HOW TO WEIGHT MULTITASK FINETUNING? FAST PREVIEWS VIA BAYESIAN MODEL-MERGING

Anonymous authors
Paper under double-blind review

ABSTRACT

When finetuning multiple tasks altogether, it is important to carefully weigh them to get a good performance, but searching for good weights can be difficult and costly. Here, we propose to aid the search with fast previews to quickly get a rough idea of different reweighting options. We use model merging to create previews by simply reusing and averaging parameters of models trained on each task separately (no retraining required). To improve the quality of previews, we propose a Bayesian approach to design new merging strategies by using more flexible posteriors. We validate our findings on vision and natural-language transformers. Our work shows the benefits of model merging via Bayes to improve multitask finetuning.

1 INTRODUCTION

Multitask finetuning has recently gained popularity due to the success of large pretrained models, but a careful weighting of tasks is crucial to get good performances (Liu et al., 2023; Xu et al., 2024; Chung et al., 2024). Similarly to the traditional multi-task learning (Caruana, 1997; Ruder, 2017), the weighting is useful in tackling data imbalance, task interference, negative transfer, and also effects of variable task difficulty (Raffel et al., 2020; Liu et al., 2023). When left unresolved, these can lead to issues, for instance, regarding safety (Jan et al., 2024). Weighting is also useful for *pre-finetuning* recently used for multi-lingual transfer and continual pretraining (Aghajanyan et al., 2021; Gemma Team, 2024a; Martins et al., 2024; Fujii et al., 2024).

Despite its importance, little has been done to address task weighting for multitask finetuning. For large models, an exhaustive search over weights is out of the question, but even if we could try a few weighting configurations, which ones should we try? There is no guide for that. Weights are often chosen arbitrarily and sometimes heuristically but these are not sufficient; see, for example, Liu et al. (2023, Sec. 6). The weighting methods used for deep learning and pretraining can be adapted to search for good weights (Ren et al., 2018; Chen et al., 2018; Raffel et al., 2020; Groenendijk et al., 2021; Du et al., 2022; Yan et al., 2022; Xie et al., 2023; Thakkar et al., 2023), but a quick guidance on reasonable search areas will still be useful to assist the search.

In this paper, we propose to aid the search with fast *previews* of performances, estimated to obtain weights that improve accuracy of multitask finetuning. We use model merging to create the previews where we train and store models on each task separately and reuse them later to create previews by simply averaging the model parameter for a wide-range of weights (Fig. 1). Our main contribution is a Bayesian approach to design new merging strategies that yield better previews over a wider range of weights. This differs fundamentally from previous work which only focus on the best performing weights (Don-Yehiya et al., 2023; Jiang et al., 2023; Feng et al., 2024; Stoica et al., 2024; Yang et al., 2024). Such strategies do not always yield good-quality previews; see Fig. 2a for an illustration.

We propose a Bayesian approach where more flexible posteriors yield better previews but with slightly higher costs (Figs. 2b and 2c). For instance, we show that Task Arithmetic (Ilharco et al., 2023) for LLMs corresponds to isotropic Gaussian posteriors, while better (and slightly more costly) Hessian-based methods (Daheim et al., 2024) employ more flexible Gaussian forms. We generalize this result to generic exponential-family posteriors and present a recipe to derive new merging strategy. We validate our method on several benchmarks on vision and natural-language transformers. For example, we experiment with image classification using Vision Transformers (ViTs) (Dosovitskiy et al., 2021) of 86M parameters and adding new languages to GEMMA with 2B parameters (Gemma Team, 2024b) for machine translation. Our results consistently show that more flexible posteriors

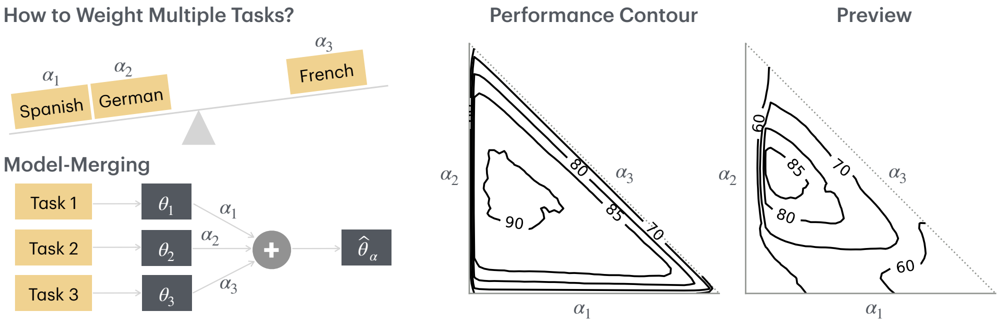
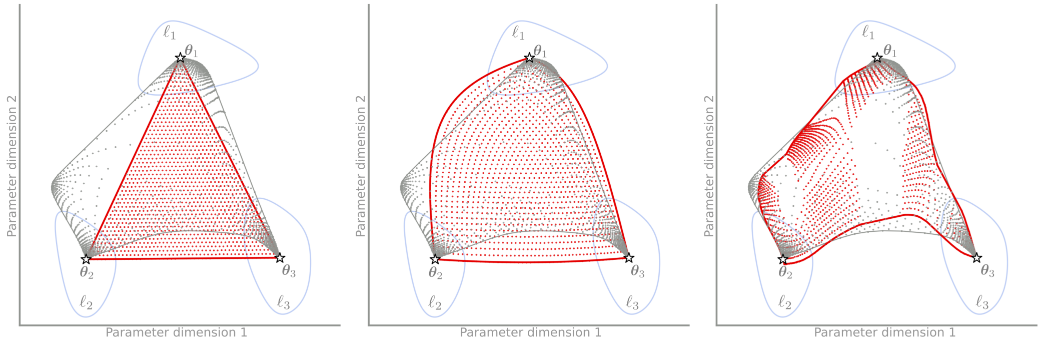


Figure 1: Our goal is to aid the search for good weights in weighted multitask finetuning. We show a performance contour for 3 tasks with weights α_1, α_2 and α_3. The well performing regions are in the middle achieving around 90% accuracy. We create a cheap preview of the contours by using model merging where previously trained models are quickly weighted with many $\boldsymbol{\alpha}$ values. The preview captures the rough shape of the true contours, encouraging a focus on the good regions.



(a) Parameter average (iso-Gaussian) (b) Full-Gaussian posterior (c) Mixture of Gaussian Posterior

Figure 2: An illustration of our Bayesian approach to improve preview quality for a toy multitask-learning problem with 3 tasks. The losses ℓ_t defined over a 2-D $\boldsymbol{\theta}$ space and are weighted by α_t varied in a fixed grid over $[0, 1]$. Panel (a) shows that parameter averaging $\sum_t \alpha_t \boldsymbol{\theta}_t$ gives poor preview (red region) of the true performances (gray contour). Each dot corresponds to a weighting option. The quality is improved in panel (b) and (c) where merging strategies using full more flexible posteriors are used, respectively. The cost is slightly increased because they need Hessians and ensembles.

produce better previews and helps us choose weights to perform more accurate multitask finetuning. Our work combines ideas from model merging and Bayesian learning to improve multitask finetuning.

2 WEIGHTED MULTITASK FINETUNING

Multitask finetuning aims to finetune on multiple tasks altogether. For example, given a Large Language Model (LLM) trained for English, we may want to finetune it on multiple languages (Muennighoff et al., 2023), for instance, German, French, Chinese, Japanese, etc. Denoting the loss of each task by $\ell_t(\boldsymbol{\theta})$ for the model parameters $\boldsymbol{\theta}$, we want to finetune over a weighted loss

$$\sum_{t=1}^{T} \alpha_t \ell_t(\boldsymbol{\theta}), \text{ where } \alpha_t > 0 \text{ for } t = 1, 2, \ldots, T. \tag{1}$$

We denote the loss-weight vector by $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \ldots, \alpha_T)$ and the finetuned parameters obtained with the weighting by $\boldsymbol{\theta}_{\boldsymbol{\alpha}}$. We will generally assume that α_t sum to 1 but this is not strictly required. Such multitask finetuning has recently become important for LLM alignment and usability. For example, it is used for improving instruction-following abilities (Chung et al., 2024; Ouyang et al., 2022), different kinds of safety tuning (Gemma Team, 2024a), combining coding tasks (Liu et al., 2023), and mixing coding and math skills into LLMs, which is useful even when they are designed for other tasks like machine translation (Martins et al., 2024).

In practice, it is important to choose $\boldsymbol{\alpha}$ carefully for reasons that are true for any multi-task learning problem (Caruana, 1997; Ruder, 2017). For instance, one issue is due to data-imbalance: different tasks may contain different types of information and some of higher quality than others. There is also task interference, for example, a model that does math well, may not necessarily be the best at languages. Additionally, learning some tasks might hurt the performances in the other tasks, and then there is variability in task difficulty: some tasks are harder to learn and we do not want those to impact the tasks that are relatively easier to learn.

The effects of these issues are often felt in practice. For example, adding too much safety data can make the model more conservative and reduce its usefulness (Bianchi et al., 2024); too much instruction finetuning can undo safety alignment and open new vulnerabilities (Qi et al., 2024; Jan et al., 2024). Such problems can be avoided by careful task weighting. Weighting is also useful during *pre-finetuning* where we try to balance multiple tasks differently during the last, say, 10% of pretraining (Aghajanyan et al., 2021; Gemma Team, 2024a; Martins et al., 2024; Fujii et al., 2024).

Despite its importance, not much work has been done to find good ways to set the weights. Decades of work exist for multitask learning but multitask finetuning is a relatively new area and is still under-explored. Similarly to multi-task learning, an exhaustive search over the whole $\boldsymbol{\alpha}$-space is not feasible when T is large. With little guidance, arbitrary values are tried to get an idea, for example, Fujii et al. (2024) try only two values of $\boldsymbol{\alpha}$ for continual pretraining. Sometimes heuristics are used and meta-learning approaches are also adopted, but the results are not always satisfactory, for example, see Liu et al. (2023, Sec. 6) who report such a result for LLMs trained for code generation.

Our goal in this paper is to provide a fast (and cheap) method to assist the search of good $\boldsymbol{\alpha}$ values. Such a guiding tool is useful to, for instance, restrict the search, to a few values and also to warm-start the optimization process. For this, we need a fast but accurate approach to quickly estimate the performance of $\boldsymbol{\theta}_{\boldsymbol{\alpha}}$ for a wide-range of $\boldsymbol{\alpha}$ values. Model merging is a useful tool for this, but the choice of merging strategy matters to get a high quality preview. We show that simple merging methods are not satisfactory because they can be quite inaccurate and may only yield good estimates for a small region in the $\boldsymbol{\alpha}$ space (see Fig. 2a). Our main technical contribution is to address this with a Bayesian approach to expand the region by designing a more accurate merging strategy. The quality is improved at the expense of cost but the approach still remains fast enough to be employed in practice. Existing methods on model merging in this space only focus on the best performing weights (Don-Yehiya et al., 2023; Jiang et al., 2023; Feng et al., 2024; Stoica et al., 2024; Yang et al., 2024). Our work instead aims to design merging strategies that work for a wide range of $\boldsymbol{\alpha}$ values.

3 FAST PREVIEWS VIA BAYESIAN MODEL-MERGING

Our goal is to create fast previews, that is, we want to estimate $\boldsymbol{\theta}_{\boldsymbol{\alpha}}$ obtained by finetuning over $\sum_t \alpha_t \ell_t$ for a wide-variety of $\boldsymbol{\alpha}$ values. The previews are useful to choose weights that give most accurate results when used to finetune jointly on all task. Our approach does this in three steps.

1. Finetune T models (denoted by $\boldsymbol{\theta}_t$) each separately over their own task $\ell_t(\boldsymbol{\theta})$.
2. Use Bayesian learning to build surrogate $\ell_t \approx \widehat{\ell}_t$ by using $\boldsymbol{\theta}_t$.
3. Create previews by finetuning over $\sum_t \alpha_t \widehat{\ell}_t$ for many $\boldsymbol{\alpha}$ values.

In step 2, we design accurate $\widehat{\ell}_t$ by using exponential-family posteriors. Such posterior always has a closed-form merging formula which enables step 3. We start by describing model merging.

3.1 MODEL MERGING AS A WEIGHTED-SURROGATE MINIMIZATION

Model merging uses simple parameter averaging but it can also be seen as minimization of a weighted sum of surrogates. For example, consider the simple averaging (SA) (Wortsman et al., 2022),

$$\widehat{\boldsymbol{\theta}}^{\text{SA}}_{\boldsymbol{\alpha}} = \sum_{t=1}^{T} \alpha_t \boldsymbol{\theta}_t \quad = \arg\min_{\boldsymbol{\theta}} \; \sum_{t=1}^{T} \alpha_t \underbrace{\left(\ell_t(\boldsymbol{\theta}_t) + \tfrac{1}{2}\|\boldsymbol{\theta} - \boldsymbol{\theta}_t\|^2\right)}_{=\widehat{\ell}_t(\boldsymbol{\theta})} + \gamma \underbrace{\tfrac{1}{2}\|\boldsymbol{\theta}\|^2}_{\widehat{\mathcal{R}}_0(\boldsymbol{\theta})}, \tag{2}$$

where the surrogate $\widehat{\ell}_t$ is a quadratic function. The term $\ell_t(\boldsymbol{\theta}_t)$ is a constant and can be ignored. The $\mathcal{R}_0$ is a regularizer with $\gamma = 1 - \sum_t \alpha_t$, which disappears if α_t sum to 1. The equality can be verified by setting the derivative to zero. Other merging techniques can also be interpreted this way. For example, Task Arithmetic (TA) (Ilharco et al., 2023) finetunes over an LLM with parameters $\boldsymbol{\theta}_{\text{LLM}}$ and can be seen as the following weighted-surrogate minimization with a regularizer,

$$\widehat{\boldsymbol{\theta}}^{\text{TA}}_{\boldsymbol{\alpha}} = \boldsymbol{\theta}_{\text{LLM}} + \sum_{t=1}^{T} \alpha_t (\boldsymbol{\theta}_t - \boldsymbol{\theta}_{\text{LLM}}) \;\; = \arg\min_{\boldsymbol{\theta}} \; \sum_{t=1}^{T} \alpha_t \tfrac{1}{2} \|\boldsymbol{\theta} - \boldsymbol{\theta}_t\|^2 + \gamma \tfrac{1}{2} \|\boldsymbol{\theta} - \boldsymbol{\theta}_{\text{LLM}}\|^2. \qquad (3)$$

Here, we removed the constant $\ell_t(\boldsymbol{\theta}_t)$ for clarity. In general, many model merging methods can be interpreted as weighted-surrogate minimization, including Wortsman et al. (2022); Matena & Raffel (2022); Jin et al. (2023); Ortiz-Jimenez et al. (2023); Daheim et al. (2024).

The interpretation highlights a major source of error when estimating performance over a wide range of $\boldsymbol{\alpha}$ values: the accuracy of the surrogates. The surrogates used above can be seen as a simplistic Taylor approximation where we assume $\nabla \ell_t(\boldsymbol{\theta}_t)$ to be zero (due to local optimality) and Hessian $\nabla^2 \ell_t(\boldsymbol{\theta}_t)$ is set to identity,

$$\begin{aligned} \ell_t(\boldsymbol{\theta}) &\approx \ell_t(\boldsymbol{\theta}_t) + \nabla \ell_t(\boldsymbol{\theta}_t)^\top (\boldsymbol{\theta} - \boldsymbol{\theta}_t) + \tfrac{1}{2} (\boldsymbol{\theta} - \boldsymbol{\theta}_t)^\top \nabla^2 \ell_t(\boldsymbol{\theta}_t) (\boldsymbol{\theta} - \boldsymbol{\theta}_t) \\ &\approx \ell_t(\boldsymbol{\theta}_t) + \tfrac{1}{2} \|\boldsymbol{\theta} - \boldsymbol{\theta}_t\|^2. \end{aligned} \qquad (4)$$

The surrogates $\widehat{\ell}_t$ are tight at only one point and their inaccuracy increases as we move away from it. Model merging can be seen as using $\sum_t \alpha_t \widehat{\ell}_t$ (along with the regularizer $\widehat{\mathcal{R}}_0$) as a proxy to estimate the results of finetuning the original $\sum_t \alpha_t \ell_t$. However, when using a wide range of $\boldsymbol{\alpha}$ values, these inaccuracy can lead to poor estimates in some regions in the $\boldsymbol{\alpha}$ space. Essentially, the errors in different $\boldsymbol{\theta}$-regions become relevant and ultimately lead to a poor estimate.

Ideally, we would like the surrogates to be designed such that they are not only locally accurate but also in a wider region. We expect such surrogates to give more accurate results and for a wider regions too, but how can we design them? Is there a general recipe to do so? We will now propose a Bayesian approach to answer this question.

3.2 A BAYESIAN APPROACH TO MODEL MERGING

The surrogate minimization approach can be seen as a special case of distributed Bayesian computation where there is a natural way to merge information distributed in different locations. We will first describe this approach and then connect it to surrogate minimization to design better surrogates.

Consider a multitask setup in a Bayesian model where there are t tasks each using a likelihood $p(\mathcal{D}_t|\boldsymbol{\theta})$ over data $\mathcal{D}_t$ and a common prior $p_0(\boldsymbol{\theta})$. For this case, there is closed-form expression to quickly get the weighted multitask posterior. We first compute T posteriors $p_t(\boldsymbol{\theta}) \propto p(\mathcal{D}_t|\boldsymbol{\theta}) p_0(\boldsymbol{\theta})$, separately over their own likelihood. The weighted posterior can be then simply be obtained by reusing the posterior p_t by using the fact that the likelihood can be written as the ratio of posterior and prior $p(\mathcal{D}_t|\boldsymbol{\theta}) = p_t(\boldsymbol{\theta})/p_0(\boldsymbol{\theta})$. This is shown below,

$$p_{\boldsymbol{\alpha}}(\boldsymbol{\theta}) \;\propto\; p_0(\boldsymbol{\theta}) \prod_{t=1}^{T} p(\mathcal{D}_t|\boldsymbol{\theta})^{\alpha_t} \;\propto\; p_0(\boldsymbol{\theta})^{\gamma} \prod_{t=1}^{T} p_t(\boldsymbol{\theta})^{\alpha_t}. \qquad (5)$$

The $\gamma = 1 - \sum_t \alpha_t$ is the same as the scalar used in front of the regularizer in Eqs. 2 and 3. Such *posterior merging* is a popular method in Bayesian literature, for example, see Bayesian committee machine (Tresp, 2000) or Bayesian data fusion (Mutambara, 1998; Durrant-Whyte & Stevens, 2001).

In fact, by choosing the Bayesian model appropriately, we can even exactly recover the solution for the weighted multitask problems. For example, suppose we want to recover the minimizer $\boldsymbol{\theta}_{\boldsymbol{\alpha}}$ by minimizing the objective $\sum_t \alpha_t \ell_t + \mathcal{R}_0$, then we can choose

$$p(\mathcal{D}_t|\boldsymbol{\theta}) \propto \exp(-\ell_t(\boldsymbol{\theta})), \qquad p_0(\boldsymbol{\theta}) \propto \exp(-\mathcal{R}_0(\boldsymbol{\theta})).$$

These choices are valid within the generalized-Bayesian framework (Zhang, 1999; Catoni, 2007; Bissiri et al., 2016). With these choices, the minimizer $\boldsymbol{\theta}_{\boldsymbol{\alpha}}$ is simply the maximum-a-posterior (MAP) solution of the merged posterior $p_{\boldsymbol{\alpha}}$, that is,

$$\boldsymbol{\theta}_{\boldsymbol{\alpha}} = \arg\max_{\boldsymbol{\theta} \in \mathbb{R}^P} \log p_{\boldsymbol{\alpha}}(\boldsymbol{\theta}) = \arg\max_{\boldsymbol{\theta} \in \mathbb{R}^P} \sum_{t=1}^{T} \alpha_t \underbrace{\log p_t(\boldsymbol{\theta})}_{=\widehat{\ell}_t(\boldsymbol{\theta})} + \gamma \underbrace{\log p_0(\boldsymbol{\theta})}_{=\widehat{\mathcal{R}}_0(\boldsymbol{\theta})}. \qquad (6)$$

Comparing this objective to Eqs. 2 and 3, we see that the Bayesian framework suggests to use the surrogate $\widehat{\ell}(\boldsymbol{\theta}) = -\log p_t(\boldsymbol{\theta})$ and regularizer $\widehat{\mathcal{R}}_0(\boldsymbol{\theta}) = -\log p_0(\boldsymbol{\theta})$. These surrogates are perfect in the sense that using them recovers the exact solution, but obviously computing them is also difficult. Our key idea is to use approximate Bayesian learning, specifically variational learning, to obtain posterior approximations and use them as surrogates.

3.3 VARIATIONAL BAYESIAN LEARNING TO BUILD EXPONENTIAL-FAMILY SURROGATES

To build surrogates $\widehat{\ell}_t$ for each task, we propose to use variational learning which finds posterior approximation $q_t(\boldsymbol{\theta})$ to the exact posterior $p_t(\boldsymbol{\theta})$,

$$q_t(\boldsymbol{\theta}) = \arg\min_{q \in \mathcal{Q}} \ \mathbb{E}_q[\ell_t(\boldsymbol{\theta})] + \mathbb{D}_{\text{KL}}[q(\boldsymbol{\theta}) \,\|\, p_0(\boldsymbol{\theta})]. \tag{7}$$

We choose $\mathcal{Q}$ to the set of exponential-family approximations or their mixtures, for example, Gaussian distribution. For such posteriors, we have good optimizers that can work at large scale (Khan & Rue, 2023). For instance, for Gaussian, the above problem can be optimized by using Adam-like optimizers (Shen et al., 2024). Even Adam can be seen as solving this problem (Khan et al., 2018) which yields a Laplace approximation. We can just use such optimizers to compute the posterior q_t.

Motivated by Eq. 6, we propose to use the approximate posterior to build the surrogate,

$$\widehat{\ell}_t(\boldsymbol{\theta}) = -\log q_t(\boldsymbol{\theta}). \tag{8}$$

As for the prior $\widehat{\mathcal{R}}_0$, we will use a Gaussian prior. These choices not only recover existing model-merging strategies but also leads to new and more accurate ones. For example, we can recover Eq. 2 by choosing q_t to be the Gaussian approximation obtained using Laplace's method at $\boldsymbol{\theta}_t$ (Khan & Rue, 2023, App. C.1) and we fix the covariance to be identity. This gives us the following,

$$q_t(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\theta} \,|\, \boldsymbol{\theta}_t, \mathbf{I}) \quad \implies \quad \widehat{\ell}_t(\boldsymbol{\theta}) = \tfrac{1}{2}\|\boldsymbol{\theta} - \boldsymbol{\theta}_t\|^2 + \text{const}. \tag{9}$$

The prior can be chosen to be isotropic Gaussian too: $p_0(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\theta} \,|\, 0, \mathbf{I})$. Similarly, Task arithmetic can be derived just by simply changing the prior to $p_0(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\theta} \,|\, \boldsymbol{\theta}_{\text{LLM}}, \mathbf{I})$.

Next, Hessian-based merging methods are obtained by using a full-Gaussian posterior, again with the Laplace's method at $\boldsymbol{\theta}_t$ and using Hessian $\mathbf{H}_t$. That is, by using $q_t(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\theta} \,|\, \boldsymbol{\theta}_t, \mathbf{H}_t)$, we get the Mahalanobis distance $\|\boldsymbol{\theta} - \boldsymbol{\theta}_t\|^2_{\mathbf{H}_t}$ as the surrogate. With this, we get a Hessian-based merging,

$$\widehat{\boldsymbol{\theta}}^{\text{Hess}}_{\boldsymbol{\alpha}} = \Big(\sum_t \alpha_t \mathbf{H}_t\Big)^{-1} \sum_t \alpha_t \mathbf{H}_t \boldsymbol{\theta}_t. \tag{10}$$

Here, we used $p_0(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\theta} \,|\, 0, \mathbf{I})$ and $\sum_t \alpha_t = 1$. If we make similar choices to the Task Arithmetic case, we recover the method proposed by Daheim et al. (2024).

In general, we can use any exponential-family form the posterior and employ them as surrogate. Such surrogates always have a closed-form solution. This is because of the form of the posterior,

$$q_t(\boldsymbol{\theta}) \propto e^{\boldsymbol{\lambda}_t^\top \mathbf{T}(\boldsymbol{\theta})} \quad \implies \quad \widehat{\ell}_t(\boldsymbol{\theta}) = -\boldsymbol{\lambda}_t^\top \mathbf{T}(\boldsymbol{\theta}) + \text{const}.$$

where we denote sufficient statistics by $\mathbf{T}(\boldsymbol{\theta})$ and natural parameter by $\boldsymbol{\lambda}_t$. For example, for Gaussian $\mathbf{T}(\boldsymbol{\theta}) = (\boldsymbol{\theta}, \boldsymbol{\theta}\boldsymbol{\theta}^\top)$ giving rise to quadratic surrogates derived earlier. The merging has a closed form solution because the minimizer of the weighted sum $\sum_t \alpha_t \widehat{\ell}_t$ is equivalent to the MAP of an exponential family distribution. The MAP solution is always available in closed-form. This is explained in App. A.1. The surrogates not only take flexible forms, but also are more globally accurate. This is because they are obtained by solving Eq. 7 which is equivalent to minimizing the KL divergence to the exact posterior p_t. Minimizing the divergence ensures that the surrogates are accurate not only locally at $\boldsymbol{\theta}_t$ but also globally in regions where q_t has high probability mass; see a discussion in (Opper & Archambeau, 2009).

3.4 IMPROVED MERGING VIA MIXTURES OF EXPONENTIAL-FAMILIES

Here, we extend to mixture of exponential-family distributions which provide more expressive posteriors and therefore even more accurate surrogates. For simplicity, we assume that $\sum_t \alpha_t = 1$,

Algorithm 1 Fast and cheap multitask previews via mixture of Gaussian merging

```
Require: $K$ different Gaussians for each of the $T$ tasks $\mathcal{N}(\boldsymbol{\theta} \,|\, \boldsymbol{\theta}_{tk}, \mathrm{diag}(1/\mathbf{h}_{tk}))$.
for all $\boldsymbol{\alpha}$ values in the preview do
    Initialize $\widehat{\boldsymbol{\theta}}_{\boldsymbol{\alpha}}$ and set $\pi_k = 1/K$ for all $k$.
    while not converged do
        For all $t, k$: compute $p_{tk} = \pi_k \mathcal{N}(\widehat{\boldsymbol{\theta}}_{\boldsymbol{\alpha}} \,|\, \boldsymbol{\theta}_{tk}, \mathrm{diag}(1/\mathbf{h}_{tk}))$; normalize $\hat{\pi}_{tk} \leftarrow \frac{p_{tk}}{\sum_{k'} p_{tk'}}$
        $\mathbf{h}_{\boldsymbol{\alpha}} \leftarrow \sum_{t,k} \hat{\pi}_{tk} \alpha_t \mathbf{h}_{tk}$
        $\widehat{\boldsymbol{\theta}}_{\boldsymbol{\alpha}} \leftarrow (\sum_{t,k} \hat{\pi}_{tk} \alpha_t \, \mathbf{h}_{tk} \boldsymbol{\theta}_{tk}) / \mathbf{h}_{\boldsymbol{\alpha}}$
    end while
end for
```

so there is no regularizer. While mode finding for mixtures is still tractable, it requires an iterative expectation-maximization (EM) procedure which should still be cheap if it converges within few steps. We assume that the k'th mixture component is an EF with natural parameter $\boldsymbol{\lambda}_{tk}$. Each component is weighted by $\pi_k > 0$ and $\sum_k \pi_k = 1$. Then, the posterior and surrogate take the following form:

$$q_t \propto \sum_{k=1}^{K} \underbrace{\pi_k e^{\boldsymbol{\lambda}_{tk}^\top \mathbf{T}(\boldsymbol{\theta})}}_{\propto p_{tk}(\boldsymbol{\theta})} \quad \Longrightarrow \quad \widehat{\ell}_t(\boldsymbol{\theta}) = -\log \sum_{k=1}^{K} \pi_k e^{\boldsymbol{\lambda}_{tk}^\top \mathbf{T}(\boldsymbol{\theta})}. \tag{11}$$

Clearly, the surrogate is much more expressive than quadratics surrogates used in model merging.

Despite the non-concavity of the objective, we can maximize it using an iterative Expectation-Maximization (EM) approach where each step has a closed-form solution. A detailed derivation is in App. A.3. As a special case, consider mixture-of-Gaussians (MoG) posterior where the updates take the following form similarly to Eq. 10:

$$\boldsymbol{\theta}^{(i+1)} \leftarrow (\mathbf{H}_{\boldsymbol{\alpha}}^{(i)})^{-1} \sum_{t,k} \hat{\pi}_{tk}^{(i)} \alpha_t \mathbf{H}_{tk} \boldsymbol{\theta}_{tk}, \text{ where } \mathbf{H}_{\boldsymbol{\alpha}}^{(i)} = \sum_{t,k} \hat{\pi}_{tk}^{(i)} \alpha_t \mathbf{H}_{tk}. \tag{12}$$

The main difference is that each component is now weighted by $\hat{\pi}_{tk}^{(i)} \propto \pi_k \mathcal{N}(\boldsymbol{\theta}^{(i)} \,|\, \boldsymbol{\theta}_{tk}, \mathbf{H}_{tk}^{-1})$, normalized over k. This update generalizes the fixed-point algorithm of Carreira-Perpiñán (2000, Section 5) which was proposed to find the modes of Gaussian mixtures.

3.5 PRACTICAL ALGORITHMS FOR FAST MULTITASK FINETUNING PREVIEWS

Here, we summarize easy recipes for fast previews of multitask finetuning. All recipes follow the three step procedure shown at the beginning of Sec. 3 and first train models independently on each task t for $t = 1, \dots, T$. These models are then reused for merging, where the merging algorithm is determined by choosing a (mixture-of-) exponential family distribution and finding its mode. For non-mixture-based distributions these are closed-form as shown in Eq. 10. These merging algorithms can then directly be used with various combinations of α_t to get many models with different task weightings for preview. For mixture-based surrogates we outline a practical solution in Alg. 1, where an EM algorithm for mode finding is shown. In practice, we only need to run this for 5-10 iterations and 10-30 mixture components which is fast but can have larger training overhead for large models due to having to train 10-30 models for each task. We use the following algorithms in experiments:

1. ADAMW-SG. For each task, compute a single Gaussian with mean at AdamW's solution (Loshchilov & Hutter, 2019) and (inverse) diagonal variance given by squared gradients computed by one extra pass through the data. Previews are calculated using Eq. 10 for all $\boldsymbol{\alpha}$.
2. IVON-HESS. For each task, we use a Gaussian with mean and diagonal variance obtained by running the variational learning method IVON (Shen et al., 2024) meaning no additional overhead. Previews are calculated by using Eq. 10 for all $\boldsymbol{\alpha}$.
3. MULTIIVON-HESS. For each task, construct a mixture-of-Gaussians obtained by independent runs of IVON. May require up to K runs. Previews are obtained using Alg. 1.

The small-scale experiments in Fig. 2 and Sec. 4.1 use full Hessian computations and exact mixture-of-Gaussian learning, and the details are described in Sec. 4.1 and Appendices B.1 and B.2.

	Figure	Model	Tasks	Simple Merging	Hessian Weighted	Mixture Weighted
CV	Fig. 3	Logistic	MNIST Imbalanced	0.2067	0.1528	**0.0463**
	Fig. 9	Logistic	MNIST Balanced	0.2015	0.1191	**0.0809**
	Fig. 4	ResNet-20	CIFAR-10	0.0252	0.0098	**0.0073**
	Fig. 5	ViT-B/32	RESISC45, GTSRB, SVHN	0.0388	**0.0263**	-
			EuroSAT, Cars, SUN397	0.0084	**0.0059**	-
NLP	Fig. 7	RoBERTa	RT, SST-2, Yelp	0.0314	**0.0281**	-
	Fig. 8	GEMMA-2B	IWSLT2017de-en, fr-en	0.5380	**0.4657**	-

Table 1: MSE between performance metrics of models obtained with multitask finetuning and models obtained with model merging for various task weightings. Model merging methods with better posterior approximations more closely match the multitask finetuned models' performances.

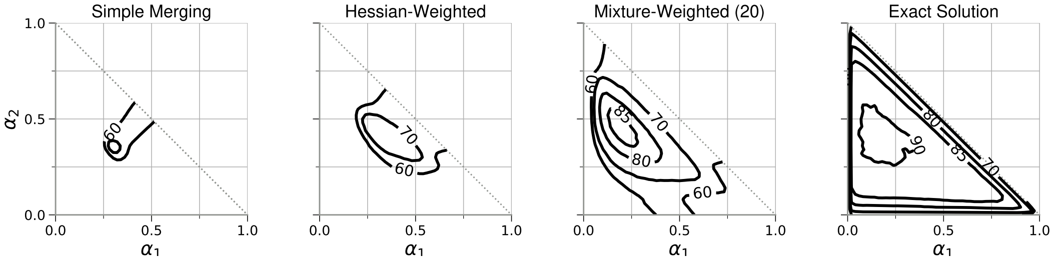


Figure 3: Multitask Learning on MNIST. As the posterior approximation gets more expressive the preview generated by the merged model resembles the exact solution better.

4 EXPERIMENTS & RESULTS

In this section, we validate our methods through several multitask finetuning experiments but first start with results on logistic regression (Sec. 4.1) which allows using exact Hessians and joint mixture-of-Gaussian learning. We then study image classification using ResNets on CIFAR-10 (Sec. 4.2). Then, we show that good previews can also be provided for image classification with vision transformers (Sec. 4.3), text classification with masked language models (Sec. 4.4), and for adding new languages for machine translation to LLMs (Sec. 4.5). An overview over our results is provided in Table 1 which shows the mean squared error (MSE) between various weightings of joint training and different merging methods: better posterior approximations result in better previews.

4.1 MULTITASK LEARNING ON MNIST

We consider MNIST broken into three tasks, each consisting of a different and disjoint subset of classes. We use a logistic regression model in two settings: one imbalanced set where number of classes per tasks vary and a balanced set. Results are shown in Fig. 3 and Fig. 9, respectively. In both cases we compare isotropic Gaussian (Simple Merging), full Gaussian (Hessian-Weighted), and mixture-of-Gaussian (Mixture-Weighted) posteriors. To compute the posterior approximations and surrogate functions, we use VON (Khan & Rue, 2023) and for mixture-of-Gaussians the joint learning algorithm from Lin et al. (2019), both with full Hessian. For Hessian-weighted merging we use Eq. 10 for all α, for the mixture-of-Gaussians we use the EM algorithm outlined in Eq. 12.

We find that better posterior approximations generally give better models but, more importantly, also more closely match the shape of the joint training solution. Simple merging would only show very few weightings as good but in fact there is a large region with good weights which is better shown by better posterior approximations. Notably, the mixture-based approach following Eq. 12 also picks up the skew to the left shown in the joint solution for imbalanced tasks.

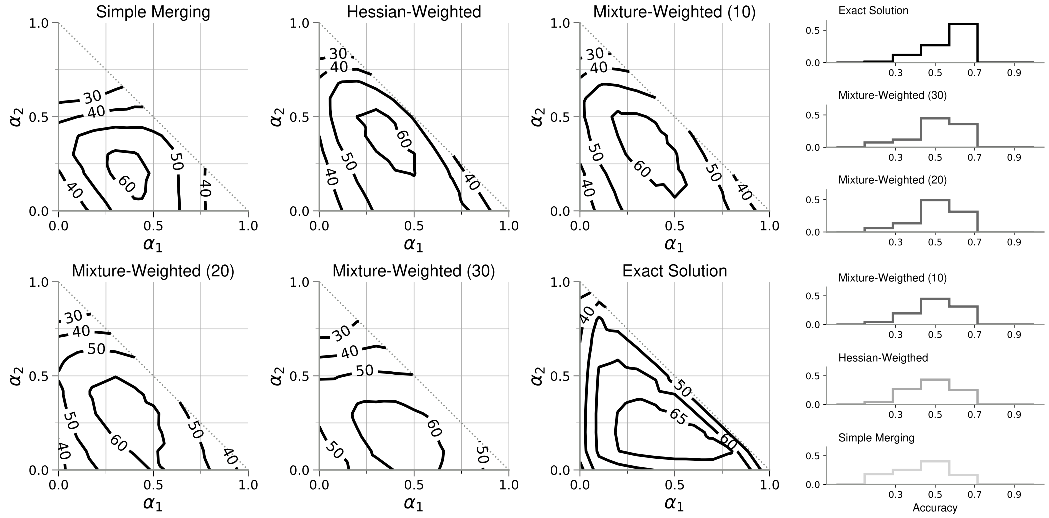


Figure 4: Results on image classification using ResNet-20 on CIFAR-10 with three tasks constructed from different sets of classes. Preview quality improves with the expressiveness of the posterior approximation. Notably, more mixture components improve the preview. Hessian-weighted previews generated with IVON-Hess and Mixture-Weighted with MultiIVON-Hess. Histograms show that the distribution of similar accuracy achieving weights also improves with better posteriors.

4.2 Image Classification on CIFAR-10

Next, we move to a neural network finetuning. We first pretrain ResNet-20 (He et al., 2016) with 260k parameters on a subset of CIFAR-10 and then finetune this checkpoint on the remaining examples. The finetuning tasks are: (1) airplane, car, ship, truck; (2) cat, dog; (3) deer, dog, frog, horse. We compare Hessian-Weighted (IVON-Hess) and Mixture-Weighted (MultiIVON-Hess) to simple averaging and Exact Solution (Joint training). Hyperparameters are in App. B.3.

Results are shown in Fig. 4 and again show that better posterior approximations yield better previews. For example, simple merging misses that performance is still good in any of the corners, especially the top and bottom right one. Hessian-Weighted merging misses the best-performing region and would suggest exploring a region slightly above it. Mixture-Weighted previews are much better and improve with the number of components. Interestingly, the best-performing region of this approach moves further down and right, that is, closer to that of the joint training solution.

4.3 Vision Transformers

Next, we experiment with multitask finetuning ViT-B/32 models (Dosovitskiy et al., 2021) based on CLIP (Radford et al., 2021) for image classification. First, we use GTSRB (Houben et al., 2013), RESISC45 (Cheng et al., 2017) and SVHN (Netzer et al., 2011); then we use EuroSAT (Helber et al., 2019), Stanford Cars (Krause et al., 2013) and SUN397 (Xiao et al., 2010). We compare Simple merging to Hessian-Weighted (AdamW-SG) and provide. Further details on training and evaluation in App. B.4. The joint solution uses a grid with spacing 0.05 to explore the possible sets of $\boldsymbol{\alpha}$.

The results are shown in Fig. 5. In both cases, the joint solution is similar and shows that almost all weightings that are not directly at the borders (where one of the tasks gets a very small weight) have good performance. While for EuroSAT, Cars, SUN397 there is a smaller region with accuracy over 84 points in accuracy, any weightings in the large contour of above 80 points will still be good. Ideally this should be reflected by a preview from merging. Hessian-based merging shows the flat triangular shape of the joint solution better than the simple method. Training times highlight the usefulness: joint training for one weighted combination takes around 51 minutes while merging takes only seconds (plus 14-17 minutes for each finetuning on separate tasks).

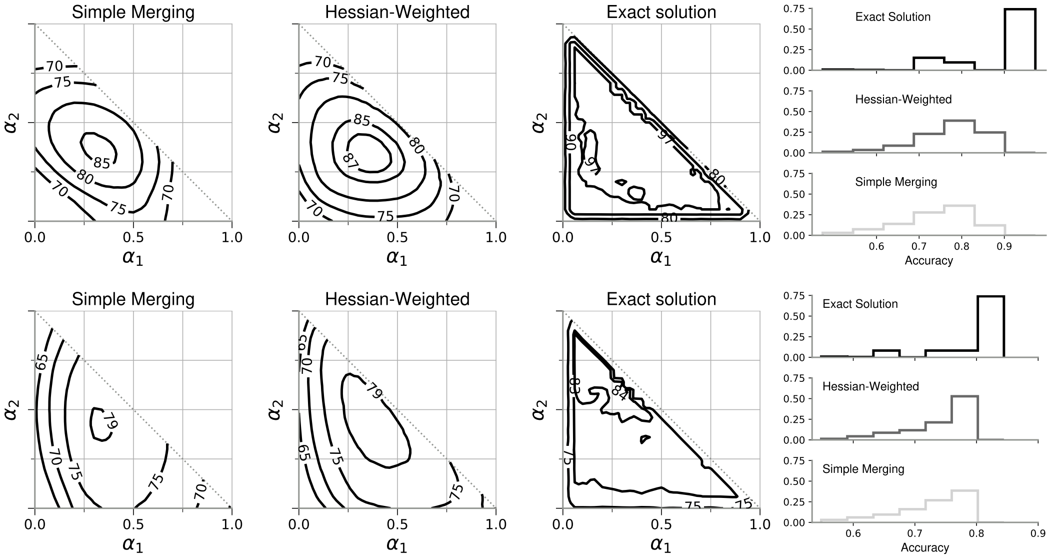
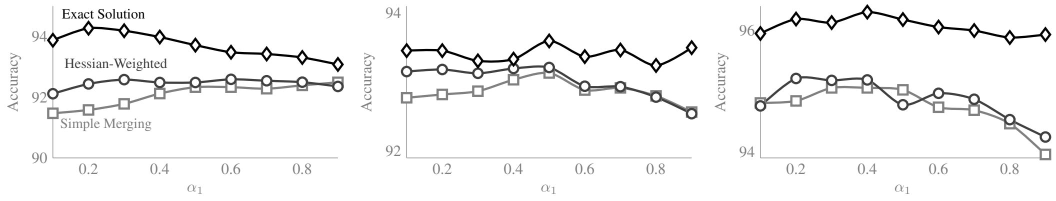


Figure 5: Results using ViT-B/32 on GTSRB, RESISC45, SVHN (top) and EuroSAT, Cars, Sun397 (bottom). The exact solution shows a large triangular area of well-performing weightings which is better captured by Hessian-weighted merging. Simple merging especially fails around the edges, whereas Hessian-Weighted (AdamW-SG) performs much better (right). Similarly we see on the histograms that the Hessian uncovers more high accuracy weights than the simple merging.



(a) SST2 & RT

(b) Yelp & RT

(c) Yelp & SST2

Figure 6: Merging of multitask finetuned RoBERTa models on pairs of sentiment analysis tasks. Model merging provides good previews of weightings for multitask finetuning but some trends (e.g. $\alpha_1 \in [0.0, 0.5]$ for SST2&RT) are only picked up by better posteriors and Hessian-Weighted merging (AdamW-SG). First-named task is weighted by α_1 and the other by $1 - \alpha_1$.

4.4 MASKED LANGUAGE MODELS

In this section, we show results when multitask finetuning masked language models for text classification. We follow Daheim et al. (2024) and train RoBERTa (Liu et al., 2019) first on the IMDB sentiment classification task Maas et al. (2011). Then, we finetune on Rotten Tomatoes (RT) (Pang & Lee, 2005), SST-2 (Socher et al., 2013), and Yelp (Zhang et al., 2015), and merge the resulting models. We use two settings: the first merges all combinations of two of the three finetuned models; the second merges all three. Due to heavy compute requirements, we run joint training with a coarser grid than model merging in the latter. We compare Simple merging and Hessian-Weighted(AdamW-SG).

Results for merging two models are shown in Fig. 6 where we see that even simple merging can often produce good previews but fails for specific weightings. For example, on SST2 and RT the best-performing factors for simple merging ($\alpha_1 = 0.9$) are the worst-performing in the joint solution. Using a diagonal Gaussian instead of an isotropic one shows a more similar trend to this joint solution. The results for merging all three tasks are shown in Fig. 7. Here, the exact solution again shows a fairly triangular shape but this is not at all reflected in the simpler merging scheme.

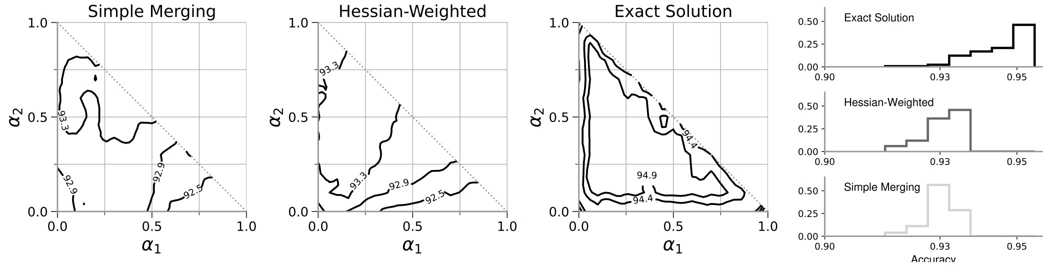
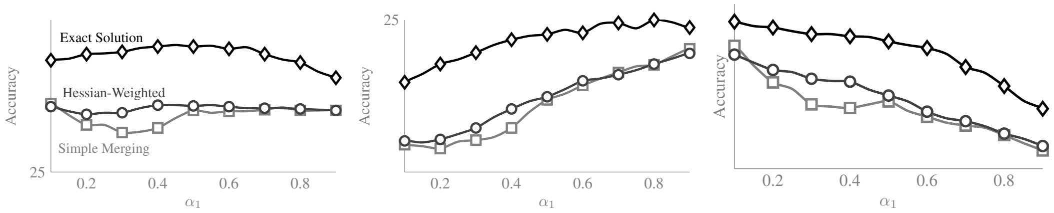


Figure 7: Preview of reweightings for RoBERTA multitask finetuned on three sentiment analysis tasks. Hessian-Weighting achieves a better preview by showing a larger region of performant weightings. The histogram shows that it more accurately recovers the trend of the weightings.



(a) DE-EN & FR-EN (b) DE-EN (c) FR-EN

Figure 8: Here, we merge LoRA-finetuned GEMMA-2B models trained on IWSLT2017de-en and IWSLT2017fr-en. Again, model merging can faithfully preview the trends of joint training with different weights. For this, better posterior approximations (Hessian-weighted with IVON-HESS) are important, for example, around $\alpha = 0.4$ on IWSLT2017fr-en.

4.5 MACHINE TRANSLATION WITH FINETUNED LLMS

Next, we show that our methods apply to LLMs with more than one billion parameters, also if they are finetuned using parameter-efficient finetuning strategies such as LoRA. In particular, we merge two GEMMA-2B-it (Gemma Team, 2024b) models finetuned on IWSLT2017 (Cettolo et al., 2017) de-en and fr-en, respectively, and compare them to training jointly on both language pairs. We use IVON-HESS for Hessian-Weighted merging. Details about the experimental set-up are in App. B.6.

Results are shown in Fig. 8. There, we find that simple merging does not always match the shape of the joint training solution, especially around $\alpha = 0.4$. Using Hessian-weighted merging improves this. Overall, this shows that the our method can also scale to larger models and datasets, even if only a small subset of the parameters is adapted during finetuning. One run of multitask finetuning for a specific weight takes around 17 hours while merging takes just 1 minute (plus 8-9 hours for finetuning on each task separately).

5 CONCLUSION

Multitask finetuning is a crucial ingredient in many neural network training recipes but good weightings between tasks are hard and expensive to find. Here, we propose to aid the search for such weightings with previews obtained from model merging, where single task models can be reused for many weight combinations. We show that model merging strategies can be derived using a Bayesian framework by defining suitable surrogate losses to the multitask objective for exponential-family-based distributions. We use this to outline various preview and merging strategies, including a new mixture-based algorithm for improved model merging. Along various experiments including image classification with Vision Transformers and machine translation with LLMs we show that model merging can effectively be used to preview multitask finetuning weightings. Flexible model merging can improve the preview quality, but also increase the cost. For example, mixture posteriors sometimes need too many mixture component to get improvements. This is not ideal for large model where storing many models is not possible. We hope to address this limitation in the future.

REFERENCES

Armen Aghajanyan, Anchit Gupta, Akshat Shrivastava, Xilun Chen, Luke Zettlemoyer, and Sonal Gupta. Muppet: Massive multi-task representations with pre-finetuning. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih (eds.), *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pp. 5799–5811. Association for Computational Linguistics, 2021. URL `https://aclanthology.org/2021.emnlp-main.468`. 1, 3

Federico Bianchi, Mirac Suzgun, Giuseppe Attanasio, Paul Rottger, Dan Jurafsky, Tatsunori Hashimoto, and James Zou. Safety-tuned LLaMAs: Lessons from improving the safety of large language models that follow instructions. In *The Twelfth International Conference on Learning Representations*, 2024. URL `https://openreview.net/forum?id=gT5hALch9z`. 3

Christopher M Bishop. *Neural networks for pattern recognition*. Oxford university press, 1995. URL `https://global.oup.com/academic/product/neural-networks-for-pattern-recognition-9780198538646`. 16

P. G. Bissiri, C. C. Holmes, and S. G. Walker. A general framework for updating belief distributions. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 2016. URL `https://rss.onlinelibrary.wiley.com/doi/abs/10.1111/rssb.12158`. 4

Miguel Carreira-Perpiñán. Mode-finding for mixtures of Gaussian distributions. *IEEE Trans. Pattern Anal. Mach. Intell. (PAMI)*, 2000. URL `https://ieeexplore.ieee.org/document/888716`. 6

Rich Caruana. Multitask learning. *Mach. Learn.*, 28(1):41–75, 1997. URL `https://doi.org/10.1023/A:1007379606734`. 1, 3

O Catoni. PAC-Bayesian supervised classification: The thermodynamics of statistical learning. *Institute of Mathematical Statistics Lecture Notes*, 2007. URL `https://www.jstor.org/stable/i20461497`. 4

Mauro Cettolo, Marcello Federico, Luisa Bentivogli, Jan Niehues, Sebastian Stüker, Katsuhito Sudoh, Koichiro Yoshino, and Christian Federmann. Overview of the IWSLT 2017 evaluation campaign. In Sakriani Sakti and Masao Utiyama (eds.), *Proceedings of the 14th International Conference on Spoken Language Translation*, pp. 2–14, Tokyo, Japan, December 14-15 2017. International Workshop on Spoken Language Translation. URL `https://aclanthology.org/2017.iwslt-1.1`. 10, 18

Zhao Chen, Vijay Badrinarayanan, Chen-Yu Lee, and Andrew Rabinovich. GradNorm: Gradient normalization for adaptive loss balancing in deep multitask networks. In *International Conference on Machine Learning (ICML)*, 2018. URL `https://proceedings.mlr.press/v80/chen18a.html`. 1

Gong Cheng, Junwei Han, and Xiaoqiang Lu. Remote sensing image scene classification: Benchmark and state of the art. *Proceedings of the IEEE*, 105(10):1865–1883, Oct 2017. ISSN 1558-2256. doi: 10.1109/jproc.2017.2675998. URL `http://dx.doi.org/10.1109/JPROC.2017.2675998`. 8, 17

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Y. Zhao, Yanping Huang, Andrew M. Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. Scaling instruction-finetuned language models. *J. Mach. Learn. Res.*, 25:70:1–70:53, 2024. URL `https://www.jmlr.org/papers/v25/23-0870.html`. 1, 2

Nico Daheim, Thomas Möllenhoff, Edoardo Ponti, Iryna Gurevych, and Mohammad Emtiyaz Khan. Model merging by uncertainty-based gradient matching. In *International Conference on Learning Representations (ICLR)*, 2024. URL `https://openreview.net/forum?id=D7KJmfEDQP`. 1, 4, 5, 9, 18

Shachar Don-Yehiya, Elad Venezian, Colin Raffel, Noam Slonim, and Leshem Choshen. Cold fusion: Collaborative descent for distributed multitask finetuning. In Anna Rogers, Jordan L. Boyd-Graber, and Naoaki Okazaki (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pp. 788–806. Association for Computational Linguistics, 2023. URL `https://aclanthology.org/2023.acl-long.46`. 1, 3

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. URL `https://openreview.net/forum?id=YicbFdNTTy`. 1, 8

Nan Du, Yanping Huang, Andrew M Dai, Simon Tong, Dmitry Lepikhin, Yuanzhong Xu, Maxim Krikun, Yanqi Zhou, Adams Wei Yu, Orhan Firat, et al. GLaM: efficient scaling of language models with mixture-of-experts. In *International Conference on Machine Learning (ICML)*, 2022. URL `https://proceedings.mlr.press/v162/du22c.html`. 1

Hugh F. Durrant-Whyte and Mike Stevens. Data fusion in decentralised sensing networks. In *International Conference on Information Fusion, 2001*, 2001. URL `https://api.semanticscholar.org/CorpusID:43837722`. 4

Shangbin Feng, Weijia Shi, Yuyang Bai, Vidhisha Balachandran, Tianxing He, and Yulia Tsvetkov. Knowledge card: Filling llms' knowledge gaps with plug-in specialized language models. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL `https://openreview.net/forum?id=WbWtOYIzIK`. 1, 3

Kazuki Fujii, Taishi Nakamura, Mengsay Loem, Hiroki Iida, Masanari Ohi, Kakeru Hattori, Hirai Shota, Sakae Mizuki, Rio Yokota, and Naoaki Okazaki. Continual pre-training for cross-lingual LLM adaptation: Enhancing japanese language capabilities. In *First Conference on Language Modeling*, 2024. URL `https://openreview.net/forum?id=TQdd1VhWbe`. 1, 3

Gemma Team. Gemma 2: Improving open language models at a practical size, 2024a. URL `https://arxiv.org/abs/2408.00118`. 1, 2, 3

Gemma Team. Gemma: Open models based on gemini research and technology, 2024b. URL `https://arxiv.org/abs/2403.08295`. 1, 10, 18

Rick Groenendijk, Sezer Karaoglu, Theo Gevers, and Thomas Mensink. Multi-loss weighting with coefficient of variations. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. URL `https://openaccess.thecvf.com/content/WACV2021/html/Groenendijk_Multi-Loss_Weighting_With_Coefficient_of_Variations_WACV_2021_paper.html`. 1

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. URL `https://ieeexplore.ieee.org/document/7780459`. 8

Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth. Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 2019. URL `https://ieeexplore.ieee.org/document/8519248`. 8, 17

Sebastian Houben, Johannes Stallkamp, Jan Salmen, Marc Schlipsing, and Christian Igel. Detection of traffic signs in real-world images: The German Traffic Sign Detection Benchmark. In *International Joint Conference on Neural Networks*, number 1288, 2013. URL `https://ieeexplore.ieee.org/document/6706807`. 8, 17

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL `https://openreview.net/forum?id=nZeVKeeFYf9`. 18

Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. Editing models with task arithmetic. In *International Conference on Learning Representations (ICLR)*, 2023. URL `https://openreview.net/forum?id=6t0Kwf8-jrj`. 1, 4

Essa Jan, Nouar AlDahoul, Moiz Ali, Faizan Ahmad, Fareed Zaffar, and Yasir Zaki. Multitask mayhem: Unveiling and mitigating safety gaps in llms fine-tuning, 2024. URL `https://arxiv.org/abs/2409.15361`. 1, 3

Junguang Jiang, Baixu Chen, Junwei Pan, Ximei Wang, Dapeng Liu, Jie Jiang, and Mingsheng Long. Forkmerge: Mitigating negative transfer in auxiliary-task learning. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 30367–30389. Curran Associates, Inc., 2023. URL `https://openreview.net/forum?id=vZHk1QlBQW`. 1, 3

Xisen Jin, Xiang Ren, Daniel Preotiuc-Pietro, and Pengxiang Cheng. Dataless knowledge fusion by merging weights of language models. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL `https://openreview.net/forum?id=FCnohuR6AnM`. 4

Mohammad Khan, Didrik Nielsen, Voot Tangkaratt, Wu Lin, Yarin Gal, and Akash Srivastava. Fast and scalable bayesian deep learning by weight-perturbation in adam. In *International conference on machine learning*, pp. 2611–2620. PMLR, 2018. URL `https://proceedings.mlr.press/v80/khan18a.html`. 5

Mohammad Emtiyaz Khan and Håvard Rue. The Bayesian learning rule. *J. Mach. Learn. Res. (JMLR)*, 2023. URL `https://jmlr.org/papers/v24/22-0291.html`. 5, 7, 16, 17

Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *2013 IEEE International Conference on Computer Vision Workshops*, pp. 554–561, 2013. doi: 10.1109/ICCVW.2013.77. URL `https://ieeexplore.ieee.org/document/6755945`. 8, 17

Wu Lin, Mohammad Emtiyaz Khan, and Mark Schmidt. Fast and simple natural-gradient variational inference with mixture of exponential-family approximations. In *International Conference on Machine Learning (ICML)*, 2019. URL `https://proceedings.mlr.press/v97/lin19b.html`. 7, 17

Bingchang Liu, Chaoyu Chen, Cong Liao, Zi Gong, Huan Wang, Zhichao Lei, Ming Liang, Dajun Chen, Min Shen, Hailian Zhou, Hang Yu, and Jianguo Li. Mftcoder: Boosting code llms with multitask fine-tuning. *arXiv preprint arXiv*, 2023. URL `https://arxiv.org/abs/2311.02303`. 1, 2, 3

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. RoBERTa: A robustly optimized BERT pretraining approach, 2019. URL `http://arxiv.org/abs/1907.11692`. arXiv:1907.11692. 9

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations (ICLR)*, 2019. URL `https://openreview.net/forum?id=Bkg6RiCqY7`. 6

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2011. URL `http://www.aclweb.org/anthology/P11-1015`. 9, 18

Pedro Henrique Martins, Patrick Fernandes, João Alves, Nuno M. Guerreiro, Ricardo Rei, Duarte M. Alves, José Pombal, Amin Farajian, Manuel Faysse, Mateusz Klimaszewski, Pierre Colombo, Barry Haddow, José G. C. de Souza, Alexandra Birch, and André F. T. Martins. Eurollm: Multilingual language models for europe, 2024. URL `https://arxiv.org/abs/2409.16235`. 1, 3

Michael S Matena and Colin A Raffel. Merging models with Fisher-weighted averaging. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. URL `https://openreview.net/forum?id=LSKlp_aceOC`. 4

Niklas Muennighoff, Thomas Wang, Lintang Sutawika, Adam Roberts, Stella Biderman, Teven Le Scao, M. Saiful Bari, Sheng Shen, Zheng Xin Yong, Hailey Schoelkopf, Xiangru Tang, Dragomir Radev, Alham Fikri Aji, Khalid Almubarak, Samuel Albanie, Zaid Alyafeai, Albert Webson, Edward Raff, and Colin Raffel. Crosslingual generalization through multitask finetuning. In Anna Rogers, Jordan L. Boyd-Graber, and Naoaki Okazaki (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pp. 15991–16111. Association for Computational Linguistics, 2023. doi: 10.18653/V1/2023.ACL-LONG.891. URL `https://doi.org/10.18653/v1/2023.acl-long.891`. 2

Arthur G. O. Mutambara. *Decentralized estimation and control for multisensor systems*. Routledge, 1998. URL `https://www.routledge.com/Decentralized-Estimation-and-Control-for-Multisensor-Systems/Mutambara/p/book/9780849318658`. 4

Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Baolin Wu, Andrew Y Ng, et al. Reading digits in natural images with unsupervised feature learning. In *NIPS workshop on deep learning and unsupervised feature learning*, volume 2011, pp. 4. Granada, 2011. URL `http://ufldl.stanford.edu/housenumbers/nips2011_housenumbers.pdf`. 8, 17

Manfred Opper and Cédric Archambeau. The variational gaussian approximation revisited. *Neural computation*, 21(3):786–792, 2009. 5

Guillermo Ortiz-Jimenez, Alessandro Favero, and Pascal Frossard. Task arithmetic in the tangent space: Improved editing of pre-trained models. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL `https://openreview.net/forum?id=0A9f2jZDGW`. 4

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 27730–27744. Curran Associates, Inc., 2022. URL `https://proceedings.neurips.cc/paper_files/paper/2022/file/b1efde53be364a73914f58805a001731-Paper-Conference.pdf`. 2

Bo Pang and Lillian Lee. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2005. URL `https://aclanthology.org/P05-1015/`. 9, 18

Xiangyu Qi, Yi Zeng, Tinghao Xie, Pin-Yu Chen, Ruoxi Jia, Prateek Mittal, and Peter Henderson. Fine-tuning aligned language models compromises safety, even when users do not intend to! In *The Twelfth International Conference on Learning Representations*, 2024. URL `https://openreview.net/forum?id=hTEGyKf0dZ`. 3

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PMLR, 2021. URL `https://proceedings.mlr.press/v139/radford21a.html`. 8, 17

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020. URL `http://jmlr.org/papers/v21/20-074.html`. 1

Mengye Ren, Wenyuan Zeng, Bin Yang, and Raquel Urtasun. Learning to reweight examples for robust deep learning. In *International Conference on Machine Learning (ICML)*, 2018. 1

Sebastian Ruder. An overview of multi-task learning in deep neural networks. *CoRR*, abs/1706.05098, 2017. URL `http://arxiv.org/abs/1706.05098`. 1, 3

Yuesong Shen, Nico Daheim, Bai Cong, Peter Nickl, Gian Maria Marconi, Clement Bazan, Rio Yokota, Iryna Gurevych, Daniel Cremers, Mohammad Emtiyaz Khan, and Thomas Möllenhoff. Variational learning is effective for large deep networks. *arXiv preprint arXiv:2402.17641*, 2024. URL `https://arxiv.org/abs/2402.17641`. 5, 6

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2013. URL `https://www.aclweb.org/anthology/D13-1170`. 9, 18

George Stoica, Daniel Bolya, Jakob Bjorner, Pratik Ramesh, Taylor Hearn, and Judy Hoffman. Zipit! merging models from different tasks without training. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL `https://openreview.net/forum?id=LEYUkvdUhq`. 1, 3

Megh Thakkar, Tolga Bolukbasi, Sriram Ganapathy, Shikhar Vashishth, Sarath Chandar, and Partha Talukdar. Self-influence guided data reweighting for language model pre-training. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 2033–2045, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.125. URL `https://aclanthology.org/2023.emnlp-main.125`. 1

Volker Tresp. A Bayesian committee machine. *Neural computation*, 2000. URL `https://direct.mit.edu/neco/article-abstract/12/11/2719/6426/A-Bayesian-Committee-Machine`. 4

Mitchell Wortsman, Gabriel Ilharco, Samir Ya Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, et al. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *International Conference on Machine Learning (ICML)*, 2022. URL `https://proceedings.mlr.press/v162/wortsman22a.html`. 3, 4

Jianxiong Xiao, James Hays, Krista A. Ehinger, Aude Oliva, and Antonio Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 3485–3492, 2010. doi: 10.1109/CVPR.2010.5539970. URL `https://ieeexplore.ieee.org/document/5539970`. 8, 18

Sang Michael Xie, Shibani Santurkar, Tengyu Ma, and Percy S Liang. Data selection for language models via importance resampling. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. URL `https://openreview.net/forum?id=uPSQv0leAu&noteId=3EMr1ZhaRY`. 1

Zhuoyan Xu, Zhenmei Shi, Junyi Wei, Fangzhou Mu, Yin Li, and Yingyu Liang. Towards few-shot adaptation of foundation models via multitask finetuning. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL `https://openreview.net/forum?id=1jbh2e0b2K`. 1

Bobby Yan, Skyler Seto, and Nicholas Apostoloff. Forml: Learning to reweight data for fairness. In *ICML Workshop*, 2022. URL `https://arxiv.org/abs/2202.01719`. 1

Enneng Yang, Zhenyi Wang, Li Shen, Shiwei Liu, Guibing Guo, Xingwei Wang, and Dacheng Tao. Adamerging: Adaptive model merging for multi-task learning. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. 1, 3

T. Zhang. Theoretical analysis of a class of randomized regularization methods. In *Conference on Learning Theory (COLT)*, COLT '99, 1999. URL `https://dl.acm.org/doi/abs/10.1145/307400.307433`. 4

Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level Convolutional Networks for Text Classification. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2015. URL `https://papers.nips.cc/paper_files/paper/2015/hash/250cf8b51c773f3f8dc8b4be867a9a02-Abstract.html`. 9, 18

A DERIVATIONS

A.1 CLOSED-FORM EXPRESSION FOR MAP OF EXPONENTIAL-FAMILY DISTRIBUTION

It is clear that minimizing $\sum_t \alpha_t \widehat{\ell}_t$ is equivalent finding MAP of $\prod_t q_t(\boldsymbol{\theta})^{\alpha_t}$. Let us denote it by $p_{\boldsymbol{\alpha}}$. The mode of $p_{\boldsymbol{\alpha}}$ is available in closed-form because we can always rewrite the posterior in an exponential form that allows us to compute the max. This is shown below where we first rewrite the posterior with a log-partition function $A(\boldsymbol{\theta})$ and an alternate sufficient statistic $\mathbf{t}(\boldsymbol{\lambda}_\alpha)$, and then simply take the derivative to get a closed-form expression for the Pareto solution,

$$p_{\boldsymbol{\alpha}} \propto \exp(\boldsymbol{\theta}^\top \mathbf{t}(\boldsymbol{\lambda}_\alpha) - A(\boldsymbol{\theta})) \implies \boldsymbol{\theta}_{\boldsymbol{\alpha}} = \arg\max_{\boldsymbol{\theta}} \log p_\alpha(\boldsymbol{\theta}) = \nabla A^{-1}\left(\mathbf{t}(\boldsymbol{\lambda}_\alpha)\right). \tag{13}$$

The alternate form is essentially following the form of the conjugate prior (see Bishop (1995, Eq. 2.229)) written to match the form of the likelihood. For instance, in the Gaussian case,

$$p_{\boldsymbol{\alpha}} \propto \exp(\boldsymbol{\theta}^\top \underbrace{\mathbf{H}_\alpha \mathbf{m}_\alpha}_{\mathbf{t}(\boldsymbol{\lambda}_\alpha)} - \underbrace{\tfrac{1}{2}\boldsymbol{\theta}^\top \mathbf{H}_\alpha \boldsymbol{\theta}}_{A(\boldsymbol{\theta})}) \quad \implies \quad \underbrace{\mathbf{H}_\alpha(\boldsymbol{\theta}_{\boldsymbol{\alpha}})}_{\nabla A(\boldsymbol{\theta}_{\boldsymbol{\alpha}})} = \underbrace{\mathbf{H}_\alpha \mathbf{m}_\alpha}_{\mathbf{t}(\boldsymbol{\lambda}_\alpha)} \quad \implies \quad \boldsymbol{\theta}_{\boldsymbol{\alpha}} = \mathbf{m}_\alpha$$

Solutions of variational objectives always have this form (Khan & Rue, 2023, Sec. 5), while convexity of $A(\boldsymbol{\theta})$ ensures that the mode always exists and can be easily found without retraining on individual objectives. In summary, we can always get a closed-form expression as follows

1. Compute $\boldsymbol{\lambda}_t$ of individual objectives.
2. Aggregate them $\boldsymbol{\lambda}_\alpha = \sum_t \alpha_t \boldsymbol{\lambda}_t$.
3. Compute $\boldsymbol{\theta}_{\boldsymbol{\alpha}}$ using Eq. 13.

A.2 CLOSED FORM SOLUTION FOR THE BETA-BERNOULLI MODEL

To illustrate the process, we discuss another example of the Beta-Bernoulli model to model coin-flips $y_t \in \{0, 1\}$ with probability π,

$$p(y_t \,|\, \pi) \propto \pi^{y_t}(1-\pi)^{1-y_t}, \quad p(\pi) \propto \pi^{a_0 - 1}(1-\pi)^{b_0 - 1}.$$

The unknown is then modeled as $\theta = \log(\pi/(1-\pi))$, to get the posterior which is also a Beta distribution with parameters

$$a_t = y_t + a_0, \quad b_t = 1 - y_t + b_0, \quad p(\theta \,|\, \mathcal{D}_t) \propto \pi^{a_t - 1}(1-\pi)^{b_t - 1} \propto \exp(\boldsymbol{\lambda}_t^\top \mathbf{T}(\theta))$$

where $\mathbf{T}(\theta) = (\log\pi, \log(1-\pi))$ and $\boldsymbol{\lambda}_t = (a_t - 1, b_t - 1)$. With this, the aggregate posterior p_α is a Beta distribution natural parameter $\boldsymbol{\lambda}_\alpha = (a_\alpha - 1, b_\alpha - 1)$ where a_α and b_α are simply a weight average of a_t and b_t respectively. To get the maximum of p_α, we write it in an exponential form,

$$p_\alpha(\theta) \propto \exp(\theta \underbrace{a_\alpha - 1}_{t(\boldsymbol{\lambda}_\alpha)} - \underbrace{(a_\alpha + b_\alpha - 2)\log(1 + e^\theta)}_{A(\theta)}) \quad \implies \quad \underbrace{\frac{a_\alpha + b_\alpha - 2}{1 + e^{-\theta_{\text{PO}}}}}_{\nabla A(\boldsymbol{\theta}_{\text{PO}})} = \underbrace{a_\alpha - 1}_{\mathbf{t}(\boldsymbol{\lambda}_\alpha)},$$

from which we get $\pi_{\text{PO}} = (a_\alpha - 1)/(a_\alpha + b_\alpha - 2)$.

A.3 DERIVATION OF THE EM ALGORITHM FOR MIXTURE POSTERIORS

To do so, we use the EM algorithm by viewing the summation over k in Eq. 11 as marginalization over a discrete variable $z_k \in \{1, 2, \dots, K\}$ of the joint $p(\boldsymbol{\theta}, z_t = k) = p_{tk}(\boldsymbol{\theta})$. Then, given parameters $\boldsymbol{\theta}^{(i)}$ at each iteration i, we maximize the EM lower bound. The posterior over z_k is

$$p(z_t = k \,|\, \boldsymbol{\theta}^{(i)}) = \hat{\pi}_{tk}^{(i)} = \frac{p_{tk}(\boldsymbol{\theta}^{(i)})}{\sum_{k'} p_{tk'}(\boldsymbol{\theta}^{(i)})}.$$

Using this, we can write the following lower bound,

$$\sum_{t=1}^{T} \alpha_t \log\left(\sum_{k=1}^{K} \frac{p_{tk}(\boldsymbol{\theta})}{\hat{\pi}_{tk}^{(i)}} \hat{\pi}_{tk}^{(i)}\right) \geq \sum_{t=1}^{T}\sum_{k=1}^{K} \alpha_t \hat{\pi}_{tk}^{(i)} \log p_{tk}(\boldsymbol{\theta}) + \text{c} = \underbrace{\sum_{t=1}^{T}\sum_{k=1}^{K} \alpha_t \hat{\pi}_{tk}^{(i)} \boldsymbol{\lambda}_{tk}^\top}_{=\boldsymbol{\lambda}_{\boldsymbol{\alpha}}^{(i)}} \mathbf{T}(\boldsymbol{\theta}) + \text{c}.$$

The above corresponds to the log of an exponential family (denoted by $p^{(i)}_{\boldsymbol{\alpha}}$) with natural parameter $\boldsymbol{\lambda}^{(i)}_{\boldsymbol{\alpha}}$, which gives us the following iterative procedure:

$$\boldsymbol{\theta}^{(i+1)} \leftarrow \arg\max_{\boldsymbol{\theta}} \sum_{t=1}^{T} \sum_{k=1}^{K} \hat{\pi}^{(i)}_{tk} \alpha_t \boldsymbol{\lambda}_{tk}^\top \mathbf{T}(\boldsymbol{\theta}), \tag{14}$$

where we use the posterior $\hat{\pi}^{(i)}_{tk} = p(z_t = k \,|\, \boldsymbol{\theta}^{(i)}) \propto p_{tk}(\boldsymbol{\theta}^{(i)})$ which is obtained by normalizing over k. The iterates $\boldsymbol{\theta}^{(i)}$ converge to a local maximum which provides a solution for $\widehat{\boldsymbol{\theta}}_{\boldsymbol{\alpha}}$. For $K = 1$, the algorithm reduces to the exponential-family case.

B EXPERIMENTAL SETUP

B.1 ILLUSTRATIVE 2D EXAMPLE

The individual functions in Fig. 2 are of the form $\ell_t(\boldsymbol{\theta}) = \log\left(\sum_{i=1}^{N} \exp(\mathbf{a}_{it}^\top \boldsymbol{\theta} + b_{it})\right)$ where $\mathbf{a}_{it}, b_{it}$ are chosen randomly from normal distributions for $t = 1, 2$ and uniformly for $t = 3$. We approximate the Gibbs distributions $\exp(-\ell_t(\boldsymbol{\theta}))$ using the mixture-of-Gaussian algorithm described in Lin et al. (2019, Section 4.1) with full Hessians, and we use the EM algorithm described in Eq. 12 to find the mode of the mixture.

B.2 MERGING LOGISTIC REGRESSION MODELS

For parameter averaging we train each model using gradient-descent with learning-rate $\rho = 3.0$ for 2500 iterations, and use $\sum_t \alpha_t \boldsymbol{\theta}_t$ to obtain each $\hat{\boldsymbol{\theta}}_{\boldsymbol{\alpha}}$. For the full-Gaussian method, which we use for Hessian-weighted merging, we implement the variational online Newton method described in Khan & Rue (2023, Section 1.3.2). We set the learning-rate $\rho_t = 0.1$, perform 3 Monte-Carlo samples to estimate the expected gradient and Hessian and run for 25 iterations. The parameters of the merged model are obtained via Eq. 10. The mixture of full-Gaussian trains each model by the method described in Lin et al. (2019, Section 4.1) with a 20 component mixture. We set the algorithm's learning-rate of $\beta = 0.02$ for the mean $\boldsymbol{\mu}$ and precision $\boldsymbol{\Sigma}^{-1}$, $\beta = 3 \times 10^{-6}$ for the mixture weights π, while Monte-Carlo samples number of iterations are the same as full-Gaussian. The test-accuracy in Fig. 3 and Fig. 9 is plotted on a grid $\boldsymbol{\Delta}$ with uniform spacing 0.02. The tasks are for imbalanced (T1: $\{0, 1\}$, T2: $\{2, 3, 4\}$ and T3: $\{5, 6, 7, 8, 9\}$); and for balanced: (T1: $\{0, 1, 2\}$, T2: $\{3, 4, 5, 6\}$, T3: $\{7, 8, 9\}\}$).

B.3 MERGING VISION MODELS ON CIFAR-10

We pretrain the ResNet-20 model by running the IVON optimizer for 1000 epochs with 5 Monte-Carlo samples to estimate expected gradients and Hessians and use IVON-HESS for previews. The hyperparmeters of IVON are set as follows: learning-rate $\alpha = 0.1$, momentum $\beta = (0.9, 0.9999)$, weight-decay $\delta = 10^{-3}$ and temperature/sample-size weighting $\lambda = 50000$. The batch-size is set to 50 and the estimated Hessian is initialized to 0.1.

The individual models are also finetuned with IVON, initialized at the pretrained posterior for $\{25, 50, 75, 100, 125, 150\}$ steps over 5 random seeds to obtain a soup of 30 models for each task. This corresponds to the MULTIIVON-HESS method. Each step processes 1000 examples, where the batch-size is set to 50. No weight-decay is used for finetuning, and we use a smaller learning-rate $\alpha = 0.01$. For the batch solution, we finetune on all data for 250 steps with the same hyperparameters.

The merged models $\hat{\boldsymbol{\theta}}_{\boldsymbol{\alpha}}$ are computed using Alg. 1, where we take the models across 5 random seeds with $\{100, 150\}$, $\{75, 100, 125, 150\}$ and $\{25, 50, 75, 100, 125, 150\}$ steps for the mixtures of size 10, 20 and 30. The test-accuracy in Fig. 4 is plotted on a grid $\boldsymbol{\Delta}$ with uniform spacing 0.1.

B.4 MERGING VISION TRANSFORMERS

The pretrained and finetuned checkpoints of ViT-B-32 a model based on CLIP (Radford et al., 2021) on these downstream tasks (RESISC45 (Cheng et al., 2017), GTSRB (Houben et al., 2013), SVHN (Netzer et al., 2011), EuroSAT (Helber et al., 2019), Stanford Cars (Krause et al., 2013) and SUN397 (Xiao et al., 2010)) were obtained based on the code from `https://github.com/mlfoundations/task_vectors`. The squared-gradients approximation for the Hessian-Weighted merge with ADAMW-SG is computed by $\sum_i \nabla \ell_i(\boldsymbol{\theta}_t)^2$, where i is a sum over data examples from the training data.

To generate the exact solution contours we start from the pretrained checkpoint and finetune on the joint datasets with weights obtained by sampling from a grid $\boldsymbol{\Delta}$ with spacing 0.05. The optimizer is AdamW, with learning rate of 10^{-5}, set $(\beta_1, \beta_2) = (0.9, 0.999)$ and decay the learning rate to 0 using a cosine decay with 500 warmup steps. Training is done for 15 epochs on RESISC45,GTSRB and SVHN, while for EuroSAT, Stanford Cars and SUN397 this was set to 35, in both experiments batch size is 128.

B.5 MERGING MASKED LANGUAGE MODELS

We pretrain RoBERTa with 125M parameters using AdamW on the IMDB dataset for sentiment classification (Maas et al., 2011). We use a learning rate of 10^{-5} and set $(\beta_1, \beta_2) = (0.9, 0.999)$ and decay the learning rate to 0 using a cosine decay with 100 warmup steps. Training is done for 2 epochs with a batch size of 16.

We then finetune this model on Rotten Tomatoes (Pang & Lee, 2005), SST-2 (Socher et al., 2013), and Yelp (Zhang et al., 2015), and train with a learning rate of $5 \cdot 10^{-6}$ using a batch size of 16 and for 5, 5, and 2 epochs each. We subsample the data of Yelp by taking the first 20% of the training data to ease computational burden.

We do not use any weight decay in pretraining or finetuning. The squared gradient approximation is calculated by doing one pass over the training data of each model and squaring the single-example gradients for ADAMW-SG.

For the batch solution, we finetune for 3 epochs on the concatenation of the above-mentioned training data after pretraining on IMDB as described above. Again, we use a learning rate of $5 \cdot 10^{-6}$ and a batch size of 16. Evaluation is done by averaging the accuracies over each individual dataset to weigh each dataset the same.

The simplex in Fig. 7 is obtained by sampling from a grid $\boldsymbol{\Delta}$ with spacing 0.05. For the joint solution we use a spacing of 0.1 due to the heavy computational load. The simple merged models are obtained using Eq. 2. For diagonal Gaussians, we use the Hessian-based weighting of Daheim et al. (2024).

B.6 MERGING LLMS FOR MACHINE TRANSLATION

We finetune GEMMA-2B-it (Gemma Team, 2024b) on the IWSLT2017 de-en and fr-en splits (Cettolo et al., 2017). Due to the model size we use LoRA (Hu et al., 2022) to finetune the models which amounts to ca. 0.9M of new trainable parameters. The rest of the network is kept frozen. Accordingly, only the LoRA weights are merged and the base model untouched.

We train the models using IVON with a learning rate of 0.05, $(\beta_1, \beta_2) = (0.9, 0.99995)$, an effective sample size of $1 \cdot 10^7$ for the single-task and $2 \cdot 10^7$ for the multitask model. We clip gradients element-wise to $1 \cdot 10^3$ and to unit norm and use a weight decay of 10^{-6}.

For the Hessian-weighted merging we use IVON-HESS. A comparison to using squared gradients instead is found in App. C.2.

For all experiments, we use a grid with equal spacing of $\alpha_1 \in [0.0, 0.05, \dots, 1.0]$ and always set $\alpha_2 = 1.0 - \alpha_1$.

C ADDITIONAL RESULTS

C.1 BALANCED SETTINGS FOR MULTITASK LEARNING

For the balanced setting there is no skew and the better combinations seem to concentrate around the center of the simplex, which we see in Figure Fig. 9 is captured by all methods, however the more complex posterior approximation allows Hessian-Weighting and Mixture-Weighting to show that

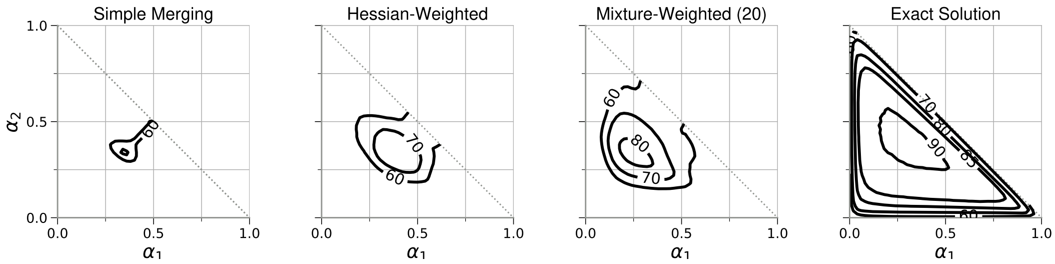
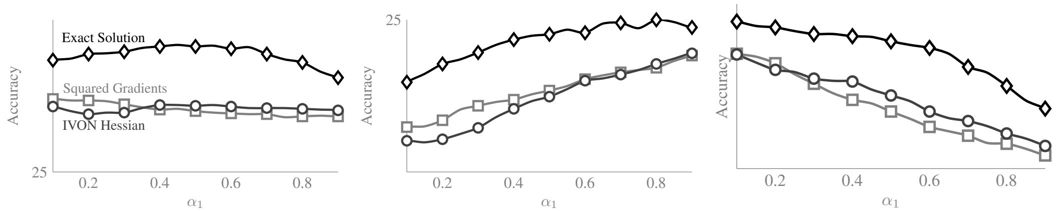


Figure 9: Similar to the previous setting, on balanced tasks improving the posterior approximation also will produce previews that capture more the the exact solution.



(a) DE-EN & FR-EN

(b) DE-EN

(c) FR-EN

Figure 10: Here, we merge LoRA-finetuned GEMMA-2B models trained on IWSLT2017de-en and IWSLT2017fr-en. We show a comparison of using the squared gradient approximation of the (diagonal) Fisher and the diagonal Hessian approximation obtained with IVON for Hessian-weighted merging. Both methods perform similarly and could be used effectively for previews.

multiple combinations even beyond the center are also interesting which Simple Merging fails to convey.

C.2 COMPARISON OF HESSIAN APPROXIMATIONS FOR LLM MERGING

Fig. 10 shows a comparison of using the squared gradient approximation of the (diagonal) Fisher and the diagonal Hessian approximation obtained with IVON for Hessian-weighted merging. Both methods are comparable and provide good previews for multitask finetuning. However, the Hessian approximation from IVON comes for free during training while the squared gradient approximation incurs overhead due to requiring an additional forward pass over at least a subset of the training data after training.