

# Knowledge Adaptation as Posterior Correction

**Mohammad Emtiyaz Khan**

EMTIYAZ.KHAN@RIKEN.JP

*RIKEN Center for Advanced Intelligence Project  
1-4-1 Nihonbashi, Chuo-ku, Tokyo 103-0027, Japan*

## Abstract

Adaptation is the holy grail of intelligence, but even the best AI models (like GPT) lack the adaptivity of toddlers. So the question remains: how can machines adapt quickly? Despite a lot of progress on model adaptation to facilitate continual and federated learning, as well as model merging, editing, unlearning, etc., little is known about the mechanisms by which machines can naturally learn to adapt in a similar way as humans and animals. Here, we show that all such adaptation methods can be seen as different ways of ‘correcting’ the approximate posteriors. More accurate posteriors lead to smaller corrections, which in turn imply quicker adaptation. The result is obtained by using a dual-perspective of the Bayesian Learning Rule of Khan and Rue (2023) where interference created during adaptation is characterized by the natural-gradient mismatch over the past data. We present many examples to demonstrate the use of posterior-correction as a natural mechanism for the machines to learn to adapt quickly.

## 1. Introduction

Quick adaptation is essential for survival in nature so much so that it can be equated to intelligence (Sternberg, 2019). Despite this, there is no Artificial Intelligence (AI) system to this day whose adaptive abilities are considered anywhere close to those of humans and animals. Large Language Models (LLMs) like GPT and Gemini can do many tasks better than humans, but they lack the adaptivity of toddlers. We cannot simply teach such models new things by just talking or interacting with them. Hence, there are no simple ways to quickly fix their harmful behavior reported nowadays on a daily basis. There is no magical tool to just permanently add, remove, or modify the knowledge of LLMs. The most trusted way is to simply retrain the whole system which is too costly to fix their day to day mistakes. So the question remains: how can such models adapt quickly? Quick adaptation is important not only to reduce the cost but also to improve the safety, security, and sustainability of AI.

A lot has been done recently to instill adaptivity in large AI models but efforts are fragmented with no apparent similarities in the methodologies used (Fig. 1). For example, Continual Learning methods adapt to streaming data over time (Kirkpatrick et al., 2017), while Federated Learning methods assimilate knowledge spread across space (McMahan et al., 2017). While both deal with distributed information, fundamentally different methodologies are preferred, for instance, replay of old data is common in continual learning (Rebuffi et al., 2017; Pan et al., 2020; Buzzega et al., 2020) but discouraged in federated learning because of privacy. Such differences can make it harder to connect the two sub-fields at a fundamental level. The same is true of other sub-fields, such as, model merging and editing (Wortsman et al., 2022a,b; Ilharco et al., 2023; Mitchell et al., 2022), knowledge

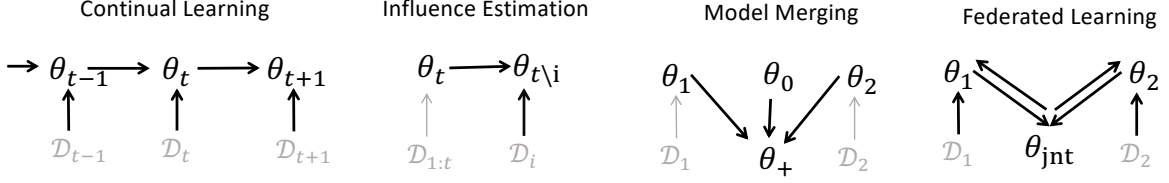


Figure 1: Four popular adaptation cases to adapt model parameters  $\theta$ . Black arrows indicate the flow of adapted knowledge, while the gray arrow indicate pre-training. Continual learning adapts  $\theta_t$  to  $\theta_{t+1}$  to include new data  $\mathcal{D}_{t+1}$ . Influence estimation gives a quick estimate of the ‘unlearned’ model obtained by removing  $i$ ’th data  $\mathcal{D}_i$ . Model Merging improves a pre-trained LLM  $\theta_0$  by merging back the fine-tuned models  $\theta_1$  and  $\theta_2$ . Finally, Federated Learning aims to obtain a joint model  $\theta_{\text{jnt}}$  by using locally trained models.

distillation (Hinton et al., 2015), fine-tuning (Hu et al., 2022), etc. All of these efforts use adaptation to avoid retraining as much as possible, but they remain disconnected from each other. Our goal in this paper is to unify these approaches to unravel a common adaptation mechanism behind them. The underlying mechanism can then be used to not only explain the effectiveness of these approaches but also to find new ways to improve them.

## 2. Posterior Correction

We present a new ‘posterior correction’ approach to unify existing adaptation schemes. The approach relies on a variational reformulation where the goal of learning is to find accurate approximations of the posterior distribution (Khan and Rue, 2023). For example, an Empirical Risk Minimization (ERM) problem over model parameters  $\theta \in \Theta$  is reformulated as Variational Learning (VL) over candidate distribution  $q(\theta) \in \mathcal{Q}$ , that is, we reformulate

$$\theta_t = \arg \min_{\theta \in \Theta} \sum_{i=0}^t \ell_i(\theta) \quad \text{as} \quad q_t = \arg \min_{q \in \mathcal{Q}} \sum_{i=1}^t \mathbb{E}_q[\ell_i] + \mathbb{D}_{\text{KL}}[q \parallel p_0]. \quad (1)$$

Here, ERM uses the loss  $\ell_i$  defined over  $t$  data examples for  $i = 1, 2, \dots, t$ , as well as a regularizer  $\ell_0$  which can either be added explicitly or implemented implicitly through an algorithm. In contrast, the VL problem uses expectation of losses and a prior  $p_0 \propto \exp(-\ell_0)$  used in the Kullback-Leibler (KL) divergence term. Despite these differences, the ERM solution  $\theta_t$  can be recovered from VL by using a Gaussian  $q_t$  and applying the delta method, as described in Sec. 3. We will use the above reformulation to derive adaptation strategies proposed for ERM as a special case of the posterior-correction approach for VL.

We start with a simple case to set the idea. Suppose we receive a new example  $t + 1$  with loss  $\ell_{t+1}$ , then how can we quickly adapt  $q_t$  to recover  $q_{t+1}$ ? Bayes’ rule suggest to use  $q_t$  as the prior and update the new posterior as follows:  $p_{t+1} \propto e^{-\ell_{t+1}} q_t$ . However, this is intractable unless the loss and prior form a conjugate pair, like Gaussians (Bishop, 2006, pp. 117). A tractable alternative is to use VL to keep the update within the set  $\mathcal{Q}$ ,

$$\hat{q}_{t+1} = \arg \min_{q \in \mathcal{Q}} \mathbb{E}_q[\ell_{t+1}] + \mathbb{D}_{\text{KL}}[q \parallel q_t]. \quad (2)$$

Unfortunately, this does not exactly recover the true  $q_{t+1}$ . We will now show that  $q_{t+1}$  can be recovered exactly if we add a correction term in the above update.

Table 1: Site functions. We denote  $\bar{\mathbf{H}}_{i|t} = \mathbb{E}_{q_t}[\nabla^2 \ell_i]$  and its diagonal by the vector  $\bar{\mathbf{h}}_i$ . Element-wise multiplication and divisions are denoted by  $\mathbf{a} \cdot \mathbf{b}$  and  $\mathbf{a}/\mathbf{b}$  respectively.

| Form of $q_t$                                                                                                     | Expression for the site $\hat{\ell}_{i t}$                                                                                                                                                                         |
|-------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Gaussian $q_t^{\text{iso}} = \mathcal{N}(\boldsymbol{\theta} \mathbf{m}_t, \mathbf{I})$                           | $\hat{\ell}_{i t}^{\text{iso}} = \boldsymbol{\theta}^\top \mathbb{E}_{q_t}[\nabla \ell_i]$                                                                                                                         |
| Gaussian $q_t^{\text{diag}} = \mathcal{N}(\boldsymbol{\theta} \mathbf{m}_t, \text{diag}(\frac{1}{\mathbf{s}_t}))$ | $\hat{\ell}_{i t}^{\text{diag}} = \boldsymbol{\theta}^\top \mathbb{E}_{q_t}[\nabla \ell_i] + \frac{1}{2}(\boldsymbol{\theta} - \mathbf{m}_t)^\top [\bar{\mathbf{h}}_i \cdot (\boldsymbol{\theta} - \mathbf{m}_t)]$ |
| Gaussian $q_t^{\text{full}} = \mathcal{N}(\boldsymbol{\theta} \mathbf{m}_t, \mathbf{S}_t^{-1})$                   | $\hat{\ell}_{i t}^{\text{full}} = \boldsymbol{\theta}^\top \mathbb{E}_{q_t}[\nabla \ell_i] + \frac{1}{2}(\boldsymbol{\theta} - \mathbf{m}_t)^\top \bar{\mathbf{H}}_{i t}(\boldsymbol{\theta} - \mathbf{m}_t)$      |
| Bernoulli $P(\theta = 1) = p_1$                                                                                   | $\hat{\ell}_{i t}^{\text{bern}} = \theta \nabla_{p_1} \mathbb{E}_{q_t}[\ell_i]$                                                                                                                                    |

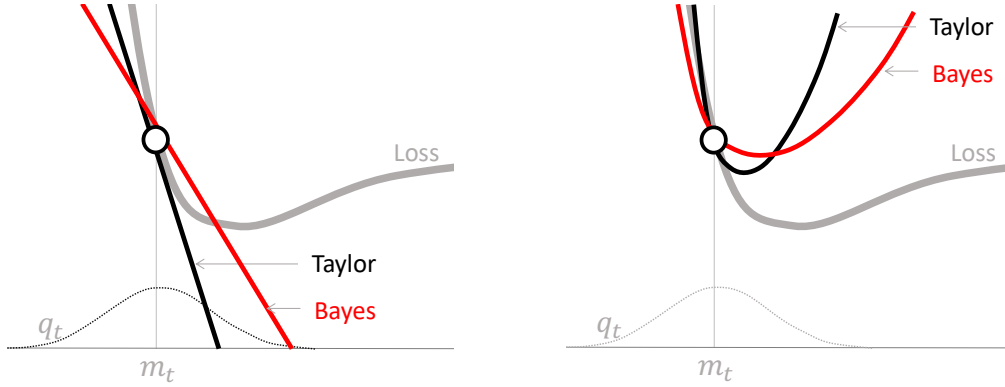


Figure 2: The left panel compares the site  $\hat{\ell}_{i|t}^{\text{iso}}$  to the 1<sup>st</sup>-order Taylor of  $\ell_i$  at  $\mathbf{m}_t$ . The right panel compares  $\hat{\ell}_{i|t}^{\text{full}}$  to the 2<sup>nd</sup>-order Taylor. The site functions use expectations of the gradients and Hessians over  $q_t$  and capture a more global information around  $\mathbf{m}_t$ .

To correct the update, we will rely on a *dual* perspective proposed by Khan and Rue (2023, Sec. 5.4). They consider  $\mathcal{Q}$  to be the set of minimal exponential-family distribution  $q(\boldsymbol{\theta}) = \exp(\langle \mathbf{T}(\boldsymbol{\theta}), \boldsymbol{\lambda} \rangle - A(\boldsymbol{\lambda}))$  with sufficient statistics  $\mathbf{T}(\boldsymbol{\theta})$ , natural parameter  $\boldsymbol{\lambda}$ , log-partition function  $A(\cdot)$ , and inner product denoted by  $\langle \cdot, \cdot \rangle$ . For such families, they show that any solution  $q_t$  can be written as a product of local *site* functions  $\hat{\ell}_{i|t}$  for each  $\ell_i$ ,

$$q_t(\boldsymbol{\theta}) \propto \prod_{i=0}^t \exp\left(-\hat{\ell}_{i|t}(\boldsymbol{\theta})\right), \quad \text{where } \hat{\ell}_{i|t}(\boldsymbol{\theta}) = \langle \mathbf{T}(\boldsymbol{\theta}), \tilde{\nabla} \mathbb{E}_{q_t}[\ell_i] \rangle. \quad (3)$$

Here,  $\tilde{\nabla}$  denotes the *natural gradient* with respect to  $\boldsymbol{\lambda}$  evaluated at the natural parameter of  $q_t$ . A derivation is given in Sec. 4.1 where we also allow non-constant base measure.<sup>1</sup> Table 1 shows a few examples of the sites  $\hat{\ell}_{i|t}$  where see Gaussian sites resembles those obtained using Taylor’s method but they use derivatives evaluated and averaged over the samples from  $q_t$ ; also see Fig. 2. The site functions contain more global information than Taylor’s surrogates and they also apply more generally, for instance, to discontinuous loss functions and discrete variables. More details are included in Sec. 4.1.

1. For non-constant base measure  $h(\boldsymbol{\theta})$ , we just need to modify  $\hat{\ell}_{0|t} = \langle \mathbf{T}(\boldsymbol{\theta}), \tilde{\nabla} \mathbb{E}_{q_t}[\ell_0 + \log h] \rangle - \log h$ .

We use Eq. 3 to correct the update in Eq. 2. In the first line below, we begin with the definition of  $q_{t+1}$ , then expand the KL term where we divide and multiply by  $q_t$  and its dual form respectively, and rearrange terms to get the correction shown in the last line:

$$\begin{aligned}
q_{t+1} &= \arg \min_{q \in \mathcal{Q}} \mathbb{E}_q[\ell_{t+1}] + \sum_{i=1}^t \mathbb{E}_q[\ell_i] + \mathbb{D}_{\text{KL}}[q \parallel p_0] \\
&= \arg \min_{q \in \mathcal{Q}} \mathbb{E}_q[\ell_{t+1}] + \sum_{i=1}^t \mathbb{E}_q[\ell_i] + \mathbb{E}_q \left[ \log \left( \frac{q}{e^{-\ell_0}} \times \frac{\prod_{i=0}^t e^{-\hat{\ell}_{i|t}}}{q_t} \right) \right] \\
&= \arg \min_{q \in \mathcal{Q}} \mathbb{E}_q[\ell_{t+1}] + \mathbb{D}_{\text{KL}}[q \parallel q_t] + \underbrace{\sum_{i=0}^t \mathbb{E}_q[\ell_i - \hat{\ell}_{i|t}]}_{\text{Correction}}
\end{aligned} \tag{4}$$

Adding the third term corrects the update in Eq. 2 to ensure the recovery of the exact  $q_{t+1}$  (instead of  $\hat{q}_{t+1}$ ). Corrections are required in all the past  $\ell_i$  as well as the regularizer  $\ell_0$ . Essentially, past  $\ell_i$  are represented in  $q_t$  through sites  $\hat{\ell}_{i|t}$  and, when a new  $\ell_{t+1}$  is added, the representation needs to change to  $\hat{\ell}_{i|t+1}$ . The correction term gives a precise mathematical characterization of the *interference* created during adaptation, which is an age-old problem (Sutton, 1986) without a precise agreed-upon mathematical definition. Other views of posterior corrections are discussed in Sec. 4.2 where it is used to derive a Bayes’ filter and a new definition of information gain, and also interpreted as natural-gradient mismatch. In addition, a method to ‘boost’ training trajectories is also discussed.

The main result of this paper is to show that many existing adaptation methods tailored to handle specific adaptation cases, can all be seen as specific instances of posterior correction. Smaller correction imply less interference which can be handled quickly during adaptation. This key message is demonstrated by covering multiple adaptation scenarios, such as, continual learning, influence estimation, model merging, and federated learning.

### 3. Knowledge Adaptation as Posterior Correction

To derive existing methods as posterior correction, we will reformulate an ERM problem as a VL problem. For instance, for a  $q^{\text{iso}}$  family, VL in Eq. 1 reduces to an ERM over the mean  $\mathbf{m}$  by using the 1<sup>st</sup>-order delta method (Khan and Rue, 2023, App. C),

$$\mathbb{E}_q[\ell_i] \approx \ell_i(\mathbf{m}) \quad \implies \quad \mathbf{m}_t = \arg \min_{\mathbf{m} \in \Theta} \sum_{i=0}^t \ell_i(\mathbf{m}) + \text{const.} \tag{5}$$

This is because the entropy of  $q^{\text{iso}}$  is constant, therefore the KL term in VL only contributes the  $\ell_0$  term. This reformulation is suitable for ERMs that aim for 1<sup>st</sup>-order stationarity conditions, for instance, by using stochastic gradient descent. For algorithms that aim 2<sup>nd</sup>-order optimality, we can use a 2<sup>nd</sup>-order delta method over a  $q^{\text{full}}$  family and assume that the covariance is set to the Hessian at the ERM solution  $\mathbf{m}_t$  which corresponds to Laplace’s method (Khan and Rue, 2023, App. C.1). Similarly, for methods such as Adam, SAM, or RMSprop, a  $q^{\text{diag}}$  family is suitable. The explicit choice to not optimize for the covariance is made in non-Bayesian methods, and by making the same choice in the VL problem, we can recover them by solving a variational formulation.

### 3.1 Continual Learning as Posterior Correction

In continual learning, the goal is to quickly update  $\boldsymbol{\theta}_t$  to  $\boldsymbol{\theta}_{t+1}$  when a new loss  $\ell_{t+1}$  is available. A popular strategy for this is to regularize the problem. Three kinds of regularization methods are commonly employed: (i) parameter-space regularization, (ii) function or prediction-space regularization, (iii) experience or memory replay. Here, we derive these as special cases of the regularization used for posterior correction used in Eq. 4, denoted by

$$\mathcal{K}_t(q) = \mathbb{D}_{\text{KL}}[q \parallel q_t] + \sum_{i=0}^t \mathbb{E}_q[\ell_i - \hat{\ell}_{i|t}]. \quad (6)$$

Here, the KL term performs the parameter-space regularization while the correction term in the data-space is related to the other two regularization methods. Posterior correction naturally combines these seemingly different approaches in one single method.

Recovering parameter-space regularization methods is straightforward. For a Gaussian family,  $\mathcal{K}_t$  without the correction term is simply the variational continual learning. The KL term in such cases reduces to a quadratic regularizers which recovers the popular Elastic-Weight Consolidation (EWC) which uses  $\frac{1}{2}(\boldsymbol{\theta} - \boldsymbol{\theta}_t)^\top \mathbf{H}_t(\boldsymbol{\theta} - \boldsymbol{\theta}_t)$  with a diagonal Hessian/Fisher  $\mathbf{H}_t$  (Kirkpatrick et al., 2017). By employing an arbitrary exponential-family in the KL term, posterior correction can enable generic parameter-space regularization commonly used for online learning (van der Hoeven et al., 2018).

Next, we consider the function- or prediction- space regularizations. These can be recovered through the correction term. For example, consider a squared-loss

$$\ell_i(\boldsymbol{\theta}) = \mathcal{L}[y_i, \hat{y}_i(\boldsymbol{\theta})] = \frac{1}{2} \|y_i - \hat{y}_i(\boldsymbol{\theta})\|^2$$

over input-output pair  $(\mathbf{x}_i, y_i)$  given a linear predictor  $\hat{y}_i(\boldsymbol{\theta}) = \mathbf{x}_i^\top \boldsymbol{\theta}$ . For a  $q^{\text{iso}}$  family, the correction term simplifies as ‘prediction matching’ among the new and old model,

$$\begin{aligned} \ell_i(\boldsymbol{\theta}) - \tilde{\ell}_{i|t}^{\text{iso}}(\boldsymbol{\theta}) &= \ell_i(\boldsymbol{\theta}) - \boldsymbol{\theta}^\top \mathbb{E}_{q_t}[\nabla \ell_i] \\ &= \frac{1}{2} \|y_i - \mathbf{x}_i^\top \boldsymbol{\theta}\|^2 - \boldsymbol{\theta}^\top \mathbf{x}_i(\mathbf{x}_i^\top \mathbf{m}_t - y_i) \\ &= \frac{1}{2} \mathcal{L}[\hat{y}_{i|t}(\mathbf{m}_t), \hat{y}_i(\boldsymbol{\theta})] + \text{const.} \end{aligned} \quad (7)$$

This is a function-space regularization used by Benjamin et al. (2019) for neural networks. A visualization of corrections as prediction mismatch is shown in the left panel of Fig. 3. The right panel additionally compares mismatches of two base models. The model with less mismatch is closer to the new model and can facilitate faster adaptation.

In general, this result can be extended to a general nonlinear model  $f_i(\boldsymbol{\theta})$  (such as neural networks) trained on a loss function  $\mathcal{L}(y, \hat{y})$  derived using Bregman divergence. As shown in Sec. 4.3, we can write the correction as a sum of a prediction-matching term and an error term arising due to the non-linearity of the model,

$$\ell_i(\boldsymbol{\theta}) - \tilde{\ell}_{i|t}^{\text{iso}}(\boldsymbol{\theta}) \approx \underbrace{\mathcal{L}[\hat{y}_{i|t}, \hat{y}_i(\boldsymbol{\theta})]}_{\text{Pred. matching}} + r_{i|t} \underbrace{\left[ f_i(\boldsymbol{\theta}) - \hat{f}_i^{\text{lin}}(\boldsymbol{\theta}) \right]}_{\text{Linearization error}}. \quad (8)$$

Here, we denote by  $\hat{y}_{i|t} = \mathbb{E}_{q_t}[\hat{y}_i(\boldsymbol{\theta})]$  the average posterior predictions under  $q_t$  and by  $r_{i|t} = \hat{y}_{i|t} - y_i$  its residual. The term  $\hat{f}_i^{\text{lin}}(\boldsymbol{\theta})$  is a 1<sup>st</sup>-order linearization of  $f_i(\boldsymbol{\theta})$  at  $\mathbf{m}_t$ .

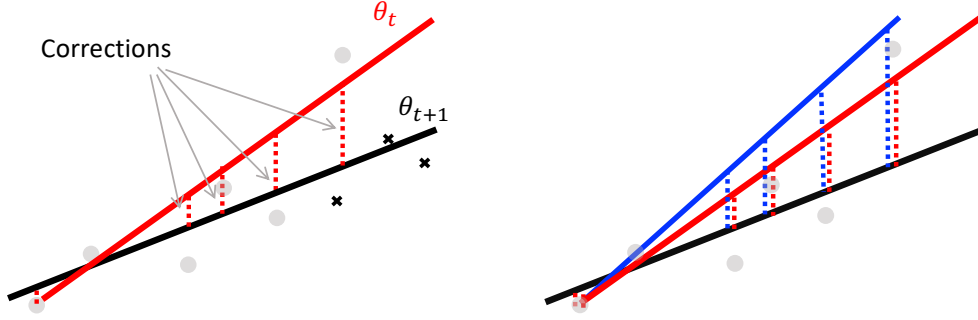


Figure 3: Illustration of correction as prediction mismatch for linear regression with  $q^{\text{iso}}$  family. The old model is trained on the gray ‘o’ and the new model additionally includes the black ‘x’ too. Corrections are mismatches over old examples (dashed red lines). The right figure shows an additional model (in blue) which has larger mismatches.

The second term disappears when the model is linear, leading to a purely mismatch-based correction as in Eq. 7. The approximation is due to the delta method and can be ignored.

The first term favors prediction matching of the past and future, while the second term deals with the residuals obtained because of past mistakes. The expression shows that the mistakes are propagated due to non-linearity of the model. The expression suggests that, for linear cases, prediction matching is sufficient, but otherwise we need additional mechanisms to fix past mistakes. A straightforward approach to reduce this error would be to use a better posterior form. For instance, when using the  $q^{\text{full}}$  family, the  $\hat{f}_i^{\text{lin}}(\theta)$  is replaced by a better quadratic surrogate, as shown in Sec. 4.3, thereby reducing the error. In fact, for squared loss, the correction term completely vanishes. The need for correction is reduced with more flexible posterior, which can speed up the adaptation process.

Flexible posteriors however can be costly, for example, computing full covariances is infeasible for larger neural networks. The expression suggests a cheaper alternative which is to simply use replay of past examples to improve continual learning. A similar approach is derived by Daxberger et al. (2023) by using the K-prior framework of Khan and Swaroop (2021), where they combine the three types of regularization methods. We show in Sec. 4.4 that K-prior, which also involves prediction matching, is a special case of posterior correction with  $q^{\text{iso}}$  family. Posterior correction generalizes K-prior by allowing the use of an arbitrary posterior distribution. This extends the prediction matching to other kinds of matching, such as, gradient and Hessian matching. It also provides a simpler way to mix and match various regularization methods as discussed in Sec. 4.4.

In Sec. 4.4, we also briefly discuss connections to other approaches covered under the K-prior framework, such as, knowledge distillation (Hinton et al., 2015), incremental Support Vector Machines (Cauwenberghs and Poggio, 2001; Liang and Li, 2009), Similarity Control (Vapnik and Izmailov, 2015), and memory-based continual learning (Lopez-Paz and Ranzato, 2017). Finally, we show an application to sequential Bayesian inference by deriving a method by Bui et al. (2017) for sequential Sparse Variational Gaussian Process as a special instance of posterior correction.

### 3.2 Influence Estimation as Posterior Correction

Influence estimation aims to estimate the influence of data examples on the models trained over them. For example, given a model  $\theta_t$  trained on  $t$  data examples, we may want to estimate the new model  $\theta_{t \setminus j}$  trained on all the data except the  $j$ 'th example. Influence estimates provide a simple expression, often obtained using a Taylor's approximation evaluated at  $\theta_t$ . These estimates can also be derived as different instances of posterior correction. Here, we will show two results to derive TracIn (Pruthi et al., 2020) and Influence Function (IF) (Jaekel, 1972; Cook, 1977; Koh and Liang, 2017).

We start by rewriting influence calculation as posterior correction. Taking a variational reformulation, we assume that  $q_t$  is given and our goal is to estimate the posterior  $q_{t \setminus j}$  trained on all example except the  $j$ 'th one. This is shown below in the first line and the following lines are obtained similarly to Eq. 4 to get the correction,

$$\begin{aligned}
q_{t \setminus j} &= \arg \min_{q \in \mathcal{Q}} \mathbb{E}_q[-\ell_j] + \sum_{i=1}^t \mathbb{E}_q[\ell_i] + \mathbb{D}_{\text{KL}}[q \parallel p_0] \\
&= \arg \min_{q \in \mathcal{Q}} \mathbb{E}_q[-\ell_j] + \mathbb{D}_{\text{KL}}[q \parallel q_t] + \sum_{i=0}^t \mathbb{E}_q[\ell_i - \hat{\ell}_{i|t}] \\
&= \arg \min_{q \in \mathcal{Q}} \mathbb{E}_q[-\hat{\ell}_{j|t}] + \mathbb{D}_{\text{KL}}[q \parallel q_t] + \sum_{i=0:t, i \neq j} \mathbb{E}_q[\ell_i - \hat{\ell}_{i|t}]
\end{aligned} \tag{9}$$

The second line characterizes the interference caused by the removal of  $\ell_j$  and the third line rearranges the terms corresponding to the  $j$ 'th example.

The TracIn estimator uses the gradient  $\nabla \ell_j(\theta_t)$  as the influence estimator. This follows as a special case of the above by using the  $q^{\text{iso}}$  family and solving the following objective where we ignore the correction term,

$$\begin{aligned}
q_{t \setminus j} &\approx \arg \min_{q \in \mathcal{Q}} \mathbb{E}_q[-\hat{\ell}_{j|t}] + \mathbb{D}_{\text{KL}}[q \parallel q_t] \\
&\implies \mathbf{m}_{t \setminus j} \approx \arg \min_{\mathbf{m}} \mathbf{m}^\top \mathbb{E}_{q_t}[-\nabla \ell_j] + \frac{1}{2} \|\mathbf{m} - \mathbf{m}_t\|^2 \\
&\implies \mathbf{m}_{t \setminus j} \approx \mathbf{m}_t + \nabla \ell_j(\mathbf{m}_t)
\end{aligned}$$

where the last line is obtained by using the delta method to approximate  $\mathbb{E}_{q_t}[\nabla \ell_i] \approx \nabla \ell_i(\mathbf{m}_t)$ . Renaming  $\mathbf{m}_t$  as  $\theta_t$ , we get the TracIn estimator  $\theta_{t \setminus j} - \theta_t \approx \nabla \ell_j(\theta_t)$ .

Unlike TracIn, the IF uses a Newton step which can be derived by approximating the correction term instead of ignoring it. Therefore it is expected to be more accurate. Specifically, we approximate the correction for all examples from  $i = 1, 2, \dots, t$  by using the 2<sup>nd</sup>-order Taylor's expansion of  $\ell_i$  at  $\mathbf{m}_t$ , as shown below:

$$\begin{aligned}
\mathbb{E}_q[\ell_i - \hat{\ell}_{i|t}] &\approx \mathbb{E}_q \left[ \theta^\top \nabla \ell_i(\mathbf{m}_t) + \frac{1}{2} (\theta - \mathbf{m}_t)^\top \nabla^2 \ell_i(\mathbf{m}_t) (\theta - \mathbf{m}_t) - \theta^\top \mathbb{E}_{q_t}[\nabla \ell_i] \right] \\
&\approx \mathbb{E}_q \left[ \frac{1}{2} (\theta - \mathbf{m}_t)^\top \nabla^2 \ell_i(\mathbf{m}_t) (\theta - \mathbf{m}_t) \right] \\
&= \frac{1}{2} (\mathbf{m} - \mathbf{m}_t)^\top \nabla^2 \ell_i(\mathbf{m}_t) (\mathbf{m} - \mathbf{m}_t) + \text{const.}
\end{aligned} \tag{10}$$

The second line is obtained again with  $\mathbb{E}_{q_t}[\nabla \ell_i] \approx \nabla \ell_i(\mathbf{m}_t)$  and simplifying. Assuming a quadratic regularizer  $\ell_0 = \frac{1}{2} \delta \|\theta\|^2$ , we can also simplify the sum as shown in Eq. 35. With

this, the influence procedure in Eq. 9 simplifies to

$$\mathbf{m}_{t \setminus j} \approx \arg \min_{\mathbf{m}} -\mathbf{m}^\top \mathbb{E}_{q_t}[\nabla \ell_j] + \frac{1}{2}(\mathbf{m} - \mathbf{m}_t)^\top \mathbf{H}_{t \setminus j}(\mathbf{m} - \mathbf{m}_t) \approx \mathbf{m}_t + \mathbf{H}_{t \setminus j}^{-1} \nabla \ell_j(\mathbf{m}_t) \quad (11)$$

where  $\mathbf{H}_{t \setminus j} = \sum_i \nabla^2 \ell_i(\mathbf{m}_t)$  where the sum runs over  $i \neq j$ . This is the IF for removal of an example (Nickl et al., 2023, Eq. 2). The derivation clearly shows the approximations used by TracIn and IF: the former ignores the corrections while the latter reduces it by using Taylor’s approximation. As a result, we expect IF to be a better estimate.

A recent proposal by Nickl et al. (2023) called the Memory-Perturbation Equation generalizes the influence estimation by using the Bayesian Learning Rule. This approach too can be seen as a special instance of posterior correction where a generic exponential family is used but the correction term is ignored. Let us denote the natural parameters of  $q_t$  and  $q_{t \setminus j}$  by  $\boldsymbol{\lambda}_t$  and  $\boldsymbol{\lambda}_{t \setminus j}$ , respectively. Then, by using the definition of  $\hat{\ell}_{t|j}$  from Eq. 3 and ignoring the correction term, we can rewrite Eq. 9 as follows:

$$\boldsymbol{\lambda}_{t \setminus j} \approx \arg \min_{\boldsymbol{\mu}} -\langle \boldsymbol{\mu}, \tilde{\nabla} \mathbb{E}_{q_t}[\ell_i] \rangle + \mathbb{D}_{\text{KL}}[q \parallel q_t] \implies \boldsymbol{\lambda}_{t \setminus j} \approx \boldsymbol{\lambda}_t + \tilde{\nabla} \mathbb{E}_{q_t}[\ell_i].$$

The second expression is obtained by noting that the natural gradient of the KL term is simply  $\boldsymbol{\lambda}_{t \setminus j} - \boldsymbol{\lambda}_t$  (Khan and Rue, 2023, Eq. 23). This is precisely the Memory-Perturbation Equation shown in Nickl et al. (2023, Eq. 6). Overall, these connections suggests two ways to improve influence estimation: either we can expand the class of posterior family or we could explicitly try to reduce the correction.

### 3.3 Model Merging as Posterior Correction

Model merging is a popular technique to improve capabilities of LLMs without retraining them from scratch. Given a pre-trained model with parameters  $\boldsymbol{\theta}_{\text{pre}}$  on loss  $\ell_{\text{pre}}$ , we ‘fine-tune’ models  $\boldsymbol{\theta}_i$  on individual tasks  $\ell_i$ . For instance, given an English-speaking<sup>2</sup> LLM, we may fine-tune it on other languages and merge them all to get a model that can also speak other languages. The merging is done via a simple addition, for instance, Task-Arithmetic (TA) (Ilharco et al., 2023) uses the following merging (with weights  $\alpha_i > 0$ ):

$$\boldsymbol{\theta}_{\text{ta}} = \boldsymbol{\theta}_{\text{pre}} + \sum_{i=1}^t \alpha_i (\boldsymbol{\theta}_i - \boldsymbol{\theta}_{\text{pre}}), \quad (12)$$

which, despite its simplicity, works extremely well for reasons unknown. In what follows, we will derive this as a special case of posterior correction with the family  $q^{\text{iso}}$ . We will then show that better approximation of the correction leads to better merging.

As before, we start by writing merging in a variational formulation denoting the pre-trained model by  $q_{\text{pre}}$ , for instance, it can be a  $q^{\text{iso}}$  candidate with mean  $\boldsymbol{\theta}_{\text{pre}}$ . Similarly, we denote by  $q_i$  the model fine-tuned on  $\ell_i$  but with a prior  $q_{\text{pre}}$ , that is, by minimizing  $\mathbb{E}_q[\ell_i] + \mathbb{D}_{\text{KL}}[q \parallel q_{\text{pre}}]$ ; we assume both  $q_i$  to be in the same family as  $q_{\text{pre}}$ . The use of KL term is similar to the proximal term which is implemented via initialization. The goal is

---

2. Note that this is an anthropomorphization to simplify things. LLMs do not speak like people do.

then to obtain the model that is trained from scratch on all the data. A popular strategy for merging posteriors is to use the following rule:

$$q_{\text{ba}} = q_{\text{pre}} \prod_{i=1}^t \left( \frac{q_i}{q_{\text{pre}}} \right)^{\alpha_i}, \quad (13)$$

which we term ‘Bayesian Arithmetic (BA)’ to draw an analogy to Task Arithmetic because the two have similar forms with addition and subtraction in TA replaced by multiplication and division in BA. The above rule is commonly used in various distributed Bayesian computation, such as, Bayesian data fusion (Mutambara, 1998; Durrant-Whyte, 2001; Wu et al., 2022), Bayesian Committee Machine (Tresp, 2000), Consensus Monte Carlo (Scott et al., 2022), and approximate inference (Vehtari et al., 2020; Ashman et al., 2022).

Clearly, the posterior merging does not recover the exact posterior fitted jointly, and we will now apply our posterior correction to write an exact expression for the error. We note that  $q_i$  takes the following dual form for  $q_i \propto q_{\text{pre}} \exp(-\hat{\ell}_{i|i})$  where the  $\hat{\ell}_{i|i}$  is the site function for  $\ell_i$  at  $q_i$ . Similarly, let  $q_{\text{pre}} \propto \exp(-\hat{\ell}_{\text{pre|pre}})$  where  $\ell_{\text{pre}}$  is assumed to contain the prior  $p_0$  used for training from scratch. With this, we can write the retraining of the model jointly on  $\ell_{\text{pre}}$ , as well as all  $\ell_i$ , as follows,

$$\begin{aligned} q_{\text{retrained}} &= \arg \min_{q \in \mathcal{Q}} \mathbb{E}_q \left[ \ell_{\text{pre}} + \sum_{i=1}^t \alpha_i \ell_i \right] + \mathbb{D}_{\text{KL}}[q \| p_0] \\ &= \arg \min_{q \in \mathcal{Q}} \mathbb{D}_{\text{KL}}[q \| q_{\text{ba}}] + \sum_{i=1}^t \alpha_i \underbrace{\mathbb{E}_q[\ell_i - \hat{\ell}_{i|i}]}_{\text{Task correction}} + \underbrace{\mathbb{E}_q[\ell_{\text{pre}} - \hat{\ell}_{\text{pre|pre}}]}_{\text{LLM correction}}. \end{aligned} \quad (14)$$

This is obtained in the same way as Eq. 4 but now using the dual forms of  $q_{\text{pre}}$  and all  $q_i$ .

It is now easy to check that TA model  $\theta_{\text{ta}}$  in Eq. 12 is obtained from  $q_{\text{ba}}$  by restricting all posteriors to  $q^{\text{iso}}$  family, that is,  $q_i = \mathcal{N}(\theta | \mathbf{m}_i, \mathbf{I})$ . Then,  $\mathbf{m}_i = \theta_i$  if we estimate  $q_1$  after using the delta method and products of all the Gaussian will simply give the addition as it is in Eq. 12. Therefore, TA can be seen as posterior correction with  $q^{\text{iso}}$  but ignoring all the correction terms in Eq. 14. To improve TA, we can approximate the correction terms, for example, by using the 2<sup>nd</sup>-order Taylor expansion. This gives us a better Hessian-weighted version of TA. Essentially, we proceed similarly to Eq. 10 to get

$$\begin{aligned} \ell_i(\theta) - \hat{\ell}_{i|i}(\theta) &\approx \frac{1}{2}(\theta - \mathbf{m}_i)^\top \mathbf{H}_i(\theta - \mathbf{m}_i) \\ \ell_{\text{pre}}(\theta) - \hat{\ell}_{\text{pre|pre}}(\theta) &\approx \frac{1}{2}(\theta - \mathbf{m}_{\text{pre}})^\top \mathbf{H}_{\text{pre}}(\theta - \mathbf{m}_{\text{pre}}) \end{aligned} \quad (15)$$

where  $\mathbf{H}_i = \nabla^2 \ell_i(\mathbf{m}_i)$  and  $\mathbf{H}_{\text{pre}} = \nabla^2 \ell_{\text{pre}}(\mathbf{m}_{\text{pre}})$ . Plugging these in Eq. 14, we get the following estimate of the merged mean, as shown in App. B,

$$\mathbf{m}_{\text{ha}} = \mathbf{m}_{\text{pre}} + \sum_{i=1}^t \alpha_i \mathbf{H}_{\text{ha}}^{-1} (\mathbf{I} + \mathbf{H}_i) (\mathbf{m}_i - \mathbf{m}_{\text{pre}}), \quad (16)$$

where  $\mathbf{H}_{\text{ha}} = \mathbf{I} + \mathbf{H}_{\text{pre}} + \sum_i \alpha_i \mathbf{H}_i$ .

This is exactly the update derived by Daheim et al. (2024, Eq. 12) by using *gradient mismatch*. In fact, it is easy to show that gradient mismatch is a special case of posterior correction. This is obtained by simply taking the derivatives of the correction term,

$$\nabla_{\mathbf{m}} \mathbb{E}_q[\ell_i - \hat{\ell}_{i|t}] = \mathbb{E}_q[\nabla \ell_i] - \mathbb{E}_{q_t}[\nabla \ell_i] \approx \nabla \ell_i(\mathbf{m}) - \nabla \ell_i(\mathbf{m}_t). \quad (17)$$

Posterior correction generalizes their approach and relaxes the assumptions made by them (they assume a quadratic regularizer). Similarly to previous section, the result can also be directly obtained by expanding the posterior form from  $q^{\text{iso}}$  to  $q^{\text{full}}$ .

### 3.4 Federated Learning as Posterior Correction

The goal of federated learning (McMahan et al., 2017) is to learn a joint model  $\theta_{\text{jnt}}$  at a server by communicating with local models  $\theta_i$  for  $i = 1, 2, \dots, t$  trained separately at  $t$  clients. The local models use local losses  $\ell_i$  which are not accessible at the server. This restriction forces the server to adapt all the available information and iteratively arrive at the correct solution. We will now show that the methods used in federated learning can be seen as an iterative posterior correction to eventually drive the error to zero. This is unlike the methods discussed so far which all used approximations for the correction term and therefore do not achieve perfect correction. Federated learning fixes this.

To show this, we start with a variational formulation for the joint-model learning and write the dual form of the solution using Eq. 3,

$$\begin{aligned} q_{\text{jnt}} &= \arg \min_{q \in \mathcal{Q}} \mathbb{E}_q \left[ \sum_{i=1}^t \ell_i \right] + \mathbb{D}_{\text{KL}}[q \parallel p_0] \\ &\propto p_0 \prod_{i=1}^t \exp \left( -\hat{\ell}_{i|\text{jnt}} \right), \text{ where } \hat{\ell}_{i|\text{jnt}}(\theta) = \langle \mathbf{T}(\theta), \tilde{\nabla} \mathbb{E}_{q_{\text{jnt}}}[\ell_i] \rangle. \end{aligned} \quad (18)$$

Here, for simplicity, we have assume that  $q_{\text{jnt}}$  and  $p_0$  are both the same exponential family which implies  $p_0 \propto \exp(-\hat{\ell}_{0|\text{jnt}})$  and simplifies the discussion. However, the connection we discuss is valid even when this condition does not hold.

The dual form factorizes across local models which suggests a natural adaptation strategy to exploit the structure. Since each local  $\hat{\ell}_{i|\text{jnt}}$  contributes a factor, we can learn a local model  $q_i$  to estimate a local approximation  $\hat{\ell}_{i|i}$  and slowly try to make it similar to the desired  $\hat{\ell}_{i|\text{jnt}}$ . To formalize this, consider the following estimate of  $q_{\text{jnt}}$  obtained by using natural-gradients at local models  $q_i$ ,

$$\hat{q}_{\text{jnt}} \propto p_0 \prod_{i=1}^t \exp \left( -\hat{\ell}_{i|i} \right) \quad \text{where } \hat{\ell}_{i|i}(\theta) = \langle \mathbf{T}(\theta), \tilde{\nabla} \mathbb{E}_{q_i}[\ell_i] \rangle. \quad (19)$$

This differs from Eq. 18 because – instead of computing natural gradients at  $q_{\text{jnt}}$  – here we compute them at a local  $q_i$  (highlighted in red). We can then write the correction for the posterior  $\hat{q}_{\text{jnt}}$  by plugging it in the first line of Eq. 18 and simplifying similarly to Eq. 4,

$$q_{\text{jnt}} = \arg \min_{q \in \mathcal{Q}} \sum_{i=1}^t \mathbb{E}_q \left[ \ell_i - \hat{\ell}_{i|i} \right] + \mathbb{D}_{\text{KL}}[q \parallel \hat{q}_{\text{jnt}}] \implies q_{\text{jnt}} \propto \hat{q}_{\text{jnt}} \prod_{i=1}^t \exp \left( \ell_{i|\text{jnt}} - \hat{\ell}_{i|i} \right). \quad (20)$$

A derivation for the second expression is included in App. C. The expression suggests that more accurate  $\hat{q}_{\text{jnt}}$  yields smaller corrections, which is also faster to adapt to get to  $q_{\text{jnt}}$ . The goal of local  $q_i$  should be to continue to reduce the correction. One straightforward strategy could be to locally minimize at each client  $i$ ,

$$q_i = \arg \min_{q_i \in \mathcal{Q}} \mathbb{E}_{q_i} [\ell_i - \hat{\ell}_{i|i}] + \mathbb{D}_{\text{KL}}[q_i \parallel \hat{q}_{\text{jnt}}]. \quad (21)$$

We can then iterate between Eq. 19 and Eq. 21. This strategy is exactly the Partition Variational Inference (PVI) method of Ashman et al. (2022). For a comparison, see the version discussed in Swaroop et al. (2025, Eq. 5). Essentially, by using Eq. 19, we can see that each local update of Eq. 21 simply replaces the old factor  $\hat{\ell}_{i|i}$  in  $\hat{q}_{\text{jnt}}$  by a fresh one,

$$q_i = \arg \min_{q_i \in \mathcal{Q}} \mathbb{E}_{q_i} \left[ \ell_i + \sum_{j=1:t, j \neq i} \hat{\ell}_{j|t} \right] + \mathbb{D}_{\text{KL}}[q_i \parallel p_0]. \quad (22)$$

Proceeding in this manner, if the algorithm converges, we recover the joint model both locally and globally, that is,  $q_{\text{jnt}} = \hat{q}_{\text{jnt}} = q_i$  for all  $i$ . The PVI update is essentially a scheme to iteratively minimize the gradient mismatch, eventually driving it to 0. The connection ultimately shows that PVI aims for a perfect posterior correction.

There are better alternatives to PVI and they too are connected to posterior correction. A recent work by Swaroop et al. (2025) connects PVI to Federated Alternating Direction Method of Multipliers (ADMM) (Gabay and Mercier, 1976; Glowinski and Marroco, 1975) and its variants for federated deep learning, for example, FedADMM (Gong et al., 2022; Wang et al., 2022; Zhou and Li, 2023) and FedDyn (Acar et al., 2021). They modify the PVI update to include a learning rate  $1/\gamma$  in front of the KL term in Eq. 21. This simple change makes it similar to the update of Tseng (1991) and improves convergence. The scheme simply replaces the stale  $\hat{\ell}_{i|t}^{\text{old}}$  by a moving average  $\gamma \ell_{i|t} + (1 - \gamma) \ell_{i|t}^{\text{old}}$ .

Möllenhoff et al. (2025) further show that ADMM can also be derived as a special case of a more general *Bayesian Duality*. Essentially, they replace the global update shown in Eq. 19 by the following which employs a learning rate  $\alpha = \gamma/(\gamma + t)$ ,

$$\hat{q}_{\text{jnt}} \propto \left[ \prod_{i=1}^t q_i^{1/t} \right]^\alpha \left[ p_0 \prod_{i=1}^t \exp \left( -\hat{\ell}_{i|i} \right) \right]^{(1-\alpha)} \quad (23)$$

This schedule is shown to work well for ADMM and Möllenhoff et al. (2025) also show improvements over PVI. In our view, such updates further stabilizes the local update of individual  $\hat{\ell}_{i|t}$  and improves convergence. This connection also enable us to write posterior correction as optimizing a Lagrangian, therefore using an algorithm similar to the well-known primal-dual methods; interested readers are encouraged to see Möllenhoff et al. (2025). These extensions bring new insight into the existing Bayesian literature (Vehtari et al., 2020; Ashman et al., 2022) which is currently missing a rigorous connection to decade old work in convex duality and distributed optimization. Altogether, with such connections, posterior correction opens a new way to design better adaptive algorithms.

## 4. Method Details and Further Connections

In this section, we give further details on the dual perspective of the Bayesian Learning Rule (BLR), as well as alternate interpretations of posterior correction. Then, we discuss a few more details regarding the connections to continual learning methods. Specifically, we derive expressions for correction terms, derive K-priors as a special case, and discuss application to online variational inference.

### 4.1 A Dual Perspective of The Bayesian Learning Rule (BLR)

Posterior correction is derived by using a dual perspective of the Bayesian Learning Rule (BLR) by Khan and Rue (2023). The perspective is similar to those used in the optimization literature, for example, the closed-circuit of Rockafellar (1967, Fig. 2) which also have parallels in the Bayesian literature, for example, the representer theorem by Kimeldorf and Wahba (1970). Both use ‘dual’ representation of solutions. The representer theorem in Kernel methods generalizes them to a pure functional form by exploiting convex duality (Schölkopf et al., 2001). The dual-perspective of the BLR extends such results to the VL problem.

The form in Eq. 3 represents the solution  $q_t$  as a product of multiple factors, traditionally called sites in the approximate Bayesian literature (Minka, 2001). The expression is derived by using the expectation parameters  $\boldsymbol{\mu} = \mathbb{E}_q[\mathbf{T}(\boldsymbol{\theta})]$ . The pair  $(\boldsymbol{\lambda}, \boldsymbol{\mu})$  live in two spaces dual to each other, that is, they are connected by a bijective Legendre transform (Khan and Rue, 2023, Sec 2.2). The natural-gradient are simply gradients in the  $\boldsymbol{\mu}$ -space which simplifies their computations. The dual form is obtained by simply setting  $\tilde{\nabla}$  of Eq. 1 to 0,

$$\sum_{i=0}^t \tilde{\nabla} \mathbb{E}_{q_t}[\ell_i] - \tilde{\nabla} \mathcal{H}(q_t) = 0 \implies \boldsymbol{\lambda}_t = \sum_{i=0}^t \tilde{\nabla} \mathbb{E}_{q_t}[-\ell_i] \implies q_t \propto \prod_{i=0}^t e^{-\langle \mathbf{T}(\boldsymbol{\theta}), \tilde{\nabla} \mathbb{E}_{q_t}[\ell_i] \rangle}. \quad (24)$$

The first expression follows by rewriting the KL, the second equation from the fact that  $\tilde{\nabla} \mathcal{H}(q_t) = -\boldsymbol{\lambda}_t$ , and the third equation is obtained by applying  $\exp(\langle \mathbf{T}(\boldsymbol{\theta}), \cdot \rangle)$  on both sides. For non-constant base measures  $h(\boldsymbol{\theta})$ , only a small change in the second step is required due to the change in the natural gradient of the entropy term (Khan and Rue, 2023, App. B). We choose to handle it through the site of  $\ell_0$ .

The derivation reveals the underlying dual structure: The left side in the second expression belongs to the  $\boldsymbol{\lambda}$  space while the gradients in the right are computed in the dual  $\boldsymbol{\mu}$  space. This is akin to the dual structure used in Rockafellar (1967, Fig. 2) and other forms of representer theorems. A formal derivation can be done via a Lagrangian formulation which shows that natural gradients can also be seen as Lagrange multipliers. We skip this discussion as it is not our main focus in this paper, but interested readers may see similar formulations in Khan et al. (2013); Adam et al. (2021); Möllenhoff et al. (2025).

As shown in Table 1, the site functions for Gaussian take a similar form to surrogates obtained by using Taylor’s method. We now derive some of those. We first consider the  $q_t^{\text{iso}} = \mathcal{N}(\boldsymbol{\theta} | \mathbf{m}_t, \mathbf{I})$  for which we have  $\mathbf{T}(\boldsymbol{\theta}) = \boldsymbol{\theta}$  and  $\boldsymbol{\mu}_t = \mathbb{E}_{q_t}[\boldsymbol{\theta}] = \mathbf{m}_t$ . With this, we get

$$\tilde{\ell}_{i|t}^{\text{iso}}(\boldsymbol{\theta}) = \langle \boldsymbol{\theta}, \nabla_{\mathbf{m}} \mathbb{E}_{q_t}[\ell_i] \rangle = \boldsymbol{\theta}^\top \mathbb{E}_{q_t}[\nabla \ell_i], \quad (25)$$

where we pushed the gradient inside by using the Bonnet’s theorem (Bonnet, 1964; Khan and Rue, 2023). We can also write a similar expression for a  $q^{\text{full}} = \mathcal{N}(\boldsymbol{\theta} | \mathbf{m}_t, \mathbf{S}_t^{-1})$  for which

we have  $\mathbf{T}(\boldsymbol{\theta}) = (\boldsymbol{\theta}, \boldsymbol{\theta}\boldsymbol{\theta}^\top)$ . Using this, we get

$$\begin{aligned}\hat{\ell}_{i|t}^{\text{full}}(\boldsymbol{\theta}) &= \langle \mathbf{T}(\boldsymbol{\theta}), \tilde{\nabla} \mathbb{E}_{q_t}[\ell_i] \rangle \\ &= \boldsymbol{\theta}^\top \mathbb{E}_{q_t} [\nabla \ell_i - (\nabla^2 \ell_i) \mathbf{m}_t] + \frac{1}{2} \boldsymbol{\theta}^\top \mathbb{E}_{q_t} [\nabla^2 \ell_i] \boldsymbol{\theta} \\ &= \boldsymbol{\theta}^\top \mathbb{E}_{q_t} [\nabla \ell_i] + \frac{1}{2} (\boldsymbol{\theta} - \mathbf{m}_t)^\top \mathbb{E}_{q_t} [\nabla^2 \ell_i] (\boldsymbol{\theta} - \mathbf{m}_t) + \text{const.}\end{aligned}\tag{26}$$

where in the first line we used the expression for the natural gradients given in Khan and Rue (2023, Eq. 11). In the derivations above, we have ignored the constants but these expressions are essentially a result of a 1<sup>st</sup>-order Taylor expansion in the (lifted)  $\boldsymbol{\mu}$ -space used in the BLR during optimization. For instance, a BLR step uses the following 1<sup>st</sup>-order expansion of  $\mathbb{E}_q[\ell_i]$  for an arbitrary  $q$  with the parameter  $\boldsymbol{\mu}$ :

$$\mathbb{E}_q[\ell_i] \approx \mathbb{E}_{q_t}[\ell_i] + \langle \boldsymbol{\mu} - \boldsymbol{\mu}_t, \tilde{\nabla} \mathbb{E}_{q_t}[\ell_i] \rangle = \mathbb{E}_q \left[ \langle \mathbf{T}(\boldsymbol{\theta}), \tilde{\nabla} \mathbb{E}_{q_t}[\ell_i] \rangle \right] + \text{const.}$$

Such approximation in the  $\boldsymbol{\mu}$ -space taken by the BLR yield the 1<sup>st</sup> and 2<sup>nd</sup> order expansion in the  $\boldsymbol{\theta}$ -space as special cases.

## 4.2 Posterior Correction as Bayes' Filter and Natural-Gradient Mismatch

The posterior correction in Eq. 4 is derived using the variational form but the update can also be written in a form similar to Bayes' rule. Essentially, by using the dual form of  $q_{t+1}$ , we can directly express it in terms of  $q_t$  as follows,

$$q_{t+1} \propto \left( e^{-\hat{\ell}_{t+1|t+1}} \right) \times q_t \times \prod_{i=0}^t \left( e^{-\hat{\ell}_{i|t+1} + \hat{\ell}_{i|t}} \right)\tag{27}$$

The recursive form is similar to Bayesian filtering but there is an additional third term to account for the interference introduced in the past  $\ell_i$  due to the inclusion of  $\ell_{t+1}$  (a similar update for federated learning is in Eq. 20). Using this, we can also write the divergence to  $q_{t+1}$  from  $q_t$  by simply rearranging to get  $\mathbb{E}_{q_{t+1}}[\log(q_{t+1}/q_t)]$ ,

$$\mathbb{D}_{\text{KL}}[q_{t+1} \parallel q_t] = \mathbb{E}_{q_{t+1}} \left[ -\hat{\ell}_{t+1|t+1} \right] - \sum_{i=0}^t \mathbb{E}_{q_t} \left[ \hat{\ell}_{i|t+1} - \hat{\ell}_{i|t} \right] + A(\boldsymbol{\lambda}_t) - A(\boldsymbol{\lambda}_{t+1}).\tag{28}$$

The last two terms account for the normalizing constants of the two distributions.

This expression can be seen as an extension of the Information Gain (Lindley, 1956). The gain is defined for the exact posterior  $p_t$  and, when updated to  $p_{t+1}$  by using new  $\ell_{t+1}$ , it is equal to  $\mathbb{E}_{p_{t+1}}[-\ell_{t+1}]$ . The first term in Eq. 28 is similar but uses the site instead of the loss. The second and third terms are due to the correction. To the best of our knowledge, no such closed-form expression exist to the date to characterize information gain of variational posteriors. For small gains, adaptation should be quick. Large ones may take longer. This is how posterior correction quantifies the feasibility of quick adaptation.

Another view of posterior correction is to see it as the natural-gradient mismatch. This is obtained by simply writing the natural parameters  $\boldsymbol{\lambda}_{t+1}$  of  $q_{t+1}$  in Eq. 27,

$$\boldsymbol{\lambda}_{t+1} = \tilde{\nabla} \mathbb{E}_{q_{t+1}}[-\ell_{t+1}] + \boldsymbol{\lambda}_t - \underbrace{\sum_{i=0}^t \left( \tilde{\nabla} \mathbb{E}_{q_{t+1}}[\ell_i] - \tilde{\nabla} \mathbb{E}_{q_t}[\ell_i] \right)}_{\text{Mismatch}}.\tag{29}$$

The mismatch is yet another characterization of the interference. Posterior correction essentially replaces the “stale” natural gradients by the fresh new ones. For example, the PVI algorithm discussed in Sec. 3.4 implements this exact operation for posterior correction.

Finally, we discuss how to apply posterior correction during training. We consider a simple case where we want to boost the training of another model given a ‘check-point’ with a dual form

$$q_{\text{chk}} \propto p_0 \prod_{i=1}^t \exp(-\hat{\ell}_{i|\text{chk}}).$$

For simplicity, we assume that the training data for the checkpoint is the same as the training which we want to boost. We also assume that it involves the same prior  $p_0$  which takes the same exponential form as the model. However, we note that the procedure described below works for much more general cases. In fact, we can also have multiple check points stored during training construct a Bayesian Arithmetic average, for instance, as shown in Eq. 13 and Eq. 23. The case below is just chosen for the simplicity sake.

We can use  $q_{\text{chk}}$  to boost the training trajectories of the BLR where we simply replace the prior  $p_0$  by the  $q_{\text{chk}}$  and correct the losses  $\ell_i$  accordingly. This is shown below where we simplify the BLR update of Khan and Rue (2023, Eq. 22) taken at  $q_{\text{old}}$ ,

$$\begin{aligned} q_{\text{new}} &= \arg \min_{\boldsymbol{\mu}} \boldsymbol{\mu}^\top \tilde{\nabla} \left( \sum_{i=1}^t \mathbb{E}_{q_{\boldsymbol{\mu}}}[\ell_i] + \mathbb{D}_{\text{KL}}[q_{\boldsymbol{\mu}} \| p_0] \right) + \frac{1}{\rho} \mathbb{D}_{\text{KL}}[q_{\boldsymbol{\mu}} \| q_{\text{old}}] \\ &= \arg \min_{\boldsymbol{\mu}} \sum_{i=1}^t \boldsymbol{\mu}^\top \tilde{\nabla} \mathbb{E}_{q_{\boldsymbol{\mu}}}[\ell_i] + \frac{1}{\alpha} \mathbb{D}_{\text{KL}}[q_{\boldsymbol{\mu}} \| p_0^\alpha q_{\text{old}}^{1-\alpha}] \\ &= \arg \min_{\boldsymbol{\mu}} \sum_{i=1}^t \boldsymbol{\mu}^\top \tilde{\nabla} \mathbb{E}_{q_{\boldsymbol{\mu}}}[\ell_i - \hat{\ell}_{i|\text{chk}}] + \frac{1}{\alpha} \mathbb{D}_{\text{KL}}[q_{\boldsymbol{\mu}} \| q_{\text{chk}}^\alpha q_{\text{old}}^{1-\alpha}]. \end{aligned} \tag{30}$$

The first line is obtained by simply plugging in the VL objective. The second line is simplified by noting that the KL between  $q_{\boldsymbol{\mu}}$  and  $p_0$  contains conjugate term so it can be simply taken out and merged with the last KL term by redefining  $\alpha = \rho/(1+\rho)$ . An update of this form is in Khan and Rue (2023, Eq. 59) and a derivation for the conjugate prior case is given in Nickl et al. (2023, Eq. 26). The final step is obtained by also noting the fact that  $\hat{\ell}_{\text{chk}}$  is also a conjugate form, so it can simply be inserted inside the first term.

The derivation above shows the strength of the posterior correction. Not only that it can be applied to trained model, it can also be used to boost the training. The check-point also need not belong to a trained model. As long as we can express them in a dual form, we can use their stored knowledge to boost training of other models. This particular feature makes posterior correction a fundamental mechanism to reuse and repurpose the knowledge learned during model training.

### 4.3 Posterior Correction for Continual Learning

We start with the derivation of Eq. 8 which shows that the correction term for continual learning can be written as a sum of two terms. The loss function considered is a Bregman divergence (or equivalently using an exponential family) which takes the following form,

$$\mathcal{L}[y_i, \hat{y}_i(\boldsymbol{\theta})] = -y_i f_i(\boldsymbol{\theta}) + A(f_i(\boldsymbol{\theta}))$$

where  $A(\cdot)$  is the convex function that generates the Bregman divergence. Typical examples include cross-entropy loss which commonly in multi-class classification with neural networks. There, the model output (also called logits) are vectors of  $f_i^k(\boldsymbol{\theta})$  for each class  $k$ , the function  $A(\cdot)$  is  $\log \sum_{k=1}^K \exp(f_i^k(\boldsymbol{\theta}))$ , the log-sum-exp function over all  $K$  classes. The predictions are obtained by simply taking a softmax over  $f_i^k(\boldsymbol{\theta})$ . For simplicity, we will present the derivation for a scalar  $f_i(\boldsymbol{\theta})$ .

For such loss functions, the correction term can be simplified as follows,

$$\begin{aligned} \ell_i(\boldsymbol{\theta}) - \hat{\ell}_{i|t}^{\text{iso}}(\boldsymbol{\theta}) &= \mathcal{L}[y_i, \hat{y}_i(\boldsymbol{\theta})] - \boldsymbol{\theta}^\top \mathbb{E}_{q_t} [\nabla \mathcal{L}[y_i, \hat{y}_i]] \\ &= -y_i f_i(\boldsymbol{\theta}) + A(\boldsymbol{\theta}) - \boldsymbol{\theta}^\top \mathbb{E}_{q_t} [\nabla f_i(\boldsymbol{\theta}) (\hat{y}_i(\boldsymbol{\theta}) - y_i)] \\ &\approx -\hat{y}_{i|t} f_i(\boldsymbol{\theta}) + A(\boldsymbol{\theta}) + r_{i|t} f_i(\boldsymbol{\theta}) - r_{i|t} \boldsymbol{\theta}^\top \nabla f_i(\mathbf{m}_t) \\ &= \mathcal{L}[\hat{y}_{i|t}, \hat{y}_i(\boldsymbol{\theta})] + r_{i|t} \left[ f_i(\boldsymbol{\theta}) - \hat{f}_i^{\text{lin}}(\boldsymbol{\theta}) \right] + \text{const.} \end{aligned} \quad (31)$$

The first and second line follows from the definition of the loss and site. In the third line, we subtract and add  $\hat{y}_{i|t} f_i(\boldsymbol{\theta})$ , then rearrange to write it in terms of the residual  $r_{i|t} = \hat{y}_{i|t} - y_i$ . The approximation is due to the 1<sup>st</sup>-order delta method to simplify the expectation with respect to  $q_t$  over  $f_i(\boldsymbol{\theta})$  in the last term. The final line is obtained by using  $\hat{f}_i^{\text{lin}}(\boldsymbol{\theta}) = (\boldsymbol{\theta} - \mathbf{m}_t)^\top \nabla f_i(\mathbf{m}_t)$  with a term that is constant with respect to  $\boldsymbol{\theta}$ . The only assumption made is the delta method for the expectation which might be useful for Bayesian cases but otherwise can be safely ignored. Other than that, the expression holds for general models, such as, neural networks.

The derivation directly extends to  $q^{\text{full}}$ , where we get an additional quadratic term,

$$\begin{aligned} \ell_i(\boldsymbol{\theta}) - \hat{\ell}_{i|t}^{\text{full}}(\boldsymbol{\theta}) &= \ell_i(\boldsymbol{\theta}) - \hat{\ell}_{i|t}^{\text{iso}}(\boldsymbol{\theta}) - \frac{1}{2}(\boldsymbol{\theta} - \mathbf{m}_t)^\top \mathbf{H}_{i|t}(\boldsymbol{\theta} - \mathbf{m}_t) \\ &\approx \mathcal{L}[\hat{y}_{i|t}, \hat{y}_i(\boldsymbol{\theta})] + r_{i|t} \left[ f_i(\boldsymbol{\theta}) - \hat{f}_i^{\text{quad}}(\boldsymbol{\theta}) \right] - \frac{1}{2} \beta_{i|t} \|\nabla f_i^{\text{lin}}\|^2, \end{aligned} \quad (32)$$

where we denote  $\hat{f}_i^{\text{quad}}(\boldsymbol{\theta}) = \hat{f}_i^{\text{lin}}(\boldsymbol{\theta}) + \frac{1}{2}(\boldsymbol{\theta} - \mathbf{m}_t)^\top \nabla^2 f_i(\mathbf{m}_t)(\boldsymbol{\theta} - \mathbf{m}_t)$ , and  $\beta_{i|t} = \mathbb{E}_{q_t}[A''(f_i(\boldsymbol{\theta}))]$ . A derivation is in App. A which involves writing the Hessian of the loss in terms of the Generalized Gauss-Newton (GGN) matrix and the Hessian of the model output. Applying the delta method, then gives this approximation. The linearization error is now reduced due to the use of a better quadratic surrogate, as opposed to the linear surrogate used in Eq. 31. The last term further reduces the error and is due to the GGN matrix.

We discuss some examples to illustrate the form of the correction. For linear regression, as we saw before in Eq. 7, that for the  $q^{\text{iso}}$  family the correction is simply the prediction mismatch, but for the  $q^{\text{full}}$  the correction entirely vanishes due to the GGN term,

$$\ell_i(\boldsymbol{\theta}) - \hat{\ell}_{i|t}^{\text{full}}(\boldsymbol{\theta}) = \|\mathbf{x}_i^\top \boldsymbol{\theta} - \mathbf{x}_i^\top \boldsymbol{\theta}_t\|^2 - \frac{1}{2} \|(\boldsymbol{\theta} - \mathbf{m}_t)^\top \mathbf{x}_i\|^2 = 0.$$

This makes sense because a full Gaussian posterior is the exact posterior and there is no need for a correction. Therefore, posterior correction simply reduces to Bayes' rule. Consider another example for logistic regression where  $\hat{y}_i(\boldsymbol{\theta}) = \sigma(\mathbf{x}_i^\top \boldsymbol{\theta})$  with  $\sigma(\cdot)$  denoting the sigmoid function. The  $q^{\text{iso}}$  family yields the usual prediction matching term but the  $q^{\text{full}}$  family reduces this by comparing the linear outputs,

$$\ell_i(\boldsymbol{\theta}) - \hat{\ell}_{i|t}^{\text{full}}(\boldsymbol{\theta}) = \mathcal{L}[\sigma(\mathbf{x}_i^\top \boldsymbol{\theta}), \sigma(\mathbf{x}_i^\top \boldsymbol{\theta}_t)] - \frac{1}{2} \beta_{i|t} \|\mathbf{x}_i^\top \boldsymbol{\theta} - \mathbf{x}_i^\top \mathbf{m}_t\|^2 \quad (33)$$

The second term essentially attempts to improve the current posterior  $q_t$  by removing the stale  $\beta_{i|t}$  and replace it by the fresh ones by using the prediction mismatch term. By using a more flexible posterior, we reduce the correction required for accurate adaptation. This idea was first used in K-priors by Khan and Swaroop (2021, Fig. 3a) through gradient matching but it is also connected to PVI where we replace old surrogates by new ones during federated learning. In general, all such procedures are generalized via posterior correction. We next show that the correction term generalizes K-priors to exponential-family posteriors.

#### 4.4 Knowledge-Adaptation Priors as Posterior Correction

The prior shown in Eq. 6 is a generalization of the K-prior by Khan and Swaroop (2021). We will now show this for the K-prior presented in Khan and Swaroop (2021, Eq. 8) for a linear model  $f_i(\boldsymbol{\theta}) = \mathbf{x}_i^\top \boldsymbol{\theta}$  over the Bregman loss  $\ell_i(\boldsymbol{\theta}) = \mathcal{L}(y_i, \hat{y}_i(\boldsymbol{\theta}))$  and quadratic regularizer  $\ell_0(\boldsymbol{\theta}) = \frac{1}{2}\delta\|\boldsymbol{\theta}\|^2$ . We have already derived the correction terms for  $i = 1, 2, \dots, t$ . Therefore, we just need the correction term for  $\ell_0$  and the KL term. We derive this below.

The correction term for  $\ell_0$  in the case of  $q^{\text{iso}}$  takes a bit more effort because the base measure is not constant:  $h(\boldsymbol{\theta}) = -\frac{1}{2}\|\boldsymbol{\theta}\|^2 - \frac{1}{2}P\log(2\pi)$ . For such a case, we use

$$\begin{aligned}\ell_0(\boldsymbol{\theta}) - \hat{\ell}_{0|t}^{\text{iso}}(\boldsymbol{\theta}) &= \ell_0(\boldsymbol{\theta}) - \boldsymbol{\theta}^\top \mathbb{E}_{q_t}[\nabla \ell_0 + \nabla \log h] + \log h(\boldsymbol{\theta}) \\ &= \frac{1}{2}\delta\|\boldsymbol{\theta}\|^2 - \delta\boldsymbol{\theta}^\top \mathbf{m}_t + \boldsymbol{\theta}^\top \mathbf{m}_t - \frac{1}{2}\|\boldsymbol{\theta}\|^2 \\ &= \frac{1}{2}\delta\|\boldsymbol{\theta} - \mathbf{m}_t\|^2 - \frac{1}{2}\|\boldsymbol{\theta} - \mathbf{m}_t\|^2 + \text{const.}\end{aligned}\tag{34}$$

Adding this to the KL term, we get

$$\mathbb{E}_q[\ell_0 - \hat{\ell}_{0|t}] + \mathbb{D}_{\text{KL}}[q \| q_t] = \frac{1}{2}\delta\|\mathbf{m} - \mathbf{m}_t\|^2 + \text{const.}\tag{35}$$

Substituting in Eq. 6, we obtain

$$\mathcal{K}_t^{\text{iso}}(\mathbf{m}) = \sum_{i=t}^t \mathbb{E}_q(\mathcal{L}[\hat{y}_{i|t}, \hat{y}_i(\boldsymbol{\theta})]) + \delta\frac{1}{2}\|\mathbf{m} - \mathbf{m}_t\|^2\tag{36}$$

which is exactly Khan and Swaroop (2021, Eq. 8) with respect to  $\mathbf{m}$ , if we use the delta method:  $\hat{y}_{i|t} = \mathbb{E}_{q_t}[\hat{y}_i(\boldsymbol{\theta})] \approx \hat{y}_i(\mathbf{m}_t)$  and  $\mathbb{E}_q(\mathcal{L}[\hat{y}_i(\mathbf{m}_t), \hat{y}_i(\boldsymbol{\theta})]) \approx \mathcal{L}[\hat{y}_i(\mathbf{m}_t), \hat{y}_i(\mathbf{m})]$ .

The prior in Eq. 6 extends K-priors to generic exponential-family posterior forms. We will call this prior the *Variational* K-prior. The new family of K-priors extends the prediction matching idea to other type of matching. For instance, the derivatives of the correction term in Eq. 6 can be written as gradient mismatch as shown in Eq. 17. Similarly, if we use  $q^{\text{full}}$ , the correction terms implement both gradient and Hessian matching. In general, such matching of predictions, gradients, or Hessians naturally emerges through natural-gradient mismatch in Eq. 29. All we need to do is to choose an appropriate family to match the desired natural gradients.

The general form of variational K-prior also makes it easy to mix and match different regularization methods. For instance, using the correction term expression given in Eq. 32 for  $q^{\text{full}}$  family, we can make a decision to store specific examples for memory replay. For instance, examples whose residuals  $r_{i|t}$  are high can accumulate errors through the nonlinear term, so it is better to include them through replay. For the other examples, it might be

enough to simply store a representation of the inputs. Such a mixture would give rise to a memory set where we pick a set of examples for replay  $\mathcal{M}_{\text{rep}}$  and then another *disjoint* set for prediction matching  $\mathcal{M}_{\text{pred}}$ , and use the following correction term,

$$\sum_{i \in \mathcal{M}_{\text{rep}}} \left( \ell_i - \hat{\ell}_{i|t} \right) + \sum_{j \in \mathcal{M}_{\text{pred}}} \left( \mathcal{L} [\hat{y}_{j|t}, \hat{y}_j(\boldsymbol{\theta})] - \frac{1}{2} \beta_{i|t} \|\nabla f_i^{\text{lin}}\|^2 \right).$$

The two sets have to be disjoint so as to not double count the contribution of prediction matching. The benefit of using the second term above is that those examples do not need labels and they can be summarized using arbitrary input locations, for instance, similar to a core-set or set of inducing inputs. Daxberger et al. (2023) used a memory construction similarly to the above for K-prior to get good improvement on the ImageNet dataset. Similarly, Pan et al. (2020) and Khan and Swaroop (2021) used a quantity similar to  $\beta_{i|t}$  to pick examples to do prediction matching on. They all show consistent improvement and point this as a viable option to design better memory replay methods.

We now briefly discuss connections to other approaches covered under the K-prior framework. For example Knowledge Distillation (KD) (Hinton et al., 2015) considers a teacher-student learning scenario which can also be written as a posterior correction. The KD objective uses a convex combination for  $\gamma \in (0, 1)$  which can be rewritten as,

$$\gamma \sum_{i=1}^t \mathcal{L} [\hat{y}_i(\boldsymbol{\theta}_t), \hat{y}_i(\boldsymbol{\theta})] + (1 - \gamma) \sum_{i=1}^t \mathcal{L} [\hat{y}_i(\boldsymbol{\theta}_t), \hat{y}_i(\boldsymbol{\theta})] = \sum_{i=1}^t \mathcal{L} [\hat{y}_i(\boldsymbol{\theta}_t), \hat{y}_i(\boldsymbol{\theta})] + \gamma \sum_{i=1}^t r_i(\boldsymbol{\theta}_t) f_i(\boldsymbol{\theta}),$$

where  $r_i(\boldsymbol{\theta})$  are the residuals. Essentially, the parameter  $\gamma$  is used to down-weight the mistakes made by the teacher. The objective shares similarities to Eq. 31 where we explicitly ‘correct’ the mistake not just down-weight them. In KD, we have the luxury to train on all examples, but it is not always possible to store the whole training set. Posterior correction can handle such cases where we need to pay attention to specific mistakes of the teacher and ensure they do not get transferred to the teacher. Note that, typically, the architectures or teacher and students are different, so one need to choose an appropriate parameterization and divergence function to define a valid KL term. We do not go into details of this since this is out of scope for our work.

Posterior correction is also closely related to dual approaches in SVM, such as, incremental Support Vector Machines (Cauwenberghs and Poggio, 2001; Liang and Li, 2009), Similarity Control (Vapnik and Izmailov, 2015), and their extensions to neural networks, for example, (Lopez-Paz and Ranzato, 2017). The correction term can be written in a trust-region form with constraints  $\mathbb{E}_q[\ell_i] = \mathbb{E}_q[\hat{\ell}_i]$ . The dual problem can be connected to posterior correction via a Lagrangian. The continual learning case can be written in a similar fashion.

#### 4.5 Posterior Correction for Sequential and Online Variational Inference

It is well-known that the recursive elegance of Bayes’ rule is lost when approximations are used. The problem of an incorrect  $\hat{q}_{t+1}$  when Eq. 2 is used occurs in almost all approximate inference problems. This is true for even the simplest variational (Bayesian) learning on conjugate models where mean-field approximations are used (Sato, 2001; Ghahramani and

Attias, 2000; Honkela and Valpola, 2003; Hoffman et al., 2010; Wang et al., 2011), but also more recent variants of those Broderick et al. (2013); Nguyen et al. (2018); Zeno et al. (2018); Chérif-Abdellatif et al. (2019); Jones et al. (2024). Other approximations also suffer from this issue, for example, those proposed in Oppé and Winther (1999); Winther and Solla (1998); Heskes and Zoeter (2002); Csató and Oppé (2002), among many others. We expect similar problems to exist even when amortized inference is used (Archer et al., 2015; Kim et al., 2020; Krishnan et al., 2017; Campbell et al., 2021).

The posterior correction is a useful approach to fix such issues in sequential and online variational inference. As an example, we show that an existing method of Bui et al. (2017) is in fact an example of posterior correction for Sparse Variational Gaussian Process (SVGP). The method attempts to handle a challenging case of updating both the inducing input as well as hyperparameters but we will consider a simpler version to show similarities to posterior correction. SVGP uses a set of inducing variables, denoted by  $\mathbf{u}$ , to model the function  $f$ . To keep things simple, we will use a notation that fits in our framework. We will denote the posterior by  $q(\mathbf{u}, f)$  and the prior by  $p_0^\delta(\mathbf{u}, f)$  where  $\delta$  is the set of kernel hyperparameter. We will denote the negative likelihoods by  $\ell_t(f)$  which are assumed to be Gaussian. With this notation, we assume that  $q_t$  and  $\delta_t$  is given and our goal is to update  $q_{t+1}$  as well as  $\delta_{t+1}$ , but we assume the inducing variables  $\mathbf{u}$  remain at the same locations.

Because the model is conjugate, the sites are not approximate, that is,  $\hat{\ell}_{i|t} = \ell_i$ . With this, we can write the following (denoting the normalizing constant by  $\mathcal{Z}_t(\delta_t)$ ),

$$q_t = \frac{1}{\mathcal{Z}_t(\delta_t)} p_0^{\delta_t} \prod_{i=1}^t \exp(-\ell_t)$$

With the goal of updating  $q_{t+1}$  can be expressed as a posterior correction,

$$\begin{aligned} q_{t+1} &= \arg \min_{q \in \mathcal{Q}} \sum_{i=1}^{t+1} \mathbb{E}_q[\ell_i] + \mathbb{D}_{\text{KL}}[q \| p_0] + \log \mathcal{Z}_{t+1}(\delta) \\ &= \arg \min_{q \in \mathcal{Q}} \mathbb{E}_q[\ell_{t+1}] + \mathbb{D}_{\text{KL}}[q \| q_t] + \mathbb{E}_q[\ell_0^\delta - \ell_0^{\delta_t}] + \log \frac{\mathcal{Z}_{t+1}(\delta)}{\mathcal{Z}_t(\delta_t)} \\ &= \arg \min_{q \in \mathcal{Q}} \log \frac{\mathcal{Z}_{t+1}(\delta)}{\mathcal{Z}_t(\delta_t)} + \mathbb{E}_q \left[ \log \left( \frac{p_0^{\delta_t}}{p_0^\delta} \frac{q}{q_t} \frac{1}{e^{-\ell_{t+1}}} \right) \right] \end{aligned} \quad (37)$$

where in the first line we added the explicit normalizing constant to show dependency on  $\delta$ . The last line is written to show equivalence to Bui et al. (2017, Eq. 5, 2<sup>nd</sup> line). Modification to handle non-conjugate likelihood is straightforwardly obtained by adding back the corrections for all the past  $\ell_i$ . The procedure can also benefit from adding memory replay, similarly to previous sections. A recent extension by Chang et al. (2023) explores such directions using the dual representations and finds good improvements. Additionally, a procedure by Adam et al. (2021) is also a special instance of posterior correction to speed up hyperparameter learning in GPs.

## 5. Conclusion and Discussion

The ability to quickly adapt is the holy grail of intelligence and we propose posterior correction as a natural mechanism for instill this ability in learning algorithms. The suggestion

is that, when adapting, the algorithm should aim to correct its posterior distribution. The posterior could be explicit (for example, for a Bayesian learners) or implicit otherwise (as Khan and Rue (2023) have shown). Nevertheless, the suggestion is that all learners should aim to correct this very same object. The proposal shares this view with those of Zellner (1988) who derived Bayes’ rule as an ‘optimal’ information processing rule. There, the information is neither created nor destroyed, simply preserved. Our proposal extends this principle to approximate posteriors obtained by using the Bayesian learning rule of Khan and Rue (2023). The message fits well into the current trend of deep learning where exact posteriors are impossible to compute. The new proposal makes it possible to apply the existing Bayesian principle to all such new cases.

The paper also discusses posterior-corrections methods. Various approximations with Taylor’s method seem useful when computation and memory are limited, but new primal-dual techniques are also attractive to eventually aim for a near-perfect correction. But, two questions still remain: what should the algorithm remember and what new experiences should it seek? We do not provide immediate answers to these but do set the scene to search for them. Essentially, memory should be chosen to minimize the corrections needed in the future, while new experiences should be chosen to enable easy-enough corrections (or at least not too daunting). Both seem extremely difficult because the future is unknown, but an algorithm still has the freedom to explore the brave new world by choosing its future and fixing its past. Posterior correction sets the stage for all such explorations for algorithms that could some day have a natural human-like ability to adapt.

## Acknowledgement

I would like to thank my family for their support throughout this work which started in 2019, and continued through pandemic. This work is supported by JST CREST Grant Number JP-MJCR2112. I benefited immensely by having many insightful discussions with many past and current members of the ABI team at RIKEN AIP as well as the CREST-ANR Bayes-Duality Project. I specifically want to thank my colleagues Dr. T. Möllenhoff (RIKEN AIP), Dr. Siddharth Swaroop (Harvard University), Ang Ming Liang (University College London), Nico Daheim (TU Darmstadt), Dharmesh Tailor (University of Amsterdam). Many thanks to Prof. Håvard Rue (KAUST) for his mentorship and encouragement throughout.

## A. Expected Hessian for a Bregman Loss

For Bregman losses, we know that the Hessian involves a Generalized Gauss-Newton (GGN) matrix and the Hessian of the model output (Martens, 2020, Eq. 6). Using this, we can then apply delta method to simplify as follows,

$$\begin{aligned} \mathbf{H}_{i|t} &= \mathbb{E}_{q_t} [\nabla^2 \mathcal{L}[y_i, \hat{y}_i(\boldsymbol{\theta})]] \\ &= \mathbb{E}_{q_t} \left[ \nabla f_i(\boldsymbol{\theta}) \nabla^2 \mathcal{L}[y_i, z] \Big|_{z=f_i(\boldsymbol{\theta})} \nabla f_i(\boldsymbol{\theta})^\top + \nabla \mathcal{L}[y_i, z] \Big|_{z=f_i(\boldsymbol{\theta})} \nabla^2 f_i(\boldsymbol{\theta}) \right] \\ &\approx \nabla f_i(\mathbf{m}_t) \beta_{i|t} \nabla f_i(\mathbf{m}_t)^\top + r_{i|t} \nabla^2 f_i(\mathbf{m}_t). \end{aligned} \quad (38)$$

where we use the fact that the first derivative of the Bregman loss with respect to  $z$  at  $z = f_i(\boldsymbol{\theta})$  is simply the residual of the prediction  $\hat{y}_i(\boldsymbol{\theta})$ ,

$$\nabla \mathcal{L}[y_i, z]|_{z=f_i(\boldsymbol{\theta})} = A'(f_i(\boldsymbol{\theta})) - y_i = \hat{y}_i(\boldsymbol{\theta}) - y_i \implies \nabla^2 \mathcal{L}[y_i, z]|_{z=f_i(\boldsymbol{\theta})} = A''(f_i(\boldsymbol{\theta})), \quad (39)$$

where we denote  $r_{i|t} = \mathbb{E}_{q_t}[\hat{y}_i(\boldsymbol{\theta}) - y_i]$  and  $\beta_{i|t} = \mathbb{E}_{q_t}[A''(f_i(\boldsymbol{\theta}))]$ .

## B. Derivation of Eq. 16 for Model Merging

We give a detailed derivation of Eq. 16. We first derive expressions for the correction terms by using the 2<sup>nd</sup>-order Taylor expansion and apply the delta method to cancel the first term with the last term,

$$\ell_i - \hat{\ell}_{i|i} \approx \boldsymbol{\theta}^\top \nabla \ell_i(\mathbf{m}_i) + \frac{1}{2}(\boldsymbol{\theta} - \mathbf{m}_i)^\top \nabla^2 \mathbf{H}_i(\boldsymbol{\theta} - \mathbf{m}_i) - \boldsymbol{\theta}^\top \mathbb{E}_{q_i}[\nabla \ell_i] \approx \underbrace{\frac{1}{2}(\boldsymbol{\theta} - \mathbf{m}_i)^\top \mathbf{H}_i(\boldsymbol{\theta} - \mathbf{m}_i)}_{:= -\log \hat{q}_i(\boldsymbol{\theta})}$$

We denote the terms by the log of a Gaussian factor  $\hat{q}_i$ . Approximation for the correction of  $\ell_{\text{pre}}$  is obtained in a similar way, and we will denote it by  $\hat{q}_{\text{pre}}$ . Plugging these in Eq. 14, we can rewrite the posterior merging as

$$\begin{aligned} q_{\text{retrained}} &= \arg \min_{q \in \mathcal{Q}} \mathbb{D}_{\text{KL}}[q \parallel q_{\text{ba}}] + \sum_{i=1}^t \alpha_i \mathbb{E}_q[\ell_i - \hat{\ell}_{i|i}] + \mathbb{E}_q[\ell_{\text{pre}} - \hat{\ell}_{\text{pre}|\text{pre}}] \\ &\approx \arg \min_{q \in \mathcal{Q}} \mathbb{D}_{\text{KL}}[q \parallel q_{\text{ba}}] - \sum_{i=1}^t \alpha_i \mathbb{E}_q[\log \hat{q}_i] - \mathbb{E}_q[\log \hat{q}_{\text{pre}}] \\ &= \arg \min_{q \in \mathcal{Q}} \mathbb{D}_{\text{KL}} \left[ q \parallel \frac{q_{\text{pre}} \hat{q}_{\text{pre}}}{Z_t} \prod_{i=1}^t \left( \frac{q_i \hat{q}_i}{q_{\text{pre}}} \right)^{\alpha_i} \right] \end{aligned}$$

where  $Z_t$  is the normalizing constant. The solution is when  $q$  is equal to the distribution at in the right argument of the KL term. Then, using the formula for Gaussian multiplication and division,

$$\begin{aligned} \mathcal{N}(\boldsymbol{\theta} | \mathbf{m}_{1+2-3}, \mathbf{H}_{1+2-3}^{-1}) &= \frac{\mathcal{N}(\boldsymbol{\theta} | \mathbf{m}_1, \mathbf{H}_1^{-1}) \mathcal{N}(\boldsymbol{\theta} | \mathbf{m}_2, \mathbf{H}_2)}{\mathcal{N}(\boldsymbol{\theta} | \mathbf{m}_3, \mathbf{H}_3)} \\ \implies \mathbf{m}_{1+2-3} &= \mathbf{H}_{1+2-3}^{-1} (\mathbf{H}_1 \mathbf{m}_1 + \mathbf{H}_2 \mathbf{m}_2 - \mathbf{H}_3 \mathbf{m}_3) \\ \mathbf{H}_{1+2-3} &= \mathbf{H}_1 + \mathbf{H}_2 - \mathbf{H}_3, \end{aligned}$$

we recover Eq. 16 as shown below:

$$\begin{aligned} \mathbf{H}_{\text{ha}} &= (\mathbf{I} + \mathbf{H}_{\text{pre}}) + \sum_{i=1}^t \alpha_i (\mathbf{I} + \mathbf{H}_i - \mathbf{I}) = \mathbf{I} + \mathbf{H}_{\text{pre}} + \sum_i \alpha_i \mathbf{H}_i \\ \mathbf{H}_{\text{ha}} \mathbf{m}_{\text{ha}} &= \mathbf{m}_{\text{pre}} + \mathbf{H}_{\text{pre}} \mathbf{m}_{\text{pre}} + \sum_{i=1}^t \alpha_i (\mathbf{m}_i + \mathbf{H}_i \mathbf{m}_i - \mathbf{m}_{\text{pre}}) \\ &= \mathbf{H}_{\text{ha}} \mathbf{m}_{\text{pre}} + \sum_{i=1}^t \alpha_i (\mathbf{I} + \mathbf{H}_i) (\mathbf{m}_i - \mathbf{m}_{\text{pre}}) \end{aligned}$$

### C. Derivation of Eq. 20 for Federated Learning

The posterior form in Eq. 20 is obtained by simply taking the natural gradient of the objective and setting it to zero:

$$\sum_{i=1}^t \left( \tilde{\nabla} \mathbb{E}_{q_{\text{jnt}}}[\ell_i] - \tilde{\nabla} \mathbb{E}_{q_t}[\ell_i] \right) + \boldsymbol{\lambda}_{\text{jnt}} - \hat{\boldsymbol{\lambda}}_{\text{jnt}} = 0 \quad \implies \quad \boldsymbol{\lambda}_{\text{jnt}} = \hat{\boldsymbol{\lambda}}_{\text{jnt}} - \sum_{i=1}^t \left( \tilde{\nabla} \mathbb{E}_{q_{\text{jnt}}}[\ell_i] - \tilde{\nabla} \mathbb{E}_{q_t}[\ell_i] \right)$$

Here, we denote the natural parameters of  $q_{\text{jnt}}$  and  $\hat{q}_{\text{jnt}}$  by  $\boldsymbol{\lambda}_{\text{jnt}}$  and  $\hat{\boldsymbol{\lambda}}_{\text{jnt}}$ , respectively. We use the expression for the natural gradient of the KL term as  $\boldsymbol{\lambda}_{\text{jnt}} - \hat{\boldsymbol{\lambda}}_{\text{jnt}}$  (Khan and Rue, 2023, Eq. 23). Applying  $\exp(\langle \mathbf{T}(\boldsymbol{\theta}), \cdot \rangle)$  on the both sides, we get the desired result.

### References

- D. A. E. Acar, Y. Zhao, R. Matas, M. Mattina, P. Whatmough, and V. Saligrama. Federated learning based on dynamic regularization. In *International Conference on Learning Representations (ICLR)*, 2021.
- V. Adam, P. Chang, M. E. E. Khan, and A. Solin. Dual parameterization of sparse variational Gaussian processes. In *Advances in neural information processing systems*, 2021.
- E. Archer, I. M. Park, L. Buesing, J. Cunningham, and L. Paninski. Black box variational inference for state space models. *arXiv preprint arXiv:1511.07367*, 2015.
- M. Ashman, T. D. Bui, C. V. Nguyen, S. Markou, A. Weller, S. Swaroop, and R. E. Turner. Partitioned variational inference: A framework for probabilistic federated learning. *arXiv preprint arXiv:2202.12275*, 2022.
- A. Benjamin, D. Rolnick, and K. Kording. Measuring and regularizing networks in function space. In *International Conference on Learning Representations*, 2019.
- C. M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, Berlin, Heidelberg, 2006. ISBN 0387310738.
- G. Bonnet. Transformations des signaux aléatoires a travers les systemes non linéaires sans mémoire. In *Annales des Télécommunications*, volume 19, pages 203–220. Springer, 1964.
- T. Broderick, N. Boyd, A. Wibisono, A. C. Wilson, and M. I. Jordan. Streaming variational Bayes. *Advances in neural information processing systems*, 26, 2013.
- T. D. Bui, C. Nguyen, and R. E. Turner. Streaming sparse Gaussian process approximations. *Advances in Neural Information Processing Systems*, 30, 2017.
- P. Buzzega, M. Boschini, A. Porrello, D. Abati, and S. Calderara. Dark experience for general continual learning: a strong, simple baseline. In *Advances in Neural Information Processing Systems*, volume 33, pages 15920–15930, 2020.
- A. Campbell, Y. Shi, T. Rainforth, and A. Doucet. Online variational filtering and parameter learning. *Advances in Neural Information Processing Systems*, 34:18633–18645, 2021.

- G. Cauwenberghs and T. Poggio. Incremental and decremental support vector machine learning. In *Advances in Neural Information Processing Systems*, volume 13. MIT Press, 2001.
- P. E. Chang, P. Verma, S. T. John, A. Solin, and M. E. Khan. Memory-based dual Gaussian processes for sequential learning. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 4035–4054. PMLR, 2023.
- B.-E. Chérif-Abdellatif, P. Alquier, and M. E. Khan. A generalization bound for online variational inference. In *Asian conference on machine learning*, pages 662–677. PMLR, 2019.
- R. D. Cook. Detection of Influential Observation in Linear Regression. *Technometrics*, 1977.
- L. Csató and M. Opper. Sparse on-line Gaussian processes. *Neural computation*, 14(3): 641–668, 2002.
- N. Daheim, T. Möllenhoff, E. M. Ponti, I. Gurevych, and M. E. Khan. Model merging by uncertainty-based gradient matching. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024.
- E. Daxberger, S. Swaroop, K. Osawa, R. Yokota, R. E. Turner, J. M. Hernández-Lobato, and M. E. Khan. Improving continual learning by accurate gradient reconstructions of the past. *Transactions on Machine Learning Research*, 2023.
- H. Durrant-Whyte. Data fusion in decentralised sensing networks. In *Fourth International Conference on Information Fusion, 2001*, 2001.
- D. Gabay and B. Mercier. A dual algorithm for the solution of nonlinear variational problems via finite element approximation. *Computers & Mathematics with Applications*, 2(1):17–40, 1976.
- Z. Ghahramani and H. Attias. Online variational Bayesian learning. In *Slides from talk presented at NIPS workshop on Online Learning*, 2000.
- R. Glowinski and A. Marroco. Sur l’approximation, par éléments finis d’ordre un, et la résolution, par pénalisation-dualité d’une classe de problèmes de Dirichlet non linéaires. *Revue française d’automatique, informatique, recherche opérationnelle. Analyse numérique*, 9(R2):41–76, 1975.
- Y. Gong, Y. Li, and N. M. Freris. FedADMM: A robust federated deep learning framework with adaptivity to system heterogeneity. In *International Conference on Data Engineering (ICDE)*, 2022.
- T. M. Heskes and O. Zoeter. Expectation propagation for approximate inference in dynamic Bayesian networks. In *Conference on Uncertainty in Artificial Intelligence*, 2002. URL <https://api.semanticscholar.org/CorpusID:10954974>.

- G. Hinton, O. Vinyals, and J. Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.
- M. Hoffman, F. Bach, and D. Blei. Online learning for latent Dirichlet allocation. *advances in neural information processing systems*, 23, 2010.
- A. Honkela and H. Valpola. On-line variational Bayesian learning. In *4th International Symposium on Independent Component Analysis and Blind Signal Separation*, volume 76, page 102, 2003.
- E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen. LoRA: Low-rank adaptation of large language models. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022.
- G. Ilharco, M. T. Ribeiro, M. Wortsman, L. Schmidt, H. Hajishirzi, and A. Farhadi. Editing models with task arithmetic. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023.
- L. A. Jaeckel. The Infinitesimal Jackknife. Technical report, Bell Lab., 1972.
- M. Jones, P. Chang, and K. Murphy. Bayesian online natural gradient (BONG). *arXiv preprint arXiv:2405.19681*, 2024.
- M. E. Khan and H. Rue. The Bayesian learning rule. *Journal of Machine Learning Research*, 24(281):1–46, 2023. URL <http://jmlr.org/papers/v24/22-0291.html>.
- M. E. Khan and S. Swaroop. Knowledge-adaptation priors. *Advances in neural information processing systems*, 34:19757–19770, 2021.
- M. E. Khan, A. Aravkin, M. Friedlander, and M. Seeger. Fast dual variational inference for non-conjugate latent Gaussian models. In *International Conference on Machine Learning*, 2013.
- G.-H. Kim, Y. Jang, H. Yang, and K.-E. Kim. Variational inference for sequential data with future likelihood estimates. In *International Conference on Machine Learning*, pages 5296–5305. PMLR, 2020.
- G. S. Kimeldorf and G. Wahba. A correspondence between Bayesian estimation on stochastic processes and smoothing by splines. *The Annals of Mathematical Statistics*, 41(2): 495–502, 1970.
- J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017.
- P. W. Koh and P. Liang. Understanding Black-Box Predictions via Influence Functions. In *International Conference on Machine Learning*, 2017.

- R. Krishnan, U. Shalit, and D. Sontag. Structured inference networks for nonlinear state space models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31, 2017.
- Z. Liang and Y. Li. Incremental support vector machine learning in the primal and applications. *Neurocomputing*, 72(10-12):2249–2258, 2009.
- D. V. Lindley. On a measure of the information provided by an experiment. *The Annals of Mathematical Statistics*, 27(4):986–1005, 1956.
- D. Lopez-Paz and M. Ranzato. Gradient episodic memory for continual learning. In *Proceedings of the 31st International Conference on Neural Information Processing Systems, NIPS’17*, page 6470–6479, Red Hook, NY, USA, 2017. Curran Associates Inc. ISBN 9781510860964.
- J. Martens. New insights and perspectives on the natural gradient method. *Journal of Machine Learning Research*, 21(146):1–76, 2020. URL <http://jmlr.org/papers/v21/17-678.html>.
- B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas. Communication-efficient learning of deep networks from decentralized data. In A. Singh and X. J. Zhu, editors, *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, AISTATS 2017, 20-22 April 2017, Fort Lauderdale, FL, USA*, volume 54 of *Proceedings of Machine Learning Research*, pages 1273–1282. PMLR, 2017.
- T. Minka. Expectation propagation for approximate Bayesian inference. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, 2001.
- E. Mitchell, C. Lin, A. Bosselut, C. Finn, and C. D. Manning. Fast model editing at scale. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022.
- A. G. O. Mutambara. *Decentralized estimation and control for multisensor systems*. Routledge, 1998.
- T. Möllenhoff, S. Swaroop, F. Doshi-Velez, and M. E. Khan. Federated ADMM from Bayesian Duality. 2025. URL <https://arxiv.org/abs/2506.13150>.
- C. V. Nguyen, Y. Li, T. D. Bui, and R. E. Turner. Variational continual learning. In *International Conference on Learning Representations*, 2018. URL <https://openreview.net/forum?id=BkQqq0gRb>.
- P. Nickl, L. Xu, D. Tailor, T. Möllenhoff, and M. E. Khan. The memory-perturbation equation: Understanding model’s sensitivity to data. *Advances in Neural Information Processing Systems*, 36:26923–26949, 2023.
- M. Oppor and O. Winther. A Bayesian approach to online learning. In *Online learning in neural networks*, pages 363–378. Cambridge University Press, 1999.

- P. Pan, S. Swaroop, A. Immer, R. Eschenhagen, R. Turner, and M. E. Khan. Continual deep learning by functional regularisation of memorable past. In *Advances in Neural Information Processing Systems*, volume 33, pages 4453–4464. Curran Associates, Inc., 2020.
- G. Pruthi, F. Liu, S. Kale, and M. Sundararajan. Estimating Training Data Influence by Tracing Gradient Descent. In *Advances in Neural Information Processing Systems*, 2020.
- S.-A. Rebuffi, A. Kolesnikov, G. Sperl, and C. H. Lampert. iCaRL: Incremental classifier and representation learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 2001–2010, 2017.
- R. T. Rockafellar. Duality and stability in extremum problems involving convex functions. *Pacific J. Math.*, 21(1):167–187, 1967.
- M.-A. Sato. Online model selection based on the variational Bayes. *Neural Computation*, 13(7):1649–1681, 2001.
- B. Schölkopf, R. Herbrich, and A. J. Smola. A generalized representer theorem. In *International conference on computational learning theory*, pages 416–426. Springer, 2001.
- S. L. Scott, A. W. Blocker, F. V. Bonassi, H. A. Chipman, E. I. George, and R. E. McCulloch. Bayes and big data: The consensus monte carlo algorithm. In *Big Data and Information Theory*, pages 8–18. Routledge, 2022.
- R. J. Sternberg. A theory of adaptive intelligence and its relation to general intelligence. *Journal of Intelligence*, 7(4):23, 2019.
- R. S. Sutton. Two problems with backpropagation and other steepest-descent learning procedures for networks. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 8, 1986.
- S. Swaroop, M. E. Khan, and F. Doshi-Velez. Connecting federated ADMM to Bayes. In *International Conference on Learning Representations (ICLR)*, 2025.
- V. Tresp. A Bayesian committee machine. *Neural Computation*, 12(11):2719–2741, 2000.
- P. Tseng. Applications of a splitting algorithm to decomposition in convex programming and variational inequalities. *SIAM J. Control Optim.*, 29(1):119–138, 1991.
- D. van der Hoeven, T. van Erven, and W. Kotłowski. The many faces of exponential weights in online learning. In *Conference On Learning Theory*, pages 2067–2092. PMLR, 2018.
- V. Vapnik and R. Izmailov. Learning using privileged information: similarity control and knowledge transfer. *Journal of Machine Learning Research*, 16(1):2023–2049, 2015.
- A. Vehtari, A. Gelman, T. Sivula, P. Jylänki, D. Tran, S. Sahai, P. Blomstedt, J. P. Cunningham, D. Schiminovich, and C. P. Robert. Expectation propagation as a way of life: A framework for Bayesian inference on partitioned data. *Journal of Machine Learning Research*, 21(17):1–53, 2020.

- C. Wang, J. Paisley, and D. M. Blei. Online variational inference for the hierarchical dirichlet process. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 752–760. JMLR Workshop and Conference Proceedings, 2011.
- H. Wang, S. Marella, and J. Anderson. FedADMM: A federated primal-dual algorithm allowing partial participation. *Conference on Decision and Control (CDC)*, 2022.
- O. Winther and S. A. Solla. Optimal Bayesian online learning. *Theoretical aspects of neural computation*. Berlin: Springer Verlag, 1998.
- M. Wortsman, G. Ilharco, S. Y. Gadre, R. Roelofs, R. Gontijo-Lopes, A. S. Morcos, H. Namkoong, A. Farhadi, Y. Carmon, S. Kornblith, and L. Schmidt. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *International Conference on Machine Learning*, 2022a. URL <https://proceedings.mlr.press/v162/wortsman22a.html>.
- M. Wortsman, G. Ilharco, J. W. Kim, M. Li, S. Kornblith, R. Roelofs, R. Gontijo-Lopes, H. Hajishirzi, A. Farhadi, H. Namkoong, and L. Schmidt. Robust fine-tuning of zero-shot models. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition*, 2022b. <https://arxiv.org/abs/2109.01903>.
- P. Wu, T. Imbiriba, V. Elvira, and P. Closas. Bayesian data fusion with shared priors, 2022. URL <http://arxiv.org/abs/2212.07311>. arXiv:2212.07311.
- A. Zellner. Optimal information processing and Bayes’s theorem. *The American Statistician*, 42(4):278–280, 1988.
- C. Zeno, I. Golan, E. Hoffer, and D. Soudry. Task agnostic continual learning using online variational Bayes. *arXiv preprint arXiv:1803.10123*, 2018.
- S. Zhou and G. Y. Li. Federated learning via inexact ADMM. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(8):9699–9708, 2023.