Multi-Domain Transfer Learning for Image Classification

Yijia Wang
Department of Computer Science
Stanford University
yijawang@stanford.edu

Abstract

The latest advancements in large pre-trained models offer a cost-efficient and high-performance approach to developing models for downstream domains via transfer learning. However, balancing target domain knowledge acquisition with general knowledge preservation is challenging. In this work, we propose a multi-domain transfer learning method leveraging recent efficient knowledge fusion techniques in deep neural network training. Specifically, we formulate the task as a sequential fine-tuning paradigm where the pre-trained foundation model serves as the initial starting point, and each fine-tuned model based on the previous domain serves as the starting point for the next. We apply tangent linearization for improved model editing performance and PCGrad for conflicting gradient calibration. Experiments with the CLIP model on multiple image classification tasks show that our proposed sequential training strategy significantly outperforms separate fine-tuning and weight interpolation. Our results demonstrate that tangent linearization and conflicting gradient calibration enhance the model's ability to preserve knowledge across domains, providing a flexible method for balancing target domain knowledge acquisition with general knowledge preservation.

1. Introduction

Large pre-trained vision models, developed through data and resource intensive training, embed vast prior information that is essential for a wide range of domains. When facing new tasks from different domains, a common and cost-effective practice is to perform transfer learning with these large pre-trained models, to leverage their valuable prior knowledge. In scenarios where multiple downstream tasks need to be addressed, one solution is to train a model specialized in each task separately. An alternative solution is to train a more versatile model that is capable of solving a broad scope of tasks. The latter approach offers several potential advantages, including better resource efficiency, simpler deployment, higher maintainability and scalability, potential positive transfer, and increased robustness to distribution shift. In this work, we focus on the latter approach.

A key challenge in combining information from different domains, is to strategically leverage common knowledge while minimizing conflicts in model weight updates. Several research efforts have been made to address this challenge by interpolating model weights [10][20][32] or strategically calibrating weight updates to alleviate conflicts [18][20][36].

In this work, we explore multi-domain transfer learning on image classification tasks by leveraging large pre-trained vision models, such as CLIP [23]. Our proposed idea is to fine-tune a versatile model on multiple target tasks in tangent space through linearization [20]. During the optimization process, we perform conflicting weights calibration [36][18] to balance general knowledge and domain-specific knowledge preservation. We sequentially fine-tune each domain and iteratively transfer to a new domain by initializing the model with the weights fine-tuned on the previous domain.

We conduct experiments on 8 image classification tasks, including EuroSAT [7], RESISC45 [1], Cars [14], DTD [2], GTSRB [8], MNIST [3], SUN397 [34] [33], and SVHN [19]. The first 4 tasks are selected as target tasks that are explicitly optimized during the fine-tuning process, and last 4 tasks are selected as reference tasks which are not involved in the fine-tuning process and solely used as auxiliary tasks to evaluate the model's generalizability. Our goal is to train a model that maximizes combined performance on target tasks while minimizing performance degradation on reference tasks. The inputs to our model are 2D images, and we use a fine-tuned CLIP ViT-B/32 model to output predicted classes. The experimental results show that our proposed approach significantly improves image classification accuracy on target domains ($6\% \sim 14\%$ higher average accuracy), while reducing performance degradation in out-of-domain reference tasks ($10\% \sim 20\%$ lower average accuracy degradation). We further conduct a thorough analysis of various factors that impact the performance, and provide insights into effective configuration strategies to better suit

different application scenarios.

2. Related Work

Most related work can be classified into 6 categories: weight interpolation, tangent linearization, gradients calibration, curriculum learning, continual learning, and multi-domain image classification. In this section, we summarize each category and highlight their relations with our approach.

Weights Interpolation This approach fuses information in weight space directly by interpolating the weights before and after fine-tuning. Existing researches on large pre-trained models such as CLIP show that weights interpolation [10] [32] can effectively preserve knowledge in pre-trained model and improve robustness to distribution shift for downstream domain-specific models. This approach provides an intuitive and convenient way to fuse knowledge. However, in a multi-domain setup, it could potentially cause significant performance degradation to each target domain.

In our work, we apply weights interpolation among models that are separately fine-tuned from a common pre-trained checkpoint, using it as a baseline for our approach.

Tangent Linearization approach This approach calculates task vectors [9] by fine-tuning the models in their tangent space [20]. The task vectors are obtained by taking the difference between the fine-tuned model's weights and the initial model's weights. The task vector represents specialized domain knowledge, which can be added together and applied to pre-trained model weights to fuse information in weight space. This draws a natural connection with the weights interpolation approach. It generalizes equally weighted interpolation among different domain expert models, while providing the flexibility to adjust the relative contribution of the pre-trained model. Furthermore, linearized fine-tuning has been shown to amplify weight disentanglement and reduce interference across different tasks, facilitating the preservation of general knowledge embedded in the pre-trained model and the distinct information from various domains. Based on these properties, tangent linearization can be naturally applied in sequential fine-tuning setup as a simple yet effective approach for multi-domain transfer learning. Although it introduces increased computational complexity by a constant factor [20], regular fine-tuning efforts generally suffice to achieve efficient and stable training. The high expressivity and well-initialized weights of the pre-trained model potentially contribute to its robustness, and reduce the sensitivity to hyperparameter variations.

In our work, the starting pre-trained model in a transfer learning process is not restricted to the original foundation model, but can also include fine-tuned models from the upstream domain tasks.

Gradients Calibration These approaches adjust the scale [18] or direction [36] of conflicting gradients during model weights updates. They aim to preserve the gradients of consistent directions, which indicate positive synergies among different tasks, while calibrating inconsistent gradients by reducing their magnitudes or adjusting their directions. Gradient calibration can thus be applied in multi-domain transfer learning to balance common knowledge and domain knowledge preservation.

In our work, we apply gradients calibration in a sequential training setup, rather than a simultaneous multi-task training setup. We use the task vectors from other domains as the reference to calibrate the gradients of the task currently being trained. Although this requires extra memory to store these task vectors, it improves the model performance significantly.

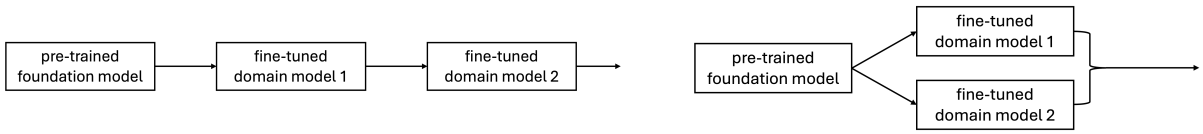
Curriculum Learning It aims to improve machine learning efficiency by gradually increasing the complexity of the training experience, inspired by the natural pattern of the human learning process [25]. This has been shown to help the models achieve better optimization results effectively [22]. However, it requires properly defining the complexity of experience to enhance learning efficiency. Curriculum learning provides valuable guidance for determining the sequential training order among multiple tasks.

Our work applies curriculum learning at the task level instead of the data sample level. We use the magnitude of achievable accuracy by the vanilla training setup as the reference of task difficulty and schedule the sequential training order with gradually increasing complexity.

Continual Learning It involves sequentially training a model based on a stream of tasks, enabling it to adapt through experiences of changing distribution. However, a key challenge is the stability-plasticity trade-off, where the learning plasticity or memory stability of previous tasks could potentially undermine each other [29]. One common strategy is to interpolate experiences by strategically mixing old and new data samples [24]. Another strategy is to interpolate the knowledge between old and new models in weight space [13].

In our work, we concentrate on avoiding conflicts among different domains during the sequential training process, by leveraging the effectiveness of tangent linearization for weight dis-entanglement and gradients calibration techniques for mitigating gradients conflicts.

Multi-domain Image Classification Image classification [15] is a fundamental task in the field of computer vision, for which deep neural network models have been surpassing human performance [6] and becoming the standard method. The advancement of large pre-trained vision models further provides a resource-efficient and high-performance approach for cross-domain and multi-domain transfer learning



(a) Sequential fine-tuning

(b) Separate fine-tuning and interpolation

Figure 1: Comparison of fine-tuning diagrams. (a) our proposed method; (b) baseline method.

in image classification tasks. In this area, numerous previous research efforts have been made focusing on cross-domain **representation learning** [4, 16, 35, 37].

Each of the aforementioned work contributes valuable insights to specific setups. In our work, we leverage the advancement in recent large pre-trained vision models editing and training techniques, to achieve a **more efficient fine-tuning process**, by incorporating the performance of multiple target domains and reference domains. Due to discrepancies in specifications, the experimental results of the above works are not directly comparable with our results. Instead, we measure the effectiveness of our method by comparing it with the baseline approaches proposed in [20], and report the results in Section 5.3.

3. Methods

In this section, we describe our proposed method. In particular, we introduce 3 variations of the implementation in Section 3.3, the approach to determine sequential training order in Section 3.4, the loss function and evaluation metric in 3.5, and code implementation in 3.6.

Considering the multi-domain knowledge fusion nature of the given problem, we apply the transfer learning technique in the sequential training setup. This approach is a simple yet efficient way to achieve multi-domain transfer learning in image classification tasks. We leverage the valuable prior knowledge embedded in the large pre-trained vision models. We further incorporate the effectiveness of tangent linearization and PGGrad in alleviating conflicts of multi-domain knowledge, to achieve better performance among multiple domains. We also adopt curriculum learning to achieve more efficient optimization during sequential training.

Other than the sequential fine-tuning setup, we also experiment with the separate fine-tuning and interpolation setup as a baseline method. An illustration of these 2 training paradigms is shown in Figure 1a and 1b. The specification details of this baseline method are shown in 5.2. The performance advantage of our proposed method is demonstrated in Section 5.3.

3.1. Tangent Linearization

Around the initialization weights $\boldsymbol{\theta}_0$, the tangent linearization method approximates the output of a neural network using a first-order Taylor expansion: $f(\boldsymbol{x};\boldsymbol{\theta}) \approx f(\boldsymbol{x};\boldsymbol{\theta}_0) + (\boldsymbol{\theta}-\boldsymbol{\theta}_0)^T\nabla_{\boldsymbol{\theta}} f(\boldsymbol{x};\boldsymbol{\theta}_0)$. Under fine-tuning setup, the learning rate and number of epochs are usually set as small enough to keep the weights close to the model initialization weights, ensuring the effectiveness of this approximation. This approximation is equivalent to a neural tangent kernel (NTK) [12] predictor, with kernel $k_{NTK}(\boldsymbol{x},\boldsymbol{x}') = \nabla_{\boldsymbol{\theta}} f(\boldsymbol{x};\boldsymbol{\theta}_0)^T\nabla_{\boldsymbol{\theta}} f(\boldsymbol{x}';\boldsymbol{\theta}_0)$, in which the relationship between weights and functions is linear [20, 31]. In the linear regime, this kernel remains constant with respect to weights, enhancing the dis-entanglement between different task data. In our fine-tuning setup, we directly optimize the results based on tangent linear approximation [20].

3.2. PCGrad

PCGrad method [36] projects conflicting gradients to orthogonal space through below procedures. Denote the gradient for task i as $\boldsymbol{g}_i$, and the gradient for task j is $\boldsymbol{g}_j$. PCGrad first determines whether $\boldsymbol{g}_i$ conflicts with $\boldsymbol{g}_j$ by checking whether the cosine similarity between them is negative. If the they conflict, it replaces $\boldsymbol{g}_i$ by its projection onto the normal plane of $\boldsymbol{g}_j$: $\boldsymbol{g}_i = \boldsymbol{g}_i - \frac{\boldsymbol{g}_i\cdot\boldsymbol{g}_j}{\|\boldsymbol{g}_j\|^2}\boldsymbol{g}_j$, otherwise the original gradient remains unchanged.

3.3. Method Specification

We have implemented 3 variations based on the sequential transfer learning setup. We use the model weights fine-tuned from the previous domain as the initial weights for the next transfer learning.

- Seq-TanLin (S-TL): We sequentially fine-tune the models under tangent linearization setup.
- Seq-TanLin-pcg$\{c\}$ (S-TL-p$\{c\}$): We sequentially fine-tune the models under tangent linearization setup. Inspired by the idea of PCGrad [36], when the directions of model gradients disagree with the weight update directions from previous tasks, we apply Gram Schmidt method [26] to calibrate the current gradient towards the orthogonal direction with previous tasks. However, when subtracting the orthogonal projection, we scale the magnitude of orthogonal projection to be subtracted by coefficient c: $\boldsymbol{g}_i = \boldsymbol{g}_i - c\frac{\boldsymbol{g}_i\cdot\boldsymbol{g}_j}{\|\boldsymbol{g}_j\|^2}\boldsymbol{g}_j$. The first task is calibrated with the task vectors of other tasks obtained through B-TL setup that is introduced

Table 1: Dataset details for target image classification tasks.

Dataset Name	EuroSAT	RESISC45	Cars	DTD
Domain	Land use and land cover	Remote sensing image scene	Car image	Textual image
Train:Val:Test Size	21600:2700:2700	17010:1890:6300	7330:814:8041	3384:376:1880
Num Classes	10	45	196	47
Input Resolution	64×64	256×256	360×240	$300 \times 300 \sim 640 \times 640$

Table 2: Dataset details for reference image classification tasks.

Dataset Name	GTSRB	MNIST	SUN397	SVHN
Domain	Traffic sign	Handwritten digits	Scene understanding	Street view house numbers
Train:Val:Test Size	23976:2664:12630	55000:5000:10000	17865:1985:19850	68257, 5000, 26032
Num Classes	43	10	397	10
Input Resolution	$15 \times 15 \sim 250 \times 250$	28×28	At most 120,000 pixels	32×32

in Section 5.2. This pre-calibration ensures the start of the sequential training is more aligned with the direction that is inherently less conflicting with downstream domains, and it demonstrates better empirical performance.

- Seq-TanLin-pcg$\{c\}$-multiround$\{n\}$ (S-TL-p$\{c\}$-$\{n\}$): This is a extension of S-TL-p$\{c\}$, by running the sequential training multiple rounds. We define one round of training as iterating all the target tasks once. In this setup, we divide the number of training epochs for each task per round by n, but run n rounds of training in total. Each round starts with the checkpoint obtained from the last task in the previous round. We always use the latest available task vector for each task as the reference to conduct conflicting gradients calibration. This multi-round training regime keeps the total number of training epochs for each task the same as S-TL-p$\{c\}$. Higher n indicates higher calibration frequencies within the same number of training epochs.

3.4. Sequential Training Order

We sequentially fine-tune the models by the order of EuroSAT, RESISC45, Cars, and DTD. This is the ascending order of the task complexity, measured by the descending magnitude of achievable accuracy with the vanilla single-domain fine-tuning setup without tangent linearization.

3.5. Loss and evaluation metric

We use Cross Entropy loss as training objective, and use the average accuracy to evaluate and compare the performances of different approaches.

3.6. Code Implementation

We implement our methods based on the implementations of [20], [36] and [18]. Specifically, we take reference to below Github repositories: [5], [27], [30] and [17]. We structure our code based on [5], by reusing the fine-tuning and evaluation script for the baseline methods. We develop the gradients calibration scripts under sequential training setup by referring to the implementation design of [30] and algorithm logics of [27] and [17]. We design and build the training and evaluation scripts of the proposed method from scratch, including the sequential training setup, the gradient calibration, as well as the flexible and automated pipeline to conduct thorough hyperparameters analysis. Our code is implemented in Python [28] and Pytorch [21].

4. Datasets

Data Description We incorporate 8 image classification datasets in the current experimental setup. Specifically, we select EuroSAT [7], RESISC45 [1], Cars [14], DTD [2] as target datasets; and GTSRB [8], MNIST [3], SUN397 [34] [33], SVHN [19] as the reference tasks. The target tasks are explicitly optimized during the fine-tuning process, and the reference tasks are solely used as auxiliary tasks to evaluate the generalizability of the model. We aim to maximize the performances on the target datasets, and minimize the performance degradation on the reference tasks.

The details of target image classification datasets are shown in Table 1, and the details of reference image classification datasets are shown in Table 2.

Data Downloading and Processing We download the datasets by following the instructions of their respective reference papers. Each dataset is partitioned into train, validation, and test splits, following the specification of [9] [20]. For the target tasks, the validation split is used for hyperparameter search, and the test split is used for evaluation accuracy score reporting. For the reference tasks, we only use the test split to report the evaluation accuracy scores. The input images are preprocessed through the standard CLIP model preprocess function [11]. No extra preprocessing,

feature extraction, or data augmentation is conducted other than the above-listed procedures.

5. Experiments

In this section, we introduce the details of the experiments, results, and analysis. Specifically, we describe the training setups in 5.1, introduce the baseline methods in 5.2, present the experimental results and analysis in 5.3, finally we further study the factors that impact the performance in 5.4.

5.1. Training Setup

Calibration Strength Different from [36] which calibrates the conflicting gradients with fixed strength, in our setup we introduce the coefficient c ($0 \leq c \leq 1$) in S-TL-p{c} as a tunable hyperparameter that balances the strength of conflict calibration and domain-specific knowledge preservation. A higher value of c prioritizes more on conflict avoidance, and a lower value of c prioritizes more on domain information preservation. Configuring $c = 1$ will yield an equivalent setup as [36]. In our current experimental configuration, we experiment with 0 (S-TL), 0.5 (S-TL-p0.5), and 1 (S-TL-p1). In the multi-round setup, we report results based on n=11 (S-TL-p0.5-11).

Regular Hyperparameters The number of total training epochs for each task, learning rate, and optimizers follow the setup of [20] and are kept consistent on the same task across all experimental setups that are used to compare the performance of different training methods. Specifically, the optimizer is AdamW, the learning rate is $1e-5$, the batch size is 128, the total training epochs for EuroSAT, RESISC45, Cars, and DTD are respectively 12, 11, 35, 76. We confirm that further increasing training epochs still keeps improving evaluation accuracy scores, hence the models have not been overfitted to training data under this setup.

We use the same hyperparameters as the reference paper that proposes the baseline method in the main experimental results reporting to facilitate comparison. We further present an analysis of the impact of hyperparameters such as calibration strength, learning rate, and the number of total training epochs in Section 5.4.

5.2. Baseline method

We compare our approach with 3 baseline methods.

- Baseline-standard (B-Stand): We separately fine-tune each target image classification task under the standard setup, and combine these specialized models into a unified model by weights interpolation.
- Baseline-TanLin (B-TL): We separately fine-tune each target image classification task under the tangent linearization setup, and combine these specialized models into a unified model by weights interpolation.
- Seq-standard (S-Stand): We sequentially fine-tune each target image classification task under the standard setup, and obtain a single final model.

For the implementation of the B-Stand and B-TL methods, we follow the setup of [20]. The task vector interpolation coefficients among different specialized domain tasks are set to be equal. The coefficient to interpolate task vectors with pre-trained models is tuned by maximizing average target task accuracy, through grid search between [0, 1] (inclusive) with step size 0.05. This is a generalization of direct equally weighted interpolation among domain expert models, with the flexibility of adjusting the contribution of pre-trained model weights. When the task vector interpolation coefficient is 0.25, it's equivalent to the simple average of fine-tuned domain expert models. For the S-Stand method, we follow the same sequential fine-tuning order as in Section 3, and the first task is pre-calibrated with the task vectors of other tasks obtained through B-TL setup.

5.3. Experimental Results

We report results using average accuracy scores, following the evaluation metric of baseline paper [20] for easier comparison. Figure 2 shows the bar plot of accuracy scores by dataset. To facilitate comparison, we further summarize the average accuracy scores on target, reference, and all tasks in Table 3. Below we summarize our main findings.

Table 3: Summary of average accuracy scores. The rows "tgt", "ref", and "all", represent the average scores of target tasks, reference tasks, and all tasks. The columns represent (1) CLIP: the pre-trained CLIP ViT-B/32 model without fine-tuning; (2) $\sim$ (8): our proposed method in Section 3.3 and baseline methods in Section 5.2. The best performance has been highlighted in **bold**.

	Baseline				Our Methods			
Acc	CLIP	B-Stand	B-Lin	S-Stand	S-TL	S-TL-p1	S-TL-p0.5	S-TL-p0.5-11
tgt	0.525	0.716	0.798	0.796	0.850	0.847	**0.852**	0.846
ref	0.439	0.201	0.301	0.263	0.352	0.387	0.379	**0.396**
all	0.482	0.458	0.549	0.530	0.601	0.617	0.615	**0.621**

Fine-tuning improves domain-specific performance but degrades out-of-domain generalizability. Compared with the out-of-box CLIP model, on average all fine-tuning methods improve the accuracy scores on target tasks ($19\% \sim 33\%$), but all suffer from performance degradation on reference tasks ($4\% \sim 24\%$). This indicates the fine-tuning processes inject domain-specific knowledge to the model, but lose some common knowledge useful for generalization capabilities.

Sequential training setup significantly outperforms separate fine-tuning and interpolation setup. With other

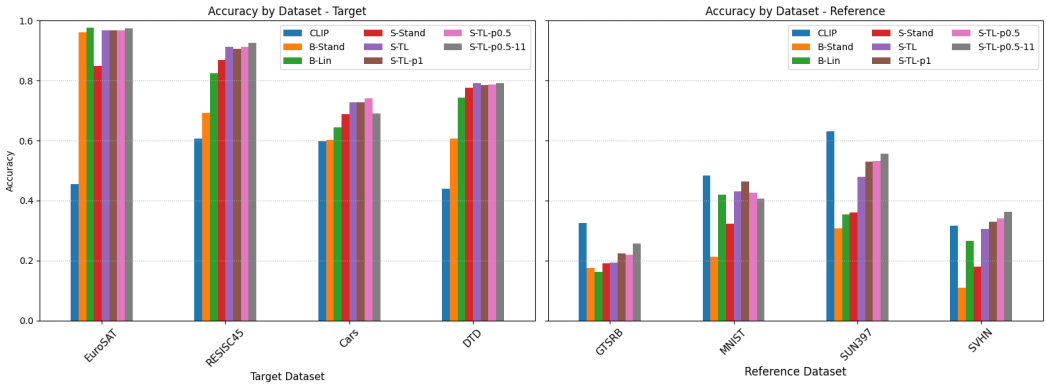


Figure 2: Accuracy scores by dataset.

conditions being the same, the sequential training setup brings $6\% \sim 8\%$ more improvement in target tasks and $5\% \sim 6\%$ less degradation in reference tasks.

The tangent linearization fine-tuning regime significantly outperforms the standard fine-tuning regime. Under both separate fine-tuning and interpolation setup, and sequential training setup, tangent linearization fine-tuning approach brings respectively 8% and 5% more improvement on target tasks, as well as 10% and 9% less degradation in reference tasks.

Conflicting gradient calibration effectively balances domain-specific and general knowledge preservation. Among all approaches, the sequential tangent linearization fine-tuning setup with PCGrad calibration achieves the most improvement on the target tasks, while suffering the least degradation on the reference tasks. When the gradient calibration coefficient is set to 1, it better preserves information for the reference domain, i.e. less degradation (5%) on the reference tasks. When set to 0.5, it better preserves information for the target domain, i.e. more improvement (33%) on target tasks. Keeping calibration strength the same as 0.5, when the calibration frequency is increased to 11, the reference task accuracy further increases by 1.7%, with only 0.6% degradation in target task accuracy, demonstrating calibration frequency as another effective factor in balancing knowledge preservation, like the calibration strength coefficient.

Our proposed approaches show promising performance improvements over the baseline methods. Compared with the 3 baseline methods, our best-performing setup S-TL-p0.5-11 shows $6\% \sim 14\%$ higher average accuracy on target tasks, and $10\% \sim 20\%$ lower average degradation on reference tasks.

5.4. Ablation Study

In this section, we conduct a thorough analysis of the factors that impact the performance of the proposed methods.

Sequential Training Order Based on S-TL-p0.5, we collect experimental results for all possible sequential orders. The statistics of the results are shown in Table 4. We can see that the standard deviations of average accuracy scores are less than 0.01, indicating the method is relatively robust to sequential training orders. The gaps between the maximum and minimum accuracy scores are around 0.03, indicating some space for performance improvement from strategic sequential training ordering. The best performances on average accuracy scores on target and all tasks are achieved under the curriculum learning setup as reported in Section 5.3, where the sequential training order is by ascending order of task complexity. The best performance on reference tasks is achieved with order [RESISC45, Cars, EuroSAT, DTD], whose average accuracy scores on target and all tasks are 0.842 and 0.611.

Table 4: Summary statistics for final average accuracy scores from all 24 sequential training orders. The rows represent target tasks (tgt), reference tasks (ref), and all tasks (all). The columns represent mean, standard deviation, minimum, and maximum values.

	Mean	Std Dev	Min	Max
tgt	0.8383	0.0092	0.8181	0.8516
ref	0.3691	0.0090	0.3522	0.3802
all	0.6037	0.0071	0.5855	0.6153

Gradients Calibration Strength This is an important factor that impacts the balance between the target domain and reference domain knowledge preservation during the training process. In Figure 3, we show the plots of average accuracy scores versus the **calibration frequency**,

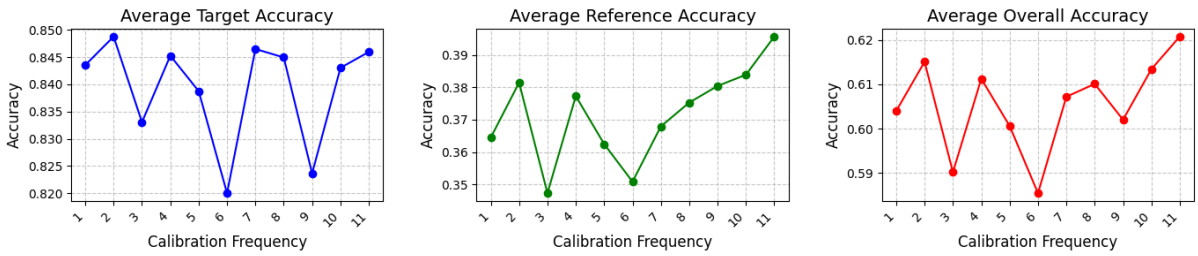


Figure 3: Accuracy scores by calibration frequency.

with the total number of training epochs fixed. We can see that higher calibration frequency yields better reference task performance, while roughly preserving the target task performance, and achieves better overall performance. This effect is similarly shown with the **calibration coefficient**. In Table 3 we can see that as the calibration coefficient increases, by varying among 0 (S-TL), 0.5 (S-TL-p0.5), and 1 (S-TL-p1), the target task performance holds roughly constant, but the reference task performance improves significantly, and it achieves increasing overall performance.

This is potentially because the PCGrad calibration only projects the conflicting gradients while preserving the non-conflicting gradients. Higher calibration strength guides the gradient updates toward the direction that more effectively preserves the common knowledge that brings positive synergy among multiple domains, and this common knowledge is likely valuable for more general tasks as well. The better preservation of this common knowledge is beneficial for preserving reference task performance. The target task performance holds relatively constant across calibration frequency, potentially because the model capacity achieved with over-parameterization is large enough to provide ideal solution space for each task, even given tighter gradient update constraints. In addition, the tangent linearization technique effectively alleviates conflicts among the tasks that are explicitly optimized during the training process. Therefore, even with a lower calibration strength, it could still achieve a relatively good balance in knowledge preservation among different target tasks, compared with higher calibration strengths.

Alternative Calibration Method Our proposed framework provides flexibility in applying alternative calibration methods to detect and calibrate the conflicting gradients among different domains. One example is to check whether the signs of gradients agree with each other, which indicates the consistency of weight update directions needed by different tasks [18]. When conflicts occur, we can scale the magnitude of the conflicting gradients by $(1 - c)$. When $c = 1$ it zeros out the conflicting gradients, which is equivalent to full-strength calibration. When $c = 0$ it scales the weights by 1, which is equivalent to no calibration. A performance comparison among the sign agreement calibration method with calibration coefficient of 0.5, our proposed method, and baseline methods, is shown in Table 5. We can see that, the sign agreement method outperforms all the baseline methods. Compared with the S-TL setup which is without conflicting gradients calibration, the sign agreement method achieves better performance on the reference task and the overall result, but worse performance on the target task. It under-performs the PCGrad method in all the target, reference, and overall tasks. This is potentially because, PCGrad redirects the gradients' directions without constraining the gradient magnitudes, which still sufficiently effectively explores the large solution space enabled by the model over-parameterization. However, the sign agreement method keeps the gradient direction but only shrinks the gradient magnitude, which poses stronger constraints, more heavily reducing the expressivity of the model and the explorable solution space that could lead to ideal performances during optimization.

Table 5: Summary of average accuracy scores. The rows "tgt", "ref', and "all", represent the average scores of target tasks, reference tasks, and all tasks. The columns represent (1) CLIP: the pre-trained CLIP ViT-B/32 model without fine-tuning; (2) $\sim$ (6): our proposed method in Section 3.3 and baseline methods in Section 5.2; (7) Sign agreement method. The best performance has been highlighted in **bold**.

	Baseline				Our Methods		Sign Agr
Acc	CLIP	B-Stand	B-Lin	S-Stand	S-TL	S-TL-p0.5	S-TL-a0.5
tgt	0.525	0.716	0.798	0.796	0.850	**0.852**	0.845
ref	0.439	0.201	0.301	0.263	0.352	**0.379**	0.375
all	0.482	0.458	0.549	0.530	0.601	**0.615**	0.610

Learning Rate and Total Training Epochs They impact the balance between target knowledge and reference knowledge by adjusting the extent of deviation from original pre-trained weights during the fine-tuning process. The smaller learning rate and less total training epochs tend to make the model weights stay closer to the start point, hence natu-

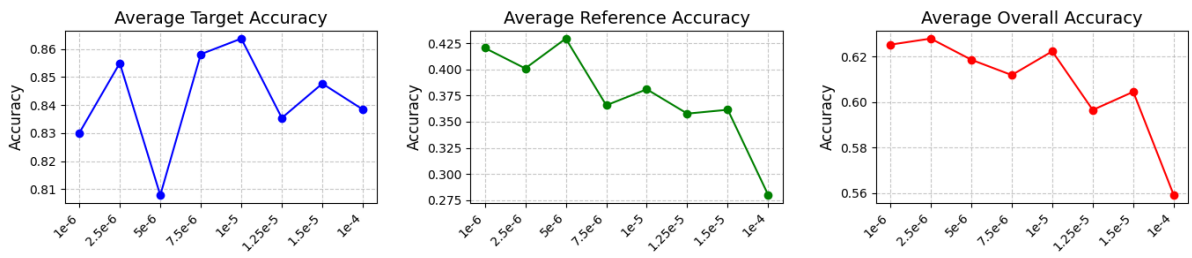


Figure 4: Accuracy scores by learning rates, based on S-TL-p0.5-20 setup.

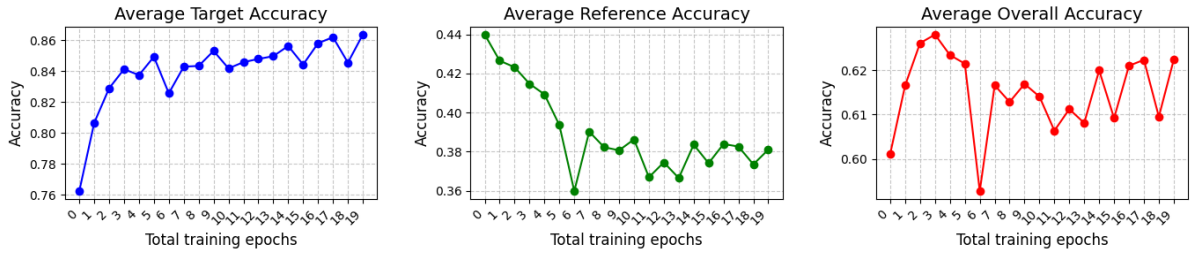


Figure 5: Accuracy scores by total training epochs, S-TL-p0.5-20 setup.

rally preserving more prior knowledge embedded in the pre-trained weights. Due to the nature of the high transferability of pre-trained foundation models, smaller deviations from it tend to preserve better performance on reference tasks. This can be verified by the "Average Reference Accuracy" plots in Figure 4 and Figure 5. In our experiment, the optimal learning rates are around $1e-6 \sim 1e-5$, depending on the objective balance between target and reference tasks. The higher number of total training epochs tends to increase the target task performance, but marginal gains decay gradually. The total performance is determined collectively by the increasing trend of target performance and the decreasing trend of reference performance. The best value depends on the ideal balance between the target domain and reference domain knowledge preservation. When setting the learning rate as $2.5e-6$ under the S-TL-p0.5-20 setup, it achieves the average accuracy scores on the target task, reference task, and the overall result of 0.855, 0.401, 0.628. This is the best performance of all setups in our experiments, outperforming all the other setups in all the 3 metrics.

6. Conclusion and Future work

In this work, we propose a sequential training paradigm based on tangent linearization and conflicting gradients calibration techniques. Our experiments focus on multi-domain transfer learning for image classification tasks. The experimental results demonstrate that sequential training setup significantly outperforms separate fine-tuning and interpolation configuration. The tangent linearization fine-tuning method yields substantial performance gains by effectively magnifying weight dis-entanglements across different domains. Conflicting gradient calibration provides an effective approach to balance domain-specific and general knowledge retention. The combination of the above techniques yields the best performance in our experiments, potentially due to their effectiveness in alleviating conflicts during knowledge fusion across different domains. Within the gradient calibration setup, the calibration coefficient and the calibration frequency are the key factors in balancing conflict mitigation and domain knowledge preservation. In the general sequential training context, the learning rate and the total number of training epochs also play crucial roles in balancing the performance of target and reference tasks, by adjusting the extent to which the fine-tuned model weights deviate from the original foundation model weights.

Although tangent linearization and PCGrad introduce increased computational complexity, regular fine-tuning efforts generally suffice to achieve efficient and stable learning performance. The high expressivity and well-initialized weights of the pre-trained model might be critical to this robustness, and reducing sensitivity to hyperparameter variations.

In future studies, we will investigate the combined impact of different hyperparameters. We will also aim to explore more foundation models across a wider range of tasks, such as image segmentation, object detection, image captioning, and vision question answering. Additionally, we plan to extend our research to include more modalities and domains, such as natural language, speech, video, robotics, and reinforcement learning.

References

[1] G. Cheng, J. Han, and X. Lu. Remote sensing image scene classification: Benchmark and state of the art. *Proceedings of the IEEE*, 105(10):1865–1883, 2017.

[2] M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, , and A. Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[3] L. Deng. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012.

[4] N. Dvornik, C. Schmid, and J. Mairal. Selecting relevant features from a multi-domain representation for few-shot classification. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part X 16*, pages 769–786. Springer, 2020.

[5] gortizji. Tangenttaskarithmetic. `https://github.com/gortizji/tangent_task_arithmetic`, 2023. commit=0aa0e1869d8b4b111fa92f2217b8ae863c084fc6.

[6] K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015.

[7] P. Helber, B. Bischke, A. Dengel, and D. Borth. Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 12(7):2217–2226, 2019.

[8] S. Houben, J. Stallkamp, J. Salmen, M. Schlipsing, and C. Igel. Detection of traffic signs in real-world images: The German Traffic Sign Detection Benchmark. In *International Joint Conference on Neural Networks*, number 1288, 2013.

[9] G. Ilharco, M. T. Ribeiro, M. Wortsman, S. Gururangan, L. Schmidt, H. Hajishirzi, and A. Farhadi. Editing models with task arithmetic. *arXiv preprint arXiv:2212.04089*, 2022.

[10] G. Ilharco, M. Wortsman, S. Y. Gadre, S. Song, H. Hajishirzi, S. Kornblith, A. Farhadi, and L. Schmidt. Patching open-vocabulary models by interpolating weights. *Advances in Neural Information Processing Systems*, 35:29262–29277, 2022.

[11] G. Ilharco, M. Wortsman, R. Wightman, C. Gordon, N. Carlini, R. Taori, A. Dave, V. Shankar, H. Namkoong, J. Miller, H. Hajishirzi, A. Farhadi, and L. Schmidt. Openclip, July 2021. If you use this software, please cite it as below.

[12] A. Jacot, F. Gabriel, and C. Hongler. Neural tangent kernel: Convergence and generalization in neural networks. *Advances in neural information processing systems*, 31, 2018.

[13] J. Kozal, J. Wasilewski, B. Krawczyk, and M. Woźniak. Continual learning with weight interpolation. *arXiv preprint arXiv:2404.04002*, 2024.

[14] J. Krause, M. Stark, J. Deng, and L. Fei-Fei. 3d object representations for fine-grained categorization. In *4th International IEEE Workshop on 3D Representation and Recognition (3dRR-13)*, Sydney, Australia, 2013.

[15] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017.

[16] Y. Liu, X. Tian, Y. Li, Z. Xiong, and F. Wu. Compact feature learning for multi-domain image classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7193–7201, 2019.

[17] lucasmansilla. Dgvgs. `https://github.com/lucasmansilla/DGvGS/commits/main`, 2022. commit=c2b69f06c6a62963570ca75c9c4148381102f9b1.

[18] L. Mansilla, R. Echeveste, D. H. Milone, and E. Ferrante. Domain generalization via gradient surgery. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6630–6638, 2021.

[19] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, A. Y. Ng, et al. Reading digits in natural images with unsupervised feature learning. In *NIPS workshop on deep learning and unsupervised feature learning*, volume 2011, page 7. Granada, Spain, 2011.

[20] G. Ortiz-Jimenez, A. Favero, and P. Frossard. Task arithmetic in the tangent space: Improved editing of pre-trained models. *Advances in Neural Information Processing Systems*, 36, 2024.

[21] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

[22] E. A. Platanios, O. Stretcu, G. Neubig, B. Poczos, and T. M. Mitchell. Competence-based curriculum learning for neural machine translation. *arXiv preprint arXiv:1903.09848*, 2019.

[23] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[24] D. Rolnick, A. Ahuja, J. Schwarz, T. Lillicrap, and G. Wayne. Experience replay for continual learning. *Advances in neural information processing systems*, 32, 2019.

[25] P. Soviany, R. T. Ionescu, P. Rota, and N. Sebe. Curriculum learning: A survey. *International Journal of Computer Vision*, 130(6):1526–1565, 2022.

[26] G. Strang. *Introduction to linear algebra*. SIAM, 2022.

[27] tianheyu927. Pcgrad. `https://github.com/tianheyu927/PCGrad/tree/master`, 2020. commit=c5fbd7c856526373828074f06875230f7f3ee79e.

[28] G. Van Rossum and F. L. Drake Jr. Python tutorial, 1995.

[29] L. Wang, X. Zhang, H. Su, and J. Zhu. A comprehensive survey of continual learning: Theory, method and application. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

[30] WeiChengTseng. Pytorch-pcgrad. `https://github.com/WeiChengTseng/Pytorch-PCGrad/commits/master`, 2021. commit=e987ac603fa1accd386820a985a6dc2fd92dec5b.

[31] L. Weng. Some math behind neural tangent kernel. *Lil'Log*, Sep 2022.

[32] M. Wortsman, G. Ilharco, J. W. Kim, M. Li, S. Kornblith, R. Roelofs, R. G. Lopes, H. Hajishirzi, A. Farhadi, H. Namkoong, et al. Robust fine-tuning of zero-shot models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7959–7971, 2022.

[33] J. Xiao, K. A. Ehinger, J. Hays, A. Torralba, and A. Oliva. Sun database: Exploring a large collection of scene categories. *International Journal of Computer Vision*, 119:3–22, 2016.

[34] J. Xiao, J. Hays, K. A. Ehinger, A. Oliva, and A. Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *2010 IEEE computer society conference on computer vision and pattern recognition*, pages 3485–3492. IEEE, 2010.

[35] B. Yang, S. Hu, Q. Guo, and D. Hong. Multisource domain transfer learning based on spectral projections for hyperspectral image classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 15:3730–3739, 2022.

[36] T. Yu, S. Kumar, A. Gupta, S. Levine, K. Hausman, and C. Finn. Gradient surgery for multi-task learning. *Advances in Neural Information Processing Systems*, 33:5824–5836, 2020.

[37] Y. Zhu, F. Zhuang, J. Wang, J. Chen, Z. Shi, W. Wu, and Q. He. Multi-representation adaptation network for cross-domain image classification. *Neural Networks*, 119:214–221, 2019.