# Upweighting Easy Samples in Fine-Tuning Mitigates Forgetting

**Sunny Sanyal** [*1] **Hayden Prairie** [*1] **Rudrajit Das** [*2] **Ali Kavis** [*1] **Sujay Sanghavi** [1]

## Abstract

Fine-tuning a pre-trained model on a downstream task often degrades its original capabilities, a phenomenon known as "catastrophic forgetting". This is especially an issue when one does not have access to the data and recipe used to develop the pre-trained model. Under this constraint, most existing methods for mitigating forgetting are inapplicable. To address this challenge, we propose a *sample weighting scheme for the fine-tuning data* solely based on the pre-trained model's losses. Specifically, we upweight the easy samples on which the pre-trained model's loss is low and vice versa to limit the drift from the pre-trained model. Our approach is orthogonal and yet complementary to existing methods; while such methods mostly operate on parameter or gradient space, we concentrate on the sample space. We theoretically analyze the impact of fine-tuning with our method in a linear setting, showing that it stalls learning in a certain subspace, which inhibits overfitting to the target task. We empirically demonstrate the efficacy of our method on both language and vision tasks. As an example, when fine-tuning Gemma 2 2B on MetaMathQA, our method results in only a $0.8\%$ drop in accuracy on GSM8K (another math dataset) compared to standard fine-tuning, while preserving $5.4\%$ more accuracy on the pre-training datasets.

## 1. Introduction

In the modern era of large-scale machine learning, one of the central goals is to design models capable of performing multiple tasks. Traditionally, this is achieved by training an appropriately large model over datasets of multiple tasks, ensuring that the model jointly learns multiple tasks at once. Unfortunately, it is not viable to repeat this process with every new additional task due to the scale of contemporary models, necessitating effective strategies that can essentially learn without full retraining. A resource-efficient convention in machine learning is to take a *pre-trained* model which is trained on some vast and diverse dataset, and *fine-tune* it on a new dataset/task. Such pre-trained models are typically large and expensive to train from scratch but perform well on a variety of tasks while offering a versatile basis for learning a new task.

Fine-tuning is a delicate process that should ideally serve multiple objectives simultaneously; we would like to use the base model and its capabilities to facilitate learning a strong model on the downstream task, and in the meantime, preserve the existing abilities of the pre-trained model. On this particular front, the major challenge in standard, unregulated fine-tuning is the *catastrophic forgetting* phenomenon. In broad terms, it describes the performance decline of the pre-trained model on previously observed data/tasks after fine-tuning on a new one. When the learning process for the downstream task interferes with the previously-learned representations beyond tolerable margins, the pre-trained model loses its prior capabilities and significantly underperforms on previously-learned tasks.

Mitigating catastrophic forgetting is an active area of research with many fundamental questions awaiting solutions. The key idea is to constrain the fine-tuning process to prevent the degeneration of the learned representations while guiding the learning of the new task to augment existing capabilities. The literature on the topic offers various approaches based on the available knowledge pertaining to the pre-training process. In fact, pre-training-specific data availability and how it is treated predominantly dictates the success of mitigating forgetting. In many real-life scenarios, however, the data and the training recipe used for generating the pre-trained model are not available (Radford et al., 2021; Touvron et al., 2023a;b; Grattafiori et al., 2024; Jiang et al., 2023). Naturally, one needs to approach the forgetting phenomenon accordingly to design realistic methods.

Therefore, we focus on the case in which we have ***no ac-***

[*]Equal contribution. Rudrajit was a PhD student at the University of Texas at Austin when this work was done. [1]University of Texas at Austin [2]Google Research. Correspondence to: Sunny Sanyal <sanyal.sunny@utexas.edu>, Hayden Prairie <haydenprairie@utexas.edu>, Rudrajit Das <dasrudrajit@google.com>, Ali Kavis <kavis@austin.utexas.edu>, Sujay Sanghavi <sanghavi@mail.utexas.edu>.

*Proceedings of the $42^{nd}$ International Conference on Machine Learning*, Vancouver, Canada. PMLR 267, 2025. Copyright 2025 by the author(s).

***cess** to the pre-training-specific information* during the fine-tuning process; we call it the ***data-oblivious*** setting. The only piece of information available during fine-tuning is indeed the pre-trained model. Therefore, one needs to devise a strategy to regulate and guide the fine-tuning process to preserve the pre-trained model capabilities while learning the new task in the absence of prior knowledge. Under this challenging setting, we present an answer to the question:

*Can we design a principled method that mitigates forgetting during fine-tuning in the data-oblivious setting?*

In this paper, we propose **F**ine-tuning with Pre-trained **L**oss-**O**riented **W**eighting (**FLOW**) to mitigate catastrophic forgetting in the data-oblivious setting. Our key insight is upweighting the "easy" samples on which the pre-trained model's loss is low and vice versa. We believe that boosting the samples on which the pre-trained model performs well (i.e., has low loss) will introduce supervised bias to the gradient updates in favor of the pre-trained model. Intuitively, this will prevent the parameters from deviating too much from the initial pre-trained state, thus mitigating forgetting.

Some prior papers assign more importance to samples with *larger losses* to accelerate the training process (Loshchilov & Hutter, 2015; Shrivastava et al., 2016; Katharopoulos & Fleuret, 2017; Kawaguchi & Lu, 2020; Das et al., 2024). We follow the reciprocal reasoning; we tweak the fine-tuning process in favor of the pre-trained model by assigning *larger* weights to samples with *smaller* pre-trained loss values. We elaborate on this while stating our **contributions** next.

1. To mitigate forgetting, we propose **FLOW**, which fine-tunes the pre-trained model using a sample-wise weighted loss. Inspired by robust optimization ideas, we derive the $i^{\text{th}}$ sample's weight to be $\exp(-\ell_i/\tau)$, where $\ell_i$ is the $i^{\text{th}}$ sample's pre-trained loss and $\tau$ is a parameter which we set as median$(\ell_i)$ in practice. Thus, our method is essentially *parameter-free*.
2. We demonstrate the superiority of FLOW over relevant baselines (model averaging, $\ell_2$ regularization, LoRA, etc.) in both vision and language model experiments. For instance, ResNet-50 fine-tuned with FLOW on six image classification datasets achieves $\sim \mathbf{17}\%$ higher average accuracy (over pre-training and fine-tuning data) than standard fine-tuning, while also surpassing other relevant baselines (see Table 1). When fine-tuning Gemma 2 2B on math datasets, the corresponding improvement of FLOW over standard fine-tuning is $\sim \mathbf{4}\%$ (see Table 2).
3. We also empirically show that combining FLOW with existing methods for mitigating forgetting *improves the performance* of the base methods (see Tables 3 and 4).
4. We theoretically analyze the effect of fine-tuning with FLOW for linear models. In particular, the covariance matrix of the fine-tuning data weighted by FLOW has a small eigenvalue and *training is stalled* along the corresponding eigenvector, *impeding overfitting to the fine-tuning task* (see Remark 7.4).

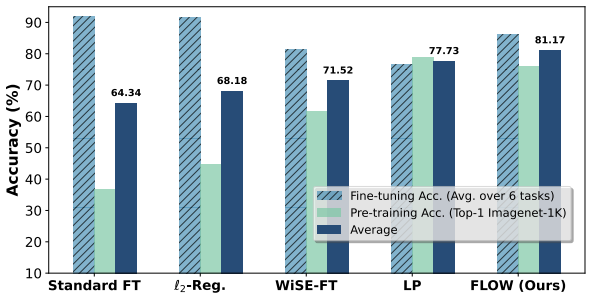


*Figure 1.* FLOW versus standard fine-tuning (FT) and relevant baselines for a ResNet-50 model pre-trained on ImageNet-1K (from Table 1). **FLOW achieves the best average accuracy** (between pre-training and target fine-tuning accuracies).

We end this section with a preview of the comparison of our method FLOW with some relevant baselines (in the data-oblivious setting) in Figure 1.

## 2. Related Work

### 2.1. Mitigating Catastrophic Forgetting

We begin by summarizing the vast literature on catastrophic forgetting with a focus on prior works most relevant to our proposed setting. For a streamlined presentation, we survey prior work in two settings – data-aware and data-oblivious. Due to space limitations, we refer the readers to Appendix A for a more detailed and explanatory review of the literature.

#### 2.1.1. Data-aware approaches

The majority of the approaches for mitigating forgetting assume task-specific knowledge access to different extents; either (a subset of) the pre-training dataset itself or some information/statistic computed from pre-training data. Below, we describe the data-aware approaches based on how they make use of task-specific knowledge.

**Regularization-based methods.** This line of work aims to preserve existing capabilities by keeping the parameters close to the pre-trained model. The key idea is to introduce task-specific regularization to penalize modifications along the "important" directions for the old tasks (Ahn et al., 2019). Kirkpatrick et al. (2016) introduces the elastic weight consolidation (EWC) algorithm, which estimates the important directions by approximating the Fisher information matrix. Several variants of EWC have been proposed (Schwarz et al., 2018; Ritter et al., 2018; Lee et al., 2020; Liu et al., 2018).

Zenke et al. (2017); Aljundi et al. (2018) infer the importance of each parameter by their variational effect on the outputs. In a similar spirit, Lee et al. (2017) aims to match the posteriors of the pre-trained and fine-tuned models.

**Optimization-driven methods.** Another perspective to mitigating forgetting is guiding the optimization process by constraining the algorithms directly as opposed to manipulating the loss function. The core idea is to keep track of "important directions" for the old tasks, and train on the new task "orthogonally." This could be done by storing prior data samples or gradients in a buffer (Lopez-Paz & Ranzato, 2017; Farajtabar et al., 2020; Chaudhry et al., 2019a) or by incrementally expanding the subspace of important directions without storing task-specific information (Zeng et al., 2019; Wang et al., 2021; 2023b).

**Replay-based methods.** A more direct approach is to store old task samples in buffers and introduce them into the training process for the new task to refresh task-specific representations periodically. There are several components to such methods. Some prior work focus on *data selection* based on the nature of old data access (Rebuffi et al., 2017; Aljundi et al., 2019; Bang et al., 2021; Chaudhry et al., 2019b; Isele & Cosgun, 2018; De Lange & Tuytelaars, 2021; Borsos et al., 2020; Tiwari et al., 2021) (e.g., streaming versus on-demand). Another important perspective is the *re-introduction strategy* of the stored information into the fine-tuning process (Silver & Mercer, 2002; Li & Hoiem, 2016; Triki et al., 2017; Lee et al., 2019b; Dhar et al., 2019; Rebuffi et al., 2017; Riemer et al., 2019; Chaudhry et al., 2019b; De Lange & Tuytelaars, 2021; Tiwari et al., 2021).

**Architecture-driven methods.** Another technique to limit interference between tasks is to allocate a separate trainable set of parameters per task. This could be done by initializing a sub-networks per new task (Rusu et al., 2016; Aljundi et al., 2017; Collier et al., 2020; Rajasegaran et al., 2019; Ramesh & Chaudhari, 2021; Wang et al., 2023a; 2022a), gradually expanding the parameters of a base network (Yoon et al., 2018; Ostapenko et al., 2019; Hung et al., 2019), or segregating a fixed model into task-specific subsets (Mallya et al., 2018; Kang et al., 2022; Serra et al., 2018; Wortsman et al., 2020; Mallya & Lazebnik, 2017; Mustafa B Gurbuz, 2022; Jung et al., 2020). The main downside with this line of work is that task identities must be known for inference to (de)activate relevant sub-networks (Aljundi et al., 2017).

#### 2.1.2. Data-oblivious approaches

In the less-explored data-oblivious setting, it is particularly challenging to devise a principled approach, as there is no access to any data-specific information, except for the pre-trained model. One line of work explores the simple idea of "model averaging" (MA) which essentially does a convex combination of the parameters of the pre-trained model and that of the fully fine-tuned model for the new task. MA and more sophisticated model merging variants have been studied in relevant context to forgetting (Lubana et al., 2021; Wortsman et al., 2021; Ilharco et al., 2023; Lin et al., 2023; Kleiman et al., 2025). Some recent works Chen et al. (2024b); Panda et al. (2024) introduce different strategies to selectively update a subset of parameters in a pre-training data-agnostic manner. Finally, Biderman et al. (2024) has shown that LoRA (Hu et al., 2022) could be effective for mitigating catastrophic forgetting in transformers. Unlike the methods discussed above which focus on the parameter or gradient space, ours focuses on the sample space.

### 2.2. Sample Selection and Weighting

Sample-wise importance selection/weighting has been studied in optimization papers (Needell et al., 2014; Zhao & Zhang, 2015; Alain et al., 2015; Stich et al., 2017) and ML papers (Loshchilov & Hutter, 2015; Shrivastava et al., 2016; Katharopoulos & Fleuret, 2017; 2018; Kawaguchi & Lu, 2020; Das et al., 2024) to speed up the optimization/training process by reducing the variance of the gradient updates. Such papers advocate focusing on "hard" samples with high-gradient norms or losses. In contrast, we focus on "easy" samples to mitigate forgetting. Another line of work focuses on robust learning under uncertain data distributions. Distributionally robust optimization (DRO) proposes to minimize the worst-case weighted loss, where the sample weights are constrained or regularized (Ben-Tal et al., 2013; Levy et al., 2020; Duchi & Namkoong, 2021; Qi et al., 2021). Some recent works (Xie et al., 2024; Chen et al., 2024a; Anonymous, 2025) propose dynamic sample-weighting strategies for LLM training based on the previously discussed ideas.

## 3. Notation and Definitions

$\mathbb{1}(.)$ denotes the indicator variable. For any $n \in \mathbb{N}$, the set $\{1, \dots, n\}$ is denoted by $[n]$. Vectors and matrices are in lowercase and uppercase bold font, respectively. The $\ell_p$ norm of a vector $\mathbf{v}$ is denoted by $\|\mathbf{v}\|_p$. The inner product between two vectors $\mathbf{v}$ and $\mathbf{v}'$ is denoted as $\langle \mathbf{v}, \mathbf{v}' \rangle$. A set of $n$ linearly independent $n$-dimensional vectors $\{\mathbf{u}_1, \dots, \mathbf{u}_n\}$ is said to be an orthonormal basis for $\mathbb{R}^n$ if $\langle \mathbf{u}_i, \mathbf{u}_j \rangle = \mathbb{1}(i = j)$. A vector $\mathbf{v} = [\mathrm{v}_1, \dots, \mathrm{v}_n]^\top$ is said to belong to the $n$-dimensional probability simplex $\mathbf{\Delta}_n$ if $\sum_{i=1}^n \mathrm{v}_i = 1$ and $\mathrm{v}_i \geq 0 \; \forall \, i \in [n]$. For any $n \in \mathbb{N}$, $\mathbf{I}_n$ denotes the identity matrix of dimension $n$.

## 4. Proposed Algorithm

Our proposed algorithm consists of two main steps: *(i)* computing weights for the samples based on their respective pre-trained loss values; and *(ii)* fine-tuning with a weighted loss wherein the per-sample losses are scaled by their respec-

tive weights. The sample-wise weights are computed once and used throughout the entire fine-tuning process. We formally state our proposed fine-tuning protocol in Algorithm 1 and delve into its design details in the sequel.

**Algorithm 1** **F**ine-tuning with Pre-trained **L**oss-**O**riented **W**eighting (**FLOW**)

**Input:** Pre-trained model $\boldsymbol{\theta}^*$, dataset $\{(\mathbf{x}_i, \mathrm{y}_i)\}_{i=1}^n$ for the new task, and temperature parameter $\tau$.

$f_i(\boldsymbol{\theta}) \to i^{\text{th}}$ sample's loss at $\boldsymbol{\theta}$, with a non-negative loss function (e.g., cross-entropy loss).

1. Compute sample weights: $w_i = \exp\left(-\frac{f_i(\boldsymbol{\theta}^*)}{\tau}\right)$.
2. Weighted loss: $\mathcal{L}(\boldsymbol{\theta}) = \sum_{i=1}^n w_i f_i(\boldsymbol{\theta})$.
3. Fine-tune with weighted loss: $\widehat{\boldsymbol{\theta}}^* := \arg\min_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta})$.

**Output:** Fine-tuned model $\widehat{\boldsymbol{\theta}}^*$.

**Remark 4.1.** *Depending on the setting, our model might have task-specific components, such as per-task prediction heads (e.g., in vision). Algorithm 1 can be slightly modified in the presence of task-specific components to enhance performance. Refer to Appendix B for these modifications.*

**Remark 4.2.** *As a heuristic prescription, we set* $\tau = \text{median}\,(f_i(\boldsymbol{\theta}^*))$ *in all our experiments (unless otherwise stated), which leads to consistently good performance.* ***Thus, our algorithm is essentially parameter-free in practice****.*

**Algorithm design.** Our main intuition is that we can control forgetting by not drifting away too much from the pre-trained model (i.e., $\boldsymbol{\theta}^*$) during fine-tuning. In the presence of pre-training data, this is done by introducing data-dependent constraints on the parameter space or gradient space. Since we have *no access* to pre-training data, we redirect our focus towards *strategies on the* ***sample space*** *depending only on the* ***pre-trained model****.*

To that end, we propose to infer the easiness of each sample of the fine-tuning dataset with respect to the pre-trained model, based on the per-sample losses $f_i(\boldsymbol{\theta}^*)$'s (see Alg. 1). We say that the $i^{\text{th}}$ sample is "easy" if $f_i(\boldsymbol{\theta}^*)$ is "small".[1] Intuitively, prioritizing the "easy" samples during fine-tuning would limit the drift from $\boldsymbol{\theta}^*$. On the other hand, over-focusing on the "easy" samples would probably lead to poor performance on the fine-tuning task. Thus, it is important to strike a balance.

Let us formalize these ideas mathematically. For fine-tuning on the new task, let us consider the objective function $\mathcal{L}_{\boldsymbol{\pi}}(\boldsymbol{\theta}) = \sum_{i=1}^n \pi_i f_i(\boldsymbol{\theta})$, where $\boldsymbol{\pi} = [\pi_1, \ldots, \pi_n]^\top$ is a **static** design-choice $\in \boldsymbol{\Delta}_n$ (i.e., $\sum_{i=1}^n \pi_i = 1$ and $\pi_i \geq 0$ $\forall\, i \in [n]$) which we allow to only depend on the pre-trained model's losses $\{f_i(\boldsymbol{\theta}^*)\}_{i=1}^n$ (and *not* the current model's losses $\{f_i(\boldsymbol{\theta})\}_{i=1}^n$). We would like to design $\boldsymbol{\pi}$ so that:

1. for all $i \neq j$ such that $f_i(\boldsymbol{\theta}^*) \leq f_j(\boldsymbol{\theta}^*)$, $\pi_i \geq \pi_j$,
2. $\boldsymbol{\pi}$ does not concentrate around one or a few samples but rather spreads uniformly over the samples.

These two requirements can be enforced by minimizing the following function (w.r.t. $\boldsymbol{\pi}$) involving negative *entropic regularization*:

$$g(\boldsymbol{\pi}) = \sum_{i=1}^n \pi_i f_i(\boldsymbol{\theta}^*) + \tau \sum_{i=1}^n \pi_i \log \pi_i. \tag{1}$$

Here $\tau > 0$ is a parameter controlling the extent of the second requirement which is facilitated by the entropy term. We now state the minimizer of $g(\boldsymbol{\pi})$ (proof is in Appendix C).

**Proposition 4.3.** *Let* $\boldsymbol{\pi}^* = [\pi_1^*, \ldots, \pi_n^*]^\top = \arg\min_{\boldsymbol{\pi} \in \boldsymbol{\Delta}_n} g(\boldsymbol{\pi})$. *Then we have* $\pi_i^* = \frac{1}{Z} \exp\left(-\frac{f_i(\boldsymbol{\theta}^*)}{\tau}\right)$, *where* $Z$ *is the normalizing factor.*

Modulo the normalizing factor $Z$ (it does not matter when optimizing w.r.t. $\boldsymbol{\theta}$), note that $w_i$ and $\mathcal{L}(\boldsymbol{\theta})$ in Algorithm 1 are equivalent to $\pi_i^*$ and $\mathcal{L}_{\boldsymbol{\pi}^*}(\boldsymbol{\theta})$, respectively.

**Distributionally robust optimization (DRO) perspective.** Our formulation above is motivated by prior work on DRO (Qi et al., 2021), but it is **exactly the opposite** of DRO in spirit. Specifically, in our setting, Qi et al. (2021) consider the following min-max problem:

$$\min_{\boldsymbol{\theta}} \max_{\boldsymbol{\pi} \in \boldsymbol{\Delta}_n} \sum_{i=1}^n \pi_i f_i(\boldsymbol{\theta}) - \tau \sum_{i=1}^n \pi_i \log \pi_i. \tag{2}$$

The first term in Eq. (2) is the *worst-case* weighted loss at $\boldsymbol{\theta}$, while the second term (i.e., entropic regularization) promotes uniform weights. The optimal solution to the inner max function w.r.t. $\boldsymbol{\pi}$ turns out to be $\pi_i^* \propto \exp\left(\frac{f_i(\boldsymbol{\theta})}{\tau}\right)$. Note that this is essentially the *inverse of our weighting function* (modulo the normalizing factor) because it assigns a higher weight to samples with larger losses (i.e., the "hard" samples). The weighting function of DRO would be very conducive to forgetting because it focuses more on the "hard" samples. Further, our weighting function is static (or one-shot) as it depends only on the losses at $\boldsymbol{\theta}^*$. On the other hand, the weighting function of DRO is dynamic (i.e., it depends on the current point $\boldsymbol{\theta}$). In fact, after plugging in the optimal value of $\pi$ into Eq. (2) and simplifying, the DRO objective reduces to $\min_{\boldsymbol{\theta}} \sum_{i=1}^n \exp\left(\frac{f_i(\boldsymbol{\theta})}{\tau}\right)$; this is noticeably different from our objective $\mathcal{L}(\boldsymbol{\theta})$ in Algorithm 1.

[1] This is not a formal definition and so "small" is not quantified.

## 5. Experimental Setup

We empirically evaluate the performance of FLOW (Algorithm 1) on vision and language tasks, showcasing its effectiveness across different model architectures and modalities.[2] Here, we explain details of our experiments: baselines, model architectures, datasets, and evaluation metrics.

**Baselines.** In our language and vision experiments, we compare FLOW against relevant baselines in the *data-oblivious setting*, namely, standard fine-tuning (fine-tuning with vanilla unweighted loss), $\ell_2$-regularization [following Kirkpatrick et al. (2016)], and WiSE-FT (Wortsman et al., 2021) (model averaging of pre-trained and standard fine-tuned models). Additionally, we compare against linear probing (fine-tuning only the classification head, keeping the body frozen) and low-rank adaptation (LoRA) (Hu et al., 2022) in language experiments. More details on the baselines can be found in Appendix G.1.

### 5.1. Vision Experiments

We compare the performance of FLOW and associated baselines in a transfer learning setup.

**Models.** We experimented with ResNet-18 and ResNet-50 (Wightman et al., workshop) pre-trained on Imagenet-1K (IN-1K).

**Datasets.** We used seven widely-used image classification datasets: CIFAR-10 (Krizhevsky, 2009), CIFAR-100 (Krizhevsky, 2009), Flowers102 (Nilsback & Zisserman, 2008), Caltech101 (Li et al., 2022), Cars (Krause et al., 2013), and Dogs (Parkhi et al., 2012).

**Evaluation metrics.** Vision models are trained with task-specific parts, such as classification head (head) and batch-norm (BN); see Appendix B for how FLOW works with with task-specific parts. Forgetting is measured by how much the model's top-1 validation accuracy on ImageNet-1K (subsequently referred to as IN-1K accuracy) reduces after fine-tuning. We report the fine-tuning performance in terms of *average fine-tuning accuracy* over all the related datasets following (Goyal et al., 2023; Ilharco et al., 2023). For IN-1K evaluation *after* fine-tuning, we replace the task-specific components of the fine-tuned model with their pre-trained counterparts. An extended discussion on experimental details, evaluation, and hyper-parameters are in Appendix G.4. We also report the **average** of IN-1K accuracy and averaged fine-tuning accuracy for each method; this is a reasonable *unified metric* to evaluate the performance of a method jointly on the pre-training and fine-tuning data.

[2]Our code is publicly available `here`.

### 5.2. Language Model Experiments

We follow a similar setup to Biderman et al. (2024); Chen et al. (2024b), where a language model's general capabilities are evaluated before and after fine-tuning on a mathematical reasoning dataset. All training for language experiments is done with HuggingFace `peft` (Mangrulkar et al., 2022), `transformers` (Wolf et al., 2020), `datasets` (Lhoest et al., 2021), and `accelerate` (Gugger et al., 2022).

**Models.** We use Gemma 2 2B (Team et al., 2024) and Llama 3.2 3B (Grattafiori et al., 2024) as our base language models. Further details on training hyper-parameters can be found in Appendix G.2.

**Datasets.** Following previous work (Biderman et al., 2024; Chen et al., 2024b), we fine-tune on MetaMathQA (Yu et al., 2023), a mathematical reasoning dataset that is bootstrapped from the training set of GSM8K (Cobbe et al., 2021) and MATH (Hendrycks et al., 2021b) using a LLM. We train with all 395K samples in MetaMathQA.

**Evaluation metrics.** To evaluate the validity of FLOW, we break down our metrics into *general capability* and *target fine-tuning* evaluations. To evaluate general capabilities, we again follow a similar setup to Chen et al. (2024b), where we use commonsense reasoning, 5-shot MMLU (Hendrycks et al., 2021a), and 3-shot MBPP (Austin et al., 2021) metrics. To evaluate the target domain, we use 5-shot GSM8K (Cobbe et al., 2021). All evaluations are performed with `lm-evaluation-harness` (Gao et al., 2024). More details on evaluation and the commonsense metric can be found in Appendix G.3. Similar to vision, we also report the **average** of general capabilities and the target fine-tuning accuracies as a *unified metric*.

## 6. Experimental Results

### 6.1. Comparing FLOW and Related Baselines

For our **vision** experiments on ResNets, Table 1 lists the accuracies of all the baselines and FLOW. We observe similar trends for both ResNet models, so we discuss the results for the larger ResNet-50 model here. The pre-trained ResNet-50 model achieves a top-1 accuracy of 79.02% on ImageNet-1K's validation set. Standard fine-tuning experiences a significant 42.11% drop in IN-1K accuracy, while achieving an average fine-tuning accuracy of 91.78% across the target datasets. In contrast, FLOW suffers only a 2.93% drop in IN-1K accuracy and exhibits a reasonable 86.25% average accuracy on target fine-tuning datasets, demonstrating a significant improvement over standard fine-tuning. Overall, FLOW's **average** on IN-1K and target domain accuracy is **16.83% higher** than standard fine-tuning.

Going beyond standard fine-tuning, our results in Table 1 show that FLOW comprehensively outperforms other base-

*Table 1.* **Performance of FLOW with ResNet vision models. Bolded** and underlined values indicate the **best** and second-best accuracies within each column (and for each model). Deltas (in color) for IN-1K and target performance are computed w.r.t. the pre-trained and standard fine-tuned models. **FLOW attains the best average accuracy** and is better than the second-best method (linear probing) by 2.94% and 3.44% for ResNet-18 and ResNet-50, respectively.

| | Method | IN-1K Acc. | Target Acc. | Average |
|---|---|---|---|---|
| **ResNet-18** | Pre-trained | **69.76** (+0.00) | – | – |
| | Standard FT | 19.58 (-50.18) | **89.07** (+0.00) | 54.60 |
| | Linear Probe | **69.76** (+0.00) | 73.57 (-15.50) | 71.63 |
| | $\ell_2$-Reg. | 34.78 (-34.98) | 88.12 (-0.95) | 61.45 |
| | WiSE-FT | 54.15 (-15.61) | 80.23 (-8.84) | 67.19 |
| | **FLOW** (Ours) | 65.21 (-4.55) | 83.93 (-5.14) | **74.57** |
| **ResNet-50** | Pre-trained | **79.02** (+0.00) | – | – |
| | Standard FT | 36.91 (-42.11) | **91.78** (+0.00) | 64.34 |
| | Linear Probe | **79.02** (+0.00) | 76.45 (-15.33) | 77.73 |
| | $\ell_2$-Reg. | 44.78 (-34.24) | 91.58 (-0.20) | 68.18 |
| | WiSE-FT | 61.65 (-17.37) | 81.38 (-10.40) | 71.52 |
| | **FLOW** (Ours) | 76.09 (-2.93) | 86.25 (-5.53) | **81.17** |

lines. Interestingly, despite its simplicity, linear probing is the second-best method. Overall, FLOW outperforms other baselines, when **averaging** IN-1K and target fine-tuning accuracies, by **a 3.44% advantage** over the closest competitor, linear probing. Although linear probing completely prevents forgetting, it learns significantly less during fine-tuning compared to FLOW. The accuracies on individual fine-tuning datasets and corresponding accuracies for IN-1K can be found in Appendix H.

Additionally, in Appendix H.1, we compare FLOW with a distillation-based approach for mitigating forgetting called "learning without forgetting" (LwF) (Li & Hoiem, 2016) on ViT-B/16, which is a larger model than ResNets and on the Food101 (Bossard et al., 2014) dataset, which is relatively larger than each of the six datasets here. In summary, FLOW outperforms LwF in terms of average accuracy; while LwF has the edge on fine-tuning performance, FLOW does better on the forgetting front.

Our **language model** results are in Table 2. Results for Gemma 2 2B show that FLOW helps preserve (and even somewhat enhance) the general capabilities of the pre-trained model. Specifically, compared to standard fine-tuning, FLOW improves general capability accuracy by 2.52% in commonsense reasoning, 3.73% in MMLU, and 10.00% in MBPP, with a minor degradation of 0.83% in GSM8K. We see a similar trend in our Llama 3.2 3B experiments. Furthermore, while alternative baselines show specific strengths (such as WiSE-FT's general capability performance and $\ell_2$-regularization's target fine-tuning performance), **FLOW outperforms all baselines, on average,** for both models, striking the best balance between preserving general capabilities and achieving good target fine-tuning performance. Additional details on commonsense reasoning results are in Appendix I.1 and an ablation for our choice of sample weighting in LLMs is in Appendix I.2.

In summary, *FLOW strikes a good balance between learning a new task and retaining knowledge from pre-training.*

### 6.2. Combining FLOW with Baselines

To complement our results in Tables 1 and 2, we investigate the performance of baselines when **combined with FLOW**. In the vision setting, we consider uniform model averaging with WiSE-FT (with $\alpha = 0.5$) and report its performance with and without FLOW in Table 3. Interestingly, averaging the pre-trained IN-1K model and the fine-tuned model obtained with FLOW **improves** over standard WiSE-FT (i.e., averaging the pre-trained IN-1K model and the standard fine-tuned model) by **4.18**% and **4.52**% for ResNet-18 and ResNet-50, respectively, in average performance.

Further, as seen in Table 4, FLOW **boosts the performance** of other baselines in language modeling. When combined with $\ell_2$-regularization, we observe improvements in general capability between $0.5\%$ and $1.80\%$, with only a $0.83\%$ reduction in GSM8K performance. Furthermore, the integration of FLOW with LoRA yields even stronger results, enhancing general capability performance by $1.07\%$ to $3.40\%$, while simultaneously improving GSM8K performance by $1.06\%$. Further details and discussion combining FLOW with $\ell_2$-regularization and LoRA are in Appendix I.1.

## 7. Theoretical Analysis

Here we consider linear pre-training and fine-tuning tasks[3] and theoretically analyze the effect of fine-tuning with our proposed method FLOW (Alg. 1). Specifically, we compare the non-asymptotic trajectories of FLOW and vanilla fine-tuning. A key insight of our analysis is that FLOW stalls training in a certain direction, impeding overfitting to the fine-tuning task (see Remark 7.4). We also demonstrate that FLOW goes beyond the simple idea of model averaging (see Remark 7.5).

We begin by describing the problem setting.

**Pre-training task:** The label $\mathrm{y} \in \mathbb{R}$ for a $d$-dimensional data point $\mathbf{x} \sim \mathcal{P}$ is given by $\mathrm{y} = \langle \boldsymbol{\theta}_*, \mathbf{x} \rangle$, where $\boldsymbol{\theta}_* \in \mathbb{R}^d$ is the ground-truth model. Let $\mathcal{D}$ denote the joint distribution of $(\mathbf{x}, \mathrm{y})$, where $\mathbf{x} \sim \mathcal{P}$. Let $\boldsymbol{\Sigma} = \mathbb{E}_{\mathbf{x}\sim\mathcal{P}}\left[\mathbf{x}\mathbf{x}^\top\right]$ be the data covariance matrix. Without loss of generality, let $\boldsymbol{\Sigma} \succeq \mathbf{I}_d$.

**Fine-tuning task:** The label $\widetilde{\mathrm{y}} \in \mathbb{R}$ for a $d$-dimensional data point $\widetilde{\mathbf{x}} \sim \widetilde{\mathcal{P}}$ is given by $\widetilde{\mathrm{y}} = \langle \widetilde{\boldsymbol{\theta}}_*, \widetilde{\mathbf{x}} \rangle$, where $\widetilde{\boldsymbol{\theta}}_* \in \mathbb{R}^d$ is the

[3] Our insights carry over to neural networks following the dynamics of linear models under gradient descent (Lee et al., 2019a).

*Table 2.* **Performance of FLOW with LLMs.** After fine-tuning Gemma 2 2B and Llama 3.2 3B on MetaMathQA, we compare the target fine-tuning performance (GSM8K) with general capability performance. **Bolded** and underlined values indicate the **best** and second-best results within each column (and for each model). Deltas (in color) for general capability metrics and fine-tuning metrics are computed w.r.t. the pre-trained and standard fine-tuned model's accuracy, respectively. We see that **FLOW, on average, has the best performance** on general capabilities and target domain, achieving within $\sim 0.8\%$ (Gemma 2 2B) and $\sim 1.4\%$ (Llama 3.2 3B) of standard fine-tuning's target performance, while significantly mitigating the degradation of general pre-training capabilities in comparison to other baselines.

| | Method | General Capability Acc. | | | Target Acc. | Average |
|---|---|---|---|---|---|---|
| | | Commonsense | MMLU | MBPP | GSM8K | |
| **Gemma 2 2B** | Pre-trained | 57.23 (+0.00) | 49.59 (+0.00) | **28.40** (+0.00) | 24.49 (-38.89) | 40.79 |
| | Standard Fine-tuning | 55.07 (-2.16) | 45.59 (-4.00) | 16.80 (-11.60) | **63.38** (+0.00) | 46.31 |
| | WiSE-FT ($\alpha = 0.5$) | 57.28 (+0.05) | **50.13** (+0.54) | 25.60 (-2.80) | 53.30 (-10.08) | 47.60 |
| | LoRA ($r = 64$) | 55.67 (-1.56) | 44.28 (-5.31) | 25.80 (-2.60) | 60.43 (-2.95) | 47.05 |
| | $\ell_2$-Regularization | 57.01 (-0.22) | 48.43 (-1.16) | 24.80 (-3.60) | 62.85 (-0.53) | 49.19 |
| | **FLOW** (Ours) | **57.59** (+0.36) | 49.31 (-0.28) | 26.80 (-1.60) | 62.55 (-0.83) | **49.98** |
| **Llama 3.2 3B** | Pre-trained | 54.48 (+0.00) | **54.34** (+0.00) | **38.00** (+0.00) | 26.01 (-40.94) | 44.28 |
| | Standard Fine-tuning | 50.68 (-3.80) | 45.29 (-9.05) | 17.80 (-20.20) | **66.95** (+0.00) | 46.10 |
| | WiSE-FT ($\alpha = 0.5$) | **54.54** (+0.04) | 53.33 (-1.01) | 34.60 (-3.40) | 57.01 (-9.94) | 50.75 |
| | LoRA ($r = 64$) | 53.10 (-1.38) | 50.95 (-3.39) | 34.00 (-4.00) | 63.84 (-3.15) | 51.66 |
| | $\ell_2$-Regularization | 53.60 (-0.88) | 51.28 (-3.06) | 33.60 (-4.40) | 66.87 (-0.08) | 52.30 |
| | **FLOW** (Ours) | 54.30 (-0.18) | 51.86 (-2.48) | 36.00 (-2.00) | 65.58 (-1.37) | **52.87** |

*Table 3.* **WiSE-FT with FLOW vs. WiSE-FT in vision.** "WiSE-FT+" denotes WiSE-FT with FLOW in the table. Comparison here is in the same setting as Table 1. Note that **WiSE-FT+ is significantly better than WiSE-FT**.

| | Method | IN-1K | Target | Average |
|---|---|---|---|---|
| **ResNet-18** | WiSE-FT | 54.15 | 80.23 | 67.19 |
| | WiSE-FT+ | 68.71 | 74.03 | **71.37** |
| **ResNet-50** | WiSE-FT | 61.65 | 81.38 | 71.52 |
| | WiSE-FT+ | 78.29 | 73.80 | **76.04** |

*Table 4.* **$\ell_2$-Reg./LoRA + FLOW vs. $\ell_2$-Reg./LoRA in language.** "$\ell_2$+" and "LoRA+" denote $\ell_2$-Reg. with FLOW and LoRA with FLOW, respectively. The results below are for Gemma 2 2B in the same setup as Table 2. We let A1, A2, A3, and B1 represent Commonsense, MMLU, MBPP, and GSM8K, respectively. We see that **$\ell_2$+ and LoRA+ are better than $\ell_2$ and LoRA**, respectively.

| Method | A1 | A2 | A3 | B1 | Avg. |
|---|---|---|---|---|---|
| $\ell_2$ | 57.01 | 48.43 | 24.80 | **62.85** | 49.19 |
| $\ell_2$+ | **57.53** | **49.38** | **26.60** | 62.02 | **49.79** |
| LoRA | 55.67 | 44.28 | 25.80 | 60.43 | 47.05 |
| LoRA+ | **56.74** | **47.68** | **28.80** | **61.49** | **49.31** |

ground-truth model. Let $\widetilde{\mathcal{D}}$ denote the joint distribution of $(\widetilde{\mathrm{x}}, \widetilde{\mathrm{y}})$, where $\widetilde{\mathrm{x}} \sim \widetilde{\mathcal{P}}$. Also, let

$$\mathbf{e} := \boldsymbol{\theta}_* - \widetilde{\boldsymbol{\theta}}_*, \overline{\mathbf{e}} := \mathbf{e}/\|\mathbf{e}\|_2,$$

and $\overline{\mathbf{e}}_\perp$ be a unit vector orthogonal to $\overline{\mathbf{e}}$. We consider the case of $\widetilde{\mathcal{P}} = \mathcal{N}(\vec{\mathbf{0}}_d, \widetilde{\boldsymbol{\Sigma}})$, where

$$\widetilde{\boldsymbol{\Sigma}} = \mathbf{I}_d + \rho\big(\overline{\mathbf{e}}\overline{\mathbf{e}}_\perp^\top + \overline{\mathbf{e}}_\perp\overline{\mathbf{e}}^\top\big), \tag{3}$$

where $\rho \in [0, 1)$ is a constant. Note that $\widetilde{\boldsymbol{\Sigma}}$ is the data covariance matrix here.

**Remark 7.1 (Regarding $\widetilde{\boldsymbol{\Sigma}}$).** *We study the case of $\widetilde{\boldsymbol{\Sigma}}$ as given in Eq. (3) because it is the minimal analytically tractable case where we can show that* FLOW *goes beyond model averaging (MA) (see Remark 7.5). Specifically, if $\rho = 0$ and $\widetilde{\boldsymbol{\Sigma}} = \mathbf{I}_d$, then* FLOW *reduces to MA. Moreover, for an arbitrary $\widetilde{\boldsymbol{\Sigma}}$, characterizing the eigen-spectrum of the matrix dictating the trajectory of* FLOW *becomes intractable. For the analysis to be tractable, we need some relationship between $\widetilde{\boldsymbol{\Sigma}}$ and $\mathbf{e}$ (i.e., the difference between the optima of the pre-training and fine-tuning tasks).*[4]

For a model parameterized by $\boldsymbol{\theta} \in \mathbb{R}^d$, let

$$\mathrm{err}_1(\boldsymbol{\theta}) := \mathbb{E}_{\mathcal{D}}\Big[\big(\mathrm{y} - \langle\boldsymbol{\theta}, \mathbf{x}\rangle\big)^2\Big] = \big(\boldsymbol{\theta} - \boldsymbol{\theta}_*\big)^\top\boldsymbol{\Sigma}\big(\boldsymbol{\theta} - \boldsymbol{\theta}_*\big),$$

$$\mathrm{err}_2(\boldsymbol{\theta}) := \mathbb{E}_{\widetilde{\mathcal{D}}}\Big[\big(\widetilde{\mathrm{y}} - \langle\boldsymbol{\theta}, \widetilde{\mathbf{x}}\rangle\big)^2\Big] = \big(\boldsymbol{\theta} - \widetilde{\boldsymbol{\theta}}_*\big)^\top\widetilde{\boldsymbol{\Sigma}}\big(\boldsymbol{\theta} - \widetilde{\boldsymbol{\theta}}_*\big) \tag{4}$$

be the population errors on the pre-training and fine-tuning tasks, respectively. Also, the total error with $\boldsymbol{\theta}$ on the two tasks is denoted by $\mathrm{err}_{\mathrm{tot}}(\boldsymbol{\theta}) = \mathrm{err}_1(\boldsymbol{\theta}) + \mathrm{err}_2(\boldsymbol{\theta})$.

We assume that initially, we learn $\boldsymbol{\theta}_*$ with the pre-training data; so $\boldsymbol{\theta}_*$ is our pre-trained model. Note that

$$\mathrm{err}_{\mathrm{tot}}(\boldsymbol{\theta}_*) = \mathrm{err}_2(\boldsymbol{\theta}_*) = \mathbf{e}^\top\widetilde{\boldsymbol{\Sigma}}\mathbf{e} = \|\mathbf{e}\|_2^2, \tag{5}$$

where the last step follows by using $\widetilde{\boldsymbol{\Sigma}}$ from Equation (3).

[4] See Appendix E for more details.

We start fine-tuning starting from $\boldsymbol{\theta}_*$. Specifically, we assume access to the population$(\widetilde{\mathbf{x}}, \widetilde{\mathbf{y}}) \sim \widetilde{\mathcal{D}}$ of the fine-tuning task, but we lose access to the pre-training data.

**Vanilla fine-tuning (FT):** We minimize $\mathrm{err}_2(\boldsymbol{\theta})$ (Eq. (4)) with gradient descent (GD) starting from $\boldsymbol{\theta}_*$ using a constant learning rate $\overline{\eta}$. Our iterate $\overline{\boldsymbol{\theta}}_K$ at the $K^{\text{th}}$ iteration is given by (using the value of $\widetilde{\boldsymbol{\Sigma}}$ from Eq. (3) and $\boldsymbol{\theta}_* - \widetilde{\boldsymbol{\theta}}_* = \mathbf{e}$):

$$\overline{\boldsymbol{\theta}}_K = \widetilde{\boldsymbol{\theta}}_* + \Big(\mathbf{I}_d - 2\overline{\eta}\Big(\mathbf{I}_d + \rho\big(\overline{\mathbf{e}}\overline{\mathbf{e}}_\perp^\top + \overline{\mathbf{e}}_\perp\overline{\mathbf{e}}^\top\big)\Big)^K\mathbf{e}. \quad (6)$$

**FLOW:** For some temperature $\tau$, the weight of $(\widetilde{\mathbf{x}}, \widetilde{\mathbf{y}}) \sim \widetilde{\mathcal{D}}$ is $w(\widetilde{\mathbf{x}}, \widetilde{\mathbf{y}}) = \exp\Big(-\frac{(\widetilde{\mathbf{y}} - \langle \boldsymbol{\theta}_*, \widetilde{\mathbf{x}}\rangle)^2}{\tau}\Big)$. We minimize

$$\widehat{\mathrm{err}}_2(\widehat{\boldsymbol{\theta}}) := \mathbb{E}_{\widetilde{\mathcal{D}}}\Big[w(\widetilde{\mathbf{x}}, \widetilde{\mathbf{y}})\big(\widetilde{\mathbf{y}} - \langle \widehat{\boldsymbol{\theta}}, \widetilde{\mathbf{x}}\rangle\big)^2\Big], \quad (7)$$

with GD starting from $\boldsymbol{\theta}_*$ using a constant learning rate $\widehat{\eta}$. Suppose our iterate at the $K^{\text{th}}$ iteration is $\widehat{\boldsymbol{\theta}}_K$.

**Theorem 7.2 (FLOW).** *Let* $\mu = \Big(\frac{\tau}{\tau + 2\|\mathbf{e}\|_2^2}\Big)^{1/2}$. *Then:*

$$\widehat{\boldsymbol{\theta}}_K = \widetilde{\boldsymbol{\theta}}_* + \Big(\mathbf{I}_d - 2\widehat{\eta}\widetilde{\boldsymbol{\Sigma}}'\Big)^K\mathbf{e}, \quad (8)$$

*where* $\widetilde{\boldsymbol{\Sigma}}' = \mu\big(\mathbf{I}_d - \mathbf{Q}\big)$ *with*

$$\mathbf{Q} = (1-\mu^2)\overline{\mathbf{e}}\overline{\mathbf{e}}^\top + \rho^2(1-\mu^2)\overline{\mathbf{e}}_\perp\overline{\mathbf{e}}_\perp^\top - \rho\mu^2\big(\overline{\mathbf{e}}\overline{\mathbf{e}}_\perp^\top + \overline{\mathbf{e}}_\perp\overline{\mathbf{e}}^\top\big). \quad (9)$$

We prove Thm. 7.2 in Appendix D. The **main technical challenge** is the evaluation of $\widetilde{\boldsymbol{\Sigma}}'$, viz., the covariance matrix of the *weighted* fine-tuning data; see Lemma F.1 for this.

Now, we are going to compare vanilla FT (6) with $\overline{\eta} = \frac{1}{2}$ and FLOW (8) with $\widehat{\eta} = \frac{1}{2\mu}$. We believe these are comparable learning rates for vanilla FT and FLOW because the resultant matrices (Eqs. (10) and (11)) dictating the convergence of both methods have exactly two non-zero eigenvalues and the corresponding eigenvectors lie in the span of $\overline{\mathbf{e}}$ and $\overline{\mathbf{e}}_\perp$. Plugging in $\overline{\eta} = \frac{1}{2}$ into Eq. (6), we get:

$$\overline{\boldsymbol{\theta}}_K = \widetilde{\boldsymbol{\theta}}_* + \mathbf{P}^K\mathbf{e}, \text{ with } \mathbf{P} = -\rho\big(\overline{\mathbf{e}}\overline{\mathbf{e}}_\perp^\top + \overline{\mathbf{e}}_\perp\overline{\mathbf{e}}^\top\big) \quad (10)$$

for vanilla FT. Plugging in $\widehat{\eta} = \frac{1}{2\mu}$ into Eq. (8), we get:

$$\widehat{\boldsymbol{\theta}}_K = \widetilde{\boldsymbol{\theta}}_* + \mathbf{Q}^K\mathbf{e}, \text{ with } \mathbf{Q} \text{ given by Eq. (9)} \quad (11)$$

for FLOW. The non-zero eigenvalues of $\mathbf{P}$ are $\mp\rho$ and the corresponding eigenvectors are $\frac{1}{\sqrt{2}}\big(\overline{\mathbf{e}} \pm \overline{\mathbf{e}}_\perp\big)$. Using this in (10) and simplifying, we get for vanilla FT:

$$\overline{\boldsymbol{\theta}}_K = \widetilde{\boldsymbol{\theta}}_* + \rho^K\Big(\mathbb{1}\big(K \text{ is even}\big)\mathbf{e} - \mathbb{1}\big(K \text{ is odd}\big)\|\mathbf{e}\|_2\overline{\mathbf{e}}_\perp\Big). \quad (12)$$

**Remark 7.3 (Vanilla FT).** *Since* $\rho < 1$, $\overline{\boldsymbol{\theta}}_K$ *converges to* $\widetilde{\boldsymbol{\theta}}_*$ *rapidly, and we cannot impede this convergence.*

Note that (we use $\boldsymbol{\Sigma} \succeq \mathbf{I}_d$ below):

$$\mathrm{err}_{\text{tot}}(\widetilde{\boldsymbol{\theta}}_*) = \mathrm{err}_1(\widetilde{\boldsymbol{\theta}}_*) = \mathbf{e}^\top\boldsymbol{\Sigma}\mathbf{e} \geq \|\mathbf{e}\|_2^2. \quad (13)$$

On the other hand, the non-zero eigenvalues and corresponding eigenvectors of $\mathbf{Q}$ are not as straightforward to compute. We do this computation in Lemma F.3 with the re-parameterization of $\mu = \sqrt{\frac{\beta(1-\rho^2)}{(1+\beta)(1-\beta\rho^2)}}$ for some $\beta \in (0, 1]$.[5] Using this in Eq. (11) and simplifying, we get for FLOW:

$$\widehat{\boldsymbol{\theta}}_K = \widetilde{\boldsymbol{\theta}}_* + \left(\frac{\widehat{\lambda}_1^K + \widehat{\lambda}_2^K\beta^2\rho^2}{1+\beta^2\rho^2}\right)\mathbf{e} - \beta\rho\left(\frac{\widehat{\lambda}_1^K - \widehat{\lambda}_2^K}{1+\beta^2\rho^2}\right)\|\mathbf{e}\|\overline{\mathbf{e}}_\perp, \quad (14)$$

where $\widehat{\lambda}_1 = \frac{1+\beta\rho^2}{1+\beta}$ and $\widehat{\lambda}_2 = \rho^2\Big(\frac{1-\beta}{1-\beta\rho^2}\Big)$.

**Remark 7.4 (FLOW's trajectory).** *Note that we can control* $\widehat{\lambda}_1$ *by varying* $\beta$. *Specifically, we can make* $\widehat{\lambda}_1$ *arbitrarily close to* 1 *by choosing a small enough* $\beta$. *On the other hand,* $\frac{1-\beta}{1-\beta\rho^2} < \frac{1+\beta\rho^2}{1+\beta} = \widehat{\lambda}_1$ *and so,* $\widehat{\lambda}_2 < \rho^2\widehat{\lambda}_1$. *Hence, beyond a certain number of iterations* $K$, *Eq. (14) becomes:*

$$\widehat{\boldsymbol{\theta}}_K \approx \boldsymbol{\theta}_K := \widetilde{\boldsymbol{\theta}}_* + \gamma(K, \beta)\Big(\mathbf{e} - \beta\rho\|\mathbf{e}\|_2\overline{\mathbf{e}}_\perp\Big), \quad (15)$$

*with* $\gamma(K, \beta) := \Big(\frac{\widehat{\lambda}_1^K}{1+\beta^2\rho^2}\Big)$. *Because we can control* $\widehat{\lambda}_1$ *by varying* $\beta$, *we can control* $\gamma(K, \beta)$. *Thus,* ***we can stall convergence along*** $\big(\mathbf{e} - \beta\rho\|\mathbf{e}\|_2\overline{\mathbf{e}}_\perp\big)$,[6] ***impeding the convergence of*** $\widehat{\boldsymbol{\theta}}_K$ ***to*** $\widetilde{\boldsymbol{\theta}}_*$.

**Remark 7.5 (FLOW goes beyond model averaging).** *If we perform model averaging between* $\boldsymbol{\theta}_*$ *and* $\widetilde{\boldsymbol{\theta}}_*$ *with parameter* $\omega \in [0, 1]$, *then our averaged model is:*

$$\boldsymbol{\theta}_{\text{avg}}(\omega) = \omega\boldsymbol{\theta}_* + (1-\omega)\widetilde{\boldsymbol{\theta}}_* = \widetilde{\boldsymbol{\theta}}_* + \omega\mathbf{e}. \quad (16)$$

*Comparing the above with Eq. (15), we see that* FLOW *goes beyond model averaging because of the component along* $\overline{\mathbf{e}}_\perp$. *But we can make* $\boldsymbol{\theta}_K$ *(Eq. (15))* $\to \boldsymbol{\theta}_{\text{avg}}(\omega)$ *by choosing* $\beta \to 0$ *and* $K$ *such that* $\gamma(K, \beta) \to \omega$. *So, we expect* **FLOW** ***to be at least as powerful as model averaging***.

As per Lemma F.4, the minimum total error on both tasks with optimally tuned model averaging is given by:

$$\min_{\omega\in[0,1]} \mathrm{err}_{\text{tot}}\big(\boldsymbol{\theta}_{\text{avg}}(\omega)\big) = \left(\frac{\overline{\mathbf{e}}^\top\boldsymbol{\Sigma}\overline{\mathbf{e}}}{\overline{\mathbf{e}}^\top\boldsymbol{\Sigma}\overline{\mathbf{e}} + 1}\right)\|\mathbf{e}\|_2^2 < \|\mathbf{e}\|_2^2, \quad (17)$$

where recall that $\boldsymbol{\Sigma}$ is the covariance matrix of the pre-training data. On the other hand, using Eqs. (5) and (13)

$$\min\Big(\mathrm{err}_{\text{tot}}(\boldsymbol{\theta}_*), \mathrm{err}_{\text{tot}}(\widetilde{\boldsymbol{\theta}}_*)\Big) = \|\mathbf{e}\|_2^2. \quad (18)$$

[5] The corresponding temperature is $\tau = \frac{2\beta(1-\rho^2)\|\mathbf{e}\|_2^2}{(1-\beta^2\rho^2)}$.

[6] This direction is the top eigenvector of $\mathbf{Q}$. Since $\widetilde{\boldsymbol{\Sigma}}' = \mu\big(\mathbf{I}_d - \mathbf{Q}\big)$, this is also the eigenvector of $\widetilde{\boldsymbol{\Sigma}}'$ with the smallest eigenvalue.

**Remark 7.6 (Error comparison).** *By comparing Eqs. (17) and (18), we see that optimally tuned model averaging attains a smaller total error than both $\boldsymbol{\theta}_*$ (i.e., the pre-trained model) and $\widetilde{\boldsymbol{\theta}}_*$ to which vanilla FT converges rapidly (Remark 7.3). More importantly, following our discussion in Remark 7.5, we conclude that* ***optimally tuned*** FLOW*'s total error is at least as good as the one in Eq. (17).*

## 8. Conclusion

In this paper, we studied the problem of catastrophic forgetting in pre-trained models during fine-tuning when we do not have access to the pre-training data. To mitigate this issue, we proposed FLOW, a method which upweights easy samples based on the pre-trained loss values. Empirically, we showed that FLOW, on average, outperforms relevant baselines and is also complementary to these baselines in both vision and language settings. We also theoretically analyzed FLOW for linear models.

**Discussion and limitations.** We would like to conclude with an overview of our work's limitations and potential future directions. In lay terms, mitigating forgetting of pre-training capabilities comes at the cost of relatively lower fine-tuning performance. FLOW maintains this balance by sacrificing performance on *hard samples from the fine-tuning data*. Table 5 indicates that FLOW has lower accuracy on samples with high pre-training loss ("hard samples") compared to standard FT. Our method selectively down-weighs samples with high pre-training losses for preserving pre-training performance. An interesting future direction is improving performance on such samples while maintaining or improving overall performance. On the theoretical side, we hope to extend our analysis to generalized linear models (GLMs) and even non-linear models.

*Table 5.* **Comparison of FLOW and Standard-FT on hard samples across three vision datasets.** We evaluate performance on the top 10% hardest samples (those with the highest pre-trained losses). Indeed, the samples with high pre-training losses have lower accuracy when using FLOW compared to standard fine-tuning (FT). This is an unsurprising outcome of our approach; we sacrifice performance on hard examples of the fine-tuning data to maintain performance on the pre-training data.

| Dataset | # of Hard Samples | Standard FT | FLOW |
|---|---|---|---|
| CIFAR-10 | 1000 | 86.60 | 30.70 |
| CIFAR-100 | 1000 | 56.40 | 21.30 |
| Stanford Cars | 805 | 71.30 | 13.18 |

## Acknowledgments

This research was supported by NSF EnCORE Tripods (2217069) and NSF AI Institute for the Foundations of Machine Learning (2019844). Research of Ali Kavis is funded in part by the Swiss National Science Foundation (SNSF) under grant number P500PT_217942. The authors are grateful to anonymous reviewers for their feedback on improving this paper.

## Impact Statement

This paper presents work whose goal is to advance the field of machine learning. There are potential societal consequences of our work, none of which we feel must be specifically highlighted here.

## References

Ahn, H., Cha, S., Lee, D., and Moon, T. Uncertainty-based continual learning with adaptive regularization. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/2c3ddf4bf13852db711dd1901fb517fa-Paper.pdf.

Alain, G., Lamb, A., Sankar, C., Courville, A. C., and Bengio, Y. Variance reduction in sgd by distributed importance sampling. *ArXiv*, abs/1511.06481, 2015. URL https://api.semanticscholar.org/CorpusID:6546520.

Aljundi, R., Chakravarty, P., and Tuytelaars, T. Expert gate: Lifelong learning with a network of experts. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7120–7129, 2017. doi: 10.1109/CVPR.2017.753.

Aljundi, R., Babiloni, F., Elhoseiny, M., Rohrbach, M., and Tuytelaars, T. Memory aware synapses: Learning what (not) to forget. In *Computer Vision – ECCV 2018: 15th European Conference, Munich, Germany, September 8–14, 2018, Proceedings, Part III*, pp. 144–161, Berlin, Heidelberg, 2018. Springer-Verlag. ISBN 978-3-030-01218-2. doi: 10.1007/978-3-030-01219-9_9. URL https://doi.org/10.1007/978-3-030-01219-9_9.

Aljundi, R., Lin, M., Goujaud, B., and Bengio, Y. Gradient based sample selection for online continual learning. *Advances in neural information processing systems*, 32, 2019.

Anonymous. Dynamic loss-based sample reweighting for improved large language model pretraining. In *The Thirteenth International Conference on Learning Represen-*

*tations*, 2025. URL https://openreview.net/forum?id=gU4ZgQNsOC.

Austin, J., Odena, A., Nye, M., Bosma, M., Michalewski, H., Dohan, D., Jiang, E., Cai, C., Terry, M., Le, Q., et al. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*, 2021.

Bang, J., Kim, H., Yoo, Y. J., Ha, J.-W., and Choi, J. Rainbow memory: Continual learning with a memory of diverse samples. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8214–8223, 2021. URL https://api.semanticscholar.org/CorpusID:232427874.

Ben-Tal, A., Den Hertog, D., De Waegenaere, A., Melenberg, B., and Rennen, G. Robust solutions of optimization problems affected by uncertain probabilities. *Management Science*, 59(2):341–357, 2013.

Biderman, D., Portes, J., Ortiz, J. J. G., Paul, M., Greengard, P., Jennings, C., King, D., Havens, S., Chiley, V., Frankle, J., Blakeney, C., and Cunningham, J. P. LoRA learns less and forgets less. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=aloEru2qCG. Featured Certification.

Bisk, Y., Zellers, R., Bras, R. L., Gao, J., and Choi, Y. Piqa: Reasoning about physical commonsense in natural language. In *Thirty-Fourth AAAI Conference on Artificial Intelligence*, 2020.

Borsos, Z., Mutny, M., and Krause, A. Coresets via bilevel optimization for continual learning and streaming. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 14879–14890. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/aa2a77371374094fe9e0bc1de3f94ed9-Paper.pdf.

Bossard, L., Guillaumin, M., and Van Gool, L. Food-101 – mining discriminative components with random forests. In *European Conference on Computer Vision*, 2014.

Caccia, L., Belilovsky, E., Caccia, M., and Pineau, J. Online learned continual compression with adaptive quantization modules. In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20. JMLR.org, 2020.

Chaudhry, A., Ranzato, M., Rohrbach, M., and Elhoseiny, M. Efficient lifelong learning with a-GEM. In *International Conference on Learning Representations*, 2019a. URL https://openreview.net/forum?id=Hkf2_sC5FX.

Chaudhry, A., Rohrbach, M., Elhoseiny, M., Ajanthan, T., Dokania, P., Torr, P., and Ranzato, M. Continual learning with tiny episodic memories. In *Workshop on Multi-Task and Lifelong Reinforcement Learning*, 2019b.

Chen, X., Wang, Z., Sow, D., Yang, J., Chen, T., Liang, Y., Zhou, M., and Wang, Z. Take the bull by the horns: Hard sample-reweighted continual training improves llm generalization. *arXiv preprint arXiv:2402.14270*, 2024a.

Chen, Y., Wang, S., Lin, Z., Qin, Z., Zhang, Y., Ding, T., and Sun, R. Mofo: Momentum-filtered optimizer for mitigating forgetting in llm fine-tuning. *arXiv preprint arXiv:2407.20999*, 2024b.

Clark, P., Cowhey, I., Etzioni, O., Khot, T., Sabharwal, A., Schoenick, C., and Tafjord, O. Think you have solved question answering? try arc, the ai2 reasoning challenge. *ArXiv*, abs/1803.05457, 2018.

Cobbe, K., Kosaraju, V., Bavarian, M., Hilton, J., Nakano, R., Hesse, C., and Schulman, J. Training verifiers to solve math word problems, 2021.

Collier, M. P., Kokiopoulou, E., Gesmundo, A., and Berent, J. Routing networks with co-training for continual learning. In *ICML 2020 Workshop on Continual Learning*, 2020.

Das, R., Chen, X., Ieong, B., Bansal, P., and Sanghavi, S. Understanding the training speedup from sampling with approximate losses. *arXiv preprint arXiv:2402.07052*, 2024.

De Lange, M. and Tuytelaars, T. Continual prototype evolution: Learning online from non-stationary data streams. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 8250–8259, October 2021.

Dhar, P., Singh, R. V., Peng, K., Wu, Z., and Chellappa, R. Learning without memorizing. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pp. 5138–5146. Computer Vision Foundation / IEEE, 2019. doi: 10.1109/CVPR.2019.00528. URL http://openaccess.thecvf.com/content_CVPR_2019/html/Dhar_Learning_Without_Memorizing_CVPR_2019_paper.html.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N. An image is worth 16x16 words: Transformers for image recognition at scale. *ICLR*, 2021.

Duchi, J. C. and Namkoong, H. Learning models with uniform performance via distributionally robust optimization. *The Annals of Statistics*, 49(3):1378–1406, 2021.

Farajtabar, M., Azizan, N., Mott, A., and Li, A. Orthogonal gradient descent for continual learning. In Chiappa, S. and Calandra, R. (eds.), *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pp. 3762–3773. PMLR, 26–28 Aug 2020. URL https://proceedings.mlr.press/v108/farajtabar20a.html.

Gao, L., Tow, J., Abbasi, B., Biderman, S., Black, S., DiPofi, A., Foster, C., Golding, L., Hsu, J., Le Noac'h, A., Li, H., McDonell, K., Muennighoff, N., Ociepa, C., Phang, J., Reynolds, L., Schoelkopf, H., Skowron, A., Sutawika, L., Tang, E., Thite, A., Wang, B., Wang, K., and Zou, A. A framework for few-shot language model evaluation, 07 2024. URL https://zenodo.org/records/12608602.

Gekhman, Z., Yona, G., Aharoni, R., Eyal, M., Feder, A., Reichart, R., and Herzig, J. Does fine-tuning llms on new knowledge encourage hallucinations? In *EMNLP*, pp. 7765–7784, 2024. URL https://aclanthology.org/2024.emnlp-main.444.

Ghosal, G., Hashimoto, T., and Raghunathan, A. Understanding finetuning for factual knowledge extraction. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024.

Goyal, S., Kumar, A., Garg, S., Kolter, Z., and Raghunathan, A. Finetune like you pretrain: Improved finetuning of zero-shot vision models. *CVPR*, 2023.

Grattafiori, A., Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., Letman, A., Mathur, A., Schelten, A., Vaughan, A., Yang, A., Fan, A., Goyal, A., Hartshorn, A., Yang, A., Mitra, A., Sravankumar, A., Korenev, A., Hinsvark, A., Rao, A., Zhang, A., Rodriguez, A., Gregerson, A., Spataru, A., Roziere, B., Biron, B., Tang, B., Chern, B., Caucheteux, C., Nayak, C., Bi, C., Marra, C., McConnell, C., Keller, C., Touret, C., Wu, C., Wong, C., Ferrer, C. C., Nikolaidis, C., Allonsius, D., Song, D., Pintz, D., Livshits, D., Wyatt, D., Esiobu, D., Choudhary, D., Mahajan, D., Garcia-Olano, D., Perino, D., Hupkes, D., Lakomkin, E., AlBadawy, E., Lobanova, E., Dinan, E., Smith, E. M., Radenovic, F., Guzmán, F., Zhang, F., Synnaeve, G., Lee, G., Anderson, G. L., Thattai, G., Nail, G., Mialon, G., Pang, G., Cucurell, G., Nguyen, H., Korevaar, H., Xu, H., Touvron, H., Zarov, I., Ibarra, I. A., Kloumann, I., Misra, I., Evtimov, I., Zhang, J., Copet, J., Lee, J., Geffert, J., Vranes, J., Park, J., Mahadeokar, J., Shah, J., van der Linde, J., Billock, J., Hong, J., Lee, J., Fu, J., Chi, J., Huang, J., Liu, J., Wang, J., Yu, J., Bitton, J., Spisak, J., Park, J., Rocca, J., Johnstun, J., Saxe, J., Jia, J., Alwala, K. V., Prasad, K., Upasani, K., Plawiak, K., Li, K., Heafield, K., Stone, K., El-Arini, K., Iyer, K., Malik, K., Chiu, K., Bhalla, K., Lakhotia, K., Rantala-Yeary, L., van der Maaten, L., Chen, L., Tan, L., Jenkins, L., Martin, L., Madaan, L., Malo, L., Blecher, L., Landzaat, L., de Oliveira, L., Muzzi, M., Pasupuleti, M., Singh, M., Paluri, M., Kardas, M., Tsimpoukelli, M., Oldham, M., Rita, M., Pavlova, M., Kambadur, M., Lewis, M., Si, M., Singh, M. K., Hassan, M., Goyal, N., Torabi, N., Bashlykov, N., Bogoychev, N., Chatterji, N., Zhang, N., Duchenne, O., Çelebi, O., Alrassy, P., Zhang, P., Li, P., Vasic, P., Weng, P., Bhargava, P., Dubal, P., Krishnan, P., Koura, P. S., Xu, P., He, Q., Dong, Q., Srinivasan, R., Ganapathy, R., Calderer, R., Cabral, R. S., Stojnic, R., Raileanu, R., Maheswari, R., Girdhar, R., Patel, R., Sauvestre, R., Polidoro, R., Sumbaly, R., Taylor, R., Silva, R., Hou, R., Wang, R., Hosseini, S., Chennabasappa, S., Singh, S., Bell, S., Kim, S. S., Edunov, S., Nie, S., Narang, S., Raparthy, S., Shen, S., Wan, S., Bhosale, S., Zhang, S., Vandenhende, S., Batra, S., Whitman, S., Sootla, S., Collot, S., Gururangan, S., Borodinsky, S., Herman, T., Fowler, T., Sheasha, T., Georgiou, T., Scialom, T., Speckbacher, T., Mihaylov, T., Xiao, T., Karn, U., Goswami, V., Gupta, V., Ramanathan, V., Kerkez, V., Gonguet, V., Do, V., Vogeti, V., Albiero, V., Petrovic, V., Chu, W., Xiong, W., Fu, W., Meers, W., Martinet, X., Wang, X., Wang, X., Tan, X. E., Xia, X., Xie, X., Jia, X., Wang, X., Goldschlag, Y., Gaur, Y., Babaei, Y., Wen, Y., Song, Y., Zhang, Y., Li, Y., Mao, Y., Coudert, Z. D., Yan, Z., Chen, Z., Papakipos, Z., Singh, A., Srivastava, A., Jain, A., Kelsey, A., Shajnfeld, A., Gangidi, A., Victoria, A., Goldstand, A., Menon, A., Sharma, A., Boesenberg, A., Baevski, A., Feinstein, A., Kallet, A., Sangani, A., Teo, A., Yunus, A., Lupu, A., Alvarado, A., Caples, A., Gu, A., Ho, A., Poulton, A., Ryan, A., Ramchandani, A., Dong, A., Franco, A., Goyal, A., Saraf, A., Chowdhury, A., Gabriel, A., Bharambe, A., Eisenman, A., Yazdan, A., James, B., Maurer, B., Leonhardi, B., Huang, B., Loyd, B., Paola, B. D., Paranjape, B., Liu, B., Wu, B., Ni, B., Hancock, B., Wasti, B., Spence, B., Stojkovic, B., Gamido, B., Montalvo, B., Parker, C., Burton, C., Mejia, C., Liu, C., Wang, C., Kim, C., Zhou, C., Hu, C., Chu, C.-H., Cai, C., Tindal, C., Feichtenhofer, C., Gao, C., Civin, D., Beaty, D., Kreymer, D., Li, D., Adkins, D., Xu, D., Testuggine, D., David, D., Parikh, D., Liskovich, D., Foss, D., Wang, D., Le, D., Holland, D., Dowling, E., Jamil, E., Montgomery, E., Presani, E., Hahn, E., Wood, E., Le, E.-T., Brinkman, E., Arcaute, E., Dunbar, E., Smothers, E., Sun, F., Kreuk, F., Tian, F., Kokkinos, F., Ozgenel, F., Caggioni, F., Kanayet, F., Seide, F., Florez, G. M., Schwarz, G., Badeer, G., Swee, G., Halpern, G., Herman, G., Sizov, G., Guangyi, Zhang, Lakshminarayanan, G., Inan, H.,

Shojanazeri, H., Zou, H., Wang, H., Zha, H., Habeeb, H., Rudolph, H., Suk, H., Aspegren, H., Goldman, H., Zhan, H., Damlaj, I., Molybog, I., Tufanov, I., Leontiadis, I., Veliche, I.-E., Gat, I., Weissman, J., Geboski, J., Kohli, J., Lam, J., Asher, J., Gaya, J.-B., Marcus, J., Tang, J., Chan, J., Zhen, J., Reizenstein, J., Teboul, J., Zhong, J., Jin, J., Yang, J., Cummings, J., Carvill, J., Shepard, J., McPhie, J., Torres, J., Ginsburg, J., Wang, J., Wu, K., U, K. H., Saxena, K., Khandelwal, K., Zand, K., Matosich, K., Veeraraghavan, K., Michelena, K., Li, K., Jagadeesh, K., Huang, K., Chawla, K., Huang, K., Chen, L., Garg, L., A, L., Silva, L., Bell, L., Zhang, L., Guo, L., Yu, L., Moshkovich, L., Wehrstedt, L., Khabsa, M., Avalani, M., Bhatt, M., Mankus, M., Hasson, M., Lennie, M., Reso, M., Groshev, M., Naumov, M., Lathi, M., Keneally, M., Liu, M., Seltzer, M. L., Valko, M., Restrepo, M., Patel, M., Vyatskov, M., Samvelyan, M., Clark, M., Macey, M., Wang, M., Hermoso, M. J., Metanat, M., Rastegari, M., Bansal, M., Santhanam, N., Parks, N., White, N., Bawa, N., Singhal, N., Egebo, N., Usunier, N., Mehta, N., Laptev, N. P., Dong, N., Cheng, N., Chernoguz, O., Hart, O., Salpekar, O., Kalinli, O., Kent, P., Parekh, P., Saab, P., Balaji, P., Rittner, P., Bontrager, P., Roux, P., Dollar, P., Zvyagina, P., Ratanchandani, P., Yuvraj, P., Liang, Q., Alao, R., Rodriguez, R., Ayub, R., Murthy, R., Nayani, R., Mitra, R., Parthasarathy, R., Li, R., Hogan, R., Battey, R., Wang, R., Howes, R., Rinott, R., Mehta, S., Siby, S., Bondu, S. J., Datta, S., Chugh, S., Hunt, S., Dhillon, S., Sidorov, S., Pan, S., Mahajan, S., Verma, S., Yamamoto, S., Ramaswamy, S., Lindsay, S., Lindsay, S., Feng, S., Lin, S., Zha, S. C., Patil, S., Shankar, S., Zhang, S., Zhang, S., Wang, S., Agarwal, S., Sajuyigbe, S., Chintala, S., Max, S., Chen, S., Kehoe, S., Satterfield, S., Govindaprasad, S., Gupta, S., Deng, S., Cho, S., Virk, S., Subramanian, S., Choudhury, S., Goldman, S., Remez, T., Glaser, T., Best, T., Koehler, T., Robinson, T., Li, T., Zhang, T., Matthews, T., Chou, T., Shaked, T., Vontimitta, V., Ajayi, V., Montanez, V., Mohan, V., Kumar, V. S., Mangla, V., Ionescu, V., Poenaru, V., Mihailescu, V. T., Ivanov, V., Li, W., Wang, W., Jiang, W., Bouaziz, W., Constable, W., Tang, X., Wu, X., Wang, X., Wu, X., Gao, X., Kleinman, Y., Chen, Y., Hu, Y., Jia, Y., Qi, Y., Li, Y., Zhang, Y., Zhang, Y., Adi, Y., Nam, Y., Yu, Wang, Zhao, Y., Hao, Y., Qian, Y., Li, Y., He, Y., Rait, Z., DeVito, Z., Rosnbrick, Z., Wen, Z., Yang, Z., Zhao, Z., and Ma, Z. The llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

Gugger, S., Debut, L., Wolf, T., Schmid, P., Mueller, Z., Mangrulkar, S., Sun, M., and Bossan, B. Accelerate: Training and inference at scale made simple, efficient and adaptable. https://github.com/huggingface/accelerate, 2022.

Hendrycks, D., Burns, C., Basart, S., Zou, A., Mazeika, M., Song, D., and Steinhardt, J. Measuring massive multitask language understanding. *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021a.

Hendrycks, D., Burns, C., Kadavath, S., Arora, A., Basart, S., Tang, E., Song, D., and Steinhardt, J. Measuring mathematical problem solving with the math dataset. *NeurIPS*, 2021b.

Hinton, G. E., Vinyals, O., and Dean, J. Distilling the knowledge in a neural network. *CoRR*, abs/1503.02531, 2015. URL http://arxiv.org/abs/1503.02531.

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., and Chen, W. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Hung, C.-Y., Tu, C.-H., Wu, C.-E., Chen, C.-H., Chan, Y.-M., and Chen, C.-S. Compacting, picking and growing for unforgetting continual learning. *Advances in neural information processing systems*, 32, 2019.

Ilharco, G., Ribeiro, M. T., Wortsman, M., Schmidt, L., Hajishirzi, H., and Farhadi, A. Editing models with task arithmetic. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=6t0Kwf8-jrj.

Isele, D. and Cosgun, A. Selective experience replay for lifelong learning. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence*, AAAI'18/IAAI'18/EAAI'18. AAAI Press, 2018. ISBN 978-1-57735-800-8.

Jiang, A. Q., Sablayrolles, A., Mensch, A., Bamford, C., Chaplot, D. S., de Las Casas, D., Bressand, F., Lengyel, G., Lample, G., Saulnier, L., Lavaud, L. R., Lachaux, M., Stock, P., Scao, T. L., Lavril, T., Wang, T., Lacroix, T., and Sayed, W. E. Mistral 7b. *CoRR*, abs/2310.06825, 2023. doi: 10.48550/ARXIV.2310.06825. URL https://doi.org/10.48550/arXiv.2310.06825.

Jung, H., Ju, J., Jung, M., and Kim, J. Less-forgetful learning for domain expansion in deep neural networks. In *AAAI Conference on Artificial Intelligence*, 2017. URL https://api.semanticscholar.org/CorpusID:19243534.

Jung, S., Ahn, H., Cha, S., and Moon, T. Continual learning with node-importance based adaptive group sparse regularization. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA, 2020. Curran Associates Inc. ISBN 9781713829546.

Kang, H., Mina, R. J. L., Madjid, S. R. H., Yoon, J., Hasegawa-Johnson, M., Hwang, S. J., and Yoo, C. D. Forget-free continual learning with winning subnetworks. In Chaudhuri, K., Jegelka, S., Song, L., Szepesvari, C., Niu, G., and Sabato, S. (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 10734–10750. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/kang22b.html.

Katharopoulos, A. and Fleuret, F. Biased importance sampling for deep neural network training. *arXiv preprint arXiv:1706.00043*, 2017.

Katharopoulos, A. and Fleuret, F. Not all samples are created equal: Deep learning with importance sampling. In Dy, J. and Krause, A. (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 2525–2534. PMLR, 10–15 Jul 2018. URL https://proceedings.mlr.press/v80/katharopoulos18a.html.

Kawaguchi, K. and Lu, H. Ordered sgd: A new stochastic optimization framework for empirical risk minimization. In *International Conference on Artificial Intelligence and Statistics*, pp. 669–679. PMLR, 2020.

Kemker, R. and Kanan, C. Fearnet: Brain-inspired model for incremental learning. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=SJ1Xmf-Rb.

Kirkpatrick, J., Pascanu, R., Rabinowitz, N. C., Veness, J., Desjardins, G., Rusu, A. A., Milan, K., Quan, J., Ramalho, T., Grabska-Barwinska, A., Hassabis, D., Clopath, C., Kumaran, D., and Hadsell, R. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114:3521 – 3526, 2016. URL https://api.semanticscholar.org/CorpusID:4704285.

Kleiman, A., Dziugaite, G. K., Frankle, J., Kakade, S., and Paul, M. Soup to go: mitigating forgetting during continual learning with model averaging, 2025. URL https://arxiv.org/abs/2501.05559.

Krause, J., Stark, M., Deng, J., and Fei-Fei, L. 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.

Krizhevsky, A. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

Lee, J., Xiao, L., Schoenholz, S., Bahri, Y., Novak, R., Sohl-Dickstein, J., and Pennington, J. Wide neural networks of any depth evolve as linear models under gradient descent. *Advances in neural information processing systems*, 32, 2019a.

Lee, J., Hong, H. G., Joo, D., and Kim, J. Continual learning with extended kronecker-factored approximate curvature. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8998–9007, 2020. URL https://api.semanticscholar.org/CorpusID:215786151.

Lee, K., Lee, K., Shin, J., and Lee, H. Overcoming catastrophic forgetting with unlabeled data in the wild. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 312–321, 2019b. URL https://api.semanticscholar.org/CorpusID:201314887.

Lee, S.-W., Kim, J.-H., Jun, J., Ha, J.-W., and Zhang, B.-T. Overcoming catastrophic forgetting by incremental moment matching. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, pp. 4655–4665, Red Hook, NY, USA, 2017. Curran Associates Inc. ISBN 9781510860964.

Levy, D., Carmon, Y., Duchi, J. C., and Sidford, A. Large-scale methods for distributionally robust optimization. *Advances in Neural Information Processing Systems*, 33: 8847–8860, 2020.

Lhoest, Q., Villanova del Moral, A., Jernite, Y., Thakur, A., von Platen, P., Patil, S., Chaumond, J., Drame, M., Plu, J., Tunstall, L., Davison, J., Šaško, M., Chhablani, G., Malik, B., Brandeis, S., Le Scao, T., Sanh, V., Xu, C., Patry, N., McMillan-Major, A., Schmid, P., Gugger, S., Delangue, C., Matussière, T., Debut, L., Bekman, S., Cistac, P., Goehringer, T., Mustar, V., Lagunas, F., Rush, A., and Wolf, T. Datasets: A community library for natural language processing. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 175–184, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. URL https://aclanthology.org/2021.emnlp-demo.21.

Li, F.-F., Andreeto, M., Ranzato, M., and Perona, P. Caltech 101, Apr 2022.

Li, Z. and Hoiem, D. Learning without forgetting. In Leibe, B., Matas, J., Sebe, N., and Welling, M. (eds.), *ECCV (4)*, volume 9908 of *Lecture Notes in Computer Science*, pp. 614–629. Springer, 2016. ISBN 978-3-319-46492-3. URL http://dblp.uni-trier.de/db/conf/eccv/eccv2016-4.html#LiH16.

Lin, Y., Lin, H., Xiong, W., Diao, S., Liu, J., Zhang, J., Pan, R., Wang, H., Hu, W., Zhang, H., et al. Mitigating the alignment tax of rlhf. *CoRR*, 2023.

Liu, X., Masana, M., Herranz, L., Van de Weijer, J., López, A. M., and Bagdanov, A. D. Rotate your networks: Better weight consolidation and less catastrophic forgetting. In *2018 24th International Conference on Pattern Recognition (ICPR)*, pp. 2262–2268, 2018. doi: 10.1109/ICPR.2018.8545895.

Lopez-Paz, D. and Ranzato, M. Gradient episodic memory for continual learning. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, pp. 6470–6479, Red Hook, NY, USA, 2017. Curran Associates Inc. ISBN 9781510860964.

Loshchilov, I. and Hutter, F. Online batch selection for faster training of neural networks. *ArXiv*, abs/1511.06343, 2015. URL https://api.semanticscholar.org/CorpusID:5324823.

Lubana, E. S., Trivedi, P., Koutra, D., and Dick, R. How do quadratic regularizers prevent catastrophic forgetting: The role of interpolation. *COLLAS*, 2021.

Mallya, A. and Lazebnik, S. Packnet: Adding multiple tasks to a single network by iterative pruning. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7765–7773, 2017. URL https://api.semanticscholar.org/CorpusID:35249701.

Mallya, A., Davis, D., and Lazebnik, S. Piggyback: Adapting a single network to multiple tasks by learning to mask weights. In *Computer Vision – ECCV 2018: 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part IV*, pp. 72–88, Berlin, Heidelberg, 2018. Springer-Verlag. ISBN 978-3-030-01224-3. doi: 10.1007/978-3-030-01225-0_5. URL https://doi.org/10.1007/978-3-030-01225-0_5.

Mangrulkar, S., Gugger, S., Debut, L., Belkada, Y., Paul, S., and Bossan, B. Peft: State-of-the-art parameter-efficient fine-tuning methods. https://github.com/huggingface/peft, 2022.

Mcclelland, J., Mcnaughton, B., and O'Reilly, R. Why there are complementary learning systems in the hippocampus and neocortex: Insights from the successes and failures of connectionist models of learning and memory. *Psychological review*, 102:419–57, 08 1995. doi: 10.1037/0033-295X.102.3.419.

Mihaylov, T., Clark, P., Khot, T., and Sabharwal, A. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *EMNLP*, 2018.

Mustafa B Gurbuz, C. D. Nispa: Neuro-inspired stability-plasticity adaptation for continual learning in sparse networks. *Proceedings of the 39th International Conference on Machine Learning*, 162, 2022. URL https://par.nsf.gov/biblio/10389701.

Needell, D., Srebro, N., and Ward, R. Stochastic gradient descent, weighted sampling, and the randomized kaczmarz algorithm. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'14, pp. 1017–1025, Cambridge, MA, USA, 2014. MIT Press.

Nilsback, M.-E. and Zisserman, A. Automated flower classification over a large number of classes. In *2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing*, pp. 722–729, 2008. doi: 10.1109/ICVGIP.2008.47.

Ostapenko, O., Puscas, M. M., Klein, T., Jähnichen, P., and Nabi, M. Learning to remember: A synaptic plasticity driven framework for continual learning. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11313–11321, 2019. URL https://api.semanticscholar.org/CorpusID:102353035.

Panda, A., Isik, B., Qi, X., Koyejo, S., Weissman, T., and Mittal, P. Lottery ticket adaptation: Mitigating destructive interference in llms, 2024. *https://arxiv.org/abs/2406.16797*, 2024.

Parkhi, O. M., Vedaldi, A., and Zisserman, A. Dogs: A dataset for recognising dog breeds from images. In *British Machine Vision Conference (BMVC)*, 2012.

Qi, Q., Guo, Z., Xu, Y., Jin, R., and Yang, T. An online method for a class of distributionally robust optimization with non-convex objectives. *Advances in Neural Information Processing Systems*, 34:10067–10080, 2021.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., and Sutskever, I. Learning transferable visual models from natural language supervision. In Meila, M. and Zhang, T. (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 8748–8763. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/radford21a.html.

Rajasegaran, J., Hayat, M., Khan, S. H., Khan, F. S., and Shao, L. Random path selection for continual learning. *Advances in neural information processing systems*, 32, 2019.

Ramesh, R. and Chaudhari, P. Model zoo: A growing brain that learns continually. In *International Conference on Learning Representations*, 2021. URL https://api.semanticscholar.org/CorpusID:245007201.

Rebuffi, S.-A., Kolesnikov, A., Sperl, G., and Lampert, C. H. icarl: Incremental classifier and representation learning. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5533–5542, 2017. doi: 10.1109/CVPR.2017.587.

Riemer, M., Cases, I., Ajemian, R., Liu, M., Rish, I., Tu, Y., and Tesauro, G. Learning to learn without forgetting by maximizing transfer and minimizing interference. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL https://openreview.net/forum?id=B1gTShAct7.

Ritter, H., Botev, A., and Barber, D. Online structured laplace approximations for overcoming catastrophic forgetting. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, pp. 3742–3752, Red Hook, NY, USA, 2018. Curran Associates Inc.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115: 211–252, 2015.

Rusu, A. A., Rabinowitz, N. C., Desjardins, G., Soyer, H., Kirkpatrick, J., Kavukcuoglu, K., Pascanu, R., and Hadsell, R. Progressive neural networks. *ArXiv*, abs/1606.04671, 2016. URL https://api.semanticscholar.org/CorpusID:15350923.

Sakaguchi, K., Bras, R. L., Bhagavatula, C., and Choi, Y. Winogrande: An adversarial winograd schema challenge at scale. *arXiv preprint arXiv:1907.10641*, 2019.

Sap, M., Rashkin, H., Chen, D., Le Bras, R., and Choi, Y. Social iqa: Commonsense reasoning about social interactions. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 4463–4473, 2019.

Schwarz, J., Czarnecki, W., Luketina, J., Grabska-Barwinska, A., Teh, Y. W., Pascanu, R., and Hadsell, R. Progress & compress: A scalable framework for continual learning. In Dy, J. and Krause, A. (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 4528–4537. PMLR, 10–15 Jul 2018. URL https://proceedings.mlr.press/v80/schwarz18a.html.

Serra, J., Suris, D., Miron, M., and Karatzoglou, A. Overcoming catastrophic forgetting with hard attention to the task. In Dy, J. and Krause, A. (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 4548–4557. PMLR, 10–15 Jul 2018. URL https://proceedings.mlr.press/v80/serra18a.html.

Shrivastava, A., Gupta, A., and Girshick, R. Training region-based object detectors with online hard example mining. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 761–769, 2016. doi: 10.1109/CVPR.2016.89.

Silver, D. L. and Mercer, R. E. The task rehearsal method of life-long learning: Overcoming impoverished data. In *Proceedings of the 15th Conference of the Canadian Society for Computational Studies of Intelligence on Advances in Artificial Intelligence*, AI '02, pp. 90–101, Berlin, Heidelberg, 2002. Springer-Verlag. ISBN 354043724X.

Stich, S. U., Raj, A., and Jaggi, M. Safe adaptive importance sampling. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, pp. 4384–4394, Red Hook, NY, USA, 2017. Curran Associates Inc. ISBN 9781510860964.

Team, G., Riviere, M., Pathak, S., Sessa, P. G., Hardin, C., Bhupatiraju, S., Hussenot, L., Mesnard, T., Shahriari, B., Ramé, A., Ferret, J., Liu, P., Tafti, P., Friesen, A., Casbon, M., Ramos, S., Kumar, R., Lan, C. L., Jerome, S., Tsitsulin, A., Vieillard, N., Stanczyk, P., Girgin, S., Momchev, N., Hoffman, M., Thakoor, S., Grill, J.-B., Neyshabur, B., Bachem, O., Walton, A., Severyn, A., Parrish, A., Ahmad, A., Hutchison, A., Abdagic, A., Carl, A., Shen, A., Brock, A., Coenen, A., Laforge, A., Paterson, A., Bastian, B., Piot, B., Wu, B., Royal, B., Chen, C., Kumar, C., Perry, C., Welty, C., Choquette-Choo, C. A., Sinopalnikov, D., Weinberger, D., Vijaykumar, D., Rogozińska, D., Herbison, D., Bandy, E., Wang, E., Noland, E., Moreira, E., Senter, E., Eltyshev, E., Visin, F., Rasskin, G., Wei, G., Cameron, G., Martins, G., Hashemi, H., Klimczak-Plucińska, H., Batra, H., Dhand, H., Nardini, I., Mein, J., Zhou, J., Svensson, J., Stanway, J., Chan, J., Zhou, J. P., Carrasqueira, J., Iljazi, J., Becker, J., Fernandez, J., van Amersfoort, J., Gordon, J., Lipschultz, J., Newlan, J., yeong Ji, J., Mohamed, K., Badola, K., Black, K., Millican, K., McDonell, K., Nguyen, K., Sodhia, K., Greene, K., Sjoesund, L. L., Usui, L., Sifre, L., Heuermann, L., Lago, L., McNealus, L., Soares, L. B., Kilpatrick, L.,

Dixon, L., Martins, L., Reid, M., Singh, M., Iverson, M., Görner, M., Velloso, M., Wirth, M., Davidow, M., Miller, M., Rahtz, M., Watson, M., Risdal, M., Kazemi, M., Moynihan, M., Zhang, M., Kahng, M., Park, M., Rahman, M., Khatwani, M., Dao, N., Bardoliwalla, N., Devanathan, N., Dumai, N., Chauhan, N., Wahltinez, O., Botarda, P., Barnes, P., Barham, P., Michel, P., Jin, P., Georgiev, P., Culliton, P., Kuppala, P., Comanescu, R., Merhej, R., Jana, R., Rokni, R. A., Agarwal, R., Mullins, R., Saadat, S., Carthy, S. M., Cogan, S., Perrin, S., Arnold, S. M. R., Krause, S., Dai, S., Garg, S., Sheth, S., Ronstrom, S., Chan, S., Jordan, T., Yu, T., Eccles, T., Hennigan, T., Kocisky, T., Doshi, T., Jain, V., Yadav, V., Meshram, V., Dharmadhikari, V., Barkley, W., Wei, W., Ye, W., Han, W., Kwon, W., Xu, X., Shen, Z., Gong, Z., Wei, Z., Cotruta, V., Kirk, P., Rao, A., Giang, M., Peran, L., Warkentin, T., Collins, E., Barral, J., Ghahramani, Z., Hadsell, R., Sculley, D., Banks, J., Dragan, A., Petrov, S., Vinyals, O., Dean, J., Hassabis, D., Kavukcuoglu, K., Farabet, C., Buchatskaya, E., Borgeaud, S., Fiedel, N., Joulin, A., Kenealy, K., Dadashi, R., and Andreev, A. Gemma 2: Improving open language models at a practical size, 2024. URL https://arxiv.org/abs/2408.00118.

Tiwari, R., Killamsetty, K., Iyer, R., and Shenoy, P. Gcr: Gradient coreset based replay buffer selection for continual learning. *CVPR*, 11 2021. doi: 10.48550/arXiv.2111.11210.

Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., Rodriguez, A., Joulin, A., Grave, E., and Lample, G. Llama: Open and efficient foundation language models, 2023a. URL https://arxiv.org/abs/2302.13971.

Touvron, H., Martin, L., Stone, K. R., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., Bikel, D. M., Blecher, L., Ferrer, C. C., Chen, M., Cucurull, G., Esiobu, D., Fernandes, J., Fu, J., Fu, W., Fuller, B., Gao, C., Goswami, V., Goyal, N., Hartshorn, A. S., Hosseini, S., Hou, R., Inan, H., Kardas, M., Kerkez, V., Khabsa, M., Kloumann, I. M., Korenev, A. V., Koura, P. S., Lachaux, M.-A., Lavril, T., Lee, J., Liskovich, D., Lu, Y., Mao, Y., Martinet, X., Mihaylov, T., Mishra, P., Molybog, I., Nie, Y., Poulton, A., Reizenstein, J., Rungta, R., Saladi, K., Schelten, A., Silva, R., Smith, E. M., Subramanian, R., Tan, X., Tang, B., Taylor, R., Williams, A., Kuan, J. X., Xu, P., Yan, Z., Zarov, I., Zhang, Y., Fan, A., Kambadur, M. H. M., Narang, S., Rodriguez, A., Stojnic, R., Edunov, S., and Scialom, T. Llama 2: Open foundation and fine-tuned chat models. *ArXiv*, abs/2307.09288, 2023b. URL https://api.semanticscholar.org/CorpusID:259950998.

Triki, A., Aljundi, R., Blaschko, M. B., and Tuytelaars, T. Encoder based lifelong learning. *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 1329–1337, 2017. URL https://api.semanticscholar.org/CorpusID:12253672.

Wang, L., Zhang, X., Li, Q., Zhu, J., and Zhong, Y. Coscl: Cooperation of small continual learners is stronger than a big one. In *Computer Vision – ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXVI*, pp. 254–271, Berlin, Heidelberg, 2022a. Springer-Verlag. ISBN 978-3-031-19808-3. doi: 10.1007/978-3-031-19809-0_15. URL https://doi.org/10.1007/978-3-031-19809-0_15.

Wang, L., Zhang, X., Yang, K., Yu, L., Li, C., HONG, L., Zhang, S., Li, Z., Zhong, Y., and Zhu, J. Memory replay with data compression for continual learning. In *International Conference on Learning Representations*, 2022b. URL https://openreview.net/forum?id=a7H7OucbWaU.

Wang, L., Zhang, X., Li, Q., Zhang, M., Su, H., Zhu, J., and Zhong, Y. Incorporating neuro-inspired adaptability for continual learning in artificial intelligence. *Nature Machine Intelligence*, 5:1–13, 11 2023a. doi: 10.1038/s42256-023-00747-w.

Wang, S., Li, X., Sun, J., and Xu, Z. Training networks in null space of feature covariance for continual learning. In *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*, pp. 184–193, 2021.

Wang, X., Chen, T., Ge, Q., Xia, H., Bao, R., Zheng, R., Zhang, Q., Gui, T., and Huang, X. Orthogonal subspace learning for language model continual learning. In Bouamor, H., Pino, J., and Bali, K. (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 10658–10671, Singapore, December 2023b. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-emnlp.715. URL https://aclanthology.org/2023.findings-emnlp.715/.

Wightman, R., Touvron, H., and Jégou, H. Resnet strikes back: An improved training procedure in timm. *NEURIPS*, workshop.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., von Platen, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Scao, T. L., Gugger, S., Drame, M., Lhoest, Q., and Rush, A. M. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp.

38–45, Online, October 2020. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/2020.emnlp-demos.6.

Wortsman, M., Ramanujan, V., Liu, R., Kembhavi, A., Rastegari, M., Yosinski, J., and Farhadi, A. Supermasks in superposition. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA, 2020. Curran Associates Inc. ISBN 9781713829546.

Wortsman, M., Ilharco, G., Kim, J. W., Li, M., Kornblith, S., Roelofs, R., Gontijo-Lopes, R., Hajishirzi, H., Farhadi, A., Namkoong, H., and Schmidt, L. Robust fine-tuning of zero-shot models. *arXiv preprint arXiv:2109.01903*, 2021. https://arxiv.org/abs/2109.01903.

Wu, Y., Chen, Y., Wang, L., Ye, Y., Liu, Z., Guo, Y., Zhang, Z., and Fu, Y. R. Incremental classifier learning with generative adversarial networks. *ArXiv*, abs/1802.00853, 2018. URL https://api.semanticscholar.org/CorpusID:3652214.

Xie, S. M., Pham, H., Dong, X., Du, N., Liu, H., Lu, Y., Liang, P. S., Le, Q. V., Ma, T., and Yu, A. W. Doremi: Optimizing data mixtures speeds up language model pretraining. *Advances in Neural Information Processing Systems*, 36, 2024.

Yoon, J., Yang, E., Lee, J., and Hwang, S. J. Lifelong learning with dynamically expandable networks. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=Sk7KsfW0-.

Yu, L., Jiang, W., Shi, H., Yu, J., Liu, Z., Zhang, Y., Kwok, J. T., Li, Z., Weller, A., and Liu, W. Metamath: Bootstrap your own mathematical questions for large language models. *arXiv preprint arXiv:2309.12284*, 2023.

Zellers, R., Holtzman, A., Bisk, Y., Farhadi, A., and Choi, Y. Hellaswag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019.

Zeng, G., Chen, Y., Cui, B., and Yu, S. Continual learning of context-dependent processing in neural networks. *Nature Machine Intelligence*, 1:364–372, 08 2019. doi: 10.1038/s42256-019-0080-x.

Zenke, F., Poole, B., and Ganguli, S. Continual learning through synaptic intelligence. In Precup, D. and Teh, Y. W. (eds.), *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 3987–3995. PMLR, 06–11 Aug 2017. URL https://proceedings.mlr.press/v70/zenke17a.html.

Zhao, P. and Zhang, T. Stochastic optimization with importance sampling for regularized loss minimization. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, pp. 1–9. JMLR.org, 2015.

# Appendix

## Table of Contents

## A. Extended Related Work

The majority of the approaches for mitigating forgetting assume task-specific knowledge access to different extents; either (a subset of) the pre-training dataset itself or some information/statistic computed from pre-training data. Below, we describe the data-aware approaches based on how they make use of task-specific knowledge.

**Regularization-based methods.** This line of work aims to preserve performance on previously learned tasks by keeping the (fine-tuned) model parameters close to the pre-trained model. The key idea is to introduce task-specific regularization in the fine-tuning phase which will penalize updates along the "important" directions for the old tasks (Ahn et al., 2019). Kirkpatrick et al. (2016) introduces the elastic weight consolidation (EWC) algorithm, which estimates the important direction per-task by calculating a diagonal approximation to the Fisher information matrix (FIM), which acts as the weight matrix for the regularization term. Several variants of EWC have been subsequently proposed (Schwarz et al., 2018; Ritter et al., 2018; Lee et al., 2020; Liu et al., 2018). Zenke et al. (2017); Aljundi et al. (2018) adopt online strategies to infer the importance of each parameter by their variational effect on the model outputs. In a spirit similar to EWC, Lee et al. (2017) incrementally matches the posterior of the pre-trained model and the new task by assuming Gaussian posteriors.

**Optimization-driven methods.** Another perspective to mitigating forgetting is guiding the optimization process by constraining the algorithms directly as opposed to manipulating the loss function. The core idea is to keep track of "important directions" for the old tasks, and train on the new task "orthogonally." This could be done by storing prior data samples or gradients in a buffer (Lopez-Paz & Ranzato, 2017; Farajtabar et al., 2020; Chaudhry et al., 2019a) or by incrementally expanding the subspace of important directions without storing task-specific information (Zeng et al., 2019; Wang et al., 2021; 2023b).

**Replay-based methods.** Drawing inspiration from the complementary learning systems theory (Mcclelland et al., 1995), a more direct approach is to introduce samples from old tasks into the training process for the new task. Samples are selected in a streaming fashion or by manually crafting a subset on demand, stored in dedicated buffers and replayed during the fine-tuning. The intuition is that the task-specific representations are refreshed periodically through historical data.

Replay-based methods consist of two fundamental components: data selection and data reiteration mechanisms. When the data is received in a streaming fashion, information has to be buffered online (Riemer et al., 2019; Chaudhry et al., 2019b; Isele & Cosgun, 2018; De Lange & Tuytelaars, 2021). In the case when datasets are available on demand, Rebuffi et al. (2017) selects samples which are "representative" of their respective class, while others focus on inducing diversity (Aljundi et al., 2019; Bang et al., 2021) and balance (Borsos et al., 2020; Tiwari et al., 2021) across buffered data. For the scenarios in which storage is limited, Caccia et al. (2020); Wang et al. (2022b) develop compression methods for buffered data. As a complimentary component to the data selection process, how the buffered data is replayed plays a significant role in the success of such methods. A fundamental idea, which has several interpretations across the board, is knowledge distillation (Hinton et al., 2015). Prior work argues that augmenting fine-tuning with knowledge distillation shows great performance on the forgetting front (Lopez-Paz & Ranzato, 2017; Chaudhry et al., 2019a; Rebuffi et al., 2017; Jung et al., 2017; Triki et al., 2017; Li & Hoiem, 2016; Lee et al., 2019b; Dhar et al., 2019).

An orthogonal research direction focuses on maintaining a generative model that could reliably output pseudo-samples that are representative of the dataset of the old tasks (Kemker & Kanan, 2018; Wu et al., 2018). Note that generative approaches are prone to scalability issues and distribution shifts.

**Architecture-driven methods.** Another technique to limit the interference between tasks is allocating a separate trainable set of parameters per task. This could be done by initializing a sub-networks per new task (Rusu et al., 2016; Aljundi et al., 2017; Collier et al., 2020; Rajasegaran et al., 2019; Ramesh & Chaudhari, 2021; Wang et al., 2023a; 2022a), gradually expanding the parameters of a base network (Yoon et al., 2018; Ostapenko et al., 2019; Hung et al., 2019), or segregating a fixed model into task-specific subset of parameters (Mallya et al., 2018; Kang et al., 2022; Serra et al., 2018; Wortsman et al., 2020; Mallya & Lazebnik, 2017; Mustafa B Gurbuz, 2022; Jung et al., 2020). While some parameters are task-specific, parts of the overall model could be shared to enable knowledge transfer. The main downside associated is that the task identity must be available during inference to (de)activate relevant sub-networks, hindering versatility. Aljundi et al. (2017) develop dedicated strategies to overcome the need for task identification by automatizing task-specific parameter activation.

FLOW **and its connection to other ML applications.** Our approach has connections to factuality in LLM training (Ghosal et al., 2024; Gekhman et al., 2024). For a fine-tuning task on a factual knowledge-based dataset, samples that are not properly represented in the pre-training distribution could be considered hard. In the data-oblivious setting, one principled way would be to rank *hardness* based on pre-training loss values; therefore, FLOW would identify such samples and guide

the fine-tuning process such that those samples will not unexpectedly deteriorate the performance.

Although fundamentally different in its objective, machine unlearning is another avenue of application for our sample weighting approach which could be interpreted as a means of *soft unlearning*. While unlearning and forgetting sound similar, they are not the same in spirit; in unlearning we deliberately induce "forgetting" on some samples, whereas in our context, forgetting is an undesirable side effect that we want to avoid. Hence, it might not be straightforward extend techniques of one into another, one could extend our sample-wise weighting scheme with appropriate modifications to help selectively unlearn specific samples.

## B. Our Algorithm in the Presence of Task-Specific Model Components

Suppose our model is parameterized by $\boldsymbol{\theta} = \mathbf{U} \cup \mathbf{V}$, where $\mathbf{U}$ is the common/shared part of the model for all tasks (i.e., this part remains the same for all tasks), and $\mathbf{V}$ is the task-specific part of the model. In particular, in our vision experiments, the models have task-specific prediction heads (i.e., softmax layers) and batch-norm (BN). The modified version of Algorithm 1 in the presence of task-specific components is stated in Algorithm 2. The main differences from Algorithm 1 are steps (i) and (iv) – these steps optimize the task-specific part for the new task with uniform weighting. It is worth mentioning that if our model consists of task-specific prediction heads – which is the case in our vision experiments – then steps (i) and (iv) are just vanilla linear probing with the pre-trained body and the body learned after fine-tuning, respectively.

**Algorithm 2** **F**ine-tuning with Pre-trained **L**oss-**O**riented **W**eighting (**FLOW**)

**Input:** Pre-trained model $\boldsymbol{\theta}_*^{(1)} = \mathbf{U}_*^{(1)} \cup \mathbf{V}_*^{(1)}$, dataset $\{(\mathbf{x}_i, \mathrm{y}_i)\}_{i=1}^n$ for the new task, and temperature parameter $\tau$.

$f_i(\mathbf{U}, \mathbf{V}) \to i^{\text{th}}$ sample's loss at $\boldsymbol{\theta} = \mathbf{U} \cup \mathbf{V}$, with a non-negative loss function (e.g., cross-entropy loss).

**Step (i)** Fine-tune task-specific part for new task with vanilla unweighted loss: $\mathbf{V}_*^{(2)} := \text{argmin}_{\mathbf{V}} \sum_{i=1}^n f_i(\mathbf{U}_*^{(1)}, \mathbf{V})$.

**Step (ii)** Compute sample weights: $w_i = \exp\Big(-f_i(\mathbf{U}_*^{(1)}, \mathbf{V}_*^{(2)})/\tau\Big)$.

**Step (iii)** Fine-tune full model **with weighted loss**: $\overline{\mathbf{U}}_*^{(2)}, \overline{\mathbf{V}}_*^{(2)} := \text{argmin}_{\mathbf{U},\mathbf{V}} \sum_{i=1}^n w_i f_i(\mathbf{U}, \mathbf{V})$.

**Step (iv)** Fine-tune task-specific part for new task *using the learned common part* with vanilla unweighted loss: $\widehat{\mathbf{V}}_*^{(2)} := \text{argmin}_{\mathbf{V}} \sum_{i=1}^n f_i(\overline{\mathbf{U}}_*^{(2)}, \mathbf{V})$.

**Output:** New model for

- Original/pre-training task is $\widehat{\boldsymbol{\theta}}_*^{(1)} = \overline{\mathbf{U}}_*^{(2)} \cup \mathbf{V}_*^{(1)}$.
- New/fine-tuning task is $\widehat{\boldsymbol{\theta}}_*^{(2)} = \overline{\mathbf{U}}_*^{(2)} \cup \widehat{\mathbf{V}}_*^{(2)}$.

**Remark B.1.** *In all our vision experiments (with task-specific parts), we set* $\tau = \text{median}(f_i(\mathbf{U}_*^{(1)}, \mathbf{V}_*^{(2)}))$ *(similar to Remark 4.2).*

## C. Proof of Proposition 4.3

*Proof.* We wish to minimize $g(\boldsymbol{\pi}) = \sum_{i=1}^n \pi_i f_i(\boldsymbol{\theta}^*) + \tau \sum_{i=1}^n \pi_i \log \pi_i$ subject to $\sum_{i=1}^n \pi_i = 1$ and $\pi_i \geq 0$ for all $i \in [n]$. The proof is a straightforward application of Lagrangian multipliers. It is enough to enforce $\sum_{i=1}^n \pi_i = 1$ only ($\pi_i \geq 0$ for all $i \in [n]$ will also follow). For that, the Lagrangian function is:

$$J(\boldsymbol{\pi}, \lambda) = \sum_{i=1}^n \pi_i f_i(\boldsymbol{\theta}^*) + \tau \sum_{i=1}^n \pi_i \log \pi_i + \lambda\Big(\sum_{i=1}^n \pi_i - 1\Big), \tag{19}$$

where $\lambda$ is the Lagrangian multiplier. Now, at the optimal point $\boldsymbol{\pi}^* = [\pi_1^*, \ldots, \pi_n^*]^\top$, we must have:

$$\left.\frac{\partial J}{\partial \pi_i}\right|_{\pi_i^*} = f_i(\boldsymbol{\theta}^*) + \tau\,(1 + \log \pi_i^*) + \lambda = 0, \tag{20}$$

for all $i \in [n]$. Simplifying, we get:

$$\pi_i^* = \frac{1}{Z} \exp\left(-\frac{f_i(\boldsymbol{\theta}^*)}{\tau}\right), \tag{21}$$

where $Z = \exp\left(\left(1 + \frac{\lambda}{\tau}\right)\right)$ is the normalizing constant. To have $\sum_{i=1}^n \pi_i = 1$, we get $Z = \sum_{j=1}^n \exp\left(-\frac{f_j(\mathbf{W}^*)}{\tau}\right)$. Also, note that we are good with the non-negativity constraints. □

## D. Proof of Theorem 7.2

*Proof.* Note that:

$$\widehat{\text{err}}_2(\widehat{\boldsymbol{\theta}}) = \left(\widehat{\boldsymbol{\theta}} - \widetilde{\boldsymbol{\theta}}_*\right)^\top \mathbb{E}_{\widetilde{\mathcal{D}}}\left[w(\widetilde{\mathbf{x}}, \widetilde{\mathbf{y}})\widetilde{\mathbf{x}}\widetilde{\mathbf{x}}^\top\right]\left(\widehat{\boldsymbol{\theta}} - \widetilde{\boldsymbol{\theta}}_*\right). \tag{22}$$

Also, after plugging in $\widetilde{\mathbf{y}} = \left\langle \widetilde{\boldsymbol{\theta}}_*, \widetilde{\mathbf{x}} \right\rangle$, we get:

$$w(\widetilde{\mathbf{x}}, \widetilde{\mathbf{y}}) = \exp\left(-\frac{\left(\langle \boldsymbol{\theta}_* - \widetilde{\boldsymbol{\theta}}_*, \widetilde{\mathbf{x}} \rangle\right)^2}{\tau}\right).$$

Recall $\mathbf{e} := \boldsymbol{\theta}_* - \widetilde{\boldsymbol{\theta}}_*$ and $\overline{\mathbf{e}} := \frac{\mathbf{e}}{\|\mathbf{e}\|_2}$. Suppose $\tau = \alpha\|\mathbf{e}\|_2^2$, for some $\alpha > 0$. Then $w(\widetilde{\mathbf{x}}, \widetilde{\mathbf{y}}) = \exp\left(-\frac{\left(\langle \overline{\mathbf{e}}, \widetilde{\mathbf{x}} \rangle\right)^2}{\alpha}\right)$, and we can focus on

$$\widetilde{\boldsymbol{\Sigma}}' := \mathbb{E}_{\widetilde{\mathbf{x}} \sim \widetilde{\mathcal{P}}}\left[\exp\left(-\frac{\left(\langle \overline{\mathbf{e}}, \widetilde{\mathbf{x}} \rangle\right)^2}{\alpha}\right)\widetilde{\mathbf{x}}\widetilde{\mathbf{x}}^\top\right]. \tag{23}$$

Let $\mu = \left(\frac{\alpha}{\alpha+2}\right)^{1/2} = \left(\frac{\tau}{\tau + 2\|\mathbf{e}\|_2^2}\right)^{1/2}$. As per Lemma F.1, we have:

$$\widetilde{\boldsymbol{\Sigma}}' = \mu(\mathbf{I}_d - \mathbf{Q}), \tag{24}$$

where

$$\mathbf{Q} = (1-\mu^2)\overline{\mathbf{e}}\overline{\mathbf{e}}^\top + \rho^2(1-\mu^2)\overline{\mathbf{e}}_\perp\overline{\mathbf{e}}_\perp^\top - \rho\mu^2\left(\overline{\mathbf{e}}\overline{\mathbf{e}}_\perp^\top + \overline{\mathbf{e}}_\perp\overline{\mathbf{e}}^\top\right). \tag{25}$$

So if we minimize $\widehat{\text{err}}_2(\widehat{\boldsymbol{\theta}})$ with GD starting from $\widehat{\boldsymbol{\theta}}_0 = \boldsymbol{\theta}_*$ and using a constant learning rate $\widehat{\eta}$, our iterate $\widehat{\boldsymbol{\theta}}_K$ at the $K^{\text{th}}$ iteration satisfies:

$$\widehat{\boldsymbol{\theta}}_K - \widetilde{\boldsymbol{\theta}}_* = \left(\mathbf{I}_d - 2\widehat{\eta}\widetilde{\boldsymbol{\Sigma}}'\right)^K\left(\boldsymbol{\theta}_* - \widetilde{\boldsymbol{\theta}}_*\right) = \left(\mathbf{I}_d - 2\widehat{\eta}\widetilde{\boldsymbol{\Sigma}}'\right)^K \mathbf{e}, \tag{26}$$

where the last step follows by recalling that $\boldsymbol{\theta}_* - \widetilde{\boldsymbol{\theta}}_* = \mathbf{e}$, and $\widetilde{\boldsymbol{\Sigma}}'$ is given by Equation (24). □

## E. Difficulty in the Analysis with a General Covariance Matrix $\widetilde{\boldsymbol{\Sigma}}$

We will first derive the *weighted* (fine-tuning) data covariance matrix $\widetilde{\boldsymbol{\Sigma}}'$ in the context of Theorem 7.2 for a general (fine-tuning) data covariance matrix $\widetilde{\boldsymbol{\Sigma}}$. Specifically, following the proof of Theorem 7.2, we have:

$$\widetilde{\boldsymbol{\Sigma}}' := \mathbb{E}_{\widetilde{\mathbf{x}} \sim \mathcal{N}(\vec{0}_d, \widetilde{\boldsymbol{\Sigma}})}\left[\exp\left(-\frac{\left(\langle \mathbf{e}, \widetilde{\mathbf{x}} \rangle\right)^2}{\tau}\right)\widetilde{\mathbf{x}}\widetilde{\mathbf{x}}^\top\right]. \tag{27}$$

Note that $\widetilde{\mathbf{x}} = \widetilde{\boldsymbol{\Sigma}}^{1/2}\mathbf{z}$, where $\mathbf{z} \sim \mathcal{N}(\vec{0}_d, \mathbf{I}_d)$. Using this above, we get:

$$\widetilde{\boldsymbol{\Sigma}}' = \widetilde{\boldsymbol{\Sigma}}^{1/2}\mathbb{E}\left[\exp\left(-\frac{\left(\langle \mathbf{e}, \widetilde{\boldsymbol{\Sigma}}^{1/2}\mathbf{z} \rangle\right)^2}{\tau}\right)\mathbf{z}\mathbf{z}^\top\right]\widetilde{\boldsymbol{\Sigma}}^{1/2} = \widetilde{\boldsymbol{\Sigma}}^{1/2}\mathbb{E}\left[\exp\left(-\frac{\left(\langle \widetilde{\boldsymbol{\Sigma}}^{1/2}\mathbf{e}, \mathbf{z} \rangle\right)^2}{\tau}\right)\mathbf{z}\mathbf{z}^\top\right]\widetilde{\boldsymbol{\Sigma}}^{1/2}, \tag{28}$$

where the last step follows by using the symmetry of $\widetilde{\boldsymbol{\Sigma}}$. Let $\tau = \alpha\left\|\widetilde{\boldsymbol{\Sigma}}^{1/2}\mathbf{e}\right\|_2^2$ for some $\alpha > 0$. Also, let $\mathbf{r} := (\widetilde{\boldsymbol{\Sigma}}^{1/2}\mathbf{e})/\|\widetilde{\boldsymbol{\Sigma}}^{1/2}\mathbf{e}\|_2$. In that case, we have:

$$\widetilde{\boldsymbol{\Sigma}}' = \widetilde{\boldsymbol{\Sigma}}^{1/2}\mathbf{M}\widetilde{\boldsymbol{\Sigma}}^{1/2}, \text{ where } \mathbf{M} := \mathbb{E}\left[\exp\left(-\frac{\left(\langle \mathbf{r}, \mathbf{z} \rangle\right)^2}{\alpha}\right)\mathbf{z}\mathbf{z}^\top\right]. \tag{29}$$

Suppose $\{\mathbf{r}_{\perp,j}\}_{j=1}^{d-1}$ is an orthonormal basis for the subspace of $\mathbb{R}^d$ orthogonal to $\mathbf{r}$; so $\langle \mathbf{r}_{\perp,j}, \mathbf{r}\rangle = 0 \; \forall \; j \in [d-1]$ and $\langle \mathbf{r}_{\perp,j}, \mathbf{r}_{\perp,k}\rangle = \mathbb{1}(j=k) \; \forall \; j,k \in [d-1]$. Note that $\{\mathbf{r}, \mathbf{r}_{\perp,1}, \ldots, \mathbf{r}_{\perp,d-1}\}$ forms an orthonormal basis for $\mathbb{R}^d$. Then, as per Lemma F.5, we have that $\mathbf{r}$ is an eigenvector of $\mathbf{M}$ with eigenvalue $\left(\frac{\alpha}{\alpha+2}\right)^{3/2}$, and each $\mathbf{r}_{\perp,j}$ is an eigenvector of $\mathbf{M}$ with eigenvalue $\left(\frac{\alpha}{\alpha+2}\right)^{1/2}$. For brevity, let $\mu = \left(\frac{\alpha}{\alpha+2}\right)^{1/2}$. Then, we can write:

$$\mathbf{M} = \mu^3 \mathbf{r}\mathbf{r}^\top + \mu \sum_{j=1}^{d-1} \mathbf{r}_{\perp,j}\mathbf{r}_{\perp,j}^\top = \mu^3 \mathbf{r}\mathbf{r}^\top + \mu\Big(\mathbf{I}_d - \mathbf{r}\mathbf{r}^\top\Big), \tag{30}$$

where the last step follows because $\{\mathbf{r}, \mathbf{r}_{\perp,1}, \ldots, \mathbf{r}_{\perp,d-1}\}$ forms an orthonormal basis for $\mathbb{R}^d$, due to which $\mathbf{r}\mathbf{r}^\top + \sum_{j=1}^{d-1} \mathbf{r}_{\perp,j}\mathbf{r}_{\perp,j}^\top = \mathbf{I}_d$. Simplifying Equation (30) a bit, we get:

$$\mathbf{M} = \mu\Big(\mathbf{I}_d - (1-\mu^2)\mathbf{r}\mathbf{r}^\top\Big). \tag{31}$$

Plugging this into Equation (29) and recalling that $\mathbf{r} := (\widetilde{\mathbf{\Sigma}}^{1/2}\mathbf{e})/\|\widetilde{\mathbf{\Sigma}}^{1/2}\mathbf{e}\|_2$, we get:

$$\widetilde{\mathbf{\Sigma}}' = \mu\mathbf{B}, \text{ where } \mathbf{B} := \left(\widetilde{\mathbf{\Sigma}} - (1-\mu^2)\frac{\widetilde{\mathbf{\Sigma}}\mathbf{e}\mathbf{e}^\top\widetilde{\mathbf{\Sigma}}}{\mathbf{e}^\top\widetilde{\mathbf{\Sigma}}\mathbf{e}}\right). \tag{32}$$

Equation (32) is the weighted covariance matrix for a general $\widetilde{\mathbf{\Sigma}}$.

**Remark E.1 (Difficulty with general $\widetilde{\mathbf{\Sigma}}$).** *It is hard to proceed with the analysis after this point because it is difficult to characterize the eigen-spectrum of $\mathbf{B}$ in general, without assuming any relation between $\widetilde{\mathbf{\Sigma}}$ and $\mathbf{e}$. This is what we meant in Remark 7.1.*

## F. Lemmas Used and Their Proofs

**Lemma F.1.** *In the proof of Theorem 7.2, recall that $\tau = \alpha\|\mathbf{e}\|_2^2$. Then, we have:*

$$\widetilde{\mathbf{\Sigma}}' := \mathbb{E}_{\widetilde{\mathbf{x}}\sim\widetilde{\mathcal{P}}}\left[\exp\left(-\frac{\big(\langle\overline{\mathbf{e}}, \widetilde{\mathbf{x}}\rangle\big)^2}{\alpha}\right)\widetilde{\mathbf{x}}\widetilde{\mathbf{x}}^\top\right] = \mu\Big(\mathbf{I}_d - (1-\mu^2)\overline{\mathbf{e}}\overline{\mathbf{e}}^\top - \rho^2(1-\mu^2)\overline{\mathbf{e}}_\perp\overline{\mathbf{e}}_\perp^\top + \rho\mu^2\big(\overline{\mathbf{e}}\overline{\mathbf{e}}_\perp^\top + \overline{\mathbf{e}}_\perp\overline{\mathbf{e}}^\top\big)\Big),$$

*where* $\mu = \left(\frac{\alpha}{\alpha+2}\right)^{1/2} = \left(\frac{\tau}{\tau+2\|\mathbf{e}\|_2^2}\right)^{1/2}$.

*Proof.* Recall that $\overline{\mathbf{e}}$ and $\overline{\mathbf{e}}_\perp$ are orthogonal to each other and both are unit-norm. Suppose $\{\overline{\mathbf{e}}_{\perp,3}, \overline{\mathbf{e}}_{\perp,4}, \ldots, \overline{\mathbf{e}}_{\perp,d}\}$ is an orthonormal basis for the $(d-2)$-dimensional subspace of $\mathbb{R}^d$ orthogonal to $\overline{\mathbf{e}}$ and $\overline{\mathbf{e}}_\perp$. Thus, $\{\overline{\mathbf{e}}, \overline{\mathbf{e}}_\perp, \overline{\mathbf{e}}_{\perp,3}, \overline{\mathbf{e}}_{\perp,4}, \ldots, \overline{\mathbf{e}}_{\perp,d}\}$ is an orthonormal basis for $\mathbb{R}^d$. Then using Lemma F.2, we can write:

$$\widetilde{\mathbf{x}} = \mathrm{z}_1\overline{\mathbf{e}} + \Big(\rho\mathrm{z}_1 + \sqrt{1-\rho^2}\mathrm{z}_2\Big)\overline{\mathbf{e}}_\perp + \sum_{j=3}^{d}\mathrm{z}_j\overline{\mathbf{e}}_{\perp,j}, \tag{33}$$

where $\{\mathrm{z}_j\}_{j=1}^d \underset{\text{iid}}{\sim} \mathcal{N}(0,1)$.

Using independence and zero-mean nature of $\{\mathrm{z}_j\}_{j=1}^d$, we get:

$$\begin{aligned}\widetilde{\mathbf{\Sigma}}' = \underbrace{\mathbb{E}\Big[\exp\Big(-\frac{\mathrm{z}_1^2}{\alpha}\Big)\mathrm{z}_1^2\Big]}_{:=\mathrm{T}_1}\overline{\mathbf{e}}\overline{\mathbf{e}}^\top + \underbrace{\mathbb{E}\Big[\exp\Big(-\frac{\mathrm{z}_1^2}{\alpha}\Big)\mathrm{z}_1\Big(\rho\mathrm{z}_1 + \sqrt{1-\rho^2}\mathrm{z}_2\Big)\Big]}_{:=\mathrm{T}_2}\big(\overline{\mathbf{e}}\overline{\mathbf{e}}_\perp^\top + \overline{\mathbf{e}}_\perp\overline{\mathbf{e}}^\top\big) \\ + \underbrace{\mathbb{E}\Big[\exp\Big(-\frac{\mathrm{z}_1^2}{\alpha}\Big)\Big(\rho\mathrm{z}_1 + \sqrt{1-\rho^2}\mathrm{z}_2\Big)^2\Big]}_{:=\mathrm{T}_3}\overline{\mathbf{e}}_\perp\overline{\mathbf{e}}_\perp^\top + \sum_{j=3}^{d}\underbrace{\mathbb{E}\Big[\exp\Big(-\frac{\mathrm{z}_1^2}{\alpha}\Big)\Big]}_{:=\mathrm{T}_4}\underbrace{\mathbb{E}\big[\mathrm{z}_j^2\big]}_{=1}\overline{\mathbf{e}}_{\perp,j}\overline{\mathbf{e}}_{\perp,j}^\top. \end{aligned} \tag{34}$$

Note that (we use the independence of $\mathrm{z}_1$ and $\mathrm{z}_2$):

$$\mathrm{T}_2 = \rho \mathrm{T}_1 + \sqrt{1-\rho^2}\mathbb{E}\Big[\exp\Big(-\frac{\mathrm{z}_1^2}{\alpha}\Big)\mathrm{z}_1\Big]\underbrace{\mathbb{E}\big[\mathrm{z}_2\big]}_{=0} = \rho \mathrm{T}_1, \tag{35}$$

and

$$\mathrm{T}_3 = \rho^2 \mathrm{T}_1 + 2\rho\sqrt{1-\rho^2}\Big[\exp\Big(-\frac{\mathrm{z}_1^2}{\alpha}\Big)\mathrm{z}_1\Big]\underbrace{\mathbb{E}\big[\mathrm{z}_2\big]}_{=0} + (1-\rho^2)\mathrm{T}_4\underbrace{\mathbb{E}\big[\mathrm{z}_2^2\big]}_{=1} = \rho^2\mathrm{T}_1 + (1-\rho^2)\mathrm{T}_4. \tag{36}$$

In the above two equations, we have again used the independence of $\mathrm{z}_1$ and $\mathrm{z}_2$. Now we will compute $\mathrm{T}_1$ and $\mathrm{T}_4$. We have:

$$\mathrm{T}_1 = \left(\frac{1}{\sqrt{2\pi}}\int_{-\infty}^{\infty}\mathrm{z}_1^2\exp\Big(-\mathrm{z}_1^2\Big(\frac{1}{\alpha}+\frac{1}{2}\Big)\Big)\mathrm{d}\mathrm{z}_1\right) = \Big(\frac{\alpha}{\alpha+2}\Big)^{3/2}, \tag{37}$$

and

$$\mathrm{T}_4 = \left(\frac{1}{\sqrt{2\pi}}\int_{-\infty}^{\infty}\exp\Big(-\mathrm{z}_1^2\Big(\frac{1}{\alpha}+\frac{1}{2}\Big)\Big)\mathrm{d}\mathrm{z}_1\right) = \Big(\frac{\alpha}{\alpha+2}\Big)^{1/2}. \tag{38}$$

Recall that $\mu = \big(\frac{\alpha}{\alpha+2}\big)^{1/2}$. Plugging this into Equations (35) to (38) gives us:

$$\mathrm{T}_1 = \mu^3, \mathrm{T}_2 = \rho\mu^3, \mathrm{T}_3 = \rho^2\mu^3 + (1-\rho^2)\mu, \text{ and } \mathrm{T}_4 = \mu. \tag{39}$$

Plugging this into Equation (34) gives us:

$$\widetilde{\mathbf{\Sigma}}' = \mu^3\overline{\mathbf{e}}\overline{\mathbf{e}}^\top + \rho\mu^3\big(\overline{\mathbf{e}}\overline{\mathbf{e}}_\perp^\top + \overline{\mathbf{e}}_\perp\overline{\mathbf{e}}^\top\big) + \Big(\rho^2\mu^3 + (1-\rho^2)\mu\Big)\overline{\mathbf{e}}_\perp\overline{\mathbf{e}}_\perp^\top + \mu\sum_{j=3}^{d}\overline{\mathbf{e}}_{\perp,j}\overline{\mathbf{e}}_{\perp,j}^\top. \tag{40}$$

Recall that $\{\overline{\mathbf{e}}, \overline{\mathbf{e}}_\perp, \overline{\mathbf{e}}_{\perp,3}, \overline{\mathbf{e}}_{\perp,4}, \ldots, \overline{\mathbf{e}}_{\perp,d}\}$ is an orthonormal basis for $\mathbb{R}^d$. Thus, $\sum_{j=3}^{d}\overline{\mathbf{e}}_{\perp,j}\overline{\mathbf{e}}_{\perp,j}^\top = \mathbf{I}_d - \overline{\mathbf{e}}\overline{\mathbf{e}}^\top - \overline{\mathbf{e}}_\perp\overline{\mathbf{e}}_\perp^\top$. Using this above, we get:

$$\widetilde{\mathbf{\Sigma}}' = \mu\Big(\mathbf{I}_d - (1-\mu^2)\overline{\mathbf{e}}\overline{\mathbf{e}}^\top - \rho^2(1-\mu^2)\overline{\mathbf{e}}_\perp\overline{\mathbf{e}}_\perp^\top + \rho\mu^2\big(\overline{\mathbf{e}}\overline{\mathbf{e}}_\perp^\top + \overline{\mathbf{e}}_\perp\overline{\mathbf{e}}^\top\big)\Big). \tag{41}$$

This finishes the proof. □

**Lemma F.2.** *Suppose $\{\overline{\mathbf{e}}, \overline{\mathbf{e}}_\perp, \overline{\mathbf{e}}_{\perp,3}, \overline{\mathbf{e}}_{\perp,4}, \ldots, \overline{\mathbf{e}}_{\perp,d}\}$ is an orthonormal basis for $\mathbb{R}^d$. If $\widetilde{\mathbf{x}} \sim \mathcal{N}(\vec{\mathbf{0}}_d, \widetilde{\mathbf{\Sigma}})$, then we can write:*

$$\widetilde{\mathbf{x}} = \mathrm{z}_1\overline{\mathbf{e}} + \Big(\rho\mathrm{z}_1 + \sqrt{1-\rho^2}\mathrm{z}_2\Big)\overline{\mathbf{e}}_\perp + \sum_{j=3}^{d}\mathrm{z}_j\overline{\mathbf{e}}_{\perp,j}, \tag{42}$$

*where $\{\mathrm{z}_j\}_{j=1}^d \underset{\text{iid}}{\sim} \mathcal{N}(0,1)$.*

*Proof.* If $\widetilde{\mathbf{x}}$ is as per Equation (42), then clearly $\widetilde{\mathbf{x}}$ is a zero-mean Gaussian. All that remains to show is that

$$\mathbb{E}\Big[\widetilde{\mathbf{x}}\widetilde{\mathbf{x}}^\top\Big] = \widetilde{\mathbf{\Sigma}} = \mathbf{I}_d + \rho\big(\overline{\mathbf{e}}\overline{\mathbf{e}}_\perp^\top + \overline{\mathbf{e}}_\perp\overline{\mathbf{e}}^\top\big).$$

Using independence and zero-mean nature of $\{\mathrm{z}_j\}_{j=1}^d$, we get:

$$\mathbb{E}\Big[\widetilde{\mathbf{x}}\widetilde{\mathbf{x}}^\top\Big] = \underbrace{\mathbb{E}\big[\mathrm{z}_1^2\big]}_{=1}\overline{\mathbf{e}}\overline{\mathbf{e}}^\top + \underbrace{\mathbb{E}\Big[\mathrm{z}_1\Big(\rho\mathrm{z}_1 + \sqrt{1-\rho^2}\mathrm{z}_2\Big)\Big]}_{:=(\mathrm{A})}\big(\overline{\mathbf{e}}\overline{\mathbf{e}}_\perp^\top + \overline{\mathbf{e}}_\perp\overline{\mathbf{e}}^\top\big) + \underbrace{\mathbb{E}\Big[\Big(\rho\mathrm{z}_1 + \sqrt{1-\rho^2}\mathrm{z}_2\Big)^2\Big]}_{:=(\mathrm{B})}\overline{\mathbf{e}}_\perp\overline{\mathbf{e}}_\perp^\top$$
$$+ \sum_{j=3}^{d}\underbrace{\mathbb{E}\big[\mathrm{z}_j^2\big]}_{=1}\overline{\mathbf{e}}_{\perp,j}\overline{\mathbf{e}}_{\perp,j}^\top. \tag{43}$$

Note that (we use the independence of $\mathrm{z}_1$ and $\mathrm{z}_2$):

$$(\mathrm{A}) = \rho \underbrace{\mathbb{E}[\mathrm{z}_1^2]}_{=1} + \sqrt{1-\rho^2}\, \underbrace{\mathbb{E}[\mathrm{z}_1]}_{=0} \underbrace{\mathbb{E}\big[\mathrm{z}_2\big]}_{=0} = \rho, \tag{44}$$

and

$$(\mathrm{B}) = \rho^2 \underbrace{\mathbb{E}\big[\mathrm{z}_1^2\big]}_{=1} + 2\rho\sqrt{1-\rho^2}\, \underbrace{\mathbb{E}\big[\mathrm{z}_1\big]}_{=0} \underbrace{\mathbb{E}\big[\mathrm{z}_2\big]}_{=0} + (1-\rho^2) \underbrace{\mathbb{E}\big[\mathrm{z}_2^2\big]}_{=1} = 1. \tag{45}$$

Plugging this into Equation (43), we get:

$$\mathbb{E}\Big[\widetilde{\mathbf{x}}\widetilde{\mathbf{x}}^\top\Big] = \overline{\mathbf{e}}\overline{\mathbf{e}}^\top + \rho\big(\overline{\mathbf{e}}\overline{\mathbf{e}}_\perp^\top + \overline{\mathbf{e}}_\perp\overline{\mathbf{e}}^\top\big) + \overline{\mathbf{e}}_\perp\overline{\mathbf{e}}_\perp^\top + \sum_{j=3}^{d} \overline{\mathbf{e}}_{\perp,j}\overline{\mathbf{e}}_{\perp,j}^\top. \tag{46}$$

Recall that $\{\overline{\mathbf{e}}, \overline{\mathbf{e}}_\perp, \overline{\mathbf{e}}_{\perp,3}, \overline{\mathbf{e}}_{\perp,4}, \ldots, \overline{\mathbf{e}}_{\perp,d}\}$ is an orthonormal basis for $\mathbb{R}^d$. Thus, $\sum_{j=3}^{d} \overline{\mathbf{e}}_{\perp,j}\overline{\mathbf{e}}_{\perp,j}^\top = \mathbf{I}_d - \overline{\mathbf{e}}\overline{\mathbf{e}}^\top - \overline{\mathbf{e}}_\perp\overline{\mathbf{e}}_\perp^\top$. Using this above, we get:

$$\mathbb{E}\Big[\widetilde{\mathbf{x}}\widetilde{\mathbf{x}}^\top\Big] = \mathbf{I}_d + \rho\big(\overline{\mathbf{e}}\overline{\mathbf{e}}_\perp^\top + \overline{\mathbf{e}}_\perp\overline{\mathbf{e}}^\top\big) = \widetilde{\boldsymbol{\Sigma}}. \tag{47}$$

This finishes the proof. □

**Lemma F.3.** *Recall that*

$$\mathbf{Q} = (1-\mu^2)\overline{\mathbf{e}}\overline{\mathbf{e}}^\top + \rho^2(1-\mu^2)\overline{\mathbf{e}}_\perp\overline{\mathbf{e}}_\perp^\top - \rho\mu^2\big(\overline{\mathbf{e}}\overline{\mathbf{e}}_\perp^\top + \overline{\mathbf{e}}_\perp\overline{\mathbf{e}}^\top\big).$$

*Let*

$$\mu = \sqrt{\frac{\beta(1-\rho^2)}{(1+\beta)(1-\beta\rho^2)}}$$

*for some $\beta \in (0,1]$. In that case, the eigenvalues of $\mathbf{Q}$ are:*

$$\widehat{\lambda}_1 = \frac{1+\beta\rho^2}{1+\beta} \text{ and } \widehat{\lambda}_2 = \rho^2\left(\frac{1-\beta}{1-\beta\rho^2}\right),$$

*and the corresponding eigenvectors are:*

$$\widehat{\mathbf{v}}_1 = \frac{1}{\sqrt{1+\beta^2\rho^2}}\overline{\mathbf{e}} - \frac{\beta\rho}{\sqrt{1+\beta^2\rho^2}}\overline{\mathbf{e}}_\perp \text{ and } \widehat{\mathbf{v}}_2 = -\frac{\beta\rho}{\sqrt{1+\beta^2\rho^2}}\overline{\mathbf{e}} - \frac{1}{\sqrt{1+\beta^2\rho^2}}\overline{\mathbf{e}}_\perp.$$

*Proof.* $\mathbf{Q}$ is a rank-2 matrix and its two eigenvectors will be in the span of $\overline{\mathbf{e}}$ and $\overline{\mathbf{e}}_\perp$. In particular, an eigenvector of $\mathbf{Q}$ is of the form $[\overline{\mathbf{e}}, \overline{\mathbf{e}}_\perp]\mathbf{b}$, where $\mathbf{b} \in \mathbb{R}^{2\times 1}$ is an eigenvector of the $2\times 2$ matrix:

$$\mathbf{A} := \begin{bmatrix} (1-\mu^2) & -\rho\mu^2 \\ -\rho\mu^2 & \rho^2(1-\mu^2) \end{bmatrix}. \tag{48}$$

Also, the corresponding eigenvalues of $\mathbf{Q}$ are the corresponding eigenvalues of $\mathbf{A}$. It can be verified that the eigenvalues of $\mathbf{A}$ are:

$$\widehat{\lambda}_1 = \frac{(1+\rho^2)(1-\mu^2)}{2} + \sqrt{\frac{(1-\rho^2)^2(1-\mu^2)^2}{4} + \rho^2\mu^4}. \tag{49}$$

and

$$\widehat{\lambda}_2 = \frac{(1+\rho^2)(1-\mu^2)}{2} - \sqrt{\frac{(1-\rho^2)^2(1-\mu^2)^2}{4} + \rho^2\mu^4}. \tag{50}$$

The corresponding eigenvectors of $\mathbf{A}$ are:

$$\widehat{\mathbf{b}}_1 = \frac{1}{\sqrt{b_{1,1}^2 + b_{1,2}^2}}\begin{bmatrix} b_{1,1} \\ b_{1,2} \end{bmatrix} \tag{51}$$

where $b_{1,1} = \frac{(1-\rho^2)(1-\mu^2)}{2} + \sqrt{\frac{(1-\rho^2)^2(1-\mu^2)^2}{4} + \rho^2\mu^4}$ and $b_{1,2} = -\rho\mu^2$, and

$$\widehat{\mathbf{b}}_2 = \frac{1}{\sqrt{b_{2,1}^2 + b_{2,2}^2}} \begin{bmatrix} b_{2,1} \\ b_{2,2} \end{bmatrix}, \tag{52}$$

where $b_{2,1} = \frac{(1-\rho^2)(1-\mu^2)}{2} - \sqrt{\frac{(1-\rho^2)^2(1-\mu^2)^2}{4} + \rho^2\mu^4}$ and $b_{2,2} = -\rho\mu^2$. Thus, the eigenvalues of $\mathbf{Q}$ are $\widehat{\lambda}_1$ and $\widehat{\lambda}_2$; the corresponding eigenvectors are $\widehat{\mathbf{v}}_1 = [\overline{\mathbf{e}}, \overline{\mathbf{e}}_\perp]\widehat{\mathbf{b}}_1$ and $\widehat{\mathbf{v}}_2 = [\overline{\mathbf{e}}, \overline{\mathbf{e}}_\perp]\widehat{\mathbf{b}}_2$. Note that:

$$\frac{(1-\rho^2)(1-\mu^2)}{2} \leq \sqrt{\frac{(1-\rho^2)^2(1-\mu^2)^2}{4} + \rho^2\mu^4} \leq \frac{(1-\rho^2)(1-\mu^2)}{2} + \rho\mu^2.$$

Let us set $\sqrt{\frac{(1-\rho^2)^2(1-\mu^2)^2}{4} + \rho^2\mu^4} = \frac{(1-\rho^2)(1-\mu^2)}{2} + \beta\rho^2\mu^2$, for some $\beta \in (0, 1]$. That gives us:

$$\mu = \sqrt{\frac{\beta(1-\rho^2)}{(1+\beta)(1-\beta\rho^2)}}. \tag{53}$$

In that case, we have:

$$\widehat{\lambda}_1 = \frac{1+\beta\rho^2}{1+\beta} \text{ and } \widehat{\lambda}_2 = \rho^2\left(\frac{1-\beta}{1-\beta\rho^2}\right). \tag{54}$$

Also,

$$b_{1,1} = \frac{1-\rho^2}{(1+\beta)(1-\beta\rho^2)}, b_{1,2} = b_{2,2} = -\frac{\beta\rho(1-\rho^2)}{(1+\beta)(1-\beta\rho^2)}, \text{ and } b_{2,1} = -\frac{\beta^2\rho^2(1-\rho^2)}{(1+\beta)(1-\beta\rho^2)}. \tag{55}$$

Therefore,

$$\widehat{\mathbf{b}}_1 = \frac{1}{\sqrt{1+\beta^2\rho^2}} \begin{bmatrix} 1 \\ -\beta\rho \end{bmatrix} \text{ and } \widehat{\mathbf{b}}_2 = \frac{1}{\sqrt{1+\beta^2\rho^2}} \begin{bmatrix} -\beta\rho \\ -1 \end{bmatrix}. \tag{56}$$

Recall that the eigenvalues of $\mathbf{Q}$ are $\widehat{\lambda}_1$ and $\widehat{\lambda}_2$, and the corresponding eigenvectors are

$$\widehat{\mathbf{v}}_1 = [\overline{\mathbf{e}}, \overline{\mathbf{e}}_\perp]\widehat{\mathbf{b}}_1 = \frac{1}{\sqrt{1+\beta^2\rho^2}}\overline{\mathbf{e}} - \frac{\beta\rho}{\sqrt{1+\beta^2\rho^2}}\overline{\mathbf{e}}_\perp \text{ and } \widehat{\mathbf{v}}_2 = [\overline{\mathbf{e}}, \overline{\mathbf{e}}_\perp]\widehat{\mathbf{b}}_2 = -\frac{\beta\rho}{\sqrt{1+\beta^2\rho^2}}\overline{\mathbf{e}} - \frac{1}{\sqrt{1+\beta^2\rho^2}}\overline{\mathbf{e}}_\perp.$$

Finally, recall that $\mu = \sqrt{\frac{\beta(1-\rho^2)}{(1+\beta)(1-\beta\rho^2)}}$. □

**Lemma F.4.** *Recall that the averaged model with parameter $\omega$ as defined in Equation (16) was*

$$\boldsymbol{\theta}_{\text{avg}}(\omega) = \omega\boldsymbol{\theta}_* + (1-\omega)\widetilde{\boldsymbol{\theta}}_* = \widetilde{\boldsymbol{\theta}}_* + \omega\mathbf{e}.$$

*We have:*

$$\min_{\omega\in[0,1]} \text{err}_{\text{tot}}\big(\boldsymbol{\theta}_{\text{avg}}(\omega)\big) = \left(\frac{\overline{\mathbf{e}}^\top\boldsymbol{\Sigma}\overline{\mathbf{e}}}{\overline{\mathbf{e}}^\top\boldsymbol{\Sigma}\overline{\mathbf{e}} + 1}\right)\|\mathbf{e}\|_2^2, \tag{57}$$

*where recall that $\boldsymbol{\Sigma}$ is the covariance matrix of the pre-training data.*

*Proof.* We have:

$$\text{err}_{\text{tot}}\big(\boldsymbol{\theta}_{\text{avg}}(\omega)\big) = \text{err}_1\big(\boldsymbol{\theta}_{\text{avg}}(\omega)\big) + \text{err}_2\big(\boldsymbol{\theta}_{\text{avg}}(\omega)\big) = \big(\boldsymbol{\theta}_{\text{avg}}(\omega) - \boldsymbol{\theta}_*\big)^\top\boldsymbol{\Sigma}\big(\boldsymbol{\theta}_{\text{avg}}(\omega) - \boldsymbol{\theta}_*\big) + \big(\boldsymbol{\theta}_{\text{avg}}(\omega) - \widetilde{\boldsymbol{\theta}}_*\big)^\top\widetilde{\boldsymbol{\Sigma}}\big(\boldsymbol{\theta}_{\text{avg}}(\omega) - \widetilde{\boldsymbol{\theta}}_*\big). \tag{58}$$

Plugging in the value of $\boldsymbol{\theta}_{\text{avg}}(\omega)$ and using the value of $\widetilde{\boldsymbol{\Sigma}}$ from Equation (3) above, we get:

$$\text{err}_{\text{tot}}(\boldsymbol{\theta}_{\text{avg}}(\omega)) = (1-\omega)^2\mathbf{e}^\top\boldsymbol{\Sigma}\mathbf{e} + \omega^2\|\mathbf{e}\|_2^2. \tag{59}$$

It can be verified (with elementary calculus) that the optimal value of $\omega$ that minimizes the RHS in Equation (59) is $\omega^* = \frac{\mathbf{e}^\top\boldsymbol{\Sigma}\mathbf{e}}{\mathbf{e}^\top\boldsymbol{\Sigma}\mathbf{e} + \|\mathbf{e}\|_2^2}$. Plugging this into Equation (59) and simplifying a bit yields the desired result. □

**Lemma F.5.** *Suppose $\alpha > 0$ and $\mathbf{r} \in \mathbb{R}^d$ is a unit-norm vector, i.e., $\|\mathbf{r}\|_2 = 1$. Let*

$$\mathbf{M} := \mathbb{E}\left[\exp\left(-\frac{\big(\langle \mathbf{r}, \mathbf{z}\rangle\big)^2}{\alpha}\right)\mathbf{z}\mathbf{z}^\top\right],$$

*where $\mathbf{z} \sim \mathcal{N}(\vec{0}_d, \mathbf{I}_d)$. $\mathbf{r}$ is an eigenvector of $\mathbf{M}$ with eigenvalue $\left(\frac{\alpha}{\alpha+2}\right)^{3/2}$. Further, the eigenvectors of $\mathbf{M}$ in the subspace of $\mathbb{R}^d$ orthogonal to $\mathbf{r}$ all have eigenvalues $\left(\frac{\alpha}{\alpha+2}\right)^{1/2}$.*

*Proof.* We have:

$$\mathbb{E}\Big[\mathbf{M}\mathbf{r}\Big] = \mathbb{E}\left[\exp\left(-\frac{\big(\langle \mathbf{r}, \mathbf{z}\rangle\big)^2}{\alpha}\right)\langle \mathbf{r}, \mathbf{z}\rangle \mathbf{z}\right]. \tag{60}$$

Suppose $\{\mathbf{r}_{\perp,j}\}_{j=1}^{d-1}$ is an orthonormal basis for the subspace orthogonal to $\mathbf{r}$; so $\langle \mathbf{r}_{\perp,j}, \mathbf{r}\rangle = 0 \;\forall\; j \in [d-1]$ and $\langle \mathbf{r}_{\perp,j}, \mathbf{r}_{\perp,k}\rangle = \mathbb{1}(j=k)\;\forall\, j,k \in [d-1]$. Then, note that:

$$\mathbf{z} = \langle \mathbf{r}, \mathbf{z}\rangle \mathbf{r} + \sum_{j=1}^{d-1} \langle \mathbf{r}_{\perp,j}, \mathbf{z}\rangle \mathbf{r}_{\perp,j}. \tag{61}$$

Since $\mathbf{z} \sim \mathcal{N}(\vec{0}_d, \mathbf{I}_d)$, $\langle \mathbf{r}, \mathbf{z}\rangle$ and $\{\langle \mathbf{r}_{\perp,j}, \mathbf{z}\rangle\}_{j=1}^{d-1}$ are i.i.d. $\mathcal{N}(0,1)$. Using all of this in Equation (60), we get:

$$\begin{aligned}
\mathbb{E}\Big[\mathbf{M}\mathbf{r}\Big] &= \mathbb{E}\left[\exp\left(-\frac{\big(\langle \mathbf{r}, \mathbf{z}\rangle\big)^2}{\alpha}\right)\big(\langle \mathbf{r}, \mathbf{z}\rangle\big)^2\right]\mathbf{r} + \sum_{j=1}^{d-1} \underbrace{\mathbb{E}\left[\exp\left(-\frac{\big(\langle \mathbf{r}, \mathbf{z}\rangle\big)^2}{\alpha}\right)\langle \mathbf{r}, \mathbf{z}\rangle\langle \mathbf{r}_{\perp,j}, \mathbf{z}\rangle\right]}_{=0\ (\langle \mathbf{r}, \mathbf{z}\rangle \text{ and } \langle \mathbf{r}_{\perp,j}, \mathbf{z}\rangle \text{ are independent})} \mathbf{r}_{\perp,j} && (62)\\
&= \mathbb{E}_{Z\sim\mathcal{N}(0,1)}\left[\exp\Big(-\frac{Z^2}{\alpha}\Big)Z^2\right]\mathbf{r} && \text{(because } \langle \mathbf{r}, \mathbf{z}\rangle \sim \mathcal{N}(0,1)) \quad (63)\\
&= \left(\frac{1}{\sqrt{2\pi}}\int_{-\infty}^{\infty} z^2 \exp\Big(-z^2\Big(\frac{1}{\alpha}+\frac{1}{2}\Big)\Big)dz\right)\mathbf{r} && (64)\\
&= \Big(\frac{\alpha}{\alpha+2}\Big)^{3/2}\mathbf{r}. && (65)
\end{aligned}$$

So $\mathbf{r}$ is an eigenvector of $\mathbf{M}$ with eigenvalue $\left(\frac{\alpha}{\alpha+2}\right)^{3/2}$.

Next, note that:

$$\begin{aligned}
\mathbb{E}\Big[\mathbf{M}\mathbf{r}_{\perp,1}\Big] = \underbrace{\mathbb{E}\left[\exp\left(-\frac{\big(\langle \mathbf{r}, \mathbf{z}\rangle\big)^2}{\alpha}\right)\langle \mathbf{r}_{\perp,1}, \mathbf{z}\rangle\langle \mathbf{r}, \mathbf{z}\rangle\right]}_{=0}\mathbf{r} + \mathbb{E}\left[\exp\left(-\frac{\big(\langle \mathbf{r}, \mathbf{z}\rangle\big)^2}{\alpha}\right)\big(\langle \mathbf{r}_{\perp,1}, \mathbf{z}\rangle\big)^2\right]\mathbf{r}_{\perp,1}\\
\sum_{j=2}^{d-1} \underbrace{\mathbb{E}\left[\exp\left(-\frac{\big(\langle \mathbf{r}, \mathbf{z}\rangle\big)^2}{\alpha}\right)\langle \mathbf{r}_{\perp,1}, \mathbf{z}\rangle\langle \mathbf{r}_{\perp,j}, \mathbf{z}\rangle\right]}_{=0} \mathbf{r}_{\perp,j}. \quad (66)
\end{aligned}$$

In the above equation, the first and last terms are 0 because $\langle \mathbf{r}, \mathbf{z}\rangle$ and $\{\langle \mathbf{r}_{\perp,j}, \mathbf{z}\rangle\}_{j=1}^{d-1}$ are i.i.d. $\mathcal{N}(0,1)$; using this fact again,

we get:

$$\mathbb{E}\Big[\mathbf{M}\mathbf{r}_{\perp,1}\Big] = \mathbb{E}_{Z\sim\mathcal{N}(0,1)}\left[\exp\Big(-\frac{Z^2}{\alpha}\Big)\right]\underbrace{\mathbb{E}_{\bar{Z}\sim\mathcal{N}(0,1)}\big[\bar{Z}^2\big]}_{=1}\mathbf{r}_{\perp,1} \tag{67}$$

$$= \left(\frac{1}{\sqrt{2\pi}}\int_{-\infty}^{\infty}\exp\Big(-z^2\Big(\frac{1}{\alpha}+\frac{1}{2}\Big)\Big)dz\right)\mathbf{r}_{\perp,1} \tag{68}$$

$$= \Big(\frac{\alpha}{\alpha+2}\Big)^{1/2}\mathbf{r}_{\perp,1}. \tag{69}$$

Similarly, we can show that for $j = \{2,\ldots,d-1\}$, we have:

$$\mathbb{E}\Big[\mathbf{M}\mathbf{r}_{\perp,j}\Big] = \Big(\frac{\alpha}{\alpha+2}\Big)^{1/2}\mathbf{r}_{\perp,j}. \tag{70}$$

So for all $j \in [d-1]$, $\mathbf{r}_{\perp,j}$ is an eigenvector of $\mathbf{M}$ with eigenvalue $\big(\frac{\alpha}{\alpha+2}\big)^{1/2}$. Thus, the eigenvectors of $\mathbf{M}$ in the subspace orthogonal to $\mathbf{r}$ all have eigenvalues $\big(\frac{\alpha}{\alpha+2}\big)^{1/2}$. □

## G. Experimental Details

In this section, we further discuss the experimental setup of FLOW's usage in both language and vision settings, specifically covering the following:

- Appendix G.1: Baseline Details.
- Appendix G.2: Language Model Hyper-Parameters.
- Appendix G.3: Language Model Evaluation Details.
- Appendix G.4: Vision Model Implementation Details.

### G.1. Baseline Details

In this section, we further discuss the baselines mentioned in Section 5.

**Linear Probing:** In our vision experiments, we define linear probing as freezing the body of the pre-trained model, initializing a new (task-specific) head and batch normalization layers, and training only the new head and batch normalization layers.

**$\ell_2$ regularization:** Based on Kirkpatrick et al. (2016), we perform $\ell_2$ regularization as a baseline in the data-oblivious setting. Specifically, the $\ell_2$-regularized loss is:

$$\mathcal{L}(\boldsymbol{\theta}) = \sum_{i=1}^{n} f_i(\boldsymbol{\theta}) + \lambda\|\boldsymbol{\theta}-\boldsymbol{\theta}^*\|_2^2 \tag{71}$$

where $f_i$ is the $i^{\text{th}}$ sample's loss, $\boldsymbol{\theta}^*$ is the pre-trained model, and $\lambda$ is the regularization parameter. Intuitively, as $\lambda$ increases, our model stays closer to the pre-trained model, mitigating forgetting at the expense of target domain performance.

**LoRA (Hu et al., 2022):** Recently, Biderman et al. (2024) showed that fine-tuning language models with LoRA (Hu et al., 2022) effectively mitigates forgetting. Following a similar setup as us, Biderman et al. (2024) fine-tuned language models on MetaMathQA (Yu et al., 2023) and then evaluated the fine-tuned model on several general capability tasks, viz., HellaSwag (Zellers et al., 2019), ARC-c (Clark et al., 2018), and WinoGrande (Sakaguchi et al., 2019), and one target domain task, viz., GSM8K (Cobbe et al., 2021). Further details about experimental hyper-parameters can be found in Appendix G.2.

**WiSE-FT (Wortsman et al., 2021):** We also consider model averaging as a baseline, specifically focusing on WiSE-FT (Wortsman et al., 2021). WiSE-FT is simply the convex combination of the model parameters shared between the two tasks, while the task-specific parts are not averaged. Specifically, we perform model averaging between the pre-trained model and the fine-tuned model. The convex combination parameter $\alpha$ of WiSE-FT is set to $0.5$ in our experiments, as we cannot optimize $\alpha$ in the data-oblivious setting.

**Learning without Forgetting (Li & Hoiem, 2016):** This work considers using a distillation-based loss to mitigate forgetting when the data from the training of previous tasks is not available. Initially, they record the responses $y_o$ on the new task images with the old tasks' model parameters, then train the model using a combination of fine-tuning loss, distillation loss, and model regularization. For new tasks, they use standard cross-entropy loss, while for old tasks they employ distillation loss that encourages the updated model's responses to match the recorded responses $y_o$ of the original model. Their proposed method has a distillation loss scaling factor $\lambda_0$ and a temperature parameter $T$ in the distillation loss, introducing extra tunable parameters. Their loss induces joint optimization of shared parameters $\theta_s$, old task parameters $\theta_o$, and new task parameters $\theta_n$ using only new task data. Note that unlike FLOW, they also update the old task parameters $\theta_o$.

### G.2. Language Model Hyper-Parameters

For both Gemma 2 2B (Team et al., 2024) and Llama 3.2 3B (Grattafiori et al., 2024), we run hyper-parameter sweeps on learning rates for each baseline. For standard fine-tuning, $\ell_2$ regularization, and FLOW, we do a learning rate sweep in [1e-4, 2e-5, 1e-5, 5e-6], and for LoRA ($r = 64$) we do a sweep in [2e-4, 2e-1], following the learning rates used in (Biderman et al., 2024). We then select the learning rate that results in the best GSM8K (Cobbe et al., 2021) accuracy, oblivious to general capability metrics. We report the hyper-parameters used for our Gemma 2 2B (Team et al., 2024) experiments in Table 6 and for Llama 3.2 3B (Grattafiori et al., 2024) in Table 7.

*Table 6.* The hyper-parameters used to train Gemma 2 2B in our experiments. Note that the learning rate selected is based on the best results on GSM8K after fine-tuning the method on MetaMathQA.

| **Hyper-parameter** | **Standard Fine-tuning** | **LoRA** ($r = 64$) | **$\ell_2$-Reg.** | **FLOW (Ours)** |
|---|---|---|---|---|
| Learning Rate | 1e-5 | 2e-4 | 5e-6 | 5e-6 |
| Learning Rate Scheduler | Cosine | | | |
| Batch Size | 128 | | | |
| Optimizer | AdamW | | | |
| Weight Decay | 0.00 | | | |
| Warmup Ratio | 0.03 | | | |
| Epochs | 2 | | | |
| Max Sequence Length | 1024 | | | |
| Seed | 42 | | | |

*Table 7.* The hyper-parameters used to train Llama 3.2 3B in our experiments. Note that the learning rate selected is based on the best results on GSM8K after fine-tuning the method on MetaMathQA.

| **Hyper-parameter** | **Standard Fine-tuning** | **LoRA** ($r = 64$) | **$\ell_2$-Reg.** | **FLOW (Ours)** |
|---|---|---|---|---|
| Learning Rate | 2e-5 | 2e-4 | 1e-5 | 1e-5 |
| Learning Rate Scheduler | Cosine | | | |
| Batch Size | 128 | | | |
| Optimizer | AdamW | | | |
| Weight Decay | 0.00 | | | |
| Warmup Ratio | 0.03 | | | |
| Epochs | 2 | | | |
| Max Sequence Length | 1024 | | | |
| Seed | 42 | | | |

For our WiSE-FT (Wortsman et al., 2021) model averaging experiments, we use $\alpha = 0.5$. For our LoRA (Hu et al., 2022) experiments, we use $\alpha = r = 64$. For $\ell_2$ regularization we use $\lambda = 1e - 3$ which is taken from (Chen et al., 2024b). Most training hyper-parameters for our language experiments are taken from Chen et al. (2024b), with the introduction of learning rate sweeps.

### G.3. Language Model Evaluation Details

As described in Section 5.2, we create a commonsense reasoning metric composed of the following six metrics: ARC-e (Clark et al., 2018), ARC-c (Clark et al., 2018), HellaSwag (Zellers et al., 2019), PIQA (Bisk et al., 2020), SIQA (Sap et al., 2019), and OBQA (Mihaylov et al., 2018). On top of the commonsense metric, we evaluate MMLU (Hendrycks et al., 2021a) and MBPP (Austin et al., 2021) to estimate the general capabilities of a language model and to measure the effects of catastrophic forgetting when fine-tuning a model on MetaMathQA (Yu et al., 2023). We additionally use GSM8K (Cobbe et al., 2021) to evaluate the target fine-tuning performance of a given fine-tuning method. We provide a brief describe each of these evaluation metrics:

1. **HellaSwag** (Zellers et al., 2019): A benchmark designed to test commonsense reasoning. HellaSwag presents a context followed by several plausible endings, and the model must choose the most appropriate continuation.

2. **ARC Easy** (Clark et al., 2018): A benchmark part of the AI2 reasoning challenge designed to test basic scientific reasoning and knowledge. ARC Easy presents 5,197 multiple-choice science questions drawn from grade 3-9 standardized tests, where each question typically includes a brief scientific scenario or statement followed by four possible answer choices.

3. **ARC Challenge** (Clark et al., 2018): A benchmark part of the AI2 reasoning challenge designed to test advanced scientific reasoning and knowledge application. ARC Challenge presents 2,590 multiple-choice science questions drawn from grade 3-9 standardized tests, where each question typically includes a scientific scenario or phenomenon followed by four possible answer choices. The questions in ARC Challenge are significantly more challenging than ARC Easy.

4. **PIQA** (Bisk et al., 2020): A benchmark designed to evaluate physical commonsense understanding in natural language. PIQA presents a goal and two possible solutions, requiring models to choose the most appropriate solution that demonstrates an understanding of everyday physical interactions.

5. **SIQA** (Sap et al., 2019): A benchmark designed to evaluate social commonsense intelligence and emotional reasoning. SIQA presents a social situation context followed by a question and three possible answers, requiring models to demonstrate an understanding of social interactions, emotional responses, and behavioral implications.

6. **Open Book QA** (Mihaylov et al., 2018): A benchmark designed to assess understanding of elementary science concepts in an open-book exam format. OBQA presents 5,957 multiple-choice questions paired with a small "book" of 1,326 core science facts, requiring models to combine these facts with common knowledge to arrive at correct answers.

7. **MMLU** (Hendrycks et al., 2021a): A benchmark designed to evaluate massive multitask language understanding. MMLU presents approximately 16,000 multiple-choice questions spanning 57 subjects including mathematics, philosophy, law, and medicine, requiring models to demonstrate broad knowledge and reasoning capabilities.

8. **MBPP** (Austin et al., 2021): A benchmark designed to evaluate basic Python programming capabilities. The entire MBPP dataset presents 974 Python programming problems, where each problem includes a natural language task description and three test cases written as assert statements, requiring models to generate functionally correct Python code solutions.

9. **GSM8K** (Cobbe et al., 2021): A benchmark designed to evaluate multi-step mathematical reasoning capabilities. The GSM8K test set contains 1,000 grade school math word problems, where each problem requires 2-8 steps to solve using basic arithmetic operations (addition, subtraction, multiplication, division).

We follow the standard evaluation process for each of these datasets and specifically use `lm-evaluation-harness` (Gao et al., 2024) to evaluate our experiments.

### G.4. Vision Model Implementation Details

We performed an extensive hyper-parameter search over six learning rates (lrs $= [0.05, 0.01, 0.005, 0.001, 0.0005, 0.0001]$), two models, and six datasets (i.e., 72 total runs per method) for standard fine-tuning, linear probing, and FLOW. We chose the best learning rates associated with the highest average score over all the target (fine-tuning) datasets. Since our method is data oblivious, we do not use the validation set of ImageNet-1K other than for evaluation. For training models with $\ell_2$-regularization, we adapted the same learning rates and other related hyperparameters used for standard fine-tuning. We searched for $\lambda$ using one dataset and ResNet50 model ($\lambda = [0.002, 0.00001, 0.00002]$) and chose 0.002 based on average accuracy over target data. We chose ($\alpha = 0.05$) for WiSE-FT following Wortsman et al. (2021). We present all the important training details in Table 8 for the ResNet models.

*Table 8.* Hyperparameter configurations for finetuning ResNet18 and ResNet50 on the image classification datasets.

| Model | Hyperparameters | CIFAR10 | CIFAR100 | Flowers102 | Caltech101 | Dogs | Cars |
|---|---|---|---|---|---|---|---|
| | # GPUs | 1 A6000 | | | | | |
| | Optimizer | SGD | | | | | |
| | LR Schedule | Cosine (except for Linear probing) | | | | | |
| | Weight Decay | 0.0005 | | | | | |
| | Seed | 42 | | | | | |
| | $\lambda$ for $\ell_2$-Reg. | 0.002 | | | | | |
| | $\alpha$ for WiSE-FT. | 0.05 | | | | | |
| | $\tau$ (temperature) for FLOW | median loss | | | | | |
| | Epochs | 20 | 25 | 25 | 30 | 30 | 30 |
| ResNet18 | LR-Standard fine-tuning | 5E-3 | 1E-2 | 5E-2 | 5E-3 | 1E-3 | 5E-2 |
| | LR-Linear probing | 5E-3 | 5E-3 | 5E-2 | 1E-2 | 5E-3 | 5E-2 |
| | LR-FLOW | 1E-3 | 5E-3 | 5E-2 | 1E-2 | 5E-3 | 1E-2 |
| ResNet50 | LR-Standard fine-tuning | 5E-3 | 1E-3 | 1E-2 | 5E-3 | 5E-4 | 5E-2 |
| | LR-Linear probing | 5E-2 | 5E-2 | 5E-2 | 5E-2 | 1E-2 | 5E-2 |
| | LR-FLOW | 5E-4 | 1E-3 | 1E-2 | 1E-2 | 5E-3 | 1E-2 |

**Hyperparameter Details for ViT-B/16.** We fine-tuned for 8 epochs on Food-101. We used a learning rate of 5e–5 for standard fine-tuning, $\ell_2$-regularization (with $\lambda = 0.002$), and our method FLOW. For linear probing, we used a higher learning rate of 1e–3. For LwF, we used the same learning rate as standard fine-tuning (5e–5), $T = 2$, and $\lambda_0$ = 5e–4. Here we used a higher value of $\tau$ for FLOW and set it equal to $80^{\text{th}}$ percentile of the pre-trained loss values.

**Datasets**

1. **ImageNet-1K** (Russakovsky et al., 2015) serves as the pre-training dataset for all our vision base models. It is a widely used large-scale image classification dataset, consisting of over a million images spanning 1000 classes.
2. **CIFAR-10** (Krizhevsky, 2009) is a widely used dataset for image classification tasks. It consists of 60,000 32x32 color images divided into ten classes, with 6,000 images per class.
3. **CIFAR-100** (Krizhevsky, 2009) extends CIFAR-10 by providing 100 classes containing 600 images each. This dataset is used for fine-grained image classification tasks.
4. **Caltech101** (Li et al., 2022) comprises images of a diverse range of objects across 101 categories with diverse set of image classes.
5. **Flowers102** (Nilsback & Zisserman, 2008) comprises 102 categories of flowers, with each category containing between 40 to 258 images. This dataset is commonly used for fine-grained image classification and flower recognition tasks.

6. **Cars** (Krause et al., 2013) refers to the Stanford Cars dataset, which includes 16,185 images of 196 classes of cars. It provides a rich resource for fine-grained car classification task.

7. **Dogs** (Parkhi et al., 2012) pertains to the Stanford Dogs dataset, containing 20,580 images of 120 breeds of dogs. This dataset is widely used for fine-grained dog breed classification and recognition tasks.

8. **Food101** (Bossard et al., 2014) is a large-scale dataset for food classification containing 101 categories with 1,000 images per class, commonly used to evaluate models on fine-grained object recognition tasks.

## H. Detailed Vision Results and Ablations

First, we present the detailed version of the results of Table 1 in Tables 9 and 10.

*Table 9.* **ResNets: Target accuracies on each of the six datasets for the results in Table 1.**

| | Method | CIFAR-10 | CIFAR-100 | Flowers-102 | Caltech-101 | Dogs | Cars | Average |
|---|---|---|---|---|---|---|---|---|
| **ResNet18** | Linear probing | 81.32 | 60.06 | 87.20 | 91.15 | 78.50 | 43.23 | 73.57 |
| | Standard FT | 96.15 | 83.42 | 92.45 | 94.02 | 80.47 | 87.91 | 89.07 |
| | $\ell_2$-Regularization | 95.53 | 81.82 | 92.11 | 94.23 | 80.27 | 84.78 | 88.12 |
| | WiSE-FT | 91.47 | 65.90 | 87.28 | 91.40 | 82.48 | 62.88 | 80.23 |
| | **FLOW** (Ours) | 88.25 | 78.95 | 90.01 | 93.05 | 86.20 | 67.17 | 83.93 |
| **ResNet50** | Linear probing | 86.62 | 67.80 | 83.64 | 93.45 | 85.76 | 41.97 | 76.45 |
| | Standard FT | 97.61 | 86.11 | 91.74 | 96.02 | 89.26 | 89.94 | 91.78 |
| | $\ell_2$-Regularization | 97.50 | 85.77 | 91.67 | 95.85 | 89.29 | 89.42 | 91.58 |
| | WiSE-FT | 94.65 | 72.55 | 71.95 | 93.73 | 92.52 | 62.89 | 81.38 |
| | **FLOW** (Ours) | 91.11 | 79.42 | 86.78 | 94.45 | 91.16 | 74.59 | 86.25 |

*Table 10.* **ResNets: Top-1 ImageNet-1K accuracy after fine-tuning on each target dataset for the results in Table 1.**

| | Method | CIFAR-10 | CIFAR-100 | Flowers-102 | Caltech-101 | Dogs | Cars | Average |
|---|---|---|---|---|---|---|---|---|
| **ResNet18** | Linear probing | 69.76 | 69.76 | 69.76 | 69.76 | 69.76 | 69.76 | 69.76 |
| | Standard FT | 19.93 | 0.39 | 6.48 | 34.17 | 56.38 | 0.17 | 19.58 |
| | $\ell_2$-Regularization | 37.86 | 29.86 | 19.34 | 46.67 | 58.34 | 16.64 | 34.78 |
| | WiSE-FT | 62.24 | 47.65 | 49.98 | 64.70 | 67.34 | 33.03 | 54.15 |
| | **FLOW** (Ours) | 69.02 | 52.64 | 67.80 | 68.32 | 67.78 | 65.74 | 65.21 |
| **ResNet50** | Linear probing | 79.02 | 79.02 | 79.02 | 79.02 | 79.02 | 79.02 | 79.02 |
| | Standard FT | 16.89 | 35.95 | 61.01 | 40.51 | 66.93 | 0.21 | 36.91 |
| | $\ell_2$-Regularization | 33.98 | 47.16 | 62.85 | 43.42 | 67.03 | 14.27 | 44.78 |
| | WiseFT ($\alpha = 0.5$) | 61.40 | 73.04 | 76.33 | 73.25 | 77.36 | 8.55 | 61.65 |
| | **FLOW** (Ours) | 78.26 | 75.13 | 78.60 | 73.38 | 78.55 | 72.64 | 76.09 |

In Table 11, we present a small ablation to compare the pre-training and fine-tuning performances with different values of $\tau$ in FLOW; recall that we prescribed selecting $\tau$ to be the median pre-training loss value. As we see, if we could tune $\tau$, then FLOW's results would be even better.

*Table 11.* FLOW **ablation with different values of $\tau$:** Pre-training and fine-tuning accuracies as a function of $\tau$ set to different percentiles of the pre-training losses. The model is ResNet-50, pre-training dataset is ImageNet-1K (IN-1K), and the fine-tuning dataset is Caltech101 (Target). So if we could tune $\tau$, then FLOW's results would further improve.

| IN-1K Accuracy | Target Accuracy | Average | $\tau$- Percentile (%) |
|---|---|---|---|
| 68.51 | 91.15 | 79.83 | 10 |
| 64.13 | 91.01 | 77.57 | 30 |
| 54.72 | 91.80 | 73.26 | 50 |
| 45.59 | 92.91 | 69.25 | 70 |
| 20.51 | 94.02 | 57.27 | 90 |

Further, in Figure 2, we present another ablation study where we compare FLOW with different values of $\tau$, WiSE-FT with different values of the convex combination parameter, and random selection where we train on a random subset of the fine-tuning data to limit the drift from the pre-trained model.

### H.1. Comparison with a Distillation-Based Method for Mitigating Forgetting

Here, we compare our method FLOW against a distillation-based method for mitigating forgetting called "learning without forgetting" (LwF) (Li & Hoiem, 2016) in vision. More details about this method can be found in Appendix G.1, but

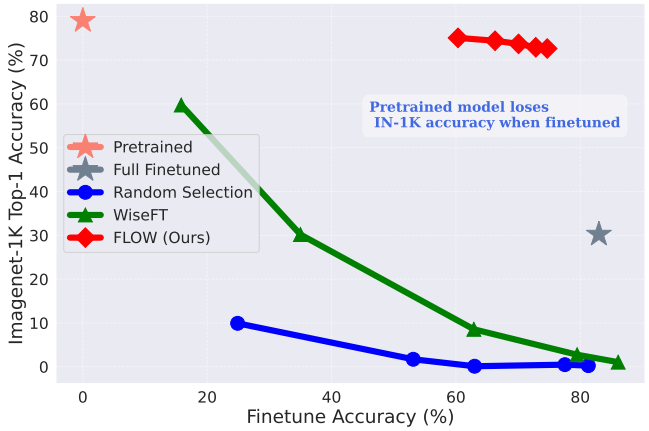


*Figure 2.* **Comparison of FLOW with different values of $\tau$ and some other baselines with different hyper-parameter values.** This plot is for ResNet-50 on the Stanford cars dataset. FLOW's plot (in red) is with $\tau = \{10, 20, 30, 40, 50\}$ percentile of the per-sample losses. As the name "random selection" implies, we just pick a random subset of the fine-tuning data and train on this subset to limit the drift from the pre-trained model. To have some correspondence with our choice of $\tau$ for FLOW, we pick random $\{10, 20, 30, 40, 50\}$ % of the data in "random selection". As we see, FLOW significantly outperforms other methods.

the important thing to note is that it updates the head corresponding to the pre-training data (which FLOW does not do) and also comes with more tunable parameters, additional memory, and evaluation overhead compared to FLOW. We consider the ViT-B/16 (Dosovitskiy et al., 2021) model pre-trained on Imagenet-1K (IN-1K) and a single large fine-tuning dataset, Food101 (Bossard et al., 2014). The evaluation metric is the same as in Section 5.1. Hyper-parameter details are in Appendix G.4.

In Table 12, we list the accuracies of the pre-trained model, standard FT, linear probing, $\ell_2$-regularization, LwF, and FLOW. Note that FLOW outperforms LwF despite its simplicity. Specifically, FLOW achieves better performance on the forgetting front, while LwF does better on the fine-tuning task.

*Table 12.* **Comparison with the distillation-based method (LwF) of Li & Hoiem (2016). Bolded** and underlined values indicate the **best** and second-best accuracies within each column (and for each model). Deltas (in color) for IN-1K and target performance are computed w.r.t. the pre-trained and standard fine-tuned models. Note that **FLOW outperforms LwF** despite not updating the head for the pre-training data, unlike LwF, and being more efficient than LwF, which comes with more tunable parameters, additional memory, and evaluation overhead.

| | Method | IN-1K Accuracy | | Target Accuracy | | Average |
|---|---|---|---|---|---|---|
| ViT-B/16 | Pre-trained | **81.10** | (+0.00) | – | | – |
| | Standard FT | 56.11 | (-24.99) | 91.60 | (+0.00) | 73.86 |
| | Linear Probe | **81.10** | (+0.00) | 83.86 | (-7.74) | 82.48 |
| | $\ell_2$-Reg. | 59.18 | (-21.92) | **91.66** | (+0.06) | 75.42 |
| | LwF | 76.39 | (-4.71) | 91.23 | (-0.37) | 83.81 |
| | **FLOW** (Ours) | 77.94 | (-3.16) | 90.57 | (-1.03) | **84.26** |

## I. Additional Language Model Results and Ablations

In this section, we discuss expanded results and further ablations of FLOW within our language experiments, specifically covering the following:

- Appendix I.1: An expanded table of results on commonsense reasoning tasks along with other baselines.
- Appendix I.2: An additional ablation on *token*-wise weighting scheme for fine-tuning with language data.
- Appendix I.3: An expanded set of plots and results for the combination of FLOW with weight averaging techniques such as Wise-FT (Wortsman et al., 2021).

### I.1. Extended Commonsense Reasoning Results

As discussed in Section 5.2 and Appendix G.3, we evaluate FLOW and the other baselines on various commonsense reasoning tasks within fine-tuning with the procedure described in Section 5.2. We include the exact results of these evaluation metrics for various baselines and FLOW in Table 13. We also include the results of commonsense reasoning metrics for the ablation combining FLOW with LoRA and $\ell_2$-regularization in Table 14.

*Table 13.* **Extended commonsense reasoning metrics for FLOW and other baselines within language modeling**. The performance on commonsense reasoning evaluations when fine-tuning Gemma 2 2B (Team et al., 2024) and Llama 3.2 3B (Grattafiori et al., 2024) on MetaMathQA (Yu et al., 2023). We include the target domain evaluation GSM8K (Cobbe et al., 2021) for convenience. The results show that FLOW can effectively mitigate catastrophic forgetting while still getting strong performance on our target fine-tuning task.

| | Method | ARC-e | ARC-c | HellaSwag | PIQA | SIQA | OBQA | Average | GSM8K |
|---|---|---|---|---|---|---|---|---|---|
| Gemma 2 2B | Pre-trained | 80.18 | 46.84 | 54.95 | 78.67 | 51.33 | 31.40 | 57.23 | 24.49 |
| | Standard Fine-tuning | 76.09 | 42.07 | 54.41 | 76.99 | 48.06 | 32.00 | 55.07 | 63.38 |
| | WiSE-FT | 79.55 | 46.42 | 56.43 | 78.24 | 51.08 | 32.00 | 57.28 | 53.30 |
| | LoRA ($r = 64$) | 77.78 | 44.37 | 54.59 | 76.99 | 50.51 | 29.80 | 55.67 | 60.43 |
| | $\ell_2$-Regularization | 79.08 | 45.99 | 56.21 | 77.20 | 50.97 | 32.60 | 57.01 | 62.85 |
| | **FLOW** (Ours) | 79.76 | 47.18 | 56.23 | 77.69 | 51.48 | 33.20 | 57.59 | 62.55 |
| Llama 3.2 3B | Pre-trained | 74.54 | 42.15 | 55.31 | 76.66 | 47.03 | 31.20 | 54.48 | 26.01 |
| | Standard Fine-tuning | 70.03 | 34.22 | 52.02 | 74.16 | 45.24 | 28.40 | 50.68 | 66.95 |
| | WiSE-FT | 75.63 | 40.79 | 55.18 | 76.93 | 47.34 | 31.40 | 54.54 | 57.01 |
| | LoRA ($r = 64$) | 71.38 | 37.88 | 55.01 | 76.55 | 47.39 | 30.40 | 53.10 | 63.84 |
| | $\ell_2$-Regularization | 73.57 | 38.91 | 54.939 | 76.12 | 47.24 | 30.80 | 53.60 | 66.87 |
| | **FLOW** (Ours) | 74.96 | 39.68 | 55.39 | 76.01 | 47.80 | 32.00 | 54.30 | 65.58 |

*Table 14.* **Extended commonsense reasoning metrics for combining FLOW with other baselines**. The performance on commonsense reasoning evaluations when fine-tuning Gemma 2 2B (Team et al., 2024) baselines in conjunction with FLOW on MetaMathQA (Yu et al., 2023). We include the target domain evaluation GSM8K (Cobbe et al., 2021) for convenience. The results show that FLOW can effectively be used in conjunction with other methods that mitigate catastrophic forgetting.

| Method | ARC-e | ARC-c | HellaSwag | PIQA | SIQA | OBQA | Average | GSM8K |
|---|---|---|---|---|---|---|---|---|
| LoRA ($r = 64$) | 77.78 | 44.37 | 54.59 | 76.99 | 50.51 | 29.80 | 55.67 | 60.43 |
| LoRA ($r = 64$) + **FLOW** | 79.50 | 45.39 | 55.27 | 77.31 | 51.18 | 31.80 | 56.74 | 61.49 |
| $\ell_2$-Regularization | 79.08 | 45.99 | 56.21 | 77.20 | 50.97 | 32.60 | 57.01 | 62.85 |
| $\ell_2$-Regularization + **FLOW** | 79.67 | 47.10 | 56.38 | 77.48 | 51.13 | 33.40 | 57.53 | 62.02 |

Table 13 shows a clear trend that FLOW, can strongly mitigate catastrophic forgetting in comparison to standard fine-tuning. For Gemma 2 2B (Team et al., 2024), we can see that FLOW only has $\sim 0.8\%$ reduction in the performance of the target fine-tuning while on average maintaining the commonsense reasoning abilities of the pre-trained model, a $\sim 2.52\%$ increase over standard fine-tuning. For Llama 3.2 3B (Grattafiori et al., 2024), we can see that FLOW can again maintain the commonsense reasoning abilities of the base pre-trained model while only having a $\sim$1.4% drop on target fine-tuning performance. Overall, FLOW strikes a strong balance between general capabilities and target fine-tuning performance compared to other baselines.

For experiments with Gemma 2 2B (Team et al., 2024), FLOW can on average maintain the best scores on commonsense reasoning tasks. Performing only $\sim 0.8\%$ and $\sim 0.3\%$ worse on GSM8K (Cobbe et al., 2021) in comparison to standard fine-tuning and $\ell_2$ regularization, FLOW can improve on commonsense reasoning metrics by $\sim 2.42\%$ and $\sim 0.58\%$ respectively. Interestingly, in our Llama 3.2 3B (Grattafiori et al., 2024) experiments, we found that WiSE-FT (Wortsman et al., 2021) performed the strongest in preventing catastrophic forgetting of commonsense capabilities ($+0.04$ over the pre-trained model); however, this came at the cost of a significant decrease in GSM8K (Cobbe et al., 2021) accuracy ($-9.94$ under standard fine-tuning). In comparison, FLOW effectively mitigated forgetting in commonsense reasoning metrics ($-0.18$ under the pre-trained model), while achieving significantly higher accuracy in GSM8K (Cobbe et al., 2021) ($-1.37$ under standard fine-tuning).

### I.2. Token-wise Sample Weighting Ablations

In the language experiments, "sample" for FLOW can be defined as an entire sequence or an individual token. The experiments in the main paper treat a sequence as a sample; in that case, the per-sample loss is the average loss over the tokens in the sequence. We call this *sequence*-wise re-weighting. Instead, one could treat a token as a sample in which case the per-sample loss is just the token's loss. We call this *token*-wise re-weighting. We run a small ablation on both *sequence*-wise and *token*-wise re-weighting by following a similar experimental setup as Section 5.2. We train a Gemma 2 2B (Team et al., 2024) on MetaMathQA (Yu et al., 2023) and evaluate it on several general capability and target domain evaluations. The results of this experiment are in Table 15.

*Table 15.* The performance of Gemma 2B 2B on general capabilities metrics compared to target domain performance (GSM8K) when training on MetaMathQA. *Pre-trained* is the base model performance of Gemma 2 2B, *Standard* is the performance after full end-to-end fine-tuning, *Sequence* is our sequence sample weighting schema with FLOW, and *Token* is our token sample weighting schema with FLOW. **Bold** and underlined values indicate the **best** and second-best results respectively within each evaluation metric.

| Method | ARC-e | ARC-c | HellaSwag | PIQA | SIQA | OBQA | MMLU | MBPP | GSM8K |
|---|---|---|---|---|---|---|---|---|---|
| Base | **80.18** | 46.84 | 54.95 | **78.67** | 51.33 | 31.40 | **49.59** | **28.40** | 24.49 |
| Standard | 76.09 | 42.07 | 54.41 | 76.99 | 48.06 | **32.80** | 45.59 | 16.80 | **63.38** |
| Sequence | 79.76 | **47.18** | **56.23** | 77.69 | **51.48** | 33.20 | 49.31 | 26.80 | 62.55 |
| Token | 79.38 | 45.90 | 53.95 | 78.29 | 51.28 | 31.80 | 48.75 | 22.00 | 23.73 |

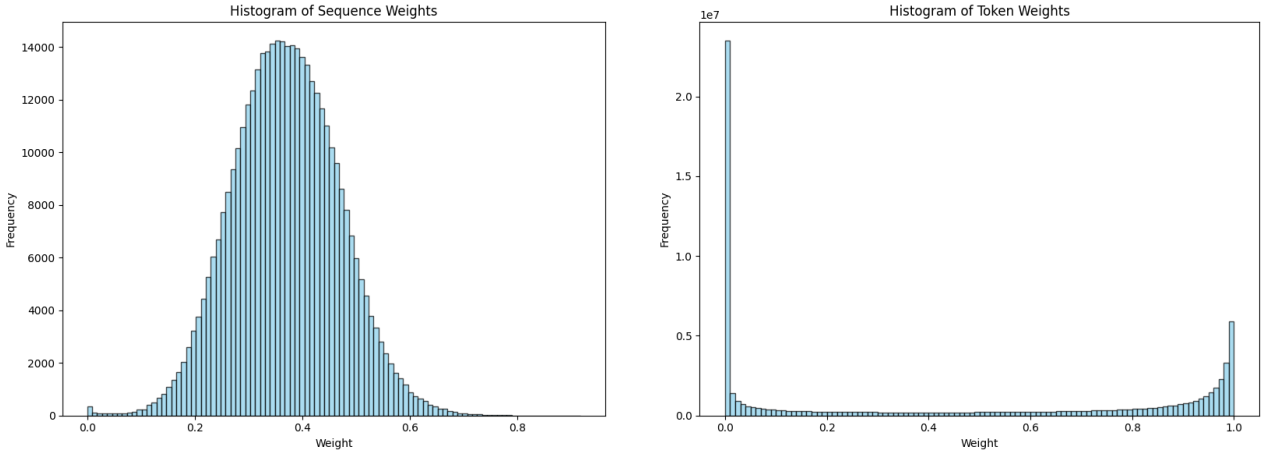


*Figure 3.* Histograms comparing the sample-wise distribution of weights in *sequence*-wise re-weighting schema for FLOW and token-wise distribution of weights *token*-wise re-weighting schema for FLOW. The *sequence*-wise weight distribution is given on the left, while the *token*-wise weight distribution is given on the right.

While *token*-wise sample re-weighting performs comparably or slightly worse than *sequence*-wise sample re-weighting in terms of the catastrophic forgetting of general capabilities of Gemma 2 2B, it struggles to effectively learn the fine-tuning target domain of GSM8K. To further understand this problem, we compare the weight distributions between *sequence*-wise and *token*-wise re-weighting schema in Figure 3. We can see that the *sequence* weights appear Gaussian, while most of the *token* weights are either 0 or 1. We speculate that *token*-wise re-weighting will force any token not commonly appearing in the pre-training data to have a high loss or perplexity, which combined with our algorithm, will heavily down-weight them to almost zero. We further speculate that these tokens are essential to improving the performance of our target fine-tuning task and that using FLOW with a *token*-wise scheme over-regularizes, preventing any meaningful learning of the target task. As *sequence*-wise re-weighting significantly outperforms *token*-wise re-weighting, we recommend using *sequence*-wise re-weighting in FLOW for language models.

### I.3. Extended Weight Averaging Results

As discussed in Section 6, we further combine FLOW with WiSE-FT (Wortsman et al., 2021) to mitigate the effects of catastrophic forgetting when fine-tuning. In this section, we report the full results of combining FLOW and WiSE-FT to prevent catastrophic forgetting with Gemma 2 2B.

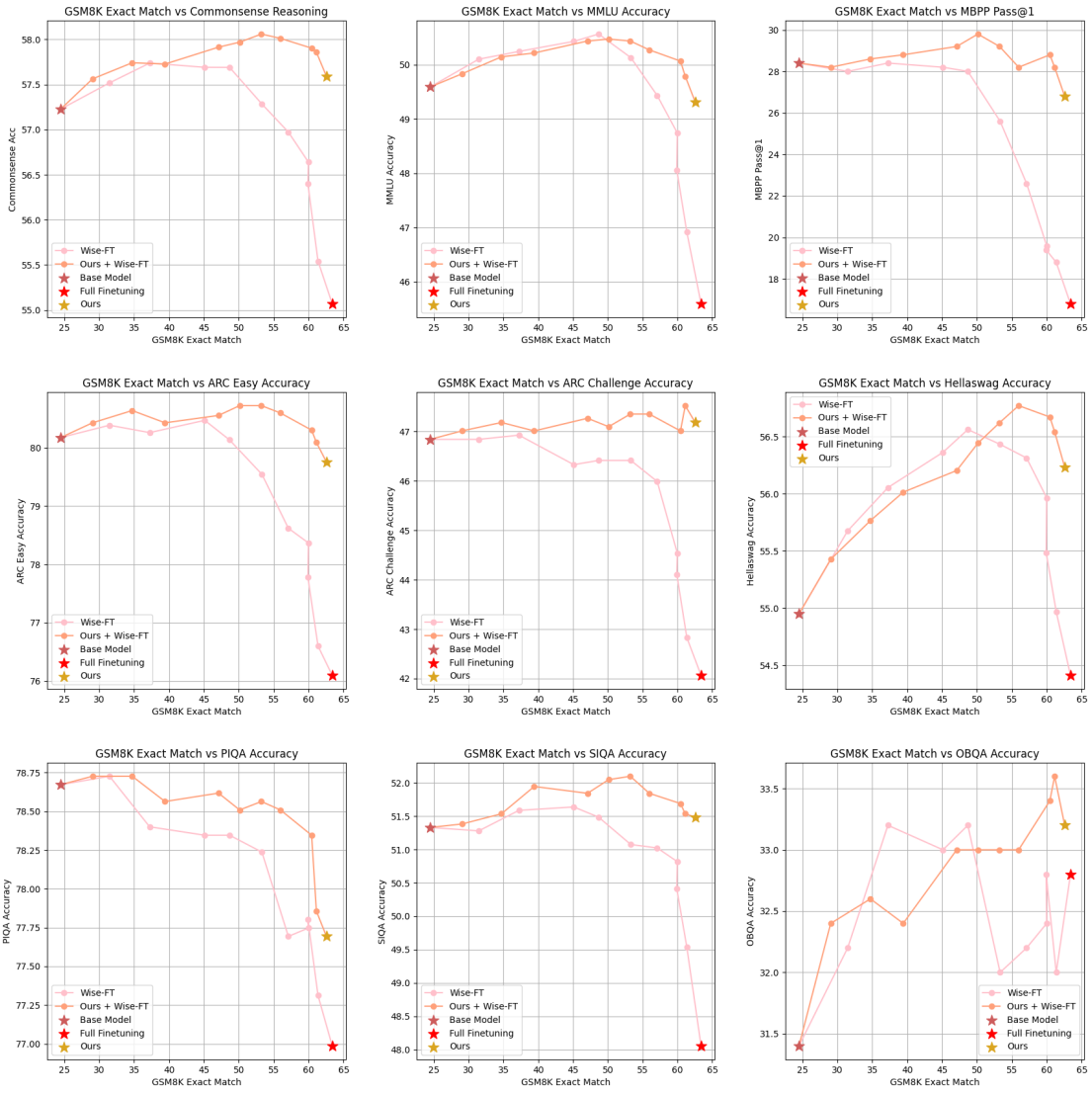


*Figure 4.* **FLOW is complementary with model averaging (WiSE-FT) in language modeling.** We compare WiSE-FT (Wortsman et al., 2021) with a standard model fine-tuning and with FLOW after fine-tuning Gemma 2 2B on MetaMathQA. We use varying $\alpha \in [0, 1]$ for WiSE-FT. The results indicate that combining Wise-FT with FLOW outperforms vanilla WiSE-FT with standard fine-tuning.