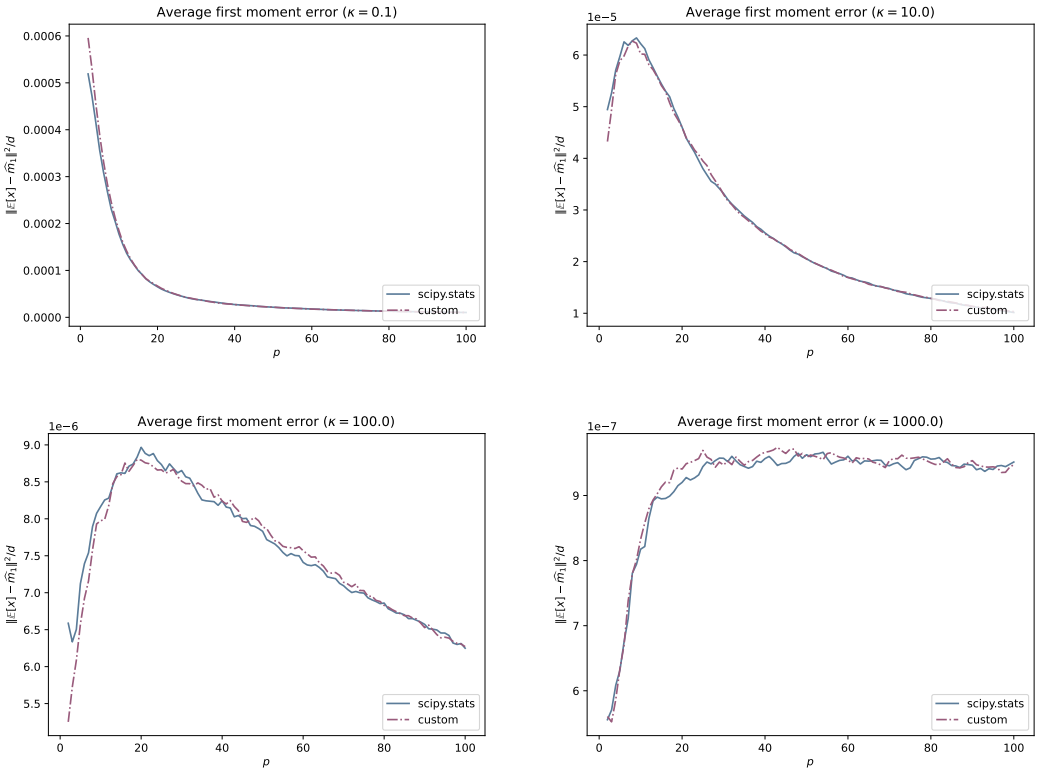

Figure 4: Comparison of sampling accuracy of our custom von Mises–Fisher sampler compared to the SciPy implementation over ranges of κ and d.