Learning Hyperparameters via a Data-Emphasized Variational Objective

Ethan Harvey
Tufts University
ethan.harvey@tufts.edu

Mikhail Petrov
Tufts University
mikhail.petrov@tufts.edu

Michael C. Hughes
Tufts University
michael.hughes@tufts.edu

Abstract

When training large models on limited data, avoiding overfitting is paramount. Common grid search or smarter search methods rely on expensive separate runs for each candidate hyperparameter, while carving out a validation set that reduces available training data. In this paper, we study gradient-based learning of hyperparameters via the evidence lower bound (ELBO) objective from Bayesian variational methods. This avoids the need for any validation set. We focus on scenarios where the model is over-parameterized for flexibility and the approximate posterior is chosen to be Gaussian with isotropic covariance for tractability, even though it cannot match the true posterior. In such scenarios, we find the ELBO prioritizes posteriors that match the prior, leading to severe underfitting. Instead, we recommend a data-emphasized ELBO that upweights the likelihood but not the prior. In Bayesian transfer learning of image and text classifiers, our method reduces the 88+ hour grid search of past work to under 3 hours while delivering comparable accuracy. We further demonstrate how our approach enables efficient yet accurate approximations of Gaussian processes with learnable lengthscale kernels.

1 INTRODUCTION

When training deep neural networks (DNNs) or other large models, significant time and effort are devoted to avoid overfitting. A common strategy is to use grid search to find hyperparameters that perform best on a validation set (Raschka, 2018). Smarter search strategies include successive halving (Karnin et al., 2013), Bayesian optimization (BO; Snoek et al., 2012; Hvarfner et al., 2024), or meta-learned BO (Wang et al., 2024). Such searches have two disadvantages. First, they require expensive separate runs for each candidate hyperparameter. Second, they need to carve out a labeled validation set, reducing data for model training. This is worrisome when available data has limited size.

A Bayesian approach to hyperparameters seems to be an elegant and pragmatic solution. Suppose we model observations $y_{1:N}$ via a likelihood $p_\eta(y_{1:N}|\theta)$, where θ is a high-dimensional parameter to be estimated and η is a hyperparameter vector. We also assume a prior $p_\eta(\theta)$. To learn both θ and η from data, we can follow the type-II maximum likelihood recipe: estimate the posterior $p_\eta(\theta|y_{1:N})$ while simultaneously learning η to maximize $p_\eta(y_{1:N}) = \int_\theta p_\eta(y_{1:N}, \theta) d\theta$. The objective $p_\eta(y_{1:N})$ is known as the *marginal likelihood* or *evidence* (MacKay, 1996; Neal, 1996; Rasmussen and Williams, 2006a). The marginal likelihood favors η values that lead to simpler models that fit the data well while avoiding overfitting (Jeffreys, 1939; MacKay, 1991; Rasmussen and Ghahramani, 2000; Grünwald, 2005). This objective naturally penalizes overcomplexity and is often praised as a Bayesian Occam's razor. Learning η to maximize the marginal likelihood via gradient ascent resolves both issues raised above: we need only one run of gradient ascent (not separate runs for each candidate η) and can use all available data for training without overfitting. No validation set is needed at all.

Unfortunately, for large flexible models of practical interest, the marginal likelihood strategy appears underutilized despite being well-known for decades. Instead, recent works in Bayesian transfer learning (Krishnan et al., 2019, 2020; Shwartz-Ziv et al., 2022; Harvey et al., 2024; Rudner et al., 2025) employ grid searches that take *multiple days*. One obvious barrier is eval-

Proceedings of the 29th International Conference on Artificial Intelligence and Statistics (AISTATS) 2026, Tangier, Morocco. PMLR: Volume 300.

Code: github.com/tufts-ml/data-emphasized-ELBO

uating the marginal likelihood. For modern DNNs, computing this high-dimensional integral is difficult even for a specific η. Learning η via gradient ascent is tougher. Immer et al. (2021) offer a route to gradient-based learning of η via a Laplace approximation of the log marginal likelihood, yet their method's runtime is expensive due to an estimated Hessian matrix. Alternatively, variational Bayesian methods (Jordan et al., 1999; Blei et al., 2017) promise a tractable objective which lower bounds the log marginal likelihood (LML), known as the evidence lower bound (ELBO). However, the ELBO is not widely used to learn η for DNNs. Blundell et al. (2015) tried to learn hyperparameters of Bayesian neural networks (BNNs) via gradients of the ELBO but found it "to not be useful, and yield worse results," although no concrete comparison was provided.

In this work, we study when and why ELBO-based methods fail to provide reliable model selection for large models like DNNs. We focus on a target scenario with two key assumptions:

- First, models are over-parameterized such that $D \gg N$, where D is the size of parameter θ and N is the number of training examples. Such models often enjoy practical success (Li et al., 2018) if η is selected to avoid overfitting.
- Second, to stay affordable we assume the approximate posterior is Gaussian with simplified covariance. With large models, we cannot afford to estimate a $D \times D$ covariance matrix. Variational methods let us explore isotropic posteriors with $D + 1$ parameters and runtime close to standard point estimation.

Our work makes two contributions for this target scenario. First, we show analytically and empirically that the ELBO objective favors posteriors and hyperparameters that underfit the data, substantiating Blundell et al.'s claim of "worse results". Second, to remedy this issue we suggest an alternative objective that we call the *data-emphasized ELBO* (DE-ELBO). By upweighting the likelihood term inside the ELBO, our DE-ELBO can jointly learn posteriors and hyperparameters that fit the data better.

We pursue two case studies to justify and validate our approach. First, Case Study A (Sec. 5) looks at a $D \gg N$ regression model where a true posterior with full covariance is analytically tractable. Here we can prove in our target scenario why the ELBO will match a prior variance and thus underfit, while our DE-ELBO delivers better data fits. Second, Case Study B (Sec. 6) on DNN transfer learning for image and text classification empirically covers many datasets, model families, and architectures like ResNets, ViTs, and ConvNeXts. In all studies, our data-emphasized ELBO yields far better practical fits than the standard ELBO. Moreover, our DE-ELBO yields competitive or better fits than recent Bayesian methods like Immer et al. (2021) or Lotfi et al. (2022), even when they use diagonal (not isotropic) covariances. There are also speed wins: our approach reduces the 88+ hour grid search of recent works (Shwartz-Ziv et al., 2022; Rudner et al., 2025) to under 3 hours while giving comparable or better accuracy. Our approach thus advances the accuracy-runtime frontier of practical Bayesian model selection.

2 BACKGROUND

Our generic probabilistic model assumes observed data $\{y_i\}_{i=1}^N$ are *i.i.d.* conditioned on parameters $\theta \in \mathbb{R}^D$:

$$p_\eta(y_{1:N}, \theta) = p_\eta(\theta) \cdot \prod_{i=1}^N p_\eta(y_i|\theta). \quad (1)$$

This template is instantiated by specifying a concrete likelihood $p_\eta(y_i|\theta)$ and prior $p_\eta(\theta)$. The subscript indicates possible dependence on hyperparameters η.

Direct estimation of the posterior $p_\eta(\theta|y_{1:N})$ or marginal likelihood is typically intractable. Instead, we can pursue an approximate posterior via variational methods (Jordan et al., 1999; Blei et al., 2017). We first select an "easy-to-use" family of distributions over the parameter θ. A member of this family is denoted $q_\psi(\theta)$, where each variational parameter ψ defines a specific distribution over θ. We then pose an optimization problem: find the variational parameter ψ that makes $q_\psi(\theta)$ as close as possible to the true (intractable) posterior.

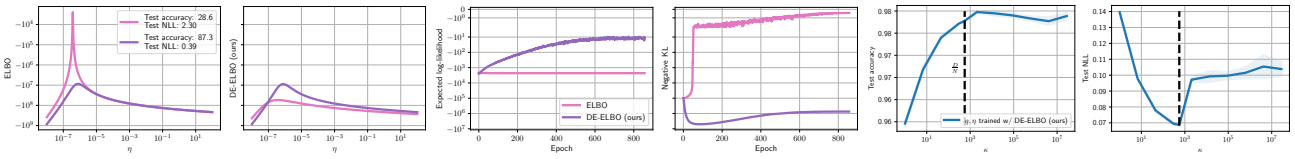
We can tractably estimate ψ by maximizing the *evidence lower bound* (ELBO; Blei et al., 2017), defined for our model p and approximate posterior q as $J_{\text{ELBO}} :=$

$$\mathbb{E}_{q_\psi(\theta)}\left[\sum_{i=1}^N \log p_\eta(y_i|\theta)\right] - D_{\text{KL}}(q_\psi(\theta)\|p_\eta(\theta)). \quad (2)$$

This objective is a function of data y, variational parameters ψ, and hyperparameters η. Maximizing J_{ELBO} is equivalent to finding q "closest" to the true posterior in the sense of Kullback-Leibler (KL) divergence (Blei et al., 2017). As its name suggests, the ELBO is a lower bound on the log of the evidence: $J_{\text{ELBO}}(y_{1:N}, \psi, \eta) \leq \log \int_\theta p_\eta(y_{1:N}, \theta) d\theta$.

Target scenario. We focus on scenarios with two key assumptions. First, we assume a model as in Eq. (1) where $D \gg N$, where D is the size of parameter θ and N is size of available training data. Second, we assume the approximate posterior q is Gaussian with an ***isotropic*** covariance matrix: $q_\psi(\theta) = \mathcal{N}(\theta|\bar{\theta}, \bar{\sigma}_q^2 I_D)$, where $\psi = \{\bar{\theta}, \bar{\sigma}_q\}$ with $\bar{\theta} \in \mathbb{R}^D$ and $\bar{\sigma}_q \in \mathbb{R}_{>0}$. This last assumption is motivated by scalability.

For gradient-based learning, we further assume both θ and η contain only continuous real values. If some



(a) Model selection as function of η (b) ELBO terms over training epochs (c) Performance as function of κ

Figure 1: *Panels (a)-(b)*: Comparing approximate posteriors q trained for ELBO (pink) and DE-ELBO (ours, purple). Task: ResNet-50 with $D >$ 1e7 trained on CIFAR-10 with $N = 1000$. (a) shows which objective prefers which q across $\eta = \lambda = \tau$. (b) shows which terms in each objective matter most over training steps. **Takeaway: When $D \gg N$, the ELBO prefers simpler q (pink) close to the prior, while our DE-ELBO favors q with higher test accuracy (purple).** *Panel (c):* Test accuracy and negative log-likelihood (NLL, lower is better) for q trained via our DE-ELBO with various κ values. Task: ConvNeXt-Tiny with $D >$ 1e7 trained on CIFAR-10 with $N = 50000$. **Takeaway: Set $\kappa = \frac{D}{N}$.**

values are discrete, we can use ELBO (or later, DE-ELBO) objectives, but not gradient-based algorithms.

ELBO for η selection. The fact that ELBO lower bounds the marginal likelihood suggests its utility for selecting hyperparameters η. Work over decades has used the ELBO to select hyperparameters in mixture models (Ueda and Ghahramani, 2002), Gaussian processes (GPs; Titsias, 2009; Damianou and Lawrence, 2013; Hensman et al., 2015; Lalchand et al., 2022), and small neural networks (KC et al., 2021). Yet none of these fit our target scenario, violating either $D \gg N$ or the assumption that q is a Gaussian with simplified covariance for tractability. Blei et al. (2017) caution that while the ELBO sometimes works in practice, "selecting based on a bound is not justified in theory." Yet Cherief-Abdellatif (2019) show that ELBO-based model selection enjoys theoretical guarantees on quality, even under misspecification.

Unfortunately, modern Bayesian deep learning efforts for our target $D \gg N$ scenario remain dominated by time-consuming grid search (Osawa et al., 2019; Shwartz-Ziv et al., 2022; Harvey et al., 2024; Rudner et al., 2025) rather than gradient-of-ELBO learning.

3 CONTRIBUTIONS

We offer two contributions to improve Bayesian model selection in our target scenario.

Contribution 1: When $D \gg N$ and q is a misspecified isotropic Gaussian, the ELBO prefers settings of ψ, η that underfit. Thus, gradient-based learning of both ψ, η via the ELBO (see Alg. C.1 with $\kappa = 1$) often yields notably subpar prediction compared to search methods for η.

Our evidence for Contribution 1 comes in analytical and experimental forms. First, in Case Study A (Sec. 5.1) we analyze a regression model whose ideal (non-isotropic) posterior yields good data fits at small N for arbitrarily large D. There, we prove in Lemma 1 that when q has an isotropic covariance matrix, the ELBO prefers the posterior variance in ψ to match the prior variance as D gets larger but N stays fixed. This too-large variance leads to underfitting.

Second, consider a DNN image classifier for CIFAR-10 data as described later in Case Study B. Fig. 1 (a) compares two different isotropic q at this task, colored pink and purple. The pink q fixes ψ parameters to values that optimize ELBO; it scores higher ELBO than the purple q across a range of η. Yet this model delivers subpar test accuracy of just 28.6%. Throughout later experiments on toy data (see Fig. 2) and Bayesian transfer learning of image and text classifiers (see Fig. 3), ELBO learning yields subpar accuracy.

To better understand the reason for underfitting, Fig. 1 (b)'s pink lines plot over epochs the two additive terms in Eq. (2) that define the ELBO. The negative KL term starts out at a large negative value and approaches zero throughout training, suggesting q is approaching the prior. In contrast, the likelihood term makes no visible progress from its modest initial value. These trace plots motivate a remedy: upweight the likelihood term.

Contribution 2: Emphasizing the data likelihood in the ELBO with upweighting factor $\kappa = \frac{D}{N}$ yields ψ, η values that fit data better when $D \gg N$ and q is misspecified. We propose the *data-emphasized ELBO* objective, $J_{\text{DE-ELBO}} :=$

$$\kappa \cdot \mathbb{E}_{q_\psi(\theta)} \left[\sum_{i=1}^{N} \log p_\eta(y_i|\theta) \right] - D_{\text{KL}}(q_\psi(\theta) \| p_\eta(\theta)) \quad (3)$$

where we have introduced a scaling factor κ on the likelihood term. $\kappa = 1$ recovers the standard ELBO; instead we recommend larger $\kappa = \frac{D}{N}$ to address the issues raised in Contribution 1. The DE-ELBO is equivalent to a standard ELBO for κN *i.i.d.* data instances, where we happen to observe κ copies of dataset $y_{1:N}$.

Gradient-based learning of both ψ, η with this objective (see Alg. C.1 with $\kappa = \frac{D}{N}$) fits data better than the

standard ELBO or other Bayesian methods. Its prediction quality rivals more expensive search methods in far less time.

Our evidence for Contribution 2 comes in analytical and experimental forms. We prove in Lemma 2 for the Case Study A regression model that DE-ELBO with $\kappa = \frac{D}{N}$ delivers posterior variance that does not collapse to the prior as $D \to \infty$. This means even when $D \gg N$, ψ, η can be learned via DE-ELBO to deliver compelling data fit. In practice, returning to Fig. 1, the purple q with ψ, η that optimize DE-ELBO with $\kappa = \frac{D}{N}$, has far better test accuracy of 87.3%. While DE-ELBO prefers this purple solution, standard ELBO clearly does not. Finally, Case Study B experiments varying κ (see Fig. 1 (c), App. B.4) show that $\kappa = \frac{D}{N}$ has competitive accuracy across datasets, far better than ELBO ($\kappa = 1$).

4 RELATED WORK

Here we survey other work on modifying ELBOs and learning hyperparameters. Although our objective is mathematically similar to previous work on modified ELBOs (Zhang et al., 2018; Aitchison, 2021; McLatchie et al., 2025), our setting of $\kappa = \frac{D}{N}$ and our purpose of enabling gradient-based learning of hyperparameters is distinct.

Upweighting data in the ELBO. Motivated by different goals than model selection, previous work has also upweighted the likelihood term of the ELBO via a κ multiplier as in Eq. (3), or equivalently downweighted the KL term. This line of work (Aitchison, 2021; Osawa et al., 2019; Zhang et al., 2018; Pitas and Arbel, 2024) refers to reweighting as *tempering*. They pursue reweighted variational ELBOs to better capture posterior uncertainty, but do not pursue gradient-based learning of η. Despite awareness that it is "favorable to tune regularization" (Zhang et al., 2018), often only a small grid of candidate η are searched, as in Osawa et al. (2019, Fig. 8) or Zhang et al. (2018), perhaps due to large costs of each separate run. Aitchison (2021) do not tune regularization hyperparameters at all.

Downweighting data in the ELBO. Other work downweights the likelihood in the ELBO for purposes other than learning η. Mandt et al. (2016)'s variational tempering method downweights data to avoid local optima in mixture models. Some Bayesian approaches that seek to counter-act model misspecification have effectively downweighted data by raising the likelihood to a power *smaller than one*. This includes the *power likelihood* (Antoniano-Villalobos and Walker, 2013), *power posterior* (Friel and Pettitt, 2008; Miller and Dunson, 2019), or "safe" Bayesian learning (Grünwald, 2012; Grünwald and van Ommen, 2017). Work on β-variational autoencoders (Higgins et al., 2017) upweights the KL term of the ELBO, reducing data influence to learn disentangled embeddings. None of these works learn η via gradient ascent as we do.

Cold posteriors. Other work in Bayesian deep learning has recommended *cold posteriors* (Wenzel et al., 2020; Kapoor et al., 2022). This work's objective is mathematically different, as it multiplies the entire log posterior by a scalar temperature, not just the log-likelihood as we do. The stated purpose of this temperature is to improve heldout prediction quality of samples from the inferred posterior over θ. Rather than seeking q from a variational method, they pursue more expensive sampling-based inference without diagonal or isotropic simplifications. They do not seek gradient-based learning of hyperparameters η.

Bayesian learning of η. Immer et al. (2021) offer a leading alternative to the ELBO for learning η. They optimize a Laplace approximation of the log marginal likelihood (LA-LML) where the covariance matrix is affordable via a diagonal empirical Fisher (diagEF) approximation. We compare to this baseline throughout Sec. 5 and 6 below. It allows gradient-based learning of η, but can take roughly 2x longer than ELBO or DE-ELBO for a single training run (timings in Tab. 2).

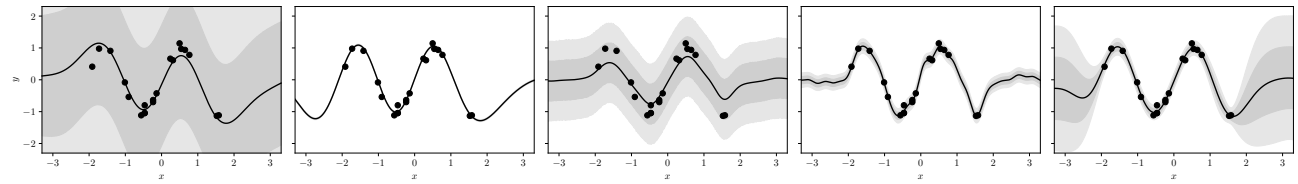
Lotfi et al. (2022) suggest a conditional log marginal likelihood (CLML) objective for selecting hyperparameters. For DNNs, they compare separate runs at distinct η, but cannot do gradient-based learning. We compare to their LA-CLML method in Sec. 5 and 6, which reuses Immer et al.'s diagEF Laplace approximation.

Gradient methods for η. Some non-ELBO works (Maclaurin et al., 2015; Lorraine et al., 2020) learn hyperparameters via gradients of validation-set performance metrics. These require large validation sets to perform well, while our DE-ELBO does not use a validation set and may be easier to implement.

Smart search for η. Stepping back, if the goal is simply to tune hyperparameters for a point estimation task, many smart search strategies have been proposed. Random search (Bergstra and Bengio, 2012), successive halving (Karnin et al., 2013; Jamieson and Talwalkar, 2016), BO (Snoek et al., 2012; Turner et al., 2021), or meta-learned BO (Wang et al., 2024) all have advantages over grid search. Yet all require separate runs for each candidate η and dividing available data into train and validation sets, unlike ELBO-based approaches.

5 CASE STUDY A: REGRESSION

The Gaussian process (GP; Rasmussen and Williams, 2006b) is a regression model often paired with a *radial basis function* (RBF) kernel. This kernel can be de-



(a) diagEF LA-LML (b) diagEF LA-CLML (c) isotropic q ELBO (d) isotropic q DE-ELBO (ours) (e) $D \times D$ covariance LML

Figure 2: Predictions using ψ, η selected by different objectives for RFF regression. We show diagEF LA-LML (Immer et al., 2021), diagEF LA-CLML (Lotfi et al., 2022), iso ELBO, iso DE-ELBO (ours), and the true posterior for η that optimize LML. We plot train data $y_{1:N}$ with the mean and two standard deviations of the predictive posterior $p(y_*|y_{1:N})$. **Takeaway: DE-ELBO best approximates the true posterior's mean and variance near data. Variance far from data is underestimated.**

fined as $k(x, x') = \sigma_k^2 \exp\left(-\frac{\|x-x'\|_2^2}{2\ell_k^2}\right)$, where ℓ_k, σ_k are lengthscale and outputscale hyperparameters. Though GP function complexity can elegantly grow with dataset size N, a downside is that fitting scales cubically with N. As a remedy, we consider *random Fourier features* (RFFs; Rahimi and Recht, 2007), a weight-space model that scales *linearly* in N yet approximates a GP. The approximation quality increases with a user-controlled size parameter R which sets the overall size: $D = R$.

GPs are notoriously sensitive to hyperparameters $\eta = \{\ell_k, \sigma_k\}$ (Rasmussen and Williams, 2006c). RFFs are also sensitive, yet tuning η well has been largely overlooked. Rahimi and Recht (2007) fix $\ell_k = 1, \sigma_k = 1$. When Liu et al. (2020, 2023) used RFF classifiers, they set $\ell_k = 2$ and tune σ_k via search with multiple runs.

We intend to show how our DE-ELBO yields affordable gradient-based learning of ℓ_k, σ_k when $D \gg N$, thus improving overall prediction quality. Our work here could be used as an efficient (*linear in* N) GP or as a drop-in way to improve Liu et al.'s distance-aware methods. Our isotropic qs are scalable complements to exact RFF posteriors (Potapczynski et al., 2021).

5.1 Model A Definition

We train on N pairs x_i, y_i of input vectors $x_i \in \mathbb{R}^H$ and targets $y_i \in \mathbb{R}$. We first map each x_i to a transformed RFF representation $\phi(x_i) \in \mathbb{R}^R$ via

$$\phi(x_i) = \sigma_k \sqrt{\frac{2}{R}} \cos\left(\frac{1}{\ell_k} A^\top x_i + b\right), \tag{4}$$

The non-learnable weights $A \in \mathbb{R}^{H \times R}$ and $b \in \mathbb{R}^R$ are randomly drawn **once** as $A_{h,r} \sim \mathcal{N}(0, 1)$ and $b_r \sim \text{Unif}(0, 2\pi)$ for all h and r. A and b remain fixed through all remaining training and prediction. Past work (Liu et al., 2020) typically has R in the range 100 to 10000. We set $R = 1024$.

To complete the regression model, we predict $\hat{y}_i = v^\top \phi(x_i)$ with weights $v \in \mathbb{R}^R$.

Contribution: RFFs for arbitrary lengthscale. Our featurization in Eq. (4) generalizes the construction of RFFs by Rahimi and Recht (2007) to any lengthscale $\ell_k > 0$ and outputscale $\sigma_k > 0$. In App. A.1, we ***prove*** that our construction is a Monte Carlo approximation of the RBF kernel. That is, for any pair of feature vectors $\phi(x_i)^\top \phi(x_j) \approx k(x_i, x_j)$, where $k(\cdot)$ is an RBF kernel whose ℓ_k, σ_k values match those used to construct $\phi(\cdot)$ in Eq. (4). The quality of this approximation increases with R. Past work has proven that RFF features approximate the RBF kernel only when $\ell_k = 1, \sigma_k = 1$ (Rahimi and Recht, 2007). To the best of our knowledge, our proof for arbitrary ℓ_k is novel.

Point estimation view. To fit the RFF model to data, empirical risk minimization seeks weights v that minimize the loss function $L(v) :=$

$$\sum_{i=1}^N \ell(y_i, v^\top \phi(x_i)) + \frac{1}{2}||v||_2^2, \tag{5}$$

where $\ell(\cdot)$ is a loss function (e.g., mean squared error). The L2 penalty on v helps avoid overfitting given many features. Ultimately, model quality is impacted by hyperparameters $\ell_k > 0, \sigma_k > 0$. Neither can be set effectively by minimizing training loss $L(\cdot)$ alone. We later show how to *learn* ℓ_k, σ_k using the DE-ELBO.

Bayesian view. We can define a probabilistic model:

$$\begin{aligned} p(v) &= \mathcal{N}(v|0_R, I_R), \\ p(y|v) &= \textstyle\prod_{i=1}^N \mathcal{N}(y_i|v^\top \phi(x_i), \sigma_y^2). \end{aligned} \tag{6}$$

This model fits into our general framework from Sec. 2: $\theta = \{v\}$, $\eta = \{\sigma_y, \ell_k, \sigma_k\}$, and $D = R$. Maximum a-posteriori (MAP) estimation of v recovers the objective in Eq. (5) when we set $\ell(\cdot)$ to $-\log p(y_i|v)$.

Ideal posterior. For the regression model in Eq. (6), the true posterior is multivariate Gaussian, with full-rank covariance $\Sigma_{\text{post}} = (I_D + \frac{1}{\sigma_y^2} \Phi^\top \Phi)^{-1}$, where $\Phi \in \mathbb{R}^{N \times D}$ stacks features $\phi(x_i)$ for each of the N train examples. For derivation, see App. A.2 and A.3.

5.2 Variational Methods for Model A

To apply the general variational recipe described in Sec. 2 to the model in Eq. (6), we first select an approximate posterior over parameter v. For simplicity and speed when D is large, we choose a Gaussian with unknown mean and isotropic covariance:

$$q(v) = \mathcal{N}(v|\bar{v}, \bar{\sigma}_q^2 I_D). \tag{7}$$

Here, the free parameters that define q are $\psi = \{\bar{v}, \bar{\sigma}_q\}$, with $\bar{v} \in \mathbb{R}^D$ and $\bar{\sigma}_q \in \mathbb{R}_{>0}$.

5.3 Theoretical Analysis of Model A

First, in Lemma 1 below we show that for over-parameterized models the assumption of isotropic q leads to undesirable underfitting when optimizing the ELBO. Next, we show the DE-ELBO avoids this issue, particularly by setting $\kappa = \frac{D}{N}$.

Lemma 1. Assuming isotropic q, as $D \to \infty$ the optimal approximate posterior variance $\bar{\sigma}_q^2$ for the ELBO ($\kappa = 1$) exactly matches the prior variance of 1.

Proof. Set $\nabla_{\bar{\sigma}_q^2} J_{\text{ELBO}} = 0$, solve for $\bar{\sigma}_q^2$, then take the limit:

$$\bar{\sigma}_q^{2*} = \frac{D}{\frac{1}{\sigma_y^2}\text{tr}(\Phi\Phi^\top) + D}, \quad \lim_{D\to\infty} \bar{\sigma}_q^{2*} = 1.$$

By construction, we have $\Phi\Phi^\top \approx K$, where K is the N-by-N matrix of kernel evaluations on the train set. For the chosen RBF kernel, $\text{tr}(K) = \sigma_k^2 N$. ■

From Lemma 1, we conclude that as parameter size D increases but N remains fixed, the variance of the approximate posterior q approaches a value of 1, matching the isotropic prior in Eq. (6). As a result, RFF regressors trained to maximize ELBO can underfit, preferring higher posterior variance than may be needed (see Fig. 2 (c)). Similar underfitting has been shown for BNNs (Coker et al., 2022).

Lemma 2. In the same setting as Lemma 1, as $D \to \infty$ the optimal approximate posterior variance for the DE-ELBO ($\kappa = \frac{D}{N}$) will be smaller than the prior variance.

Proof. Set $\nabla_{\bar{\sigma}_q^2} J_{\text{DE-ELBO}} = 0$, solve for $\bar{\sigma}_q^2$, then take the limit:

$$\bar{\sigma}_q^{2*} = \frac{D}{\frac{D}{N}\frac{1}{\sigma_y^2}\text{tr}(\Phi\Phi^\top) + D}, \quad \lim_{D\to\infty} \bar{\sigma}_q^{2*} = \frac{1}{\frac{\sigma_k^2}{\sigma_y^2} + 1} < 1. \blacksquare$$

The DE-ELBO thus helps the misspecified posterior retain dependence on the data even when $D \gg N$, avoiding collapse to the prior. The particular choice of $\kappa = \frac{D}{N}$ is key to this result via elegant cancellation in the denominator. When hyperparameters $\sigma_y, \ell_k, \sigma_k$ are *learnable*, DE-ELBO can produce high-quality fits to data even for large D, as in Fig. 2 (d).

5.4 Experiments for Model A

We compare different methods for training ψ, η for RFF models with $R = D = 1024$ on an $N = 20$ univariate regression dataset $y = \sin(3x) + \varepsilon$ where $\varepsilon \sim \mathcal{N}(0, 0.01)$. The goal here is to illustrate sensitivity to hyperparameters $\eta = \{\sigma_y, \ell_k, \sigma_k\}$ and the effective learning of q, η enabled by our approach.

Baselines. We fit isotropic (iso) posteriors q and hyperparameters η via the standard ELBO and our proposed DE-ELBO, via Alg. C.1. We compare to diagonal (diag) covariance versions of LA-LML (Immer et al., 2021) and LA-CLML (Lotfi et al., 2022).

Common training plan. For all methods on this toy dataset, we perform gradient ascent using all training data in every step without any data augmentation. We train for a specified number of iterations and verify convergence by inspection. Each run of our method and the baselines depends on the adequate selection of learning rate. All runs search over 4 candidate values, selecting the best using a method-appropriate objective (e.g., train set DE-ELBO for iso DE-ELBO). See pseudocode in App. C for details.

Learning hyperparameters $\eta = \{\sigma_y, \ell_k, \sigma_k\}$. For ELBO and DE-ELBO, at each epoch we update variational parameters ψ and hyperparameters $\sigma_y, \ell_k, \sigma_k$ via a gradient step. Gradients of each ELBO-based objective J are easily computed with respect to $\sigma_y, \ell_k, \sigma_k$ via automatic differentiation. We use the softplus reparameterization to handle positivity constraints.

5.5 Results for Model A

Fig. 2 visualizes the posterior predictive distribution for each method on the $N = 20$ toy regression task. The iso ELBO underfits with high variance, validating our Lemma 1. Our iso DE-ELBO best approximates the true predictive posterior's mean and variance near data, outperforming the other baselines even though they have flexibility to use a diagonal (not isotropic) covariance. The iso DE-ELBO does *underestimate* the variance far from data. This is expected since we chose an isotropic q for tractability not high fidelity to the posterior. App. A has more RFF results on *classification*.

6 CASE STUDY B: TRANSFER LEARNING

6.1 Model B Definition

For Case Study B, we explore transfer learning of image and text classifiers using informative priors (Li et al., 2018; Shwartz-Ziv et al., 2022) in our target scenario.

Each neural network has two parts. First, a backbone encoder $f(\cdot)$ with weights $w \in \mathbb{R}^F$ maps input vector x_i to a representation vector $z_i \in \mathbb{R}^H$. For transfer learning, we assume the backbone weights w are high-dimensional (F is very large) and that w has been pre-trained to a high-quality initial value μ on a source task. Second, a linear-boundary classifier with weights $V \in \mathbb{R}^{C \times H}$ leads to probabilities over C possible classes. We seek values of w and V that classify well on a provided *target task* dataset of N pairs x_i, y_i of features x_i and corresponding class labels $y_i \in \{1, 2, \dots C\}$.

Deep learning view. Typical DNN approaches to transfer learning (e.g., baselines in Li et al. (2018)) would pursue empirical risk minimization with regularization, training to minimize the loss $L(w, V) :=$

$$\sum_{i=1}^{N} \ell(y_i, V f_w(x_i)) + \frac{\alpha}{2}||w||_2^2 + \frac{\beta}{2}||\operatorname{vec}(V)||_2^2 \quad (8)$$

where $\ell(\cdot)$ is a cross-entropy loss indicating agreement with the true label y_i, while the L2-penalty on weights w, V favors magnitudes closer to zero, often referred to as "weight decay". Hyperparameters $\alpha \geq 0$, $\beta \geq 0$ control the strength of the L2 penalty.

Bayesian view. For this problem, we can define a joint $p(y_{1:N}, w, V)$ decomposed as in Eq. (1), where

$$\begin{aligned} p(w) &= \mathcal{N}(w|\mu_p, \lambda\Sigma_p), \\ p(V) &= \mathcal{N}(\operatorname{vec}(V)|0_{HC}, \tau I_{HC}), \\ p(y_i|w, V) &= \operatorname{Cat}(y_i|\operatorname{SM}(V f_w(x_i))). \end{aligned} \quad (9)$$

Here, $\lambda > 0, \tau > 0$ are hyperparameters, μ_p, Σ_p represent *a priori* knowledge of the mean and covariance of the backbone weights w, $\operatorname{SM}(\cdot)$ is the softmax function, and $\operatorname{Cat}(\cdot)$ is the categorical probability mass function. Pursuing MAP estimation for w and V recovers the objective in Eq. (8) when we set $\alpha = \frac{1}{\lambda}, \beta = \frac{1}{\tau}, \mu_p = 0_F, \Sigma_p = I_F$, and $\ell(\cdot)$ to $-\log p(y_i|w, V)$. In terms of our general framework, we have $\theta = \{w, V\}$, $\eta = \{\lambda, \tau\}$, and $D = F + HC$.

Need for validation set and grid search. Selecting α, β (or equivalently λ, τ) to directly minimize Eq. (8) on the training set alone is not a coherent way to guard against overfitting. Regardless of data or weights, we would select $\alpha^* = 0, \beta^* = 0$ to minimize $L(\cdot)$ as a function of α, β and thus enforce no penalty on weight magnitudes at all. We see similar results with $\kappa \gg \frac{D}{N}$ (see App. B.4). Carving out a validation set for selecting these hyperparameters is thus critical to avoid overfitting when point estimating w, V.

Backbone priors. Several recent transfer learning methods correspond to specific values of the mean and covariance μ_p, Σ_p of the backbone prior $p(w)$. Let μ represent pre-trained backbone weights from the source task. Setting $\mu_p = 0_F, \Sigma_p = I_F$ recovers a conventional approach we call L2-zero, where regularization pushes backbone weights to zero. The pre-trained μ only informs the initial value of w before stochastic gradient descent (SGD). Instead, setting $\mu_p = \mu, \Sigma_p = I_F$ recovers *L2 starting point* (L2-SP) regularization (Chelba and Acero, 2006; Li et al., 2018). Further setting Σ_p to the covariance matrix of a Gaussian approximation of the posterior over backbones from the source task recovers the "Pre-Train Your Loss" (PTYL) method (Shwartz-Ziv et al., 2022).

Table 1: Possible priors.

Method	$p(w)$	Init.
L2-zero	$\mathcal{N}(0_F, \lambda I_F)$	μ
L2-SP	$\mathcal{N}(\mu, \lambda I_F)$	μ
PTYL	$\mathcal{N}(\mu, \lambda\Sigma)$	μ

Need to specify a search space. Selecting α, β (or equivalently λ, τ) via grid search requires specifying a grid of candidates spanning a finite range. For PTYL, the optimal search space for these hyperparameters is still unclear. For the same prior and the same datasets, the search space has varied between works: PTYL's creators recommended large values from 1e0 to 1e10 (Shwartz-Ziv et al., 2022). Later works search smaller values (1e-5 to 1e-3) (Rudner et al., 2025). Our DE-ELBO with per-epoch learning of η avoids the need to set any predefined ranges.

6.2 Variational Methods for Model B

To apply the general variational recipe described in Sec. 2 to Model B, we first select an approximate posterior over parameters w, V. For tractability and speed, we choose a factorized Gaussian with unknown means and isotropic covariance controlled by scalar $\bar{\sigma}_q > 0$:

$$\begin{aligned} q(w, V) &= q(w)q(V) \\ q(w) &= \mathcal{N}(w|\bar{w}, \bar{\sigma}_q^2 I_F), \\ q(V) &= \mathcal{N}(\operatorname{vec}(V)|\operatorname{vec}(\bar{V}), \bar{\sigma}_q^2 I_{HC}). \end{aligned} \quad (10)$$

Here, the free parameters for q are $\psi = \{\bar{w}, \bar{V}, \bar{\sigma}_q\}$.

ELBO-based training. Given a training set, we optimize ψ, η to maximize the ELBO or our DE-ELBO via Alg. C.1. We evaluate the KL term inside each objective in closed-form, as both prior and q are Gaussian. To evaluate the expected log-likelihood term, we use Monte Carlo averaging of S samples from q (Xu et al., 2019; Mohamed et al., 2020). We find that just one sample ($S = 1$) per training step is sufficient and fast. We use the *reparameterization trick* (Blundell et al., 2015) to obtain gradient estimates for this likelihood term.

Learning hyperparameters $\eta = \{\lambda, \tau\}$. While we could use gradients to update the prior variances λ, τ, inspecting the closed-form of the KL term in the objective reveals an analytical update guaranteed to deliver

the best possible value of λ (in terms of ELBO or DE-ELBO) given the current value of $\psi = \{\bar{w}, \bar{V}, \bar{\sigma}_q\}$. Setting $\nabla_\lambda J = 0$ and solving for λ, we get

$$\lambda^* = \frac{1}{F}\left[\bar{\sigma}_q^2 \operatorname{tr}(\Sigma_p^{-1}) + (\mu_p - \bar{w})^\top \Sigma_p^{-1}(\mu_p - \bar{w})\right] \quad (11)$$

Similar updates can be derived for τ (see App. B.2). We use these updates in Line 6 of Alg. C.1.

6.3 Experiments for Model B

Image experiments. We fine-tune 3 architectures: ResNet-50 (He et al., 2016), ViT-B/16 (Dosovitskiy et al., 2021), and ConvNeXt-Tiny (Liu et al., 2022), on 3 datasets with 8 different dataset sizes: CIFAR-10 (Krizhevsky et al., 2010) with $N \in \{100, 1000, 10000, 50000\}$; Flower-102 (Nilsback and Zisserman, 2008) with $N \in \{510, 1020\}$; and Pet-37 (Parkhi et al., 2012) with $N \in \{370, 3441\}$. All 3 architectures were pre-trained on ImageNet.

Text experiments. We fine-tune BERT-base (Devlin et al., 2019) on News-4 (Zhang et al., 2015) with $N \in \{400, 4000, 40000, 120000\}$. BERT-base was pre-trained on BooksCorpus and English Wikipedia.

Baselines. We compare our iso DE-ELBO to its natural ablation the iso ELBO, the gradient-based diagEF LA-LML (Immer et al., 2021), and the grid search-based diag EF LA-CLML (Lotfi et al., 2022).

We also compare to MAP + grid search (GS), which does MAP point estimation of w, V via separate SGD runs for each candidate λ, τ value in a fixed grid (see App. B.1), selecting the best according to the validation set likelihood. This GS baseline represents cutting-edge work in transfer learning (Shwartz-Ziv et al., 2022; Harvey et al., 2024). We further compare to a smarter search method: MAP + Bayesian optimization (BO; Hvarfner et al., 2024).

Common training plan. For each dataset size, we draw 3 separate random training sets of size N from the full training set, stratifying by class to ensure balanced class frequencies. We run each method on all 3 sets.

Results for diagEF LA-LML, iso ELBO, and iso DE-ELBO do not use a validation set. For all other methods, we hold out $\frac{1}{5}$ of the training set for validation, stratifying by class to ensure balanced class frequencies. After selecting the best hyperparameters, we retrain the model using the selected hyperparameters on the combined training and validation set.

For all methods, we perform minibatch SGD with a Nesterov momentum parameter of 0.9. For image experiments, we use a batch size of 128 with light data augmentation (random crops and horizontal flips) and train for 6000 steps using a cosine annealing learning rate (Loshchilov and Hutter, 2016). For text experiments, we use a batch size of 32 without any data augmentation and train for 12000 steps using a cosine annealing learning rate. Each run of our method and the baselines depends on the adequate selection of learning rate. All runs search over 4 candidate values and select the best according to a method-appropriate objective. See pseudocode in App. C for details on learning rate selection for the ELBO and our DE-ELBO.

Estimating ELBO and accuracy. After training, we estimate the expected log-likelihood term of the ELBO or our DE-ELBO for model selection by averaging over 10 samples from q. To compute classifier accuracy given q, we find that just plugging in the means $\bar{w}, \bar{V}$ to make predictions gives similar accuracy to averaging over 10 posterior samples without the added runtime cost.

6.4 Results for Model B

Across text and image datasets, several backbones, and several priors, our findings are:

The runtime of iso DE-ELBO is affordable, avoiding the extreme time costs of grid search. In Tab. 2, an individual SGD run of our iso DE-ELBO has comparable cost to one SGD run of standard MAP estimation. However, the cumulative cost of selecting η via grid search is far higher than our approach: L2-SP's search from Li et al. (2018) takes over 88 hours while PTYL's search (Shwartz-Ziv et al., 2022) takes over 148 hours. Our iso DE-ELBO delivers in under 3 hours in both cases by learning η via gradients.

The accuracy of iso DE-ELBO is competitive with or better than recent Bayesian methods. Fig. 3 shows test set accuracy over time for L2-SP transfer learning methods on select datasets and architectures for both image and text tasks. Our iso DE-ELBO consistently performs competitive with or better than recent Bayesian methods like Immer et al. (2021) or Lotfi et al. (2022), even though they use diagonal (not isotropic) posteriors. Lotfi et al. (2022)'s DiagEF LA-CLML performs similar to MAP + GS. For final test accuracy, negative log-likelihood (NLL), and expected calibration error (ECE) results, see App. B.6.

7 DISCUSSION AND CONCLUSION

We have proposed a practical approach to Bayesian model selection. Our modified ELBO objective enables per-epoch updates to hyperparameters on the full training set. Our solution is intended for a specific target scenario: where $D \gg N$ and q is simpler than the true posterior for tractability. In such scenarios, we provide analytical and experimental evidence for why the standard ELBO yields poor fits to the data, while our

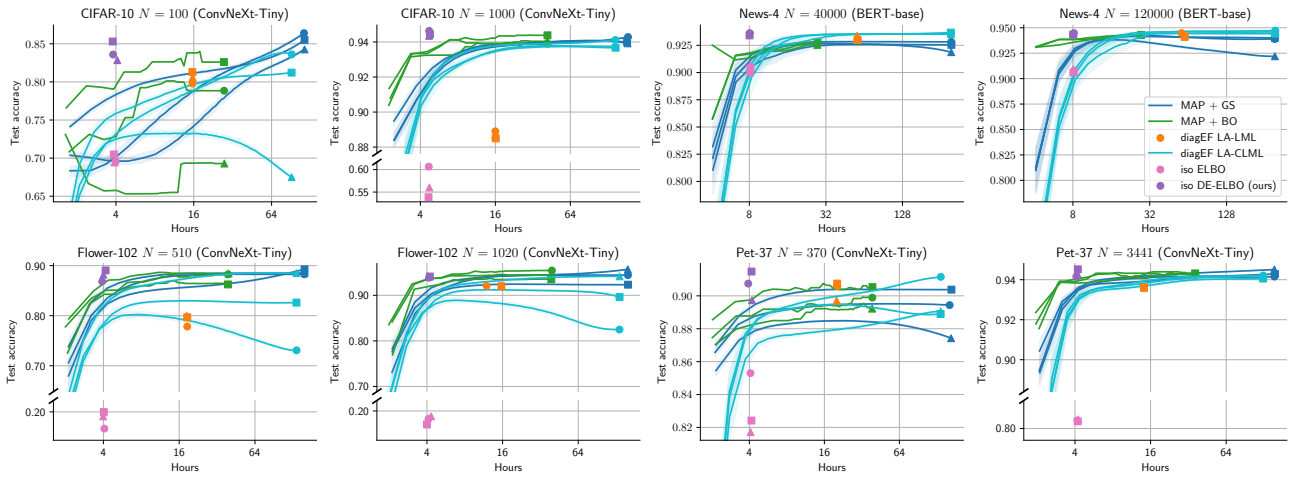


Figure 3: Test accuracy over time for L2-SP transfer learning of image and text classifiers. We run each method on 3 separate train sets of size N (3 marker styles). Each panel shows a distinct task: ConvNeXt-Tiny fine-tuned on CIFAR-10, Flower-102, and Pet-37; BERT-base fine-tuned on News-4. We compare MAP + GS, MAP + BO (Hvarfner et al., 2024), diagEF LA-LML (Immer et al., 2021), diagEF LA-CLML (Lotfi et al., 2022), iso ELBO, and iso DE-ELBO. **Takeaway: After just a few hours, iso DE-ELBO reaches as good or better performance at small data sizes and similar performance at large sizes, even when other methods are given many additional hours.** Further results in App. B examine ConvNeXt-Tiny (Fig. B.3), ViT-B/16 (Fig. B.4), ResNet-50 (Fig. B.5), and BERT-base (Fig. B.6).

Table 2: Timing for transfer learning methods. Task: ResNet-50 fine-tuned on CIFAR-10 with $N = 50000$. We compare MAP + GS, diagEF LA-LML (Immer et al., 2021), and iso DE-ELBO (ours). See App. B.1 for search details. **Takeaway: Each iso DE-ELBO run is 2x faster than diagEF LA-LML and learns λ, τ, avoiding the extreme costs of grid search.** Hardware: 4 Intel Xeon 6226R CPUs (2.90 GHz) and 1 NVIDIA A100 GPU (40 GB).

	Size of grid search (GS) space		L2-SP		PTYL	
Method	λ, τ	Learning rate	**Avg. SGD run**	**Total GS time**	**Avg. SGD run**	**Total GS time**
MAP + GS	36 (L2-SP) / 60 (PTYL)	4	37 min.	88.5 hr.	37 min.	148.7 hr.
diagEF LA-LML	Learned via gradients	4	70 min.	4.7 hr.	*PTYL prior covariance not supported*	
iso DE-ELBO (ours)	Learned via Eq. (11)	4	33 min.	2.2 hr.	36 min.	2.4 hr.

data-emphasized ELBO delivers better fits. We hope Bayesian deep learning researchers appreciate our approach's favorable comparisons to Immer et al. (2021) and Lotfi et al. (2022). We hope DNN practitioners appreciate the method's reliability at delivering accurate classifiers quickly while avoiding the variability of validation sets when N is small.

Limitations. The per-epoch updates to η in our DE-ELBO approach only work for continuous hyperparameters that explicitly appear in the prior or likelihood of a probabilistic model. Separate runs for each candidate would still be needed for discrete hyperparameters or optimization hyperparameters like learning rates, because gradients aren't available. DE-ELBO can still select among candidate runs for such values, as we do for learning rate in Alg. C.1.

Though we focus on Gaussian q with isotropic covariance, we conjecture the DE-ELBO would work reasonably with diagonal covariance. Further work is needed to consider low-rank or full-rank covariances, or non-Gaussian q. The closed-form prior variance updates are only possible because of the closed-form KL term. More rigorous theoretical understanding of the DE-ELBO is needed, including understanding of weaknesses like the underestimated variance far from data in Fig. 2 (d), a common issue in variational inference (Wilson and Izmailov, 2020). Our recommendation to set $\kappa = \frac{D}{N}$, though supported by Lemma 2 and experiments like Fig. B.1 and B.2, could use further support to understand its general applicability to other models or scenarios beyond those studied here. Especially in tasks that benefit from heavy data augmentation, adjusting the value of κ to account for this could be fruitful.

Outlook. The DE-ELBO avoids the need for any validation set and expensive separate runs. It can offer hours of saved time to practitioners, which could be used to further improve models. For example, on Pet-37 we found that L2-zero using an initialization from supervised pre-training results in an accuracy gain of 32.4 percentage points over self-supervised pre-training. Beyond saving valuable time, we hope our work sparks interest in theoretical understanding of modified ELBOs for improved model selection.

Acknowledgments

Authors EH and MCH gratefully acknowledge support in part from the Alzheimer's Drug Discovery Foundation and the National Institutes of Health (grant # R01NS134859). MCH is also supported in part by the U.S. National Science Foundation (NSF) via grant IIS # 2338962. We are thankful for computing infrastructure support provided by Research Technology Services at Tufts University, with hardware funded in part by NSF award OAC CC* # 2018149. We would like to thank Tim G. J. Rudner for helpful comments on an earlier draft of this paper.

References

Laurence Aitchison. A statistical theory of cold posteriors in deep neural networks. In *International Conference on Learning Representations (ICLR)*, 2021.

Isadora Antoniano-Villalobos and Stephen G. Walker. Bayesian Nonparametric Inference for the Power Likelihood. *Journal of Computational and Graphical Statistics*, 22(4):801–813, 2013. URL http://www.tandfonline.com/doi/abs/10.1080/10618600.2012.728511.

James Bergstra and Yoshua Bengio. Random Search for Hyper-Parameter Optimization. *Journal of Machine Learning Research (JMLR)*, 2012.

Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

David M. Blei, Alp Kucukelbir, and Jon D. McAuliffe. Variational Inference: A Review for Statisticians. *Journal of the American Statistical Association*, 112 (518):859–877, 2017.

Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight Uncertainty in Neural Networks. In *International Conference on Machine Learning (ICML)*, 2015.

Yuri Burda, Roger Grosse, and Ruslan Salakhutdinov. Importance Weighted Autoencoders. In *International Conference on Learning Representations (ICLR)*, 2016.

Ciprian Chelba and Alex Acero. Adaptation of Maximum Entropy Capitalizer: Little Data Can Help a Lot. *Computer Speech & Language*, 20(4):382–399, 2006.

Badr-Eddine Cherief-Abdellatif. Consistency of ELBO maximization for model selection. In *Proceedings of The 1st Symposium on Advances in Approximate Bayesian Inference*, 2019. URL https://proceedings.mlr.press/v96/cherief-abdellatif19a.html.

Beau Coker, Wessel P. Bruinsma, David R. Burt, Weiwei Pan, and Finale Doshi-Velez. Wide mean-field bayesian neural networks ignore the data. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2022.

Andreas Damianou and Neil Lawrence. Deep Gaussian Processes. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2013. URL http://proceedings.mlr.press/v31/damianou13a.html.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 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 Jakob. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *International Conference on Learning Representations (ICLR)*, 2021.

Nial Friel and Anthony N. Pettitt. Marginal Likelihood Estimation via Power Posteriors. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 70(3):589–607, 2008. URL https://academic.oup.com/jrsssb/article/70/3/589/7109555.

Peter Grünwald. A Tutorial Introduction to the Minimum Description Length Principle. *Advances in Minimum Description Length: Theory and Applications*, 2005.

Peter Grünwald. The Safe Bayesian: learning the learning rate via the mixability gap. In *International Conference on Algorithmic Learning Theory*, pages 169–183. Springer, 2012.

Peter Grünwald and Thijs van Ommen. Inconsistency of Bayesian Inference for Misspecified Linear Models, and a Proposal for Repairing It. *Bayesian Analysis*, 12(4):1069–1103, 2017. URL https://projecteuclid.org/journals/bayesian-analysis/volume-12/issue-4/Inconsistency-of-Bayesian-Inference-for-Misspecified-Linear-Models-and-a/10.1214/17-BA1085.full.

Gregory W. Gundersen, Michael Minyi Zhang, and Barbara E. Engelhardt. Latent variable modeling with random features. In *Artificial Intelligence and Statistics (AISTATS)*, 2021.

Ethan Harvey, Mikhail Petrov, and Michael C. Hughes. Transfer Learning with Informative Priors: Simple Baselines Better than Previously Reported. *Transactions on Machine Learning Research (TMLR)*, 2024. ISSN 2835-8856. URL https://openreview.net/

forum?id=BbvSU02jLg. Reproducibility Certification.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

James Hensman, Alexander Matthews, and Zoubin Ghahramani. Scalable Variational Gaussian Process Classification. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2015.

Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. β-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework. In *International Conference on Learning Representations (ICLR)*, 2017.

Carl Hvarfner, Erik Orm Hellsten, and Luigi Nardi. Vanilla bayesian optimization performs great in high dimensions. In *International Conference on Machine Learning (ICML)*, 2024.

Alexander Immer, Matthias Bauer, Vincent Fortuin, Gunnar Rätsch, and Khan Mohammad Emtiyaz. Scalable Marginal Likelihood Estimation for Model Selection in Deep Learning. In *International Conference on Machine Learning (ICML)*, 2021.

Kevin Jamieson and Ameet Talwalkar. Non-stochastic best arm identification and hyperparameter optimization. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2016.

Harold Jeffreys. *The Theory of Probability*. The Clarendon Press, Oxford, 1939.

Michael I. Jordan, Zoubin Ghahramani, Tommi S Jaakkola, and Lawrence K Saul. An Introduction to Variational Methods for Graphical Models. *Machine Learning*, 37:183–233, 1999.

Sanyam Kapoor, Wesley J. Maddox, Pavel Izmailov, and Andrew G. Wilson. On Uncertainty, Tempering, and Data Augmentation in Bayesian Classification. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Zohar Karnin, Tomer Koren, and Oren Somekh. Almost Optimal Exploration in Multi-Armed Bandits. In *International Conference on Machine Learning (ICML)*, 2013.

Kishan KC, Rui Li, and MohammadMahdi Gilany. Joint Inference for Neural Network Depth and Dropout Regularization. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

Ranganath Krishnan, Mahesh Subedar, and Omesh Tickoo. Efficient Priors for Scalable Variational Inference in Bayesian Deep Neural Networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, 2019.

Ranganath Krishnan, Mahesh Subedar, and Omesh Tickoo. Specifying Weight Priors in Bayesian Deep Neural Networks with Empirical Bayes. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020.

Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. CIFAR-10 (Canadian Institute for Advanced Research). 2010. URL http://www.cs.toronto.edu/~kriz/cifar.html.

Vidhi Lalchand, Wessel Bruinsma, David Burt, and Carl Edward Rasmussen. Sparse Gaussian Process Hyperparameters: Optimize or Integrate? In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Xuhong Li, Yves Grandvalet, and Franck Davoine. Explicit Inductive Bias for Transfer Learning with Convolutional Networks. In *International Conference on Machine Learning (ICML)*, 2018.

Jeremiah Liu, Zi Lin, Shreyas Padhy, Dustin Tran, Tania Bedrax Weiss, and Balaji Lakshminarayanan. Simple and Principled Uncertainty Estimation with Deterministic Deep Learning via Distance Awareness. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Jeremiah Zhe Liu, Shreyas Padhy, Jie Ren, Zi Lin, Yeming Wen, Ghassen Jerfel, Zachary Nado, Jasper Snoek, Dustin Tran, and Balaji Lakshminarayanan. A Simple Approach to Improve Single-Model Deep Uncertainty via Distance-Awareness. *Journal of Machine Learning Research (JMLR)*, 24(42):1–63, 2023.

Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

Jonathan Lorraine, Paul Vicol, and David Duvenaud. Optimizing Millions of Hyperparameters by Implicit Differentiation. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2020. URL https://proceedings.mlr.press/v108/lorraine20a/lorraine20a.pdf.

Ilya Loshchilov and Frank Hutter. SGDR: Stochastic Gradient Descent with Warm Restarts. In *International Conference on Learning Representations (ICLR)*, 2016.

Sanae Lotfi, Pavel Izmailov, Gregory Benton, Micah Goldblum, and Andrew Gordon Wilson. Bayesian Model Selection, the Marginal Likelihood, and Generalization. In *International Conference on Machine Learning (ICML)*, 2022.

David J. C. MacKay. Bayesian Model Comparison and Backprop Nets. In *Advances in Neural Information Processing Systems (NeurIPS)*, 1991.

David J. C. MacKay. Hyperparameters: Optimize, or Integrate Out? In Glenn R. Heidbreder, editor, *Maximum Entropy and Bayesian Methods*, pages 43–59. Kluwer Academic Publishers, Dordrecht, 1996.

Dougal Maclaurin, David Duvenaud, and Ryan P. Adams. Gradient-Based Hyperparameter Optimization through Reversible Learning. In *International Conference on Machine Learning (ICML)*, 2015. URL https://proceedings.mlr.press/v37/maclaurin15.html.

Wesley J. Maddox, Pavel Izmailov, Timur Garipov, Dmitry P. Vetrov, and Andrew Gordon Wilson. A Simple Baseline for Bayesian Uncertainty in Deep Learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

Stephan Mandt, James McInerney, Farhan Abrol, Rajesh Ranganath, and David Blei. Variational tempering. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2016.

Yann McLatchie, Edwin Fong, David T. Frazier, and Jeremias Knoblauch. Predictive performance of power posteriors. *Biometrika*, 112(3):asaf034, 2025.

Jeffrey W. Miller and David B. Dunson. Robust Bayesian Inference via Coarsening. *Journal of the American Statistical Association*, 2019. URL http://arxiv.org/abs/1506.06101.

Shakir Mohamed, Mihaela Rosca, Michael Figurnov, and Andriy Mnih. Monte Carlo Gradient Estimation in Machine Learning. *Journal of Machine Learning Research (JMLR)*, 21(132), 2020. URL http://jmlr.org/papers/v21/19-346.html.

Kevin S. Murphy. *Probabilistic Machine Learning: An Introduction*, chapter 6.2.3 Example: KL divergence between two Gaussians. MIT Press, 2022.

Radford M. Neal. *Bayesian Learning for Neural Networks*. Springer Science & Business Media, 1996.

Maria-Elena Nilsback and Andrew Zisserman. Automated Flower Classification over a Large Number of Classes. In *2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing (ICVGIP)*, 2008.

Kazuki Osawa, Siddharth Swaroop, Mohammad Emtiyaz E. Khan, Anirudh Jain, Runa Eschenhagen, Richard E. Turner, and Rio Yokota. Practical Deep Learning with Bayesian Principles. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

Omkar M. Parkhi, Andrea Vedaldi, Andrew Zisserman, and C. V. Jawahar. Cats And Dogs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

Konstantinos Pitas and Julyan Arbel. The Fine Print on Tempered Posteriors. In *Asian Conference on Machine Learning*, 2024.

Andres Potapczynski, Luhuan Wu, Dan Biderman, Geoff Pleiss, and John P Cunningham. Bias-Free Scalable Gaussian Processes via Randomized Truncations. In *International Conference on Machine Learning (ICML)*, 2021.

Ali Rahimi and Benjamin Recht. Random Features for Large-Scale Kernel Machines. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2007.

Sebastian Raschka. Model Evaluation, Model Selection, and Algorithm Selection in Machine Learning. *arXiv preprint arXiv:1811.12808*, 2018. URL http://arxiv.org/abs/1811.12808.

Carl Edward Rasmussen and Zoubin Ghahramani. Occam's Razor. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2000.

Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning*, chapter 5.2 Bayesian Model Selection. The MIT Press, 2006a.

Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning*. The MIT Press, 2006b.

Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning*, chapter 2.3 Varying the Hyperparameters. The MIT Press, 2006c.

Tim G. J. Rudner, Xiang Pan, Yucen Lily Li, Ravid Shwartz-Ziv, and Andrew Gordon Wilson. Fine-Tuning with Uncertainty-Aware Priors Makes Vision and Language Foundation Models More Reliable. In *ICML Workshop on Structured Probabilistic Inference & Generative Modeling (SPIGM@ICML)*, 2025. URL https://openreview.net/forum?id=37fM2QEBSE.

Ravid Shwartz-Ziv, Micah Goldblum, Hossein Souri, Sanyam Kapoor, Chen Zhu, Yann LeCun, and Andrew G. Wilson. Pre-Train Your Loss: Easy Bayesian Transfer Learning with Informative Priors. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/b1e7f61f40d68b2177857bfcb195a507-Paper-Conference.pdf.

Jasper Snoek, Hugo Larochelle, and Ryan P. Adams. Practical bayesian optimization of machine learning algorithms. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2012.

Michalis Titsias. Variational Learning of Inducing Variables in Sparse Gaussian Processes. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2009.

Ryan Turner, David Eriksson, Michael McCourt, Juha Kiili, Eero Laaksonen, Zhen Xu, and Isabelle Guyon. Bayesian Optimization is Superior to Random Search for Machine Learning Hyperparameter Tuning: Analysis of the Black-Box Optimization Challenge 2020. In *NeurIPS 2020 Competition and Demonstration Track*, 2021.

Naonori Ueda and Zoubin Ghahramani. Bayesian Model Search for Mixture Models Based on Optimizing Variational Bounds. *Neural Networks*, 15(1): 1223–1241, 2002.

Zi Wang, George E Dahl, Kevin Swersky, Chansoo Lee, Zachary Nado, Justin Gilmer, Jasper Snoek, and Zoubin Ghahramani. Pre-trained gaussian processes for bayesian optimization. *Journal of Machine Learning Research (JMLR)*, 2024. URL https://jmlr.org/papers/volume25/23-0269/23-0269.pdf.

Florian Wenzel, Kevin Roth, Bastiaan S. Veeling, Jakub Świątkowski, Linh Tran, Stephan Mandt, Jasper Snoek, Tim Salimans, Rodolphe Jenatton, and Sebastian Nowozin. How Good is the Bayes Posterior in Deep Neural Networks Really? In *International Conference on Machine Learning (ICML)*, 2020.

Andrew G. Wilson and Pavel Izmailov. Bayesian Deep Learning and a Probabilistic Perspective of Generalization. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Max A. Woodbury. *Inverting Modified Matrices*. Department of Statistics, Princeton University, 1950.

Ming Xu, Matias Quiroz, Robert Kohn, and Scott A. Sisson. Variance Reduction Properties of the Reparameterization Trick. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2019.

Guodong Zhang, Shengyang Sun, David Duvenaud, and Roger Grosse. Noisy Natural Gradient as Variational Inference. In *International Conference on Machine Learning (ICML)*, 2018.

Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level Convolutional Networks for Text Classification. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2015.

Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. Yes.
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. Yes, see Tab. 2.
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. Yes, see URL on page 1.

2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. Yes, see Sec. 5.2.
   (b) Complete proofs of all theoretical results. Yes, see App. A.4 and A.5.
   (c) Clear explanations of any assumptions. Yes, see Sec. 5.2.

3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). Yes, see URL on page 1.
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). Yes, see App. B.1.
   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). Yes, see captions of Tab. B.6, B.7, B.7, and B.8.
   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). Yes, see caption of Tab. 2.

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
   (a) Citations of the creator If your work uses existing assets. Yes, we cite the creators of used assets.
   (b) The license information of the assets, if applicable. Yes, see license at URL on page 1.
   (c) New assets either in the supplemental material or as a URL, if applicable. Yes, see URL on page 1.
   (d) Information about consent from data providers/curators. Yes, datasets are publicly available.
   (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. Not Applicable.

5. If you used crowdsourcing or conducted research with human subjects, check if you include:
   (a) The full text of instructions given to participants and screenshots. Not Applicable.
   (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. Not Applicable.
   (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. Not Applicable.

Appendix

C PSEUDOCODE

Below in Alg. C.1, we give a practical procedure for estimating an approximate posterior q and key hyperparameters given a dataset of N observations using our DE-ELBO objective.

This algorithm can handle two kinds of hyperparameters:

- model hyperparameters η that impact the prior or likelihood, via gradient/closed-form updates each epoch,
- and other hyperparameters like learning rates that impact optimization quality, via grid search.

For the former, we can do gradient-based learning of η (see bottom case of Line 6 in code below), or sometimes closed-form updates when setting the analytical gradient to zero and solving is feasible (see Case Study B for an example, especially App. B.2). For the latter, DE-ELBO can be used to select learning rate, but the objective is not an explicit function of learning rate and so gradient-based learning is not possible. Instead, the outer loop in the code below compares ultimate objective function values at a fixed grid of lr values.

For simplicity, we assume in the pseudocode that the dataset N is small enough that stochastic minibatches are not necessary. It is straightforward to extend this algorithm to minibatches (see subsection below).

The algorithm uses automatic differentiation (AD) for computing all gradients, as well as the reparameterization trick for estimating gradients of the expected log-likelihood term of the ELBO. We use the closed-form KL between the two multivariate Normal distributions (which presumes the prior and q are Normal).

Algorithm C.1 Gradient Ascent to Estimate q and Hyperparameters with the Data-Emphasized ELBO

Input:
- training set $\{y_i\}_{i=1}^N$
- likelihood $p_\eta(y_i|\theta)$
- prior $p_\eta(\theta)$
- initial value ψ_0 for parameters $\psi = \{\bar{\theta}, \bar{\sigma}_q\}$ that define the approximate posterior $q_\psi(\theta) = \mathcal{N}(\theta|\bar{\theta}, \bar{\sigma}_q^2 I_D)$
- initial value η_0 for hyperparameters η
- data-emphasis factor κ. Set $\kappa = \frac{D}{N}$ for our recommended DE-ELBO. Instead, $\kappa = 1$ recovers standard ELBO.
- set of candidate learning rates C_{lr}

Output:
- estimated parameters ψ for the approximate posterior q
- estimated hyperparameters η

Procedure:

1: **for** each lr in C_{lr}:
2: $\quad \psi \leftarrow \psi_0, \eta \leftarrow \eta_0$
3: $\quad$ **for** each epoch until converged:
4: $\quad\quad \varepsilon \sim \mathcal{N}(0_D, I_D)$ *We do just 1 MC sample per train step; could do more if affordable.*
5: $\quad\quad \psi \leftarrow \psi + \mathsf{lr}\left(\kappa \cdot \sum_{i=1}^N \underbrace{\nabla_\psi \log p_\eta(y_i|\theta = \bar{\theta} + \bar{\sigma}_q \epsilon)}_{\text{Reparameterization trick and AD}} - \underbrace{\nabla_\psi D_{\mathrm{KL}}(q_\psi(\theta)\|p_\eta(\theta))}_{\text{AD on KL of two Normals}}\right)$
6: $\quad\quad \eta \leftarrow \begin{cases} \eta^*(\psi, \{y_i\}_{i=1}^N) & \text{if closed-form update exists} \\ \eta + \mathsf{lr}\left(\kappa \cdot \sum_{i=1}^N \nabla_\eta \log p_\eta(y_i|\theta = \bar{\theta} + \bar{\sigma}_q \epsilon) - \nabla_\eta D_{\mathrm{KL}}(q_\psi(\theta)\|p_\eta(\theta))\right) & \text{otherwise} \end{cases}$
7: $\quad$ Calculate $J_{\text{DE-ELBO}}$ for final values of ψ, η on training set, using 10 MC samples for log-likelihood term
8: Return values ψ, η corresponding to lr that maximized $J_{\text{DE-ELBO}}$

Choosing the number of samples to use at each training step and for the final learning rate selection decision is up to the user. We recommend using the largest values that can complete training in affordable time limits. We used 1 and 10 samples, respectively, throughout our experiments.

C.1 Handling Positivity Constraints

The gradient updates as written in Alg. C.1 do not respect the constrained domains of some parameters. For the chosen isotropic Gaussian family for q, within ψ there is a scalar variance that must be positive: $\bar{\sigma}_q > 0$. Similarly, within η there are often continuous hyperparameters with constrained domains (e.g., lengthscales and outputscales must be positive in Case Study A, variances must be positive in Case Study B).

To handle positivity constraints, we reparameterize in terms of unconstrained parameters via a one-to-one mapping. For example: $\bar{\sigma}_q = \text{softplus}(\bar{u}_q)$, where $\bar{u}_q \in \mathbb{R}$ and the softplus function is defined as in the PyTorch documentation: https://docs.pytorch.org/docs/stable/generated/torch.nn.Softplus.html.

Other constrained parameters could be handled via similarly appropriate mappings.

C.2 Extension to Minibatches

For learning on large datasets, as in Case Study B, we need to perform gradient updates on minibatches of a few examples at a time, rather than all N examples in the training set. We approximate the sum over all data in Lines 5 and 6 of Alg. C.1 with a rescaled sum over all items in one minibatch $\mathcal{B}$:

$$\sum_{i=1}^{N} \log p_\eta(y_i|\theta_s) \approx \frac{N}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} \log p_\eta(y_i|\theta_s). \tag{12}$$

A APPENDIX FOR CASE STUDY A: FOURIER REGRESSION

A.1 RFFs with Arbitrary Lengthscale and Outputscale for Model A

Random Fourier features (RFFs) are typically used to approximate an RBF kernel with fixed lengthscale $\ell_k = 1$ and output scale $\sigma_k = 1$. Below we show that our RFF construction in Sec. 5.1 allows a Monte Carlo approximation of the RBF kernel for any $\ell_k > 0, \sigma_k > 0$. Our proof extends a proof for the base case ($\ell_k = 1, \sigma_k = 1$) found in a blogpost *Random Fourier Features* by Gregory Gundersen: https://gregorygundersen.com/blog/2019/12/23/random-fourier-features/, and likely informed work on RFFs for latent variable models in Gundersen et al. (2021).

First, recall basic notation and dimensionality assumptions. Each original feature vector is $x \in \mathbb{R}^H$. At algorithm startup, we draw weights $A \in \mathbb{R}^{H \times R}$ from $A_{h,r} \sim \mathcal{N}(0,1)$ for all $h \in [H]$ and all $r \in [R]$. Similarly, we draw bias weights $b \in \mathbb{R}^R$ from $b_r \sim \text{Unif}(0, 2\pi)$ for all $r \in [R]$. We typically assume the user-controlled size of the RFF feature space R is rather large. As $R \to \infty$, the approximate equalities marked by $\approx$ below become increasingly accurate (in expectation).

For any pair of original feature vectors x, x', after applying our RFF transformation $\phi(\cdot)$ in Eq. (4) the inner product of the pair in the transformed space is:

$$\phi(x)^\top \phi(x') = \frac{\sigma_k^2}{R} \sum_{r=1}^{R} 2 \cos\left(\frac{1}{\ell_k} A_{:,r}^\top x + b_r\right) \cos\left(\frac{1}{\ell_k} A_{:,r}^\top x' + b_r\right) \qquad \text{By definition of } \phi(\cdot) \tag{13}$$

$$= \frac{\sigma_k^2}{R} \sum_{r=1}^{R} \cos\left(\frac{1}{\ell_k} A_{:,r}^\top (x + x') + 2b_r\right) + \cos\left(\frac{1}{\ell_k} A_{:,r}^\top (x - x')\right) \qquad \text{Sum of angles formula} \tag{14}$$

$$\approx \frac{\sigma_k^2}{R} \sum_{r=1}^{R} \cos\left(\frac{1}{\ell_k} A_{:,r}^\top (x - x')\right) \qquad \mathbb{E}[\cos(t+b)] = 0 \text{ for any uniform r.v. } b \text{ with a } 2\pi \text{ interval length and any scalar } t \tag{15}$$

$$\approx \frac{\sigma_k^2}{R} \sum_{r=1}^{R} \cos\left(\frac{1}{\ell_k} A_{:,r}^\top (x - x')\right) + i \sin\left(\frac{1}{\ell_k} A_{:,r}^\top (x - x')\right) \qquad \mathbb{E}[\sin(a)] = 0 \text{ for any zero-mean Gaussian r.v. } a \tag{16}$$

$$\approx \frac{\sigma_k^2}{R} \sum_{r=1}^{R} \exp\left(i \frac{1}{\ell_k} A_{:,r}^\top (x - x')\right) \qquad \text{Euler's formula} \tag{17}$$

The above sum over R samples can be viewed as an unbiased estimate of an expectation of a complex exponential:

$$\frac{\sigma_k^2}{R}\sum_{r=1}^{R}\exp\left(i\frac{1}{\ell_k}A_{:,r}^\top(x-x')\right)\approx\sigma_k^2\mathbb{E}_{p(\omega)}\left[\exp(i\omega^\top(x-x'))\right],\qquad\text{where } p(\omega)=\mathcal{N}(\omega|0_H,\tfrac{1}{\ell_k^2}I_H).\tag{18}$$

Here, the random variable is a vector $\omega\in\mathbb{R}^H$, where each entry is distributed as $\omega_h\sim\mathcal{N}(0,\frac{1}{\ell_k^2})$ for all $h\in[H]$. Recall that one way to sample values of ω is in two steps: first draw vector $A_{:,r}\in\mathbb{R}^H$ from a standard zero-mean, identity-covariance Normal in H dimensions (which matches our RFF procedure for A), then set $\omega\leftarrow\frac{1}{\ell_k}A_{:,r}$. This two-step sampling procedure justifies the approximation in Eq. (18).

Now, we show that the expectation of the complex exponential on the right-side of Eq. (18) is equivalent to the stationary RBF kernel $k_{\text{RBF},\ell_k,\sigma_k}(\delta)$ where $\delta=x-x'$ is an H-dimensional vector. The lengthscale ℓ_k and outputscale σ_k values of the kernel match those used to construct the RFF features $\phi(\cdot)$ above.

$\sigma_k^2\mathbb{E}_{p(\omega)}[\exp(i\omega^\top\delta)]$

$$=\sigma_k^2\int p(\omega)\exp\left(i\omega^\top\delta\right)d\omega\qquad\text{Expectation as integral}\tag{19}$$

$$=\sigma_k^2\int\left(\frac{\ell_k^2}{2\pi}\right)^{H/2}\exp\left(-\frac{\ell_k^2\omega^\top\omega}{2}\right)\exp\left(i\omega^\top\delta\right)d\omega\qquad\text{By definition of } p(\omega)\tag{20}$$

$$=\sigma_k^2\left(\frac{\ell_k^2}{2\pi}\right)^{H/2}\int\exp\left(-\frac{\ell_k^2\omega^\top\omega}{2}+i\omega^\top\delta\right)d\omega\qquad\text{Simplify}\tag{21}$$

$$=\sigma_k^2\left(\frac{\ell_k^2}{2\pi}\right)^{H/2}\int\exp\left(-\frac{\ell_k^2\omega^\top\omega}{2}+i\omega^\top\delta+\frac{\delta^\top\delta}{2\ell_k^2}-\frac{\delta^\top\delta}{2\ell_k^2}\right)d\omega\qquad\text{Add and subtract same term}\tag{22}$$

$$=\sigma_k^2\left(\frac{\ell_k^2}{2\pi}\right)^{H/2}\exp\left(-\frac{\delta^\top\delta}{2\ell_k^2}\right)\int\exp\left(-\frac{\ell_k^2}{2}\left(\omega-\frac{i}{\ell_k^2}\delta\right)^\top\left(\omega-\frac{i}{\ell_k^2}\delta\right)\right)d\omega\qquad\text{Complete the square}\tag{23}$$

$$=\sigma_k^2\left(\frac{\ell_k^2}{2\pi}\right)^{H/2}\exp\left(-\frac{\delta^\top\delta}{2\ell_k^2}\right)\left(\frac{2\pi}{\ell_k^2}\right)^{H/2}\qquad\text{Closed-form integral}\tag{24}$$

$$=\sigma_k^2\exp\left(-\frac{\delta^\top\delta}{2\ell_k^2}\right)\qquad\text{Reciprocal terms cancel out}\tag{25}$$

$$=k_{\text{RBF},\ell_k,\sigma_k}(\delta)\qquad\text{By definition of RBF kernel}\tag{26}$$

This completes the proof. Our constructed RFF features with arbitrary lengthscale and outputscale in Eq. (4) provide an unbiased approximation of the corresponding RBF kernel that is increasingly accurate as $R\to\infty$.

A.2 Regression Model A Definition

A Bayesian interpretation of the RFF problem assumes a joint probabilistic model $p(y_{1:N},v)=p(v)p(y_{1:N}|v)$, with factors for the regression case

$$p(v)=\mathcal{N}(v|0_R,I_R),\qquad p(y_{1:N}|v)=\mathcal{N}(y_{1:N}|\Phi v,\sigma_y^2I_N)\tag{27}$$

where $y_{1:N}\in\mathbb{R}^N$ and $\Phi=\Phi(x_{1:N})\in\mathbb{R}^{N\times R}$. Note that including a prior variance hyperparameter different from 1 for $p(v)$ does not add an additional degree of freedom when Φ includes an outputscale hyperparameter. Both factors would be redundant in the inner product $v^\top\phi(x_i)$.

Marginal likelihood. Given the marginal Gaussian distribution for v and the conditional Gaussian distribution for $y_{1:N}$ given v, the marginal distribution of $y_{1:N}$ under our RFF regression model is:

$$p(y_{1:N})=\mathcal{N}(y_{1:N}|0_N,\sigma_y^2I_N+\Phi\Phi^\top)\qquad\text{by Eq. (2.115) from Bishop (2006).}\tag{28}$$

This is *exactly the same* marginal likelihood as the classic Gaussian process with RBF kernel regression model, where we have latent function values $f\sim\mathcal{N}(0_N,K)$ and then observed targets $y|f\sim\mathcal{N}(f,\sigma_y^2I_N)$, so long as the approximation $K\approx\Phi\Phi^\top$ is accurate.

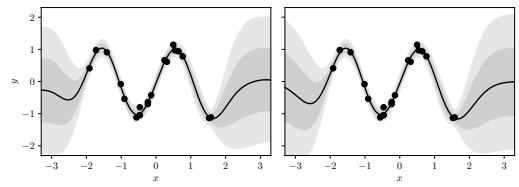
Ideal posterior. The conditional distribution of v given y is available in closed form:

$$p(v|y_{1:N}) = \mathcal{N}(v|\tfrac{1}{\sigma_y^2}\Sigma_{\text{post}}\Phi^\top y_{1:N}, \Sigma_{\text{post}}) \qquad \text{by Eq. (2.116) from Bishop (2006)} \tag{29}$$

where

$$\Sigma_{\text{post}} = (I_R + \tfrac{1}{\sigma_y^2}\Phi^\top\Phi)^{-1} \qquad \text{by Eq. (2.117) from Bishop (2006).} \tag{30}$$

The posterior predictive $p(y_*|y_{1:N})$ of the RFF regression model again should match the GP with corresponding RBF kernel, provided the user-controlled rank R is large enough. See Fig. A.1 for a visual demonstration.



(a) RFF (learned ℓ_k, σ_k) LML (b) GP (learned ℓ_k, σ_k) LML

Figure A.1: Left: A Monte Carlo approximation of the RFF predictive posterior $p(y_*|y_{1:N}) = \int_v p(y_*|v)p(v|y_{1:N})dv$ by sampling from the true posterior $p(v|y_{1:N})$. Right: The closed-form GP predictive posterior $p(y_*|y_{1:N})$.

A.3 Closed-Form Optimal ψ for Full-Rank q Under ELBO for Regression Model A

Suppose we assume a full-rank covariance for q, rather than isotropic: $q(v) = \mathcal{N}(v|\mu_q, \Sigma_q)$. In this case, the closed-form ELBO for regression Model A is

$$J_{\text{ELBO}} = -\frac{1}{2}\left[N\log(2\pi\sigma_y^2) + \frac{1}{\sigma_y^2}\|y_{1:N} - \Phi\mu_q\|_2^2 + \frac{1}{\sigma_y^2}\operatorname{tr}(\Phi\Sigma_q\Phi^\top) + \operatorname{tr}(\Sigma_q) + \|\mu_q\|_2^2 - R - \log\det(\Sigma_q)\right]. \tag{31}$$

We can solve for Σ_q by taking the gradient of J_{ELBO} with respect to Σ_q. The gradient is

$$\nabla_{\Sigma_q} J_{\text{ELBO}} = -\frac{1}{2}\left[\frac{1}{\sigma_y^2}\Phi^\top\Phi + I_R - \Sigma_q^{-1}\right]. \tag{32}$$

Setting $\nabla_{\Sigma_q} J_{\text{ELBO}} = 0$ and solving for Σ_q, we get

$$\Sigma_q^* = (I_R + \tfrac{1}{\sigma_y^2}\Phi^\top\Phi)^{-1} \tag{33}$$

which is exactly the ideal posterior's covariance matrix.

A.4 Closed-Form Optimal ψ for Isotropic q Under ELBO for Regression Model A

If we assume $\Sigma_q = \bar{\sigma}_q^2 I_R$, we can simplify the ELBO to

$$J_{\text{ELBO}} = -\frac{1}{2}\left[N\log(2\pi\sigma_y^2) + \frac{1}{\sigma_y^2}\|y_{1:N} - \Phi\mu_q\|_2^2 + \frac{\bar{\sigma}_q^2}{\sigma_y^2}\operatorname{tr}(\Phi\Phi^\top) + \bar{\sigma}_q^2 R + \|\mu_q\|_2^2 - R - \log\bar{\sigma}_q^{2R}\right]. \tag{34}$$

We can solve for $\bar{\sigma}_q^2$ by taking the gradient of J_{ELBO} with respect to $\bar{\sigma}_q^2$. The gradient is

$$\nabla_{\bar{\sigma}_q^2} J_{\text{ELBO}} = -\frac{1}{2}\left[\frac{1}{\sigma_y^2}\operatorname{tr}(\Phi\Phi^\top) + R - \frac{1}{\bar{\sigma}_q^2}R\right]. \tag{35}$$

Setting $\nabla_{\bar{\sigma}_q^2} J_{\text{ELBO}} = 0$ and solving for $\bar{\sigma}_q^2$, we get

$$\bar{\sigma}_q^{2*} = \frac{R}{\frac{1}{\sigma_y^2}\operatorname{tr}(\Phi\Phi^\top) + R}. \tag{36}$$

A.5 Closed-Form Optimal ψ for Isotropic q Under DE-ELBO for Regression Model A

Here we again assume an isotropic covariance for q, but now focus on the DE-ELBO. We can solve for $\bar{\sigma}_q^2$ by taking the gradient of $J_{\text{DE-ELBO}}$ ($\kappa = \frac{R}{N}$) with respect to $\bar{\sigma}_q^2$. The gradient is

$$\nabla_{\bar{\sigma}_q^2} J_{\text{DE-ELBO}} = -\frac{1}{2}\left[\frac{R}{N}\frac{1}{\sigma_y^2}\operatorname{tr}(\Phi\Phi^\top) + R - \frac{1}{\bar{\sigma}_q^2}R\right]. \tag{37}$$

Setting $\nabla_{\bar{\sigma}_q^2} J_{\text{DE-ELBO}} = 0$ and solving for $\bar{\sigma}_q^2$, we get

$$\bar{\sigma}_q^{2*} = \frac{R}{\frac{R}{N}\frac{1}{\sigma_y^2}\operatorname{tr}(\Phi\Phi^\top) + R}. \tag{38}$$

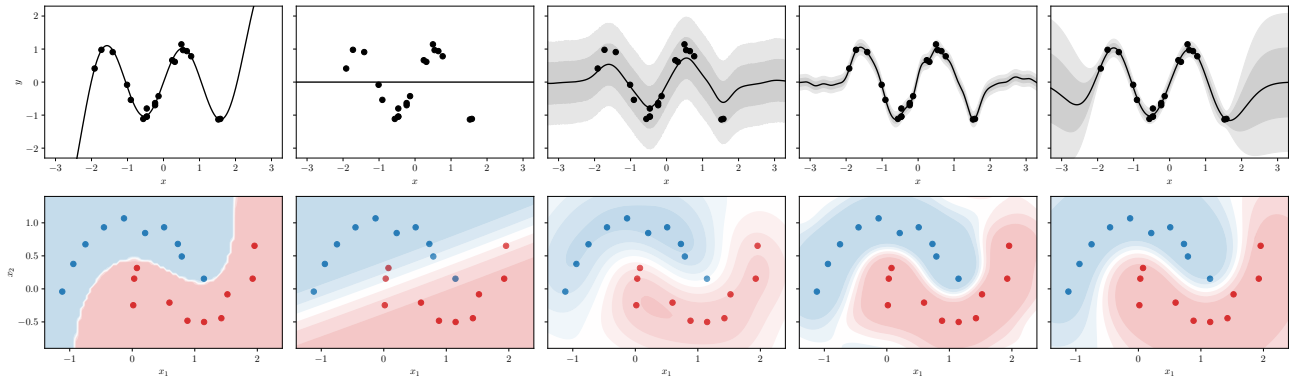
A.6 Classification Model A Definition

Building on the regression model above, we now consider an RFF classifier for when observed data are discrete labels $y_i \in \{1, 2, \ldots, C\}$. A Bayesian interpretation of this problem assumes a joint probabilistic model $p(y_{1:N}, V) = p(V)\prod_i p(y_i|V)$, with factors for the classification case

$$p(V) = \mathcal{N}(\operatorname{vec}(V)|0_{RC}, I_{RC}), \qquad p(y_i|V) = \operatorname{Cat}(y_i|\text{SM}(V\phi(x_i))). \tag{39}$$

Unlike the regression case, there is no known analytical formula for the posterior, due to the non-linear softmax function preventing conjugacy.

A.7 Results for Model A: Regression and Classification



(a) RFF ($\ell_k = 1$) MAP + GS (b) RFF ($\ell_k = 20$) MAP + GS (c) RFF (learned ℓ_k, σ_k) ELBO (d) RFF (learned ℓ_k, σ_k) DE-ELBO (ours) (e) GP (learned ℓ_k, σ_k) LML

Figure A.2: Demo of hyperparameter sensitivity and selection for RFF models. The first four columns use the RFF regression model with isotropic Gaussian q in Sec. 5.1, varying estimation and selection techniques. The last column shows the reference fit of a GP's exact posterior, a gold-standard for this toy data but less scalable. For regression, we plot the mean and two standard deviations for the predictive posterior $p(y_*|y_{1:N})$. Our DE-ELBO objective best approximates the GP, though underestimates variance far from data.

A.8 Varying κ for Model A

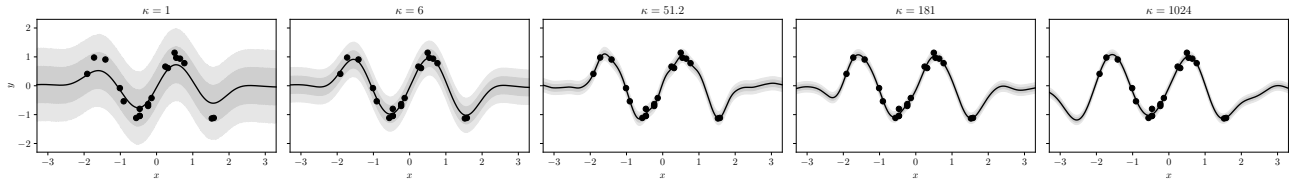


Figure A.3: Predictive posteriors on a univariate regression dataset using our *data-emphasized ELBO* (DE-ELBO) with various κ values ($\frac{D}{N} = 51.2$). We use Adam with the closed-form expected log-likelihood. The predictive posterior is slightly better compared to Adam with a Monte Carlo approximation of the expected log-likelihood (see Fig. 2 (c)).

B APPENDIX FOR CASE STUDY B: TRANSFER LEARNING

B.1 Implementation Details for Model B

For MAP + GS, we select the initial learning rate from {0.1, 0.01, 0.001, 0.0001} and α, β (or equivalently λ, τ). For L2-zero, we select $\frac{\alpha}{N} = \frac{\beta}{N}$ from {0.01, 0.001, 0.0001, 1e-5, 1e-6, 0.0}. For L2-SP, we select $\frac{\alpha}{N}$ from {0.01, 0.001, 0.0001, 1e-5, 1e-6, 0.0} and $\frac{\beta}{N}$ from {0.01, 0.001, 0.0001, 1e-5, 1e-6, 0.0}. For PTYL, we select λ from {1, 10, 100, 1000, 10000, 1e5, 1e6, 1e7, 1e8, 1e9} and $\frac{\beta}{N}$ from {0.01, 0.001, 0.0001, 1e-5, 1e-6, 0.0}.

For MAP + BO (Hvarfner et al., 2024), we select the initial learning rate from [0.1, 0.0001] and α, β (or equivalently λ, τ). For L2-SP, we select $\frac{\alpha}{N}$ from [0.01, 1e-6] and $\frac{\beta}{N}$ from [0.01, 1e-6].

For diagEF LA-LML (Immer et al., 2021), we select the initial learning rate from {0.1, 0.01, 0.001, 0.0001} and learn α, β (or equivalently λ, τ).

For diagEF LA-CLML (Lotfi et al., 2022), we select the initial learning rate from {0.1, 0.01, 0.001, 0.0001} and α, β (or equivalently λ, τ). For L2-SP, we select $\frac{\alpha}{N}$ from {0.01, 0.001, 0.0001, 1e-5, 1e-6, 0.0} and $\frac{\beta}{N}$ from {0.01, 0.001, 0.0001, 1e-5, 1e-6, 0.0}.

For iso ELBO and iso DE-ELBO (ours), we select the initial learning rate from {0.1, 0.01, 0.001, 0.0001} and learn λ, τ.

B.2 Closed-Form Optimal η for Model B

In our particular model in Eq. (9), the KL divergence between two Gaussians (Murphy, 2022) simplifies for the backbone KL term as:

$$-D_{\mathrm{KL}}(q(w)\|p(w)) = -\frac{1}{2}\left[\frac{\bar{\sigma}_q^2}{\lambda}\operatorname{tr}(\Sigma_p^{-1}) + \frac{1}{\lambda}(\mu_p - \bar{w})^\top \Sigma_p^{-1}(\mu_p - \bar{w}) - F + \log\left(\frac{\lambda^F \det(\Sigma_p)}{\bar{\sigma}_q^{2F}}\right)\right]. \tag{40}$$

Closed-form updates. To find an optimal λ value with respect to the J_{ELBO}, notice that of the 3 additive terms in Eq. (3), only the KL term between $q(w)$ and $p(w)$ involves λ. We solve for λ by taking the gradient of the KL term with respect to λ, setting to zero, and solving, with assurances of a local maximum of J_{ELBO} via a second derivative test. The gradient is

$$\nabla_\lambda - D_{\mathrm{KL}}(q(w)\|p(w)) = -\frac{1}{2}\left[-\frac{\bar{\sigma}_q^2}{\lambda^2}\operatorname{tr}(\Sigma_p^{-1}) - \frac{1}{\lambda^2}(\mu_p - \bar{w})^\top \Sigma_p^{-1}(\mu_p - \bar{w}) + \frac{F}{\lambda}\right]. \tag{41}$$

Setting $\nabla_\lambda - D_{\mathrm{KL}}(q(w)\|p(w)) = 0$ and solving for λ, we get

$$\lambda^* = \frac{1}{F}\left[\bar{\sigma}_q^2 \operatorname{tr}(\Sigma_p^{-1}) + (\mu_p - \bar{w})^\top \Sigma_p^{-1}(\mu_p - \bar{w})\right]. \tag{42}$$

In our particular model in Eq. (9), the KL divergence between two Gaussians (Murphy, 2022) simplifies for the classifier head KL term as:

$$-D_{\mathrm{KL}}(q(V)\|p(V)) = -\frac{1}{2}\left[\frac{\bar{\sigma}_q^2}{\tau}HC + \frac{1}{\tau}||\operatorname{vec}(\bar{V})||_2^2 - HC + \log\left(\frac{\tau^{HC}}{\bar{\sigma}_q^{2HC}}\right)\right]. \tag{43}$$

To find an optimal τ value with respect to the J_{ELBO}, notice that of the 3 additive terms in Eq. (3), only the KL term between $q(V)$ and $p(V)$ involves τ. We solve for τ by taking the gradient of the KL term with respect to τ, setting to zero, and solving, with assurances of a local maximum of J_{ELBO} via a second derivative test. The gradient is

$$\nabla_\tau - D_{\mathrm{KL}}(q(V)\|p(V)) = -\frac{1}{2}\left[-\frac{\bar{\sigma}_q^2}{\tau^2}HC - \frac{1}{\tau^2}||\operatorname{vec}(\bar{V})||_2^2 + \frac{1}{\tau}HC\right]. \tag{44}$$

Setting $\nabla_\tau - D_{\mathrm{KL}}(q(V)\|p(V)) = 0$ and solving for τ, we get

$$\tau^* = \frac{\bar{\sigma}_q^2 HC + ||\operatorname{vec}(\bar{V})||_2^2}{HC}. \tag{45}$$

Second derivative tests. To verify the optima found above, we perform second derivative tests. The second derivative of the negative KL term with respect to λ is:

$$\nabla_\lambda^2 - D_{\mathrm{KL}}(q(w)\|p(w)) = -\frac{1}{2}\left[\frac{2\bar{\sigma}_q^2}{\lambda^3}\operatorname{tr}(\Sigma_p^{-1}) + \frac{2}{\lambda^3}(\mu_p - \bar{w})^\top\Sigma_p^{-1}(\mu_p - \bar{w}) - \frac{F}{\lambda^2}\right] \tag{46}$$

$$= -\frac{1}{2}\left[\frac{2F}{\lambda^3}\frac{1}{F}\left(\bar{\sigma}_q^2\operatorname{tr}(\Sigma_p^{-1}) + (\mu_p - \bar{w})^\top\Sigma_p^{-1}(\mu_p - \bar{w})\right) - \frac{F}{\lambda^2}\right] \tag{47}$$

$$= -\frac{1}{2}\left[\frac{2F}{\lambda^3}\lambda^* - \frac{F}{\lambda^2}\right]. \tag{48}$$

Plugging in λ^* and simplifying, we get

$$\nabla_\lambda^2 - D_{\mathrm{KL}}(q(w)\|p(w|\lambda^*)) = -\frac{F}{2}\frac{1}{\lambda^{*2}} \tag{49}$$

This expression is always negative, indicating that λ^* is a local maximum of J_{ELBO}.

The second derivative of the negative KL term with respect to τ is:

$$\nabla_\tau^2 - D_{\mathrm{KL}}(q(V)\|p(V)) = -\frac{1}{2}\left[\frac{2\bar{\sigma}_q^2}{\tau^3}HC + \frac{2}{\tau^3}||\operatorname{vec}(\bar{V})||_2^2 - \frac{1}{\tau^2}HC\right] \tag{50}$$

$$= -\frac{1}{2}\left[\frac{2HC}{\tau^3}\left(\bar{\sigma}_q^2 + \frac{1}{HC}||\operatorname{vec}(\bar{V})||_2^2\right) - \frac{HC}{\tau^2}\right] \tag{51}$$

$$= -\frac{1}{2}\left[\frac{2HC}{\tau^3}\tau^* - \frac{HC}{\tau^2}\right]. \tag{52}$$

Plugging in τ^* and simplifying, we get

$$\nabla_\tau^2 - D_{\mathrm{KL}}(q(V)\|p(V|\tau^*)) = -\frac{HC}{2}\frac{1}{\tau^{*2}}. \tag{53}$$

This expression is always negative, indicating that τ^* is a local maximum of J_{ELBO}.

B.3 Low-rank Σ_p for Model B

The PTYL method (Shwartz-Ziv et al., 2022) uses Stochastic Weight Averaging-Gaussian (SWAG; Maddox et al., 2019) to approximate the posterior distribution $p(w|\mathcal{D}_S)$ of the source data $\mathcal{D}_S$ with a Gaussian distribution

$\mathcal{N}(\mu, \Sigma)$ where μ is the learned mean and $\Sigma = \frac{1}{2}(\Sigma_{\text{diag}} + \Sigma_{\text{LR}})$ is a representation of a covariance matrix with both diagonal and *low-rank* components. The LR covariance has the form $\Sigma_{\text{LR}} = \frac{1}{K-1} QQ^\top$, where $Q \in \mathbb{R}^{F \times K}$.

We use the Woodbury matrix identity (Woodbury, 1950), trace properties, and the matrix determinant lemma to compute the trace of the inverse, squared Mahalanobis distance, and log determinant of the low-rank covariance matrix for the KL term.

The trace and log determinant of the low-rank covariance matrix can be calculated once and used during training. Like in the PTYL method, the squared Mahalanobis distance needs to be re-evaluated every iteration of gradient descent.

Trace of the inverse. We compute the trace of the inverse of the low-rank covariance matrix using the Woodbury matrix identity and trace properties.

$$\begin{aligned}
\operatorname{tr}(\Sigma_p^{-1}) &= \operatorname{tr}((A + UCV)^{-1}) \\
&= \operatorname{tr}(A^{-1} - A^{-1}U(C^{-1} + VA^{-1}U)^{-1}VA^{-1}) && \text{Woodbury matrix identity} \\
&= \operatorname{tr}(A^{-1}) - \operatorname{tr}(A^{-1}U(C^{-1} + VA^{-1}U)^{-1}VA^{-1}) && \operatorname{tr}(A+B) = \operatorname{tr}(A) + \operatorname{tr}(B) \\
&= \operatorname{tr}(A^{-1}) - \operatorname{tr}((C^{-1} + VA^{-1}U)^{-1}VA^{-1}A^{-1}U) && \operatorname{tr}(AB) = \operatorname{tr}(BA)
\end{aligned}$$

where $A = \frac{1}{2}\Sigma_{\text{diag}}$, $C = I_K$, $U = \frac{1}{\sqrt{2K-2}}Q$, and $V = \frac{1}{\sqrt{2K-2}}Q^\top$. The last trace property, lets us compute the trace of the inverse of the low-rank covariance matrix without having to store a $F \times F$ covariance matrix.

Squared Mahalanobis distance. We compute the squared Mahalanobis distance $(\mu_p - \bar{w})^\top \Sigma_p^{-1}(\mu_p - \bar{w})$ by distributing the mean difference vector into the Woodbury matrix identity.

$$\begin{aligned}
\Sigma_p^{-1} &= (A + UCV)^{-1} \\
&= (A^{-1} - A^{-1}U(C^{-1} + VA^{-1}U)^{-1}VA^{-1}) && \text{Woodbury matrix identity}
\end{aligned}$$

Log determinant. We compute the log determinant of the low-rank covariance matrix using the matrix determinant lemma.

$$\begin{aligned}
\log\det(\Sigma_p) &= \log\det(A + UV) \\
&= \log(\det(I_K + VA^{-1}U)\det(A)) && \text{Matrix determinant lemma}
\end{aligned}$$

B.4 Varying κ for Model B

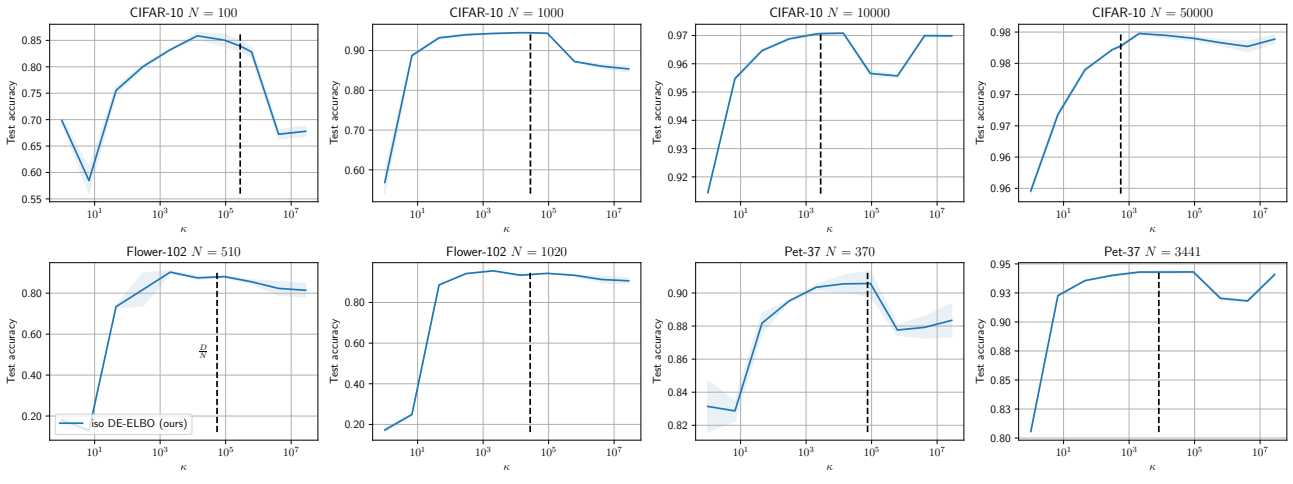


Figure B.1: Test set accuracy over κ values for L2-SP with iso DE-ELBO (ours). We report the mean (std) over 3 separately-sampled training sets. Each pannel shows a different task: ConvNeXt-Tiny fine-tuned on CIFAR-10, Flower-102, and Pet-37.

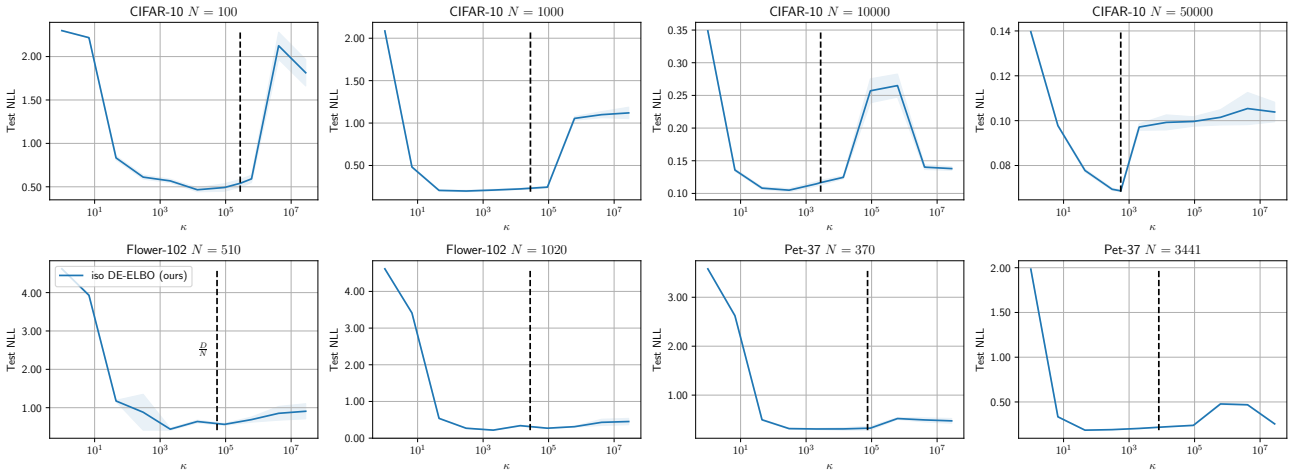


Figure B.2: Test set NLL over κ values for L2-SP with iso DE-ELBO (ours). We report the mean (std) over 3 separately-sampled training sets. Each pannel shows a different task: ConvNeXt-Tiny fine-tuned on CIFAR-10, Flower-102, and Pet-37.

B.5 Computational Time Comparisons for Model B

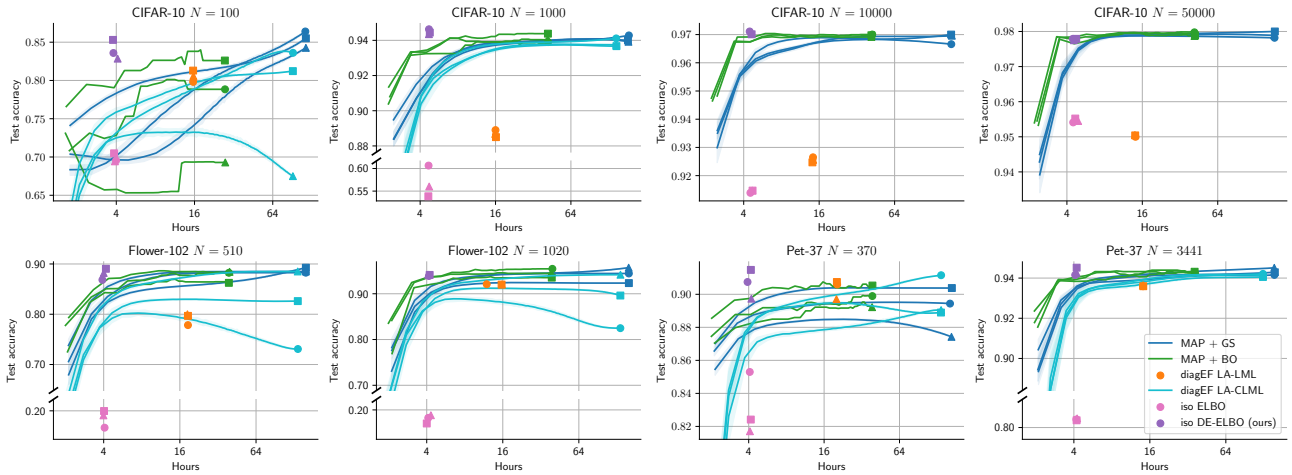


Figure B.3: Test set accuracy over time for L2-SP transfer learning methods. We run each method on 3 separate training sets of size N (3 different marker styles). Each panel shows a different task: ConvNeXt-Tiny fine-tuned on CIFAR-10, Flower-102, and Pet-37. We compare MAP + GS, MAP + BO (Hvarfner et al., 2024), diagEF LA-LML (Immer et al., 2021), diagEF LA-CLML (Lotfi et al., 2022), iso ELBO, and iso DE-ELBO (ours). To make the blue curves, we did the full grid search once (markers). Then, for each grid search size, we subsampled that number of hyperparameter configurations and selected the best using validation NLL. Averaging this over 500 subsamples for each grid size produced the blue lines. To make the green curves, we use BO to select candidate hyperparameter configurations and selected the best using validation NLL. Averaging this over 5 BO runs produced the green lines.

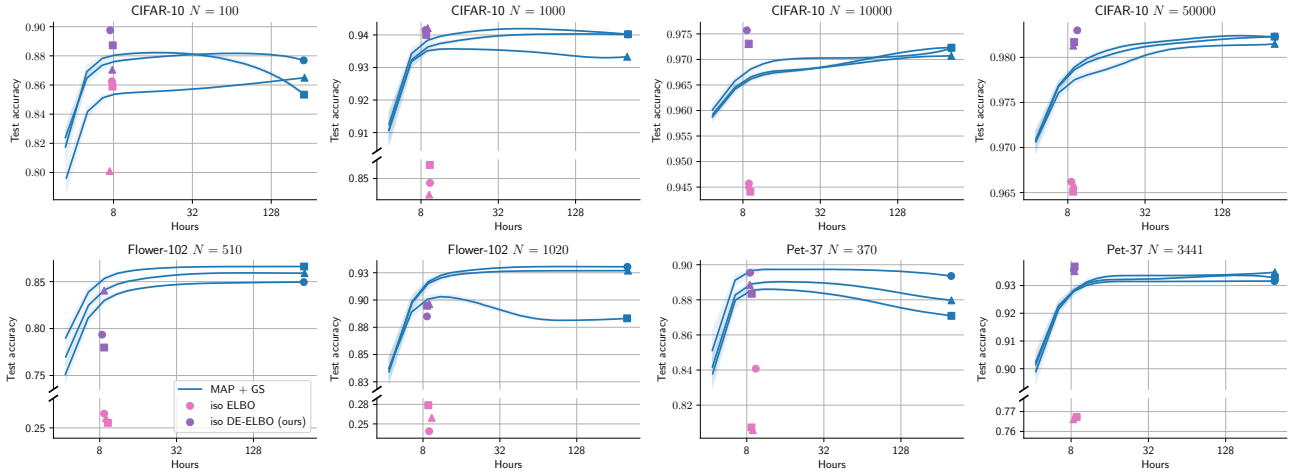


Figure B.4: Test set accuracy over time for L2-SP transfer learning methods. We run each method on 3 separate training sets of size N (3 different marker styles). Each panel shows a different task: ViT-B/16 fine-tuned on CIFAR-10, Flower-102, and Pet-37. We compare MAP + GS, iso ELBO, and iso DE-ELBO (ours). To make the blue curves, we did the full grid search once (markers). Then, for each grid search size, we subsampled that number of hyperparameter configurations and selected the best using validation NLL. Averaging this over 500 subsamples for each grid size produced the blue lines.

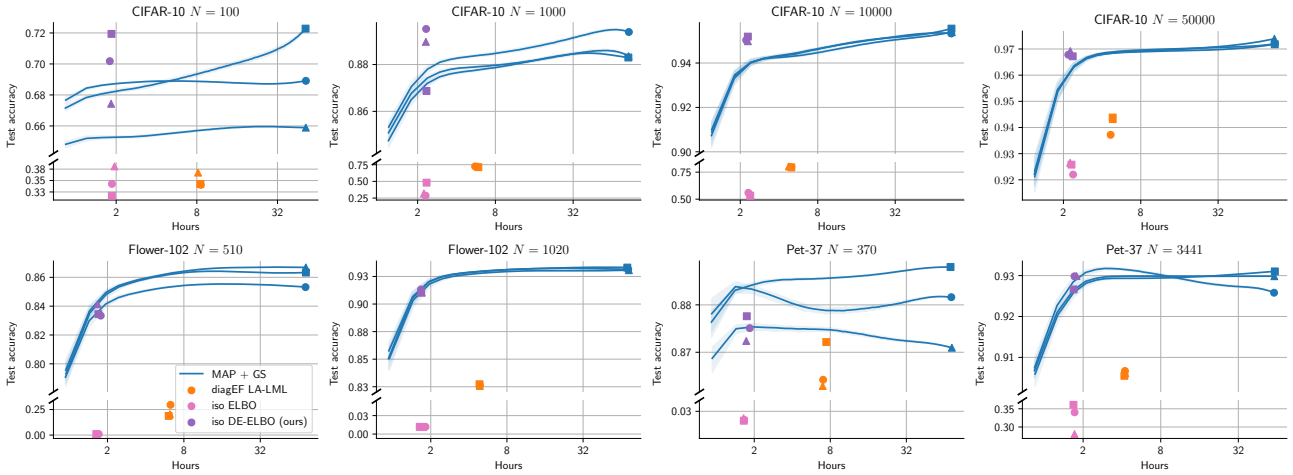


Figure B.5: Test set accuracy over time for L2-SP transfer learning methods. We run each method on 3 separate training sets of size N (3 different marker styles). Each panel shows a different task: ResNet-50 fine-tuned on CIFAR-10, Flower-102, and Pet-37. We compare MAP + GS, diagEF LA-LML (Immer et al., 2021), iso ELBO, and iso DE-ELBO (ours). To make the blue curves, we did the full grid search once (markers). Then, for each grid search size, we subsampled that number of hyperparameter configurations and selected the best using validation NLL. Averaging this over 500 subsamples for each grid size produced the blue lines.

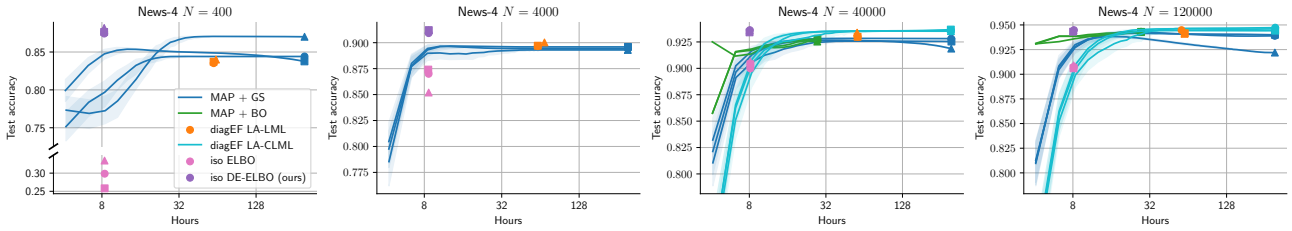


Figure B.6: Test set accuracy over time for L2-SP transfer learning methods. We run each method on 3 separate training sets of size N (3 different marker styles). Task: BERT-base fine-tuned on News-4. We compare MAP + GS, MAP + BO (Hvarfner et al., 2024), diagEF LA-LML (Immer et al., 2021), diagEF LA-CLML (Lotfi et al., 2022), iso ELBO, and iso DE-ELBO (ours). To make the blue curves, we did the full grid search once (markers). Then, for each grid search size, we subsampled that number of hyperparameter configurations and selected the best using validation NLL. Averaging this over 500 subsamples for each grid size produced the blue lines. To make the green curves, we use BO to select candidate hyperparameter configurations and selected the best using validation NLL. Averaging this over 10 BO runs produced the green lines.

B.6 Results for Model B

Table B.6: Accuracy on CIFAR-10, Flower-102, and Pet-37 test sets for different probabilistic models, methods, and backbones. We report the mean (min-max) over 3 separately-sampled training sets. MAP + GS requires 24 different SGD runs for L2-zero, 144 for L2-SP, and 240 for PTYL, while diagEF LA-LML (Immer et al., 2021) and iso DE-ELBO (ours) require 4 different SGD runs.

		CIFAR-10				Flower-102		Pet-37	
Model	**Method**	N = **100**	**1000**	**10000**	**50000**	**510**	**1020**	**370**	**3441**
				ResNet-50					
L2-zero	MAP + GS	67.7 (66.0-69.2)	87.7 (87.2-88.2)	94.6 (93.8-95.1)	97.0 (96.9-97.1)	86.4 (86.0-86.7)	92.8 (92.4-93.2)	87.6 (86.7-88.4)	93.1 (92.9-93.4)
	diagEF LA-LML	67.5 (66.6-68.5)	88.2 (87.8-88.8)	95.1 (95.1 95.2)	96.9 (96.9-96.9)	85.3 (84.7-85.8)	92.5 (92.4-92.7)	88.2 (87.8-88.5)	93.2 (93.0-93.3)
	iso DE-ELBO (ours)	62.3 (60.4-63.4)	87.1 (86.7-87.5)	91.3 (90.7-91.8)	92.9 (92.8-93.1)	81.6 (81.1-82.1)	90.0 (89.5-90.6)	83.2 (82.9-83.6)	91.8 (91.6-92.0)
L2-SP	MAP + GS	69.0 (65.9-72.3)	88.7 (88.3-89.4)	95.4 (95.3-95.6)	97.3 (97.2-97.4)	86.1 (85.3-86.7)	93.2 (93.1-93.3)	88.0 (87.1-88.8)	92.9 (92.6-93.1)
	diagEF LA-LML	35.0 (34.1-36.7)	72.0 (71.3-72.4)	76.4 (76.3-76.5)	94.1 (93.7-94.4)	22.9 (18.8-29.6)	82.7 (82.6-82.7)	86.7 (86.4-87.3)	90.6 (90.5-90.7)
	iso DE-ELBO (ours)	69.9 (67.4-71.9)	88.5 (86.9-89.5)	95.1 (95.0-95.2)	96.8 (96.7-96.9)	84.5 (84.5-84.6)	91.8 (91.6-92.1)	87.5 (87.2-87.8)	92.9 (92.7-93.0)
PTYL (SSL)	MAP + GS	57.5 (56.1-58.6)	78.4 (77.8-79.0)	90.6 (90.1-90.8)	96.6 (96.5-96.6)	81.9 (80.9-82.8)	89.7 (89.2-90.1)	58.3 (55.6-60.7)	86.3 (85.9-86.8)
	iso DE-ELBO (ours)	60.2 (59.5-60.7)	78.1 (77.5-78.8)	90.6 (90.3-90.8)	96.7 (96.6-96.7)	76.8 (76.5-77.2)	84.9 (84.6-85.1)	56.6 (56.1-57.2)	80.1 (80.0-80.3)
PTYL	MAP + GS	70.1 (69.2-71.4)	89.8 (89.5-90.3)	95.6 (95.5-95.8)	97.0 (96.8-97.2)	86.3 (85.8-86.6)	92.9 (92.6-93.1)	87.9 (87.5-88.2)	93.0 (92.8-93.2)
	iso DE-ELBO (ours)	70.0 (67.9-72.1)	89.2 (89.0-89.5)	95.1 (94.9-95.4)	96.9 (96.8-97.0)	84.6 (84.5-84.6)	91.8 (91.7-92.0)	87.5 (87.3-87.8)	92.9 (92.7-93.0)
				ViT-B/16					
L2-SP	MAP + GS	86.5 (85.3-87.7)	93.8 (93.3-94.0)	97.2 (97.1-97.2)	98.2 (98.2-98.2)	85.8 (84.9-86.6)	91.4 (88.3-93.0)	88.1 (87.1-89.4)	93.3 (93.2-93.5)
	iso DE-ELBO (ours)	88.5 (87.1-89.8)	94.1 (94.0-94.2)	97.4 (97.3-97.6)	98.2 (98.1-98.3)	80.5 (78.0-84.0)	89.2 (88.5-89.6)	88.9 (88.3-89.5)	93.6 (93.5-93.7)
				ConvNeXt-Tiny					
L2-SP	MAP + GS	85.4 (84.3-86.4)	94.1 (93.9-94.3)	96.9 (96.7-97.0)	97.9 (97.8-98.0)	88.7 (88.3-89.3)	94.2 (92.3-95.7)	89.1 (87.4-90.4)	94.3 (94.2-94.5)
	diagEF LA-LML	80.6 (79.8-81.3)	88.7 (88.5-88.9)	92.6 (92.5-92.7)	95.0 (95.0-95.1)	79.2 (77.8-80.0)	92.0 (92.0-92.1)	90.3 (89.7-90.7)	93.6 (93.6-93.6)
	iso DE-ELBO (ours)	83.9 (82.9-85.3)	94.5 (94.4-94.6)	97.1 (97.0-97.1)	97.8 (97.8-97.8)	88.0 (86.8-89.0)	94.1 (93.9-94.2)	90.7 (89.7-91.5)	94.3 (94.2-94.5)

Table B.7: NLL on CIFAR-10, Flower-102, and Pet-37 test sets for different probabilistic models, methods, and backbones. We report the mean (min-max) over 3 separately-sampled training sets. MAP + GS requires 24 different SGD runs for L2-zero, 144 for L2-SP, and 240 for PTYL, while diagEF LA-LML (Immer et al., 2021) and iso DE-ELBO (ours) require 4 different SGD runs.

		CIFAR-10				Flower-102		Pet-37	
Model	**Method**	N = **100**	**1000**	**10000**	**50000**	**510**	**1020**	**370**	**3441**
				ResNet-50					
L2-zero	MAP + GS	0.97 (0.94-1.03)	0.41 (0.39-0.44)	0.19 (0.19-0.20)	0.10 (0.10-0.10)	0.65 (0.63-0.68)	0.35 (0.32-0.38)	0.42 (0.37-0.47)	0.24 (0.24-0.25)
	diagEF LA-LML	1.05 (1.02-1.06)	0.46 (0.44-0.48)	0.20 (0.19-0.21)	0.12 (0.11-0.12)	0.71 (0.69-0.73)	0.36 (0.35-0.37)	0.40 (0.37-0.42)	0.29 (0.28-0.29)
	iso DE-ELBO (ours)	2.76 (2.51-3.18	0.54 (0.53-0.55)	0.39 (0.25-0.64)	0.28 (0.26-0.29)	0.84 (0.78-0.92)	0.44 (0.41-0.46)	0.80 (0.75-0.83)	0.27 (0.26-0.28)
L2-SP	MAP + GS	0.94 (0.85-1.02)	0.40 (0.39-0.41)	0.15 (0.15-0.16)	0.09 (0.09-0.09)	0.65 (0.63-0.68)	0.33 (0.32-0.35)	0.41 (0.36-0.45)	0.27 (0.24-0.30)
	diagEF LA-LML	2.30 (2.30-2.30)	1.18 (1.17-1.19)	0.78 (0.78-0.78)	0.18 (0.17-0.19)	4.62 (4.62-4.62)	2.60 (2.60-2.61)	1.34 (1.28-1.37)	0.41 (0.41-0.41)
	iso DE-ELBO (ours)	1.01 (0.92-1.12)	0.44 (0.42-0.46)	0.24 (0.23-0.24)	0.12 (0.12-0.12)	0.71 (0.67-0.74)	0.38 (0.38-0.39)	0.51 (0.49-0.53)	0.24 (0.24-0.24)
PTYL (SSL)	MAP + GS	1.34 (1.20-1.41)	0.50 (0.49-0.52)	0.23 (0.22-0.24)	0.11 (0.11-0.11)	0.84 (0.81-0.86)	0.49 (0.46-0.51)	1.64 (1.55-1.68)	0.54 (0.50-0.56)
	iso DE-ELBO (ours)	1.36 (1.33-1.43)	0.76 (0.74-0.78)	0.29 (0.28-0.31)	0.12 (0.12-0.12)	1.10 (1.05-1.15)	0.72 (0.70-0.74)	2.07 (2.04-2.08)	0.68 (0.67-0.69)
PTYL	MAP + GS	0.90 (0.87-0.94)	0.36 (0.34-0.37)	0.15 (0.14-0.16)	0.10 (0.10-0.10)	0.65 (0.63-0.68)	0.33 (0.32-0.34)	0.42 (0.37-0.46)	0.26 (0.21-0.31)
	iso DE-ELBO (ours)	1.00 (0.92-1.08)	0.46 (0.46-0.47)	0.23 (0.22-0.24)	0.12 (0.12-0.12)	0.71 (0.67-0.74)	0.38 (0.38-0.39)	0.51 (0.49-0.53)	0.24 (0.24-0.24)
				ViT-B/16					
L2-SP	MAP + GS	0.46 (0.41-0.51)	0.25 (0.20-0.28)	0.10 (0.10-0.10)	0.06 (0.06-0.06)	0.69 (0.64-0.72)	0.42 (0.34-0.54)	0.42 (0.38-0.47)	0.25 (0.24-0.29)
	iso DE-ELBO (ours)	0.36 (0.32-0.41)	0.24 (0.23-0.26)	0.12 (0.12-0.13)	0.07 (0.07-0.07)	0.90 (0.73-1.03)	0.51 (0.48-0.55)	0.40 (0.38-0.42)	0.30 (0.30-0.31)
				ConvNeXt-Tiny					
L2-SP	MAP + GS	0.46 (0.44-0.50)	0.22 (0.19-0.24)	0.11 (0.10-0.12)	0.07 (0.07-0.07)	0.54 (0.51-0.56)	0.28 (0.23-0.38)	0.38 (0.30-0.46)	0.18 (0.18-0.18)
	diagEF LA-LML	1.02 (0.94-1.09)	0.40 (0.40-0.41)	0.24 (0.24-0.24)	0.16 (0.16-0.16)	2.88 (2.87-2.90)	0.95 (0.94-0.96)	0.43 (0.43-0.43)	0.24 (0.24-0.24)
	iso DE-ELBO (ours)	0.54 (0.48-0.59)	0.23 (0.22-0.24)	0.12 (0.11-0.12)	0.07 (0.07-0.07)	0.58 (0.53-0.64)	0.30 (0.29-0.31)	0.32 (0.27-0.36)	0.22 (0.22-0.22)

Table B.6: ECE on CIFAR-10 test set for different probabilistic models, methods, and backbones. We report the mean (min-max) over 3 separately-sampled training sets. MAP + GS requires 24 different SGD runs for L2-zero, 144 for L2-SP, and 240 for PTYL, while diagEF LA-LML (Immer et al., 2021) and iso DE-ELBO (ours) require 4 different SGD runs.

		CIFAR-10			
Model	**Method**	$N =$ **100**	**1000**	**10000**	**50000**
		ResNet-50			
L2-SP	diagEF LA-LML	25.0 (24.0–26.6)	32.4 (30.8–33.9)	13.5 (13.0–13.8)	1.8 (1.7–1.9)
	iso ELBO	24.9 (21.7–28.1)	26.1 (18.6–38.0)	43.5 (42.6–45.2)	2.3 (1.9–2.7)
	iso DE-ELBO (ours)	11.7 (10.0–13.2)	5.5 (3.6– 6.7)	3.3 (3.2– 3.5)	1.8 (1.7–1.9)
		ConvNeXt-Tiny			
L2-SP	diagEF LA-LML	37.5 (34.2–39.9)	10.8 (10.6–11.1)	4.6 (4.5– 4.9)	2.2 (2.2–2.2)
	iso ELBO	59.7 (59.4–60.3)	43.9 (40.6–47.8)	12.1 (12.0–12.1)	1.6 (1.5–1.6)
	iso DE-ELBO (ours)	6.6 (5.6– 7.6)	3.5 (3.2– 3.6)	1.7 (1.6– 1.8)	0.8 (0.7–0.8)

Changes to training performance from running/computed mean and variance in batch normalization layers. During training, batch normalization layers keep running estimates of its computed mean and variance, which are then used for normalization during evaluation. We find that when using batch normalization, the ELBO and accuracy on the train set can change between `train()` and `eval()` mode (see Fig. B.7). We recommend using the computed mean and variance to evaluate the ELBO on the training set for model selection since this mode is used for normalization during evaluation.

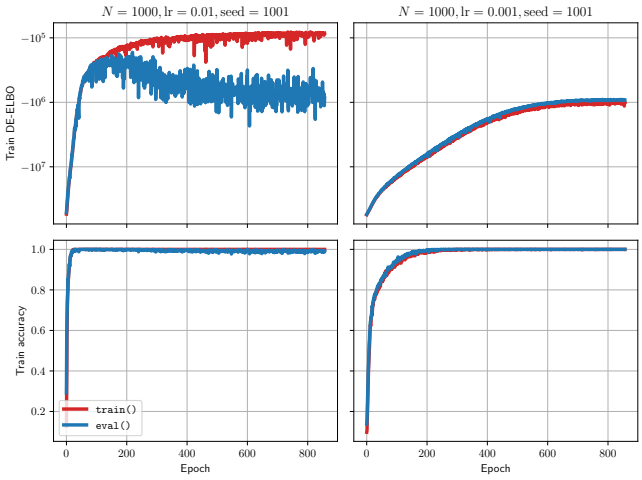


Figure B.7: Train set DE-ELBO and accuracy on CIFAR-10 with $N = 1000$ using ResNet-50 over epochs for L2-SP. We use just one sample per training step to approximate the expected log-likelihood.

Table B.7: Accuracy on News-4 test set for different probabilistic models, methods, and backbones. We report the mean (min-max) over 3 separately-sampled training sets. MAP + GS requires 24 different SGD runs for L2-zero, 144 for L2-SP, and 240 for PTYL, while diagEF LA-LML (Immer et al., 2021) and iso DE-ELBO (ours) require 4 different SGD runs.

		News-4			
Model	**Method**	$N =$ **400**	**4000**	**40000**	**120000**
		BERT-base			
L2-SP	MAP + GS	85.1 (83.8-87.0)	89.4 (89.3-89.6)	92.4 (91.9-92.8)	93.4 (92.2-94.0)
	diagEF LA-LML	84.2 (83.5-85.3)	89.8 (89.7-90.0)	93.1 (93.0-93.4)	92.5 (92.5-92.6)
	iso DE-ELBO (ours)	87.8 (87.4-88.2)	91.1 (90.9-91.2)	93.5 (93.4-93.6)	94.4 (94.3-94.5)

Table B.8: NLL on News-4 test set for different probabilistic models, methods, and backbones. We report the mean (min-max) over 3 separately-sampled training sets. MAP + GS requires 24 different SGD runs for L2-zero, 144 for L2-SP, and 240 for PTYL, while diagEF LA-LML (Immer et al., 2021) and iso DE-ELBO (ours) require 4 different SGD runs.

		News-4			
Model	**Method**	$N =$ **400**	**4000**	**40000**	**120000**
		BERT-base			
L2-SP	MAP + GS	0.09 (0.05-0.17)	0.03 (0.03-0.03)	0.02 (0.02-0.02)	0.02 (0.02-0.02)
	diagEF LA-LML	0.08 (0.08-0.09)	0.03 (0.03-0.03)	0.02 (0.02-0.02)	0.02 (0.02-0.02)
	iso DE-ELBO (ours)	0.10 (0.10-0.11)	0.08 (0.08-0.09)	0.03 (0.03-0.03)	0.02 (0.02-0.02)

B.7 Importance Weighted ELBO for Model B

For transfer learning of image classifiers from Case Study B, the results in Sec. 6 show that our *data-emphasized ELBO* outperforms the standard ELBO, favoring settings of ψ, η that produce higher test accuracy. Our hypothesis about *why* this occurs is that in our chosen target scenario, the ELBO objective itself is too loose a bound on the log marginal likelihood for reliable selection, whereas setting $\kappa = D/N$ helps account for the misspecified isotropic q to deliver better fitting models.

Recall the bounding relation between the exact log marginal likelihood and the ELBO:

$$\underbrace{\log \mathbb{E}_{q_\psi(\theta)}\left[\frac{p_\eta(y_{1:N}|\theta)p_\eta(\theta)}{q_\psi(\theta)}\right]}_{\log p_\eta(y_{1:N})} \geq \underbrace{\mathbb{E}_{q_\psi(\theta)}\left[\log \frac{p_\eta(y_{1:N}|\theta)p_\eta(\theta)}{q_\psi(\theta)}\right]}_{J_{\text{ELBO}}(y_{1:N},\psi,\eta)} \tag{54}$$

Naturally, if we could find tighter bounds than the ELBO we might hope these could be used for improved practical model selection, delivering good predictive accuracy even when q remains misspecified as isotropic compared to an ideal full-rank covariance. One possible candidate for a tighter bound than the ELBO is the *importance weighted ELBO* (IWELBO) objective of (Burda et al., 2016).

$$J_{\text{IWELBO}} := \mathbb{E}_{\theta_1,\ldots,\theta_S \sim q_\psi(\theta)}\left[\log \frac{1}{S}\sum_{s=1}^{S} \exp\left(\sum_{i=1}^{N} \log p_\eta(y_i|\theta_s) + \log p_\eta(\theta_s) - \log q_\psi(\theta_s)\right)\right] \tag{55}$$

Burda et al. show that the IWELBO is a better estimator of the log of the evidence than the ELBO, with estimation quality improving as the number of samples S increases. Burda et al.'s Theorem 1 says that, if $\frac{p(y_{1:N},\theta)}{q_\psi(\theta)}$ is bounded, then in the limit as $S \to \infty$, the IWELBO will converge to the log marginal likelihood.

We can also introduce a *data-emphasized IWELBO* (DE-IWELBO)

$$J_{\text{DE-IWELBO}} := \mathbb{E}_{\theta_1,\ldots,\theta_S \sim q_\psi(\theta)}\left[\log \frac{1}{S}\sum_{s=1}^{S} \exp\left(\kappa \cdot \sum_{i=1}^{N} \log p_\eta(y_i|\theta_s) + \log p_\eta(\theta_s) - \log q_\psi(\theta_s)\right)\right] \tag{56}$$

For simplicity, we again set $\kappa = \frac{D}{N}$ like with our DE-ELBO. Future work could investigate theoretical grounding for how to set κ in the DE-IWELBO.

In principle, a training algorithm that estimates ψ, η that maximize IWELBO with very large S should enjoy the Occam's razor benefits of the true marginal likelihood. However, for large DNNs using very large S requires S forward passes at every input image and would be prohibitively expensive.

To understand if the IWELBO offers a *practical* route forward, we thus experiment with whether the $S = 50$ IWELBO could resolve the model selection issues seen in Fig. 1. We use the same pink and purple ψ from Fig. 1. We then evaluate the $S = 50$ IWELBO and $S = 50$ DE-IWELBO at each ψ across a range of η.

Results are shown in the figure below. While we verified the IWELBO yields numerically higher values than the ELBO at each ψ, η, the differences are slight compared to the overall y-axis scale. The overall trends and takeaways for DE-IWELBO vs IWELBO match the takeaways from Fig. 1 for DE-ELBO vs ELBO.

Even with $S = 50$ samples, the IWELBO favors the low-test-accuracy ψ over the high-test-accuracy ψ. This suggests that Burda et al.'s tighter objective isn't an immediate resolution to the problems with ELBO model selection raised in Contribution 1, at least with $S = 50$. Interestingly, the DE-IWELBO with $S = 50$ favors q that produce higher test accuracy.

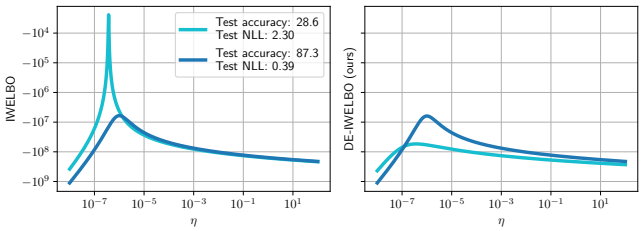


Figure B.8: Model selection comparison for the $S = 50$ IWELBO and $S = 50$ DE-IWELBO across a range of hyperparameters $\eta = \lambda = \tau$ for Model B in Eq. (9). Task: ResNet-50 fine-tuned on CIFAR-10 with $N = 1000$. The light blue curve uses the same low-test-accuracy ψ value as in Fig. 1; the dark blue curve uses the same high-test-accuracy ψ value as in Fig. 1. **Takeaway: $S = 50$ is near the limit of practical affordability for large DNNs, and even here the tighter IWELBO is not enough to overcome the underfitting of the standard ELBO raised in Fig. 1.**