Enhancing Domain Adaptation through Prompt Gradient Alignment

Hoang Phan[*]
New York University
hvp2011@nyu.edu

Lam Tran[*]
VinAI Research
lamtt12@vinai.io

Quyen Tran[*]
VinAI Research
quyentt15@vinai.io

Trung Le
Monash University
trunglm@monash.edu

Abstract

Prior Unsupervised Domain Adaptation (UDA) methods often aim to train a domain-invariant feature extractor, which may hinder the model from learning sufficiently discriminative features. To tackle this, a line of works based on prompt learning leverages the power of large-scale pre-trained vision-language models to learn both domain-invariant and specific features through a set of domain-agnostic and domain-specific learnable prompts. Those studies typically enforce invariant constraints on representation, output, or prompt space to learn such prompts. Differently, we cast UDA as a multiple-objective optimization problem in which each objective is represented by a domain loss. Under this new framework, we propose aligning per-objective gradients to foster consensus between them. Additionally, to prevent potential overfitting when fine-tuning this deep learning architecture, we penalize the norm of these gradients. To achieve these goals, we devise a practical gradient update procedure that can work under both single-source and multi-source UDA. Empirically, our method consistently surpasses other vision language model adaptation methods by a large margin on a wide range of benchmarks. The implementation is available at https://github.com/VietHoang1512/PGA.

1 Introduction

Deep learning has significantly advanced the field of computer vision, achieving remarkable performance in tasks such as image classification [1–5], object detection [6–9], and semantic segmentation [10–13]. However, the effectiveness of these deep learning models heavily relies on large amounts of labeled training data, which is often labor-intensive and expensive to collect. Moreover, the discrepancy between training data and real-world testing data can lead to substantial performance drops when models are deployed in practical settings [14–16].

To address these challenges, Unsupervised Domain Adaptation (UDA) has emerged as a pivotal solution. UDA aims to transfer knowledge from a labeled source domain to an unlabeled target domain in the presence of a domain shift, thereby enabling models to generalize well across different domains without requiring extensive labeled data for the target domain. This is often achieved by optimizing objective function on source domains and other auxiliary terms that encourage learning domain-invariant feature representations [17–20] or enhance model robustness [21–24], which mitigates the domain shift and improve the performance on unseen data. Nevertheless, aligning representations could potentially hurt the model performance due to the loss of discriminative features [25, 26].

[*] Equal contributions.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

https://doi.org/10.52202/079017-1447

Conceptually, our proposed method is orthogonal to these invariant feature learning methods, and they could complement each other.

Recent works leveraging pre-trained models like CLIP [27] for UDA can significantly bridge domain gaps and improve generalization by utilizing rich semantic information and robust visual representations through extensive pre-training on diverse image-text datasets. Following this vein, DAPL [25] first introduces domain-specific and domain-agnostic prompts to efficiently adapt pre-trained vision-language models without fine-tuning the entire model. Furthermore, MPA [28] aligns multiple prompts from different sources using an auto-encoder. While these methods could obtain superior performance on different benchmarks, we find that the main improvement comes from the strong zero-shot performance and self-training mechanism. In particular, prior works often generate pseudo-label for unlabeled images and then train the model on those samples. Consequently, finetuning a pretrained CLIP model on this dataset alone without leveraging source datasets can help refine model prediction significantly, boosting the performance from 88.1% to 90.1%, yielding a competitive result compared against MPA, as presented in Table 1.

Dataset	$\rightarrow$ **C**	$\rightarrow$ **I**	$\rightarrow$ **P**	**Avg**
Zero-shot	87.9	88.2	78.7	88.1
Simple Prompt	93.6	90.6	80.9	88.4
Self-training	92.9	94.3	83.2	90.1
MPA	97.2	96.2	80.4	91.3

Table 1: Self-training on pseudo-labeled target data is already a strong baseline.

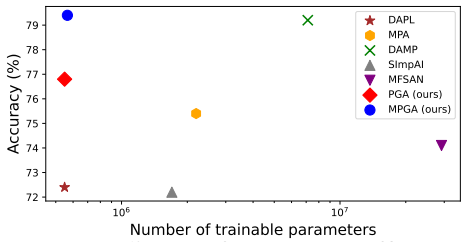

Figure 1: Baselines performance on Office-Home

Motivated by this observation, we directly optimize the main objective function not only on source domains but also on the target data, instead of only using them for auxiliary objectives as in previous work [29, 30]. We thus cast the original UDA problem as a multi-objective optimization (MOO) problem. Specifically, we minimize a vector-valued loss function, which includes the objectives of multiple source domains and the target domain. This formulation allows us to apply existing results from MOO literature for finding Pareto solutions, from which we can not optimize an objective without hurting another [31–33] or encourage positive inter-task transfer between objectives [34–37]. Note that in the context of UDA, we focus more on learning the target task, thus motivating us to apply prioritized MOO algorithms [38–40] or to incorporate predefined preferences [32, 41–43]. While those methods allow practitioners to focus more or less on the objectives at hand, they come with the cost of extensive hyperparameter tuning. Besides, recent works [44–46] argue that simple loss functions reweighting can match the performance of gradient-based MOO methods [35, 47]. Those findings suggest we focus more on the inherent conflict nature of per-objective gradients instead of attempting to remove the conflict between them [35, 34].

In this paper, we propose casting the problem of UDA as a multi-objective optimization by leveraging powerful pre-trained models. However, while obtaining impressive results on various downstream tasks, over-parameterization is still a crucial problem for transformer-based models [48, 49], which potentially causes overfitting [50–54] more severely than small-parameter architectures [55], especially in the multi-task learning context. For that reason, we propose to (i) fine-tune pre-trained model via prompt learning, which is known for being more robust [56–58] and especially more light-weight than full fine-tuning, (ii) and to leverage the gradient norm penalty to encourage model generalization [59, 24, 60, 61]. Furthermore, we introduce a gradient alignment algorithm to foster inherent consensus between per-objective gradients without modifying the gradient itself. Our proposed method, termed Prompt Gradient Alignment (PGA), and its variant for multi-source UDA, Multi-Prompt Gradient Alignment (MPGA), achieve state-of-the-art performance on different UDA benchmarks. As shown in Figure 1, PGA and MPGA outperform traditional UDA methods like MFSAN [62] and recent prompt-based UDA methods such as MPA [28] and DAMP [63] while requiring fewer trainable parameters. We also provide a generalization bound for UDA and show how theoretical insights motivate the design of our proposed method.

https://doi.org/10.52202/079017-1447

2 Related work

Unsupervised Domain Adaptation. A dominant approach to solving the UDA problem is to reduce the distribution shift between source and target domains. Following the foundational theory outlined in [64], one group of methods seeks to minimize the H-divergence between the marginal distributions of these domains [65–67]. Alternatively, other methods aim to align the moments of these distributions, as suggested in [68–70]. Additionally, adversarial learning has been employed to learn domain-invariant features. For instance, methods such as those in [19, 71] use a domain discriminator to differentiate between source and target samples, training a feature extractor to deceive this discriminator. However, as [25] highlights, these methods often struggle with a trade-off between domain alignment and classification performance, particularly in multi-source scenarios where only a single model is used.

Prompt learning-based domain adaptation is a novel approach introduced in [25], leverages the generalization capabilities of CLIP to learn both domain-agnostic and domain-specific prompts. This method effectively addresses the trade-off between domain alignment and classification performance by employing a contrastive loss. This loss aligns the representation of an image with the prompt corresponding to its ground truth class and domain, thereby encouraging the learning of domain-invariant features. Building on this foundation, MPA [28] advances the concept of multi-source UDA. It adapts the prompt learning strategy to each source-target domain pair. The prompts are aligned through a denoising auto-encoder using Euclidean distance. However, prompting is known as a brittle process where a small shift to the prompt can cause large variations in the model predictions [72–74]. Therefore, in this work, we propose to intervene in the training on the gradient space as it offers a more interpretable and controllable effect during training. Furthermore, PGA is trained in an end-to-end fashion, avoiding the sequential training for each source-target pair as in MPA.

Gradient-based multi-task learning. Due to the multi-objective nature of the multi-source domain adaptation problem, one can leverage recent methods from the multi-task learning literature [34, 35, 75] to derive an optimization procedure that benefits the learning across domains or put more weight on some specific domains via incorporating preference [32, 41, 42]. While those techniques are readily applicable in our context, we directly re-weight per-task gradients, similar to scalarization, instead of adopting multi-task learning methods for simplicity. Furthermore, our work is orthogonal to those gradient-based multi-task learning methods where we encourage the consensus among objects instead of directly manipulating their gradients to remove inherent conflicts among them.

Gradient matching is commonly used in continual learning [76–78] to measure conflict and transferability between tasks. A positive dot product between two tasks' gradients implies updating the models with one task can benefit the other. This principle is also applied in domain generalization [79, 80] to focus on invariant features. However, our approach aligns in the space of prompt gradient, a significantly smaller parameter set than the full model gradients used in previous works. Besides, to avoid the computation of costly second-order derivatives, [79] linearly approximate the inner product between gradients, which underperforms on datasets with a larger number of domains due to cumulative approximation error. Meanwhile, our method does not face this problem since we implicitly compute this term without using any approximation. More works sharing the same intuition of gradient alignment are provided in Appendix D.

3 Background

3.1 Unsupervised Domain Adaptation

Given a set of $N \geq 1$ source domains $\{D_{S,i}\}_{i=1}^{N}$ each of which is a collection of data-label pairs of domain i, i.e. $D_{S,i} = \{\boldsymbol{x}_j, y_j\}_{j=1}^{N_{S,i}}$, and one unlabelled target domain $D_T = \{\boldsymbol{x}_j\}_{j=1}^{N_T}$, where $N_{S,i}$ and N_T are respectively the number of data points in source domain i and target domain T, the goal is to learn a model that can perform well on the unlabelled target domain. In this paper, we focus on classification problems and denote K as the number of categories.

3.2 Prompt Learning on CLIP-based models

CLIP [27] is a vision-language model that consists of an image encoder f_i and a text encoder f_t, which is trained to align the visual representation $f_i(\boldsymbol{x})$ of an image $\boldsymbol{x}$ with the textual representation

https://doi.org/10.52202/079017-1447

$f_t(y)$ of the corresponding label. The textual representation is derived from a manually crafted prompt $\boldsymbol{p}_k$ in the form "A photo of a $[\text{CLASS}]_k$", where $[\text{CLASS}]_k$ is the class k's name. With great generalization capability, pre-trained CLIP models are often used for a variety of downstream tasks through prompt learning.

For zero-shot inference, K class names are forwarded through the text encoder, and the one with the highest representation similarity with the image is the predicted class:

$$y_{\text{pred}} = \text{argmax}_k P(y=k|\boldsymbol{x}), \text{ where } P(y=k|\boldsymbol{x}) = \frac{\exp(\langle f_i(\boldsymbol{x}), f_t(\boldsymbol{p}_k)\rangle/\gamma)}{\sum_{k'=1}^{K}\exp(\langle f_i(\boldsymbol{x}), f_t(\boldsymbol{p}_{k'})\rangle/\gamma)}, \quad (1)$$

and $\langle ., . \rangle$ measures the cosine similarity and γ is the temperature.

For fine-tuning, a set of learnable class-shared prompts are added to the class token to form $\boldsymbol{P}_k = [\boldsymbol{v}_1|\boldsymbol{v}_2|\cdots|\boldsymbol{v}_M][\text{CLASS}]_k$, where $\boldsymbol{v}_i$ is a vector with the same size as the word embedding, and M is the number of added prompts. These prompts are learnt by maximizing log-likelihood on downstream data, i.e. $\max \sum_i \log P(y=y_i|\boldsymbol{x}_i, \boldsymbol{P})$. Note that in this predictive probability, we abuse symbol $\boldsymbol{P}$ to refer to the learnable tokens $\boldsymbol{v}_i$, and when we drop the symbol as in 1, we refer to the zero-shot prediction using CLIP. As a result, additional information about the downstream task can be encoded in the prompts, and this design will enable knowledge transfer from pre-trained datasets.

4 Proposed method

In this section, we describe our proposed prompt gradient alignment method. Motivated by the lightweight and effective nature of prompt learning in adapting pre-trained knowledge to downstream tasks, we cast UDA as a multi-objective optimization (MOO) problem, from which we propose aligning gradients of different objectives and minimizing their norms simultaneously. Additionally, we derive a UDA generalization bound to justify the intuition of our method. The full details of our proposed method in the generalized case where we have more than one source domain are provided in Appendix B.

4.1 Prompt design

A common assumption in domain adaptation literature is that each domain can be represented by domain-specific features and those that are shared with others. To reflect this, we employ two sets of prompts for each domain: domain-agnostic prompt (or shared prompt, interchangeably) $\boldsymbol{P}_{sh}$, and domain-specific prompts $\boldsymbol{P}_{S,i}$ and $\boldsymbol{P}_T$. Here, $\boldsymbol{P}_{S,i}$ refers to prompt used for source domain i, and $\boldsymbol{P}_T$ is that for target one. In particular, following DAPL, we use $K \times M_1$ tokens to construct $\boldsymbol{P}_{sh} = [\boldsymbol{P}_{sh}^k]_{k=1}^K$, where $\boldsymbol{P}_{sh}^k = [\boldsymbol{v}_1^k|\boldsymbol{v}_2^k|\cdots|\boldsymbol{v}_{M_1}^k]$ is class-specific shared tokens. For source- and target-specific prompts, we use M_2 tokens: $\boldsymbol{P}_{S,i} = [\boldsymbol{u}_1^{S,i}|\boldsymbol{u}_2^{S,i}|\cdots|\boldsymbol{u}_{M_2}^{S,i}]$, $\boldsymbol{P}_T = [\boldsymbol{u}_1^T|\boldsymbol{u}_2^T|\cdots|\boldsymbol{u}_{M_2}^T]$. And denote $\boldsymbol{P} = [\boldsymbol{P}_{sh}, \{\boldsymbol{P}_{S,i}\}_{i=1}^N, \boldsymbol{P}_T]$ as the whole prompts used in our method. Based on this, we use a prompt of the form $[\boldsymbol{P}_{sh}^k][\boldsymbol{P}_{S,i}][\text{CLASS}]_k$ to compute the predictive distribution of a source i sample belonging to class k, and similarly $[\boldsymbol{P}_{sh}^k][\boldsymbol{P}_T][\text{CLASS}]_k$ for a target sample.

4.2 Empirical risk minimization: a simple baseline

As we introduced, to learn those prompts, we consider the cross-entropy losses applied to source data and target data with pseudo labels as a set of objectives to optimize simultaneously:

$$\mathcal{L}_{total}(\boldsymbol{P}) := \big[[\mathcal{L}_{S,i}(\boldsymbol{P})]_{i=1}^N, \mathcal{L}_T(\boldsymbol{P})\big] = \big[[\mathcal{L}_{S,i}(\boldsymbol{P}_{sh}, \boldsymbol{P}_{S,i})]_{i=1}^N, \mathcal{L}_T(\boldsymbol{P}_{sh}, \boldsymbol{P}_T)\big],$$

$$\mathcal{L}_{S,i}(\boldsymbol{P}_{sh}, \boldsymbol{P}_{S,i}) = \text{CE}(\boldsymbol{P}_{sh}, \boldsymbol{P}_{S,i}; \boldsymbol{X}_{S,i}, Y_{S,i}) = -\frac{1}{N_{S,i}}\sum_{j=1}^{N_{S,i}} \log P(y=y_j|\boldsymbol{x}_j, \boldsymbol{P}_{sh}, \boldsymbol{P}_{S,i}), \quad (2)$$

$$\begin{aligned}\mathcal{L}_T(\boldsymbol{P}_{sh}, \boldsymbol{P}_T) &= \text{CE}_\tau(\boldsymbol{P}_{sh}, \boldsymbol{P}_T; \boldsymbol{X}_T, Y_T) \\ &= -\frac{1}{N_T}\sum_{j=1}^{N_T} \mathbb{I}(P(y=\hat{y}_j|\boldsymbol{x}_j) \geq \tau) \log P(y=\hat{y_j}|\boldsymbol{x}_j, \boldsymbol{P}_{sh}, \boldsymbol{P}_T), \quad (3)\end{aligned}$$

$$\hat{y}_j = \arg\max_k P(y=k|\boldsymbol{x}_j). \quad (4)$$

In summary, the total loss consists of $N+1$ objectives. The target objective is applied to target samples whose zero-shot predictions made by CLIP are larger than a threshold τ.

Given these objectives, source- and target-specific prompts can be updated by minimizing source and target losses, respectively. Regarding domain-agnostic prompt, one can put a weighting term on the signal from source losses to compute the gradient. Formally, for $\forall i = 1 \rightarrow N$, we have:

$$\boldsymbol{g}_{sh,i}, \boldsymbol{g}_{S,i} = \nabla_{\boldsymbol{P}} \mathcal{L}_{S,i}(\boldsymbol{P}_{sh}, \boldsymbol{P}_{S,i}), \quad \boldsymbol{g}_{sh,T}, \boldsymbol{g}_T = \nabla_{\boldsymbol{P}} \mathcal{L}_T(\boldsymbol{P}_{sh}, \boldsymbol{P}_T),$$
$$\boldsymbol{P}_{S,i} = \boldsymbol{P}_{S,i} - \eta \boldsymbol{g}_{S,i}, \qquad \boldsymbol{P}_T = \boldsymbol{P}_T - \eta \boldsymbol{g}_T, \tag{5}$$
$$\boldsymbol{P}_{sh} = \boldsymbol{P}_{sh} - \eta(\boldsymbol{g}_{sh,T} + \lambda \sum_i \boldsymbol{g}_{sh,i}), \tag{6}$$

where η is the learning rate, and λ is the weighting term to control how much emphasis we want to put on the target domain. Note that we treat gradient signals from source domains equally as we assume no prior preference knowledge about them. Nevertheless, one can measure the domain similarity between each source and target domain to devise a better way to reweight source domains' objectives. However, as will be shown in the experiments, taking the average is simple yet yields superior results, hence we will leave this for future work.

4.3 Prompt gradient alignment for UDA

For simplicity, we first consider the single-source UDA setting and will present the extension to the multi-source one later in Appendix B. One problem with the method above is we ignored the potential inherent gradient conflict between objectives when updating the shared prompt. To mitigate this, one can follow gradient-based methods, such as [35, 47] to manipulate the gradients so that conflict is reduced. However, it has been shown in [44–46] that comparable performance can be obtained without such complex manipulations, but with simple re-weighting the loss functions. Therefore, to encourage consensus between these gradients without modifying them, we propose aligning gradients between source and target domains during training. Specifically, we aim to maximize their cosine similarity, $\langle \boldsymbol{g}_{sh,S}, \boldsymbol{g}_{sh,T} \rangle$, If this goal is achieved, one can expect the shared prompt to capture useful features for classes regardless of domains. Indeed, $-\boldsymbol{g}_{sh,S}$ denotes the direction that moves the shared prompt towards low-loss region of source data, and similar for $-\boldsymbol{g}_{sh,T}$. Hence, when they point to the same direction, i.e., $\langle \boldsymbol{g}_{sh,S}, \boldsymbol{g}_{sh,T} \rangle > 0$, updating the shared prompt as in Eq. 6 can reduce loss of both domains, because the aggregated gradient $\boldsymbol{g}_{sh} = \lambda \boldsymbol{g}_{sh,S} + \boldsymbol{g}_{sh,T}$ will create acute angles with both $\boldsymbol{g}_{sh,S}$ and $\boldsymbol{g}_{sh,T}$. As a result, the shared prompt can learn knowledge that benefits both domains, which is its ultimate goal.

However, there remain two important questions when implementing this gradient alignment constrain: (i) How to incorporate the cosine similarity maximization term as a regularization in the framework described in Sec. 4.2?; and (ii) How to reduce training time and space when explicitly maximizing it, as it involves the computation of Hessian matrix w.r.t the shared prompt? Our method will address these two concerns.

Consider the following loss applied on target data with $||.||$ denoting l_2-norm of a vector:

$$\begin{aligned}
\mathcal{L}_T^{\text{align}}(\boldsymbol{P}) &:= \mathcal{L}_T(\boldsymbol{P}_{sh} - \rho \frac{\boldsymbol{g}_{sh,S}}{\|\boldsymbol{g}_{sh,S}\|.\|\boldsymbol{g}_{sh,T}\|}, \boldsymbol{P}_T) \\
&\approx \mathcal{L}_T(\boldsymbol{P}_{sh}, \boldsymbol{P}_T) - \rho \frac{(\boldsymbol{g}_{sh,S})^{\mathbb{T}}.\nabla_{\boldsymbol{P}_{sh}} \mathcal{L}_T(\boldsymbol{P}_{sh}, \boldsymbol{P}_T)}{\|\boldsymbol{g}_{sh,S}\|.\|\boldsymbol{g}_{sh,T}\|} \\
&= \mathcal{L}_T(\boldsymbol{P}_{sh}, \boldsymbol{P}_T) - \rho \langle \boldsymbol{g}_{sh,S}, \boldsymbol{g}_{sh,T} \rangle,
\end{aligned} \tag{7}$$

where Eq. 7 is obtained by applying first-order Taylor expansion with ρ is a small value, and $\mathbb{T}$ is the vector transpose. It can be seen that minimizing this loss also maximizes cosine similarity between gradients of the two domains. In order to achieve this, let denote $\boldsymbol{a} = \frac{\boldsymbol{g}_{sh,S}}{\|\boldsymbol{g}_{sh,S}\|.\|\boldsymbol{g}_{sh,T}\|}$, and consider the loss's gradient w.r.t $\boldsymbol{P}_{sh}$:

$$\begin{aligned}
\boldsymbol{g}_{sh,T}^{\text{align}} &:= \nabla_{\boldsymbol{P}_{sh}} \mathcal{L}_T(\boldsymbol{P}_{sh} - \rho \boldsymbol{a}, \boldsymbol{P}_T) \\
&= \frac{d(\boldsymbol{P}_{sh} - \rho \boldsymbol{a})}{d(\boldsymbol{P}_{sh})} \nabla_{\boldsymbol{P}_{sh}} \mathcal{L}_T(\boldsymbol{P}_{sh}, \boldsymbol{P}_T) \Big|_{\boldsymbol{P}_{sh} = \boldsymbol{P}_{sh} - \rho \boldsymbol{a}} \\
&\approx \nabla_{\boldsymbol{P}_{sh}} \mathcal{L}_T(\boldsymbol{P}_{sh}, \boldsymbol{P}_T)|_{\boldsymbol{P}_{sh} = \boldsymbol{P}_{sh} - \rho \boldsymbol{a}}.
\end{aligned} \tag{8}$$

https://doi.org/10.52202/079017-1447

In the approximation of Eq. 8, we avoid the Hessian computation by dropping the derivative of $\boldsymbol{a}$ w.r.t $\boldsymbol{P}_{sh}$. Now we can practically apply deep learning optimizers, such as SGD, to minimize $\mathcal{L}_T^{\text{align}}(\boldsymbol{P})$. Specifically, we first compute gradients of the source and target losses w.r.t the shared prompt to get vector $\boldsymbol{a}$, then move the current shared prompt to the new stage: $\boldsymbol{P}_{sh} = \boldsymbol{P}_{sh} - \rho\boldsymbol{a}$. Finally, at this new stage, re-compute the loss on target data then calculate the new gradient.

In a similar way, we can derive $\mathcal{L}_S^{\text{align}}(\boldsymbol{P})$ on source data and then compute its new gradient w.r.t the shared prompt, i.e. $\boldsymbol{g}_{sh,S}^{\text{align}}$. Given these two new gradients, we can combine them to get the final update direction of the shared prompt, which will navigate it to common low-valued regions in the loss landscapes of both domains.

$$\boldsymbol{b} = \frac{\boldsymbol{g}_{sh,T}}{\|\boldsymbol{g}_{sh,S}\|.\|\boldsymbol{g}_{sh,T}\|}, \boldsymbol{g}_{sh,S}^{\text{align}} \approx \nabla_{\boldsymbol{P}_{sh}} \mathcal{L}_S(\boldsymbol{P}_{sh}, \boldsymbol{P}_S)|_{\boldsymbol{P}_{sh}=\boldsymbol{P}_{sh}-\rho\boldsymbol{b}},$$
$$\boldsymbol{g}_{sh}^{\text{align}} = \boldsymbol{g}_{sh,T}^{\text{align}} + \lambda \boldsymbol{g}_{sh,S}^{\text{align}}.$$

4.4 Prompt gradient-norm penalization for UDA

So far, we have proposed casting each domain loss as an objective in a multiple-objective optimization framework, and have suggested maximizing congruence between gradients of these objectives to reduce their inherent conflict. However, the domain loss is in the empirical form, which has been shown to be easily stuck in sharp minima and thus limiting generalization ability [81, 82], especially under distribution shifts [83]. Therefore, we argue that explicit control over the generalization of these prompts can be beneficial. Moreover, inspired by the finding in [59] that gradient norm penalization can help model favor flat minima, and by the effectiveness of such minima in the context of multi-task learning [81], we propose minimizing prompt gradient norm of each objective to enhance prompt generalization.

By following the same analysis as in Eq. 7, we can seamlessly fuse the gradient norm penalty term with the cosine similarity maximization with the loss below:

$$\begin{aligned}
\mathcal{L}_T^{\text{PGA}}(\boldsymbol{P}) &:= \mathcal{L}_T(\boldsymbol{P}_{sh} - \rho_{ga}\frac{\boldsymbol{g}_{sh,S}}{\|\boldsymbol{g}_{sh,S}\|.\|\boldsymbol{g}_{sh,T}\|} + \rho_{gn}\frac{\boldsymbol{g}_{sh,T}}{\|\boldsymbol{g}_{sh,T}\|}, \boldsymbol{P}_T + \rho_{gn}\frac{\boldsymbol{g}_T}{\|\boldsymbol{g}_T\|}) \\
&\approx \mathcal{L}_T(\boldsymbol{P}_{sh}, \boldsymbol{P}_T) - \rho_{ga}\frac{(\boldsymbol{g}_{sh,S})^{\mathbb{T}}.\nabla_{\boldsymbol{P}_{sh}}\mathcal{L}_T(\boldsymbol{P}_{sh}, \boldsymbol{P}_T)}{\|\boldsymbol{g}_{sh,S}\|.\|\boldsymbol{g}_{sh,T}\|} + \rho_{gn}(\|\boldsymbol{g}_{sh,T}\| + \|\boldsymbol{g}_T\|) \\
&= \mathcal{L}_T(\boldsymbol{P}_{sh}, \boldsymbol{P}_T) - \rho_{ga}\langle \boldsymbol{g}_{sh,S}, \boldsymbol{g}_{sh,T}\rangle + \rho_{gn}(\|\boldsymbol{g}_{sh,T}\| + \|\boldsymbol{g}_T\|),
\end{aligned}$$

where $\boldsymbol{g}_T$ is the gradient of the target loss w.r.t target-specific $\boldsymbol{P}_T$, and gn stands for gradient norm.

We then follow the derivation of Eq. 8 to come up with a practical approximation of the gradient of $\mathcal{L}_T^{\text{PGA}}(\boldsymbol{P})$

$$\begin{aligned}
\boldsymbol{g}_{sh,T}^{\text{PGA}}, \boldsymbol{g}_T^{\text{PGA}} &:= \nabla_{\boldsymbol{P}}\mathcal{L}_T^{\text{PGA}}(\boldsymbol{P}) \\
&\approx \nabla_{\boldsymbol{P}}\mathcal{L}_T(\boldsymbol{P}_{sh}, \boldsymbol{P}_T)|_{\boldsymbol{P}_{sh}=\boldsymbol{P}_{sh}-\rho_{ga}\boldsymbol{a}+\rho_{gn}\frac{\boldsymbol{g}_{sh,T}}{||\boldsymbol{g}_{sh,T}||}, \boldsymbol{P}_T=\boldsymbol{P}_T+\rho_{gn}\frac{\boldsymbol{g}_T}{||\boldsymbol{g}_T||}}.
\end{aligned}$$

Similarly, we obtain the gradient of the source objective

$$\boldsymbol{g}_{sh,S}^{\text{PGA}}, \boldsymbol{g}_S^{\text{PGA}} \approx \nabla_{\boldsymbol{P}}\mathcal{L}_S(\boldsymbol{P}_{sh}, \boldsymbol{P}_S)|_{\boldsymbol{P}_{sh}=\boldsymbol{P}_{sh}-\rho_{ga}\boldsymbol{b}+\rho_{gn}\frac{\boldsymbol{g}_{sh,S}}{||\boldsymbol{g}_{sh,S}||}, \boldsymbol{P}_S=\boldsymbol{P}_S+\rho_{gn}\frac{\boldsymbol{g}_S}{||\boldsymbol{g}_S||}}.$$

Following the same update rules in Eq. 5 and Eq. 6, the prompts can be learnt to achieve both of our two goals: inter-domain gradient alignment and flat minima enforcement, which can lead to improved performance for UDA. We will recap this with a generalization bound in the next part, and provide details for the final loss function in Appendix B.

4.5 Theoretical Analysis of PGA

We informally present an information-theoretic bound to explain why PGA works. Refer to Appendix A for the formal version. For simplicity, we will consider the single-source UDA setting and abuse N as the number of source samples. Let $\mathcal{Z}, \mathcal{P}$ be the input-label space and prompt space (or hypothesis

space), respectively. Assume the loss function $\ell : \mathcal{P} \times \mathcal{Z} \to \mathbb{R}_0^+$ is R-subgaussian [*] Denote μ, μ' as the two underlying distributions from which the source and target data is sampled, and $KL(.||.)$ as the KL-divergence. The generalization error[*] is defined as the difference between the target population loss and the source empirical loss

$$Err := \mathbb{E}_{\boldsymbol{P},D_S,D_T}[R_{\mu'}(\boldsymbol{P}) - R_{D_S}(\boldsymbol{P})] = \mathbb{E}_{\boldsymbol{P},D_S,D_T}[\mathbb{E}_{\boldsymbol{Z'}\sim\mu'}[\ell(\boldsymbol{P},\boldsymbol{Z'})] - \frac{1}{N}\sum_{i=1}^{N}\ell(\boldsymbol{P},\boldsymbol{Z}_i)].$$

Theorem 4.1. *Under the assumption R-subgaussianity, the generalization error can be upper-bounded by:*

$$|Err| \leq \sqrt{\frac{4R^2}{N}\sum_{t=1}^{\mathcal{T}}\tilde{\eta}_t^2\mathbb{E}_{\boldsymbol{P}_{t-1},D_S,D_T}[\|\boldsymbol{g}_t^{src}\|^2 + \|\boldsymbol{g}_t^{tgt}\|^2 + \|\boldsymbol{g}_t^{src} - \boldsymbol{g}_t^{tgt}\|^2]} + \sqrt{2R^2KL(\mu||\mu')},$$

where $\mathcal{T}$ is the total number of training iterations, $\tilde{\eta}_t$ is the learning rate at iteration t scaled by a scalar, $\boldsymbol{g}_t^{src} = \nabla_{\boldsymbol{P}}\mathcal{L}_S(\boldsymbol{P}_{t-1})$, $\boldsymbol{g}_t^{tgt} = \nabla_{\boldsymbol{P}}\mathcal{L}_T(\boldsymbol{P}_{t-1})$ are the gradients w.r.t $\boldsymbol{P}_{t-1}$ of source loss Eq. 2 and target loss Eq 3 where $\boldsymbol{P}_t$ is the prompt at iteration t.

As our method tries to minimize source empirical loss, gradient norms and gradient mis-alignment, from the first term in the R.H.S of Eq. 4.1, its benefit to the performance on target domain can be justified. Furthermore, the second term shows that the generalization error can be reduced by bridging the gap between the two domain distributions, which is the core of many UDA methods, such as [70, 84]. However, as stated earlier, our work is orthogonal to this line of method as we do not explicitly attempt to close such gap. Hence, an interesting future development could be taking the second term into account. Refer to Appendix A.5 for more discussion about this bound.

5 Experiments

In this section, we evaluate the efficacy of our proposed method on different UDA benchmarks, following the same protocol of recent prompt-based UDA studies [25, 28]. Before that, we start with a simple multi-objective-optimization setup to derive insights into the effectiveness of our proposed method compared to conventional empirical risk minimization (ERM).

5.1 Illustrative example

Let $\mathbf{y} \in \{-1, 1\}$ be the true label, $\mathbf{e}$ be the environmental feature and $\boldsymbol{\epsilon}$ be Gaussian noise, $\mathbf{x} \in \mathbb{R}^{300}$, and $\mathrm{p} \in (0, 1), C > 1$ be predefined scalar constants. The data-generating process is given by:

$$\mathbf{y} \sim \mathcal{U}\{-1, 1\}, \quad \mathbf{e} \sim \begin{cases} p_{\mathrm{p}}(\mathbf{e} = y \mid \mathbf{y} = y) = \mathrm{p} \\ p_{\mathrm{p}}(\mathbf{e} = -y \mid \mathbf{y} = y) = (1 - \mathrm{p}) \end{cases}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}\left(0, \mathbf{I}^{298}\right), \quad \mathbf{x} = [C * \mathbf{e}, \mathbf{y}, \boldsymbol{\epsilon}]$$

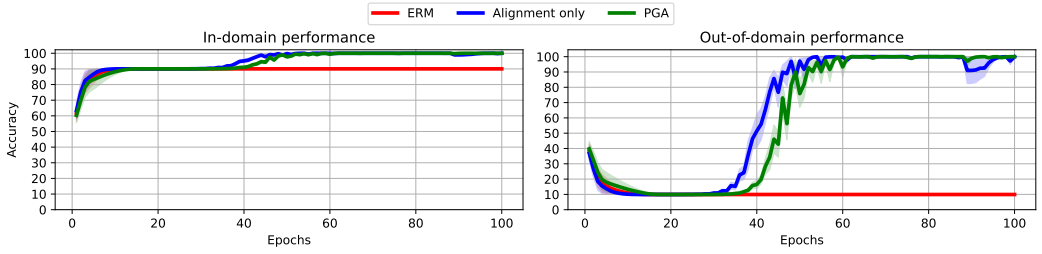

Figure 2: Performance of ERM and PGA on the in-domain data (validation set) and out-of-distribution data (test set). Average results and shaded standard errors are obtained from 10 random seeds.

The environmental feature $\mathbf{e}$ correlates with the true label $\mathbf{y}$ according to p. Similar to [55, 85], we set $\mathrm{p} = 0.9$ for the training and validation set (in-distribution) and $\mathrm{p} = 0.1$ for the test set

[*] A random variable X is R-subgaussian if for any $\rho, \log \mathbb{E}\exp(\rho(X - \mathbb{E}X)) \leq \rho^2R^2/2$.

[*] Refer to the Appendix to see why the expectation is taken over $\boldsymbol{P}, D_S, D_T$.

https://doi.org/10.52202/079017-1447

Table 2: Accuracy (%) on ImageCLEF and Office-Home. We use **bold** to denote the best method overall and underscore to denote the best method using source combined. Overall, PGA and MPGA consistently obtain the best results among source combined and multi-source scenarios, respectively.

	ImageCLEF				Office-Home				
	→ **C**	→ **I**	→ **P**	**Avg**	→ **Ar**	→ **Cl**	→ **Pr**	→ **Rw**	**Avg**
Zero-Shot									
CLIP [27]	87.9	88.2	78.7	88.1	71.2	50.4	81.4	82.6	71.4
Source Combined									
DAN [19]	93.3	92.2	77.6	87.7	68.5	59.4	79.0	82.5	72.4
DANN [18]	93.7	91.8	77.9	87.8	68.4	59.1	79.5	82.7	72.4
D-CORAL [69]	93.6	91.7	77.1	87.5	68.1	58.6	79.5	82.7	72.2
DAPL [25]	96.0	89.2	76.0	87.1	72.8	51.9	82.6	83.7	72.8
Simple Prompt [28]	93.6	90.6	80.9	88.4	70.7	52.9	82.9	83.9	72.4
PGA (Ours)	96.8	95.7	84.6	92.4	75.2	59.7	86.2	86.2	76.8
Multi-Source									
DCTN [88]	95.7	90.3	75.0	87.0	N.A.	N.A.	N.A.	N.A.	N.A.
MDDA [89]	N.A.	N.A.	N.A.	N.A.	66.7	62.3	79.5	79.6	71.0
SIMplDA [96]	93.3	91.0	77.5	87.3	70.8	56.3	80.2	81.5	72.2
MFSAN [62]	95.4	93.6	79.1	89.4	72.1	62.0	80.3	81.8	74.1
MPA [28]	97.2	96.2	80.4	91.3	74.8	54.9	86.2	85.7	75.4
MPGA (Ours)	**97.4**	**96.5**	**84.7**	**92.9**	**76.3**	**63.8**	**90.0**	**87.4**	**79.4**

(out-of-distribution). Figure 2 presents the performance of three linear classifiers trained by ERM, our gradient alignment method only and PGA. In summary, while ERM learns non-predictive features and fails to generalize beyond in-distribution data, our gradient alignment algorithm can leverage the gradient information from multiple environments to learn the core feature that helps perform well on OOD data. Besides, incorporating the gradient norm penalty term further enhances stability and robustness at convergence.

5.2 Experimental setup

Datasets. We conduct experiments using three well-established UDA datasets of varying scales: ImageCLEF [17], Office-Home [86], and DomainNet [87], respectively. Detailed descriptions of these datasets are available in Appendix C.1.

Metrics. We evaluate our model by reporting the top-1 accuracy for each target domain and the average accuracy across all domains. To further validate the effectiveness of our proposed method, we conduct experiments in two distinct settings: a source-combined setting, where data from all source domains are merged, and a multi-source setting, which utilizes individual domain identifications. Additionally, we provide pair-wise single-source domain adaptation results for the Office-Home dataset.

Baselines. Regarding prompt-based baselines, we compare our method with MPA [28], DAPL [25], Simple Prompt [28], and Zero-shot CLIP [27]. To ensure a comprehensive evaluation, we also include comparisons with various non-prompt methods such as DCTN [88], MDDA [89], MFSAN [62], T-SVDNet [90] and PFSA [91] ... As we follow the same settings as in [28] and [25], results for baselines are taken from those studies for the consistency. Note that while DAPL [25], MPA [28] and our methods employ CoOp [92] with text-end soft-prompt, other methods finetune the transformer block [63] or both image and text-end soft-prompts [93] or the whole encoders [94, 95]. Since those methods typically fine-tune many more parameters, we thus do not include them in the experimental results for the sake of fair comparison.

5.3 Experimental results

Table 2 presents the results for the ImageCLEF and Office-Home datasets. Under the source-combined scenario, PGA significantly outperforms nearly all other baseline methods on both datasets, with the exception of its own multi-source variant, MPGA. For instance, PGA surpasses the second-

best source combined method by a notable 4% in average accuracy and exceeds MPA by over 1%. Notably, in the Office-Home domain Clipart, while two prompt-based baselines, DAPL and Simple Prompt, lag behind their non-prompt counterparts, PGA still manages to achieve slightly better results than these non-prompt methods. In the multi-source setting, MPGA consistently delivers the highest performance across all domains, notably outperforming MPA, the state-of-the-art (SOTA) prompt-based method for multi-source UDA, by a substantial margin of 4% on Office-Home.

Table 3: Accuracy (%) on DomainNet. We use **bold** to denote the best method overall and underscore to denote the best method using source combine.

	DomainNet						
	→ Clp	**→ Inf**	**→ Pnt**	**→ Qdr**	**→ Rel**	**→ Skt**	**Avg**
Zero-Shot							
CLIP [27]	61.3	42.0	56.1	10.3	79.3	54.1	50.5
Source Combined							
DANN [18]	45.5	13.1	37.0	13.2	48.9	31.8	32.6
MCD [97]	54.3	22.1	45.7	7.6	58.4	43.5	38.5
DAPL [25]	62.4	43.8	59.3	10.6	81.5	54.6	52.0
Simple Prompt [28]	63.1	41.2	57.7	10.0	75.8	55.8	50.6
PGA (Ours)	66.3	49.2	63.3	11.1	81.8	60.6	55.4
Multi-Source							
M³SDA-β [98]	58.6	26.0	52.3	6.3	62.7	49.5	42.6
SImpAl101 [96]	66.4	26.5	56.6	**18.9**	68.0	55.5	48.6
LtC-MSDA [99]	63.1	28.7	56.1	16.3	66.1	53.8	47.4
T-SVDNet [90]	66.1	25.0	54.3	16.5	65.4	54.6	47.0
PFSA [91]	64.5	29.2	57.6	17.2	67.2	55.1	48.5
PTMDA [100]	66.0	28.5	58.4	13.0	63.0	54.1	47.2
MPA [28]	65.2	47.3	62.0	10.2	82.0	57.9	54.1
MPGA (Ours)	**67.9**	**50.5**	**63.8**	11.6	**82.2**	**61.0**	**56.2**

On DomainNet, as Table 3 presents, our method still obtains superior average accuracy under both source combined and multi-source, higher than the runner-up by 3.4% and 2.1%, respectively. Overall, in the domain where CLIP brings significant results compared with non-prompt baselines, our method leads to better performance, except for the difficult QuickDraw domain, as remarked by a relatively low zero-shot accuracy for CLIP-based methods, where it seems that prompt learning fails to beat non-prompt counterparts. Even though, both PGA and MPGA still outperform other prompt-based counterparts while fine-tuning fewer parameters (e.g. 500k versus 2M of MPA).

In addition, we also demonstrate our method's effectiveness under 12 pair-wise source-target settings on Office-Home in Table 4. Again, PGA acquires the highest average score and consistently beats DAPL under 12 settings while using the same parameter-efficient-finetuning method [92].

Table 4: Accuracy (%) on Office-Home[101] for unsupervised domain adaptation (ResNet-50[102]). The best accuracy is indicated in **bold**.

Method	Ar→Cl	Ar→Pr	Ar→Rw	Cl→Ar	Cl→Pr	Cl→Rw	Pr→Ar	Pr→Cl	Pr→Rw	Rw→Ar	Rw→Cl	Rw→Pr	Avg
ResNet-50[102]	34.9	50.0	58.0	37.4	41.9	46.2	38.5	31.2	60.4	53.9	41.2	59.9	46.1
DANN [19]	45.6	59.3	70.1	47.0	58.5	60.9	46.1	43.7	68.5	63.2	51.8	76.8	57.6
JAN [17]	45.9	61.2	68.9	50.4	59.7	61.0	45.8	43.4	70.3	63.9	52.4	76.8	58.3
CDAN+E [71]	50.7	70.6	76.0	57.6	70.0	70.0	57.4	50.9	77.3	70.9	56.7	81.6	65.8
BSP+CDAN [103]	52.0	68.6	76.1	58.0	70.3	70.2	58.6	50.2	77.6	72.2	59.3	81.9	66.3
SymNets [104]	47.7	72.9	78.5	64.2	71.3	74.2	63.6	47.6	79.4	73.8	50.8	82.6	67.2
ETD [105]	51.3	71.9	85.7	57.6	69.2	73.7	57.8	51.2	79.3	70.2	57.5	82.1	67.3
BNM [106]	52.3	73.9	80.0	63.3	72.9	74.9	61.7	49.5	79.7	70.5	53.6	82.2	67.9
GSDA [107]	**61.3**	76.1	79.4	65.4	73.3	74.3	65.0	53.2	80.0	72.2	**60.6**	83.1	70.3
GVB-GD [108]	57.0	74.7	79.8	64.6	74.1	74.6	65.2	55.1	81.0	74.6	59.7	84.3	70.4
RSDA-MSTN [109]	53.2	77.7	81.3	66.4	74.0	76.5	67.9	53.0	82.0	75.8	57.8	85.4	70.9
SPL [110]	54.5	77.8	81.9	65.1	78.0	81.1	66.0	53.1	82.8	69.9	55.3	**86.0**	71.0
SRDC [26]	52.3	76.3	81.0	69.5	76.2	78.0	68.7	53.8	81.7	**76.3**	57.1	85.0	71.3
DisClusterDA [111]	58.8	77.0	80.8	67.0	74.6	77.1	65.9	**56.3**	81.4	74.2	60.5	83.6	71.4
CLIP [27]	51.6	81.9	82.6	71.9	81.9	82.6	71.9	51.6	82.6	71.9	51.6	81.9	72.0
DAPL [25]	54.1	84.3	84.8	74.4	83.7	85.0	74.5	54.6	84.8	75.2	54.7	83.8	74.5
PGA (Ours)	56.1	**85.5**	**86.0**	**75.5**	**85.2**	**85.8**	**75.2**	55.7	**86.1**	75.4	56.7	85.8	**75.8**

https://doi.org/10.52202/079017-1447

5.4 Ablation study

From Table 5, we can see that (i) learning prompts using solely the target loss, the accuracy across all settings already surpasses that of Zero-shot CLIP. This confirms the reliability of pseudo labels generated by CLIP. (ii) When adding source loss and grad-norm penalization, the results improve slightly. (iii) Importantly, adding gradient alignment, the scores increase more clearly. These observations verify each of our contributions.

$\mathbf{L_T}$	$\mathbf{L_S}$	GN	GA	$\rightarrow$ C	$\rightarrow$ I	$\rightarrow$ P	Avg
×	×	×	×	87.9	88.2	78.7	88.1
✓	×	×	×	92.9	94.3	83.2	90.1
✓	✓	×	×	93.3	95.0	83.3	90.6
✓	✓	✓	×	94.3	95.3	83.2	90.9
✓	✓	✓	✓	96.8	95.7	84.6	92.4

Table 5: Ablation studies on various modules of PGA on the ImageCLEF. Each of the proposed components shows its effectiveness while combining them helps obtain the best performance.

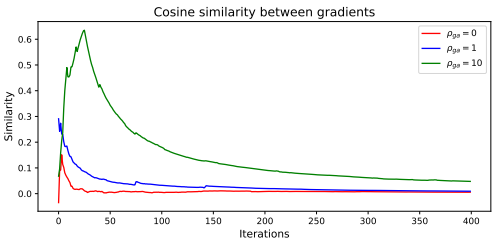

Figure 3: Evolution of the gradient similarity during training.

Furthermore, to show that gradient alignment indeed increases consensus between gradients, we plot cosine similarity along the training process with three different values of ρ_{ga} in Figure 3. First, during early training stages, there seems to be less agreement between gradients when no alignment is enforced, c.f. $\rho_{ga} = 0$. When $\rho_{ga} > 0$, we can see the similarity increase. Noticeably, there exists a point where similarity starts plummeting. This is reasonable when the model starts to converge to a Pareto solution where source and target gradients cancel each other. This is depicted more clearly in Figure 4 in the appendix where the closer the model is to the Pareto front, the more conflict the gradients are.

6 Conclusion

In this work, we have proposed a framework for UDA inspired by Multi-objective optimization thanks to the generalizability of CLIP and the lightweight nature of prompt learning. We have then devised a practical method to align per-objective gradients, which aims to encourage inherent consensus between objectives. We have further fused gradient norm penalization into the method to enhance prompt generalization. Finally, a UDA generalization bound is presented to justify the benefits of our method.

Acknowledgements: Trung Le was supported by ARC DP23 grant DP230101176 and by the Air Force Office of Scientific Research under award number FA2386-23-1-4044.

References

[1] Siddharth Srivastava and Gaurav Sharma. Omnivec: Learning robust representations with cross modal sharing. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1236–1248, 2024.

[2] Jiahui Yu, Zirui Wang, Vijay Vasudevan, Legg Yeung, Mojtaba Seyedhosseini, and Yonghui Wu. Coca: Contrastive captioners are image-text foundation models. *arXiv preprint arXiv:2205.01917*, 2022.

[3] Mitchell Wortsman, Gabriel Ilharco, Samir Ya Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, et al. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *International conference on machine learning*, pages 23965–23998. PMLR, 2022.

[4] Xi Chen, Xiao Wang, Soravit Changpinyo, AJ Piergiovanni, Piotr Padlewski, Daniel Salz, Sebastian Goodman, Adam Grycner, Basil Mustafa, Lucas Beyer, et al. Pali: A jointly-scaled multilingual language-image model. In *The Eleventh International Conference on Learning Representations*, 2022.

[5] Zihang Dai, Hanxiao Liu, Quoc V Le, and Mingxing Tan. Coatnet: Marrying convolution and attention for all data sizes. *Advances in neural information processing systems*, 34:3965–3977, 2021.

[6] Zhuofan Zong, Guanglu Song, and Yu Liu. Detrs with collaborative hybrid assignments training. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6748–6758, 2023.

[7] Wenhai Wang, Jifeng Dai, Zhe Chen, Zhenhang Huang, Zhiqi Li, Xizhou Zhu, Xiaowei Hu, Tong Lu, Lewei Lu, Hongsheng Li, et al. Internimage: Exploring large-scale vision foundation models with deformable convolutions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14408–14419, 2023.

[8] Weijie Su, Xizhou Zhu, Chenxin Tao, Lewei Lu, Bin Li, Gao Huang, Yu Qiao, Xiaogang Wang, Jie Zhou, and Jifeng Dai. Towards all-in-one pre-training via maximizing multi-modal mutual information. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15888–15899, 2023.

[9] Kemal Oksuz, Selim Kuzucu, Tom Joy, and Puneet K Dokania. Mocae: Mixture of calibrated experts significantly improves object detection. *arXiv preprint arXiv:2309.14976*, 2023.

[10] Xudong Wang, Shufan Li, Konstantinos Kallidromitis, Yusuke Kato, Kazuki Kozuka, and Trevor Darrell. Hierarchical open-vocabulary universal image segmentation. *Advances in Neural Information Processing Systems*, 36, 2024.

[11] Peng Wang, Shijie Wang, Junyang Lin, Shuai Bai, Xiaohuan Zhou, Jingren Zhou, Xinggang Wang, and Chang Zhou. One-peace: Exploring one general representation model toward unlimited modalities. *arXiv preprint arXiv:2305.11172*, 2023.

[12] Wenhui Wang, Hangbo Bao, Li Dong, Johan Bjorck, Zhiliang Peng, Qiang Liu, Kriti Aggarwal, Owais Khan Mohammed, Saksham Singhal, Subhojit Som, et al. Image as a foreign language: Beit pretraining for all vision and vision-language tasks. *arXiv preprint arXiv:2208.10442*, 2022.

[13] Serdar Erisen. Sernet-former: Semantic segmentation by efficient residual network with attention-boosting gates and attention-fusion networks. *arXiv preprint arXiv:2401.15741*, 2024.

[14] Michael A Lones. How to avoid machine learning pitfalls: a guide for academic researchers. *arXiv preprint arXiv:2108.02497*, 2021.

 https://doi.org/10.52202/079017-1447

[15] Pang Wei Koh, Shiori Sagawa, Henrik Marklund, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard Lanas Phillips, Irena Gao, et al. Wilds: A benchmark of in-the-wild distribution shifts. In *International conference on machine learning*, pages 5637–5664. PMLR, 2021.

[16] Shiori Sagawa, Pang Wei Koh, Tatsunori B Hashimoto, and Percy Liang. Distributionally robust neural networks. In *International Conference on Learning Representations*, 2019.

[17] Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I Jordan. Deep transfer learning with joint adaptation networks. In *ICML*, pages 2208–2217, 2017.

[18] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario March, and Victor Lempitsky. Domain-adversarial training of neural networks. *Journal of machine learning research*, 17(59):1–35, 2016.

[19] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *ICML*, pages 1180–1189, 2015.

[20] Mingsheng Long, Yue Cao, Zhangjie Cao, Jianmin Wang, and Michael I Jordan. Transferable representation learning with deep adaptation networks. *IEEE transactions on pattern analysis and machine intelligence*, 41(12):3071–3085, 2018.

[21] Boxiao Pan, Zhangjie Cao, Ehsan Adeli, and Juan Carlos Niebles. Adversarial cross-domain action recognition with co-attention. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 11815–11822, 2020.

[22] Zhongyi Han, Xian-Jin Gui, Chaoran Cui, and Yilong Yin. Towards accurate and robust domain adaptation under noisy environments. In *Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence*, pages 2269–2276, 2021.

[23] Jinyu Yang, Chunyuan Li, Weizhi An, Hehuan Ma, Yuzhi Guo, Yu Rong, Peilin Zhao, and Junzhou Huang. Exploring robustness of unsupervised domain adaptation in semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9194–9203, 2021.

[24] Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-aware minimization for efficiently improving generalization. In *International Conference on Learning Representations*, 2020.

[25] Chunjiang Ge, Rui Huang, Mixue Xie, Zihang Lai, Shiji Song, Shuang Li, and Gao Huang. Domain adaptation via prompt learning. *IEEE Transactions on Neural Networks and Learning Systems*, 2023.

[26] Hui Tang, Ke Chen, and Kui Jia. Unsupervised domain adaptation via structurally regularized deep clustering. In *CVPR*, pages 8725–8735, 2020.

[27] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, volume 139, pages 8748–8763, 2021.

[28] Haoran Chen, Zuxuan Wu, and Yu-Gang Jiang. Multi-prompt alignment for multi-source unsupervised domain adaptation. *Neural Information Processing Systems*, 2022.

[29] Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, and Shin Ishii. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *IEEE transactions on pattern analysis and machine intelligence*, 41(8):1979–1993, 2018.

[30] Rui Shu, Hung Bui, Hirokazu Narui, and Stefano Ermon. A dirt-t approach to unsupervised domain adaptation. In *International Conference on Learning Representations*, 2018.

[31] Ozan Sener and Vladlen Koltun. Multi-task learning as multi-objective optimization. *Advances in neural information processing systems*, 31, 2018.

[32] Xi Lin, Hui-Ling Zhen, Zhenhua Li, Qing-Fu Zhang, and Sam Kwong. Pareto multi-task learning. *Advances in neural information processing systems*, 32, 2019.

[33] Michinari Momma, Chaosheng Dong, and Jia Liu. A multi-objective/multi-task learning framework induced by pareto stationarity. In *International Conference on Machine Learning*, pages 15895–15907. PMLR, 2022.

[34] Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning. *Advances in Neural Information Processing Systems*, 33:5824–5836, 2020.

[35] Bo Liu, Xingchao Liu, Xiaojie Jin, Peter Stone, and Qiang Liu. Conflict-averse gradient descent for multi-task learning. *Advances in Neural Information Processing Systems*, 34:18878–18890, 2021.

[36] Adrián Javaloy and Isabel Valera. Rotograd: Gradient homogenization in multitask learning. In *International Conference on Learning Representations*, 2021.

[37] Zhao Chen, Jiquan Ngiam, Yanping Huang, Thang Luong, Henrik Kretzschmar, Yuning Chai, and Dragomir Anguelov. Just pick a sign: Optimizing deep multitask models with gradient sign dropout. *Advances in Neural Information Processing Systems*, 33:2039–2050, 2020.

[38] Xiaobo Xia, Jiale Liu, Shaokun Zhang, Qingyun Wu, and Tongliang Liu. Coreset selection with prioritized multiple objectives. *arXiv preprint arXiv:2311.08675*, 2023.

[39] Yuru Song, Zan Lou, Shan You, Erkun Yang, Fei Wang, Chen Qian, Changshui Zhang, and Xiaogang Wang. Learning with privileged tasks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10685–10694, 2021.

[40] Aviv Shamsian, Aviv Navon, Neta Glazer, Kenji Kawaguchi, Gal Chechik, and Ethan Fetaya. Auxiliary learning as an asymmetric bargaining game. In *International Conference on Machine Learning*, pages 30689–30705. PMLR, 2023.

[41] Pingchuan Ma, Tao Du, and Wojciech Matusik. Efficient continuous pareto exploration in multi-task learning. In *International Conference on Machine Learning*, pages 6522–6531. PMLR, 2020.

[42] Aviv Navon, Aviv Shamsian, Ethan Fetaya, and Gal Chechik. Learning the pareto front with hypernetworks. In *International Conference on Learning Representations*, 2020.

[43] Hoang Phan, Andrew Gordon Wilson, and Qi Lei. Controllable prompt tuning for balancing group distributional robustness. In *Forty-first International Conference on Machine Learning*, 2024.

[44] Vitaly Kurin, Alessandro De Palma, Ilya Kostrikov, Shimon Whiteson, and Pawan K Mudigonda. In defense of the unitary scalarization for deep multi-task learning. *Advances in Neural Information Processing Systems*, 35:12169–12183, 2022.

[45] Derrick Xin, Behrooz Ghorbani, Justin Gilmer, Ankush Garg, and Orhan Firat. Do current multi-task optimization methods in deep learning even help? *Advances in neural information processing systems*, 35:13597–13609, 2022.

[46] Yuzheng Hu, Ruicheng Xian, Qilong Wu, Qiuling Fan, Lang Yin, and Han Zhao. Revisiting scalarization in multi-task learning: A theoretical perspective. *Advances in Neural Information Processing Systems*, 36, 2024.

[47] Shiji Zhou, Wenpeng Zhang, Jiyan Jiang, Wenliang Zhong, Jinjie Gu, and Wenwu Zhu. On the convergence of stochastic multi-objective gradient manipulation and beyond. *Advances in Neural Information Processing Systems*, 35:38103–38115, 2022.

[48] Angela Fan, Edouard Grave, and Armand Joulin. Reducing transformer depth on demand with structured dropout. In *International Conference on Learning Representations*, 2019.

 https://doi.org/10.52202/079017-1447

[49] Aliakbar Panahi, Seyran Saeedi, and Tom Arodz. Shapeshifter: a parameter-efficient transformer using factorized reshaped matrices. *Advances in Neural Information Processing Systems*, 34:1337–1350, 2021.

[50] Bonan Li, Yinhan Hu, Xuecheng Nie, Congying Han, Xiangjian Jiang, Tiande Guo, and Luoqi Liu. Dropkey for vision transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22700–22709, 2023.

[51] Sandhini Agarwal, Gretchen Krueger, Jack Clark, Alec Radford, Jong Wook Kim, and Miles Brundage. Evaluating clip: towards characterization of broader capabilities and downstream implications. *arXiv preprint arXiv:2108.02818*, 2021.

[52] Jialu Wang, Yang Liu, and Xin Wang. Are gender-neutral queries really gender-neutral? mitigating gender bias in image search. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1995–2008, 2021.

[53] Yu Du, Fangyun Wei, Zihe Zhang, Miaojing Shi, Yue Gao, and Guoqi Li. Learning to prompt for open-vocabulary object detection with vision-language model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14084–14093, 2022.

[54] Michael Zhang and Christopher Ré. Contrastive adapters for foundation model group robustness. *Advances in Neural Information Processing Systems*, 35:21682–21697, 2022.

[55] Shiori Sagawa, Aditi Raghunathan, Pang Wei Koh, and Percy Liang. An investigation of why overparameterization exacerbates spurious correlations. In *International Conference on Machine Learning*, pages 8346–8356. PMLR, 2020.

[56] Lifu Tu, Caiming Xiong, and Yingbo Zhou. Prompt-tuning can be much better than fine-tuning on cross-lingual understanding with multilingual language models. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 5478–5485, 2022.

[57] Cheng-En Wu, Yu Tian, Haichao Yu, Heng Wang, Pedro Morgado, Yu Hen Hu, and Linjie Yang. Why is prompt tuning for vision-language models robust to noisy labels? In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15488–15497, 2023.

[58] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Conditional prompt learning for vision-language models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16816–16825, 2022.

[59] Yang Zhao, Hao Zhang, and Xiuyuan Hu. Penalizing gradient norm for efficiently improving generalization in deep learning. *arXiv preprint arXiv: 2202.03599*, 2022.

[60] Devansh Bisla, Jing Wang, and Anna Choromanska. Low-pass filtering sgd for recovering flat optima in the deep learning optimization landscape. In *International Conference on Artificial Intelligence and Statistics*, pages 8299–8339. PMLR, 2022.

[61] Dongxian Wu, Shu-Tao Xia, and Yisen Wang. Adversarial weight perturbation helps robust generalization. *Advances in neural information processing systems*, 33:2958–2969, 2020.

[62] Yongchun Zhu, Fuzhen Zhuang, and Deqing Wang. Aligning domain-specific distribution and classifier for cross-domain classification from multiple sources. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 5989–5996. AAAI Press, 2019.

[63] Zhekai Du, Xinyao Li, Fengling Li, Ke Lu, Lei Zhu, and Jingjing Li. Domain-agnostic mutual prompting for unsupervised domain adaptation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2024.

[64] Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Mach. Learn.*, 79(1-2):151–175, 2010.

[65] Han Zhao, Shanghang Zhang, Guanhang Wu, José M. F. Moura, Joao P Costeira, and Geoffrey J Gordon. Adversarial multiple source domain adaptation. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

[66] Han Zhao, Shanghang Zhang, Guanhang Wu, Jo ao P. Costeira, José M. F. Moura, and Geoffrey J. Gordon. Multiple source domain adaptation with adversarial learning, 2018.

[67] Hoang Phan, Trung Le, Trung Phung, Anh Tuan Bui, Nhat Ho, and Dinh Phung. Global-local regularization via distributional robustness. In *International Conference on Artificial Intelligence and Statistics*, pages 7644–7664. PMLR, 2023.

[68] Arthur Gretton, Karsten Borgwardt, Malte J. Rasch, Bernhard Scholkopf, and Alexander J. Smola. A kernel method for the two-sample problem. *arXiv preprint arXiv: 0805.2368*, 2008.

[69] Baochen Sun and Kate Saenko. Deep coral: Correlation alignment for deep domain adaptation. In *ECCV*, pages 443–450. Springer, 2016.

[70] Trung Quoc Phung, Trung Le, Long Tung Vuong, Toan Tran, Anh Tuan Tran, Hung Bui, and Dinh Phung. On learning domain-invariant representations for transfer learning with multiple sources. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021.

[71] Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I. Jordan. Conditional adversarial domain adaptation. In *NeurIPS*, pages 1647–1657, 2018.

[72] Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. Calibrate before use: Improving few-shot performance of language models. In *International conference on machine learning*, pages 12697–12706. PMLR, 2021.

[73] Ari Holtzman, Peter West, Vered Shwartz, Yejin Choi, and Luke Zettlemoyer. Surface form competition: Why the highest probability answer isn't always right. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7038–7051, 2021.

[74] Simran Arora, Avanika Narayan, Mayee F Chen, Laurel Orr, Neel Guha, Kush Bhatia, Ines Chami, and Christopher Re. Ask me anything: A simple strategy for prompting language models. In *The Eleventh International Conference on Learning Representations*, 2022.

[75] Hoang Phan, Ngoc Tran, Trung Le, Toan Tran, Nhat Ho, and Dinh Phung. Stochastic multiple target sampling gradient descent. *Advances in neural information processing systems*, 35:22643–22655, 2022.

[76] Matthew Riemer, Ignacio Cases, Robert Ajemian, Miao Liu, Irina Rish, Yuhai Tu, , and Gerald Tesauro. Learning to learn without forgetting by maximizing transfer and minimizing interference. In *International Conference on Learning Representations*, 2019.

[77] David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. *Neural Information Processing Systems*, 2017.

[78] Arslan Chaudhry, Marc'Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. Efficient lifelong learning with a-GEM. In *International Conference on Learning Representations*, 2019.

[79] Yuge Shi, Jeffrey Seely, Philip Torr, Siddharth N, Awni Hannun, Nicolas Usunier, and Gabriel Synnaeve. Gradient matching for domain generalization. In *International Conference on Learning Representations*, 2022.

[80] Pengfei Wang, Zhaoxiang Zhang, Zhen Lei, and Lei Zhang. Sharpness-aware gradient matching for domain generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3769–3778, 2023.

[81] Hoang Phan, Lam Tran, Ngoc N. Tran, Nhat Ho, Dinh Phung, and Trung Le. Improving multi-task learning via seeking task-based flat regions. *arXiv preprint arXiv: 2211.13723*, 2022.

 https://doi.org/10.52202/079017-1447

[82] Yaowei Zheng, Richong Zhang, and Yongyi Mao. Regularizing neural networks via adversarial model perturbation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8156–8165, 2021.

[83] Harsh Rangwani, Sumukh K Aithal, Mayank Mishra, Arihant Jain, and R. Venkatesh Babu. A closer look at smoothness in domain adversarial training. *International Conference on Machine Learning*, 2022.

[84] Shuang Li, Chi Liu, Qiuxia Lin, Binhui Xie, Zhengming Ding, Gao Huang, and Jian Tang. Domain conditioned adaptation network. In *AAAI*, volume 34, pages 11386–11393, 2020.

[85] Aahlad Manas Puli, Lily Zhang, Yoav Wald, and Rajesh Ranganath. Don't blame dataset shift! shortcut learning due to gradients and cross entropy. *Advances in Neural Information Processing Systems*, 36:71874–71910, 2023.

[86] Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. Deep hashing network for unsupervised domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5018–5027, 2017.

[87] Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1406–1415, 2019.

[88] Ruijia Xu, Ziliang Chen, W. Zuo, Junjie Yan, and Liang Lin. Deep cocktail network: Multi-source unsupervised domain adaptation with category shift. *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018.

[89] Sicheng Zhao, Guangzhi Wang, Shanghang Zhang, Yang Gu, Yaxian Li, Zhichao Song, Pengfei Xu, Runbo Hu, Hua Chai, and Kurt Keutzer. Multi-source distilling domain adaptation. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 12975–12983. AAAI Press, 2020.

[90] Ruihuang Li, Xu Jia, Jianzhong He, Shuaijun Chen, and Qinghua Hu. T-svdnet: Exploring high-order prototypical correlations for multi-source domain adaptation. *ICCV*, 2021.

[91] Yangye Fu, Ming Zhang, Xing Xu, Zuo Cao, Chao Ma, Yanli Ji, Kai Zuo, and Huimin Lu. Partial feature selection and alignment for multi-source domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16654–16663, June 2021.

[92] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *International Journal of Computer Vision*, 130(9):2337–2348, 2022.

[93] Zhengfeng Lai, Haoping Bai, Haotian Zhang, Xianzhi Du, Jiulong Shan, Yinfei Yang, Chen-Nee Chuah, and Meng Cao. Empowering unsupervised domain adaptation with large-scale pre-trained vision-language models. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2691–2701, 2024.

[94] Zhengfeng Lai, Noranart Vesdapunt, Ning Zhou, Jun Wu, Cong Phuoc Huynh, Xuelu Li, Kah Kuen Fu, and Chen-Nee Chuah. Padclip: Pseudo-labeling with adaptive debiasing in clip for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16155–16165, 2023.

[95] Wenlve Zhou and Zhiheng Zhou. Unsupervised domain adaption harnessing vision-language pre-training. *IEEE Transactions on Circuits and Systems for Video Technology*, 2024.

[96] Naveen Venkat, Jogendra Nath Kundu, D. K. Singh, Ambareesh Revanur, and R. Venkatesh-Babu. Your classifier can secretly suffice multi-source domain adaptation. *Neural Information Processing Systems*, 2021.

[97] Kuniaki Saito, Kohei Watanabe, Y. Ushiku, and T. Harada. Maximum classifier discrepancy for unsupervised domain adaptation. *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2017.

[98] Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, pages 1406–1415. IEEE, 2019.

[99] Hang Wang, Minghao Xu, Bingbing Ni, and Wenjun Zhang. Learning to combine: Knowledge aggregation for multi-source domain adaptation. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part VIII*, volume 12353 of *Lecture Notes in Computer Science*, pages 727–744. Springer, 2020.

[100] Chuan-Xian Ren, Yong Liu, Xiwen Zhang, and Ke-Kun Huang. Multi-source unsupervised domain adaptation via pseudo target domain. *IEEE Transactions on Image Processing*, 2022.

[101] Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. Deep hashing network for unsupervised domain adaptation. In *CVPR*, pages 5385–5394, 2017.

[102] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016.

[103] Xinyang Chen, Sinan Wang, Mingsheng Long, and Jianmin Wang. Transferability vs. discriminability: Batch spectral penalization for adversarial domain adaptation. In *ICML*, volume 97, pages 1081–1090, 2019.

[104] Yabin Zhang, Hui Tang, Kui Jia, and Mingkui Tan. Domain-symmetric networks for adversarial domain adaptation. In *CVPR*, pages 5031–5040, 2019.

[105] Mengxue Li, Yiming Zhai, You-Wei Luo, Pengfei Ge, and Chuan-Xian Ren. Enhanced transport distance for unsupervised domain adaptation. In *CVPR*, 2020.

[106] Shuhao Cui, Shuhui Wang, Junbao Zhuo, Liang Li, Qingming Huang, and Qi Tian. Towards discriminability and diversity: Batch nuclear-norm maximization under label insufficient situations. In *CVPR*, pages 3940–3949, 2020.

[107] Lanqing Hu, Meina Kan, Shiguang Shan, and Xilin Chen. Unsupervised domain adaptation with hierarchical gradient synchronization. In *CVPR*, pages 4043–4052, 2020.

[108] Shuhao Cui, Shuhui Wang, Junbao Zhuo, Chi Su, Qingming Huang, and Qi Tian. Gradually vanishing bridge for adversarial domain adaptation. In *CVPR*, pages 12455–12464, 2020.

[109] Xiang Gu, Jian Sun, and Zongben Xu. Spherical space domain adaptation with robust pseudo-label loss. In *CVPR*, pages 9101–9110, 2020.

[110] Qian Wang and Toby P. Breckon. Unsupervised domain adaptation via structured prediction based selective pseudo-labeling. In *AAAI*, pages 6243–6250, 2020.

[111] Hui Tang, Yaowei Wang, and Kui Jia. Unsupervised domain adaptation via distilled discriminative clustering. *Pattern Recognition*, 127:108638, 2022.

[112] Ziqiao Wang and Yongyi Mao. Information-theoretic analysis of unsupervised domain adaptation. In *The Eleventh International Conference on Learning Representations*, 2023.

[113] Gergely Neu. Information-theoretic generalization bounds for stochastic gradient descent. In *Annual Conference Computational Learning Theory*, 2021.

[114] Jonas Geiping, Micah Goldblum, Phillip Pope, Michael Moeller, and Tom Goldstein. Stochastic training is not necessary for generalization. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022.

 https://doi.org/10.52202/079017-1447

[115] Lingfeng Shen, Weiting Tan, Boyuan Zheng, and Daniel Khashabi. Flatness-aware prompt selection improves accuracy and sample efficiency. *Conference on Empirical Methods in Natural Language Processing*, 2023.

[116] Liangchen Liu, Nannan Wang, Dawei Zhou, Xinbo Gao, Decheng Liu, Xi Yang, and Tongliang Liu. Gradient constrained sharpness-aware prompt learning for vision-language models, 2024.

[117] Ziqiao Wang and Yongyi Mao. Two facets of sde under an information-theoretic lens: Generalization of sgd via training trajectories and via terminal states. *arXiv preprint arXiv: 2211.10691*, 2022.

[118] Yury Polyanskiy and Yihong Wu. Lecture notes on information theory. Technical report, 2019.

[119] Armen Aghajanyan, Luke Zettlemoyer, and Sonal Gupta. Intrinsic dimensionality explains the effectiveness of language model fine-tuning. *arXiv preprint arXiv:2012.13255*, 2020.

[120] Eckart Zitzler, Kalyanmoy Deb, and Lothar Thiele. Comparison of multiobjective evolutionary algorithms: Empirical results. *Evolutionary computation*, 8(2):173–195, 2000.

[121] Hui Tang and Kui Jia. A new benchmark: On the utility of synthetic data with blender for bare supervised learning and downstream domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15954–15964, 2023.

[122] Zhenbin Wang, Lei Zhang, Lituan Wang, and Minjuan Zhu. Landa: Language-guided multi-source domain adaptation. *arXiv preprint arXiv:2401.14148*, 2024.

[123] Mainak Singha, Harsh Pal, Ankit Jha, and Biplab Banerjee. Ad-clip: Adapting domains in prompt space using clip. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4355–4364, 2023.

[124] Korawat Tanwisuth, Xinjie Fan, Huangjie Zheng, Shujian Zhang, Hao Zhang, Bo Chen, and Mingyuan Zhou. A prototype-oriented framework for unsupervised domain adaptation. *Advances in Neural Information Processing Systems*, 34:17194–17208, 2021.

[125] Beier Zhu, Yulei Niu, Yucheng Han, Yue Wu, and Hanwang Zhang. Prompt-aligned gradient for prompt tuning. International Conference on Computer Vision, 2023.

[126] Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M. Hospedales. Learning to generalize: Meta-learning for domain generalization. In Sheila A. McIlraith and Kilian Q. Weinberger, editors, *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 3490–3497. AAAI Press, 2018.

[127] Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M. Hospedales. Sequential learning for domain generalization. *ECCV Workshops*, 2020.

[128] Juncheng Li, Minghe Gao, Longhui Wei, Siliang Tang, Wenqiao Zhang, Meng Li, Wei Ji, Qi Tian, Tat seng Chua, and Yueting Zhuang. Gradient-regulated meta-prompt learning for generalizable vision-language models. *IEEE International Conference on Computer Vision*, 2023.

Supplement to "Enhancing Domain Adaptation through Prompt Gradient Alignment"

Due to space constraints, some details were omitted from the main paper. We therefore include additional theoretical developments (section A), the detailed algorithm description (section B) and experimental results (section C) in this appendix.

A UDA generalization bound

Here, we provide an information-theoretic generalization bound for UDA, which can be reduced by our gradient alignment and gradient norm penalization. For simplicity, we will consider the single-source UDA setting.

To begin, we first define some additional notations: let $\mathcal{X}, \mathcal{Y}, \mathcal{P}$ be the input space, output space, and prompt space (or hypothesis space), respectively. Denote the input-label space as $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$, and the loss function as $\ell : \mathcal{P} \times \mathcal{Z} \to \mathbb{R}_0^+$ (Cross entropy in our case). Finally, denote μ, μ' as the two underlying distributions from which the source and target domains are sampled. The training data for source domain $D_S = \{\boldsymbol{Z}_i\}_{i=1}^N$ is drawn i.i.d from $\mu^{\otimes N}$, and that for target domain $D_T = \{\boldsymbol{X}'_j\}_{j=1}^M$ is from $\mu_X^{'\otimes M}$.

For each prompt parameter, the population risk in the target domain is defined as

$$R_{\mu'}(\boldsymbol{P}) := \mathbb{E}_{\boldsymbol{Z'}\sim\mu'}[\ell(\boldsymbol{P}, \boldsymbol{Z'})]. \tag{9}$$

This risk is the ultimate goal that a UDA algorithm aims to minimize. However, since μ' is unknown, and only a finite number of training data is given, we define the empirical risk in the source domain as

$$R_{D_S}(\boldsymbol{P}) := \frac{1}{N}\sum_{i=1}^{N} \ell(\boldsymbol{P}, \boldsymbol{Z}_i). \tag{10}$$

In the information-theoretic analysis framework, model parameter, $\boldsymbol{P} \in \mathcal{P}$ in our case, is a random variable that is outputted from a learning algorithm $\mathcal{A}$ characterized by some conditional distribution $P_{\boldsymbol{P}|D_S,D_T}$. Then the generalization error, measuring how close these two risks can be, has the form

$$Err := \mathbb{E}_{\boldsymbol{P},D_S,D_T}[R_{\mu'}(\boldsymbol{P}) - R_{D_S}(\boldsymbol{P})], \tag{11}$$

where the expectation is taken over $\boldsymbol{P}, D_S, D_T \sim P_{\boldsymbol{P}|D_S,D_T}, \mu^{\otimes N}, \mu_X^{'\otimes M}$.

To derive the bound, we need the following assumption on the loss function, which is commonly adopted in many information-theoretic bounds such as those in [112, 113]:

Assumption A.1. (Subgaussianity). $\ell(\boldsymbol{P}; \boldsymbol{Z'})$ is R-subgaussian [*] under $P_{\boldsymbol{P},\boldsymbol{Z}'|\boldsymbol{X}'_j=\boldsymbol{x}'_j}, \forall \boldsymbol{x}'_j \in \mathcal{X}$, for any $\boldsymbol{P} \in \mathcal{P}$.

We also present the definitions of Mutual Information, Disintegrated Mutual Information, and Conditional Mutual Information:

Definition A.2. (Mutual Information). $I(X;Y) = \mathrm{KL}(P_{X,Y}||P_X \otimes P_Y)$, where KL is the KL-divergence and $\otimes$ denote the product of two marginal distributions.

Definition A.3. (Disintegrated Mutual Information). The disintegrated mutual information between two random variables X and Y given a realization of a variable $Z = z$ is

$$I^{Z=z}(X;Y) = \mathrm{KL}(P_{X,Y|Z=z}||P_{X|Z=z} \otimes P_{Y|Z=z})$$

Definition A.4. (Conditional Mutual Information). $I(X,Y|Z) = \mathbb{E}_Z I^Z(X;Y)$.

[*] A random variable X is R-subgaussian if for any $\rho, \log \mathbb{E}\exp(\rho(X - \mathbb{E}X)) \leq \rho^2 R^2/2$.

https://doi.org/10.52202/079017-1447

Theorem A.5. *Under assumption A.1, the generalization error can be upper-bounded by*

$$|Err| \leq \sqrt{\frac{4R^2}{N}\sum_{t=1}^{\mathcal{T}}\frac{\eta_t^2}{\sigma_t^2}\mathbb{E}_{\boldsymbol{P}_{t-1},D_S,D_T}[\|\boldsymbol{g}_t^{src}\|^2 + \|\boldsymbol{g}_t^{tgt}\|^2 + \|\boldsymbol{g}_t^{src} - \boldsymbol{g}_t^{tgt}\|^2]} + \sqrt{2R^2 KL(\mu||\mu')}, \tag{12}$$

where $\mathcal{T}$ is the total number of training iterations, η_t is the learning rate at iteration t, $\boldsymbol{P}_t$ is the prompt at iteration t, $\boldsymbol{g}_t^{src} = \nabla_{\boldsymbol{P}}\mathcal{L}_{src}(\boldsymbol{P}_{t-1})$, $\boldsymbol{g}_t^{tgt} = \nabla_{\boldsymbol{P}}\mathcal{L}_{tgt}(\boldsymbol{P}_{t-1})$ are the gradients w.r.t $\boldsymbol{P}_{t-1}$ of source loss Eq.2 and target loss Eq.3, and σ_t is the standard deviation of the isotropic Gaussian noise added to the update of $\boldsymbol{P}_t$.

Remark A.6. *For the purpose of simplicity, here we consider a 'noisy' update version of prompts: $\boldsymbol{P}_t = \boldsymbol{P}_{t-1} - \eta_t \boldsymbol{g} + N_t, N_t \sim \mathcal{N}(\boldsymbol{0}, \sigma_t^2 \boldsymbol{I})$. However, note that the bound still holds for the conventional SGD update, i.e., no added noise, by following techniques in [113].*

Remark A.7. *Our methods align gradients of shared-prompt, but here we can omit its subscript in the inter-domain gradient matching term, $\|\boldsymbol{g}_t^{src} - \boldsymbol{g}_t^{tgt}\|^2$, by noting that $\boldsymbol{g}_t^{src} = [\boldsymbol{g}_t^{sh,src}, \boldsymbol{g}_t^{S}, \boldsymbol{0}]$ and $\boldsymbol{g}_t^{tgt} = [\boldsymbol{g}_t^{sh,tgt}, \boldsymbol{0}, \boldsymbol{g}_t^{T}]$. Indeed, $\|\boldsymbol{g}_t^{src} - \boldsymbol{g}_t^{tgt}\|^2 = \|\boldsymbol{g}_t^{src}\|^2 + \|\boldsymbol{g}_t^{tgt}\|^2 - 2(\boldsymbol{g}_t^{sh,src})^{\mathbb{T}}\boldsymbol{g}_t^{sh,tgt}$, where $\mathbb{T}$ denotes the vector transpose. In addition, this bound suggests maximizing the dot product between gradients; however, to stabilize training, we aim to maximize the cosine similarity instead.*

This theorem suggests that penalizing gradient norm and matching gradients across domains can improve generalization on the target domain, i.e., the first term in the R.H.S of A.5 is minimized. Note that minimizing gradient norm has been widely used in [59, 81, 114] to control the sharpness of the loss landscape, which is strongly related to the generalization capability of the model. In this work, we can empirically and theoretically verify the effectiveness of this technique in the gradient space of prompt, consistent with results in previous works [115, 116].

Regarding the second term, we do not aim for a method that can explicitly reduce the gap between source and target distributions, because we do not want to remove any domain-specific features that may be helpful for prediction. Instead, we want to capture domain-agnostic features in the shared prompt, and specific features in the domain-specific ones so that at inference, a more meaningful representation can be obtained by using these prompts. Hence, one possible direction for future work is to design and learn prompts such that domain distribution alignment can also be achieved.

Finally, this bound can grow as the number of training iterations increases unless gradient norms and the difference between source and target gradients are extremely small at final iterations. Future work could be overcoming this limitation by considering other bounds, such as ones suggested in [117].

Proof. Our bound is inspired from the bound in Theorem 5.1 in [112], which is restated as the following lemma

Lemma A.8. *Under assumption A.1, the generalization error can be upper-bounded by*

$$|Err| \leq \frac{1}{NM}\sum_{j=1}^{M}\sum_{i=1}^{N}\mathbb{E}_{\boldsymbol{X}'_j}\sqrt{2R^2 I^{\boldsymbol{X}'_j}(\boldsymbol{P};\boldsymbol{Z}_i)} + \sqrt{2R^2 KL(\mu||\mu')} \tag{13}$$

$$\leq \sqrt{\frac{2R^2}{N} I(\boldsymbol{P}; D_S|D_T)} + \sqrt{2R^2 KL(\mu||\mu')} \tag{14}$$

Now consider the 'noisy' update of the prompt as presented in Eqs. 6 and 5:

$$\boldsymbol{P}_t = \boldsymbol{P}_{t-1} - \eta_t(\nabla_{\boldsymbol{P}}\mathcal{L}_{src}(\boldsymbol{P}_{t-1}) + \nabla_{\boldsymbol{P}}\mathcal{L}_{tgt}(\boldsymbol{P}_{t-1})) + N_t \tag{15}$$

$$:= \boldsymbol{P}_{t-1} - \eta_t \boldsymbol{g}_t^{src} - \eta_t \boldsymbol{g}_t^{tgt} + N_t. \tag{16}$$

Assume that we obtain the final prompts after $\mathcal{T}$ iterations, then following the chain rule of mutual information and data processing inequality, we have

$$
\begin{aligned}
I(\boldsymbol{P}_{\mathcal{T}}; D_S|D_T) &= I(\boldsymbol{P}_{\mathcal{T}-1} - \eta_{\mathcal{T}}\boldsymbol{g}_{\mathcal{T}}^{src} - \eta_{\mathcal{T}}\boldsymbol{g}_{\mathcal{T}}^{tgt} + N_{\mathcal{T}}; D_S|D_T) && (17)\\
&\leq I\left(\boldsymbol{P}_{\mathcal{T}-1}, -\eta_{\mathcal{T}}\boldsymbol{g}_{\mathcal{T}}^{src} - \eta_{\mathcal{T}}\boldsymbol{g}_{\mathcal{T}}^{tgt} + N_{\mathcal{T}}; D_S|D_T\right) && (18)\\
&= I(\boldsymbol{P}_{\mathcal{T}-1}; D_S|D_T) + I(-\eta_{\mathcal{T}}\boldsymbol{g}_{\mathcal{T}}^{src} - \eta_{\mathcal{T}}\boldsymbol{g}_{\mathcal{T}}^{tgt} + N_{\mathcal{T}}; D_S|D_T, \boldsymbol{P}_{\mathcal{T}-1}) && (19)\\
&\vdots && (20)\\
&\leq \sum_{t=1}^{\mathcal{T}} I(-\eta_t\boldsymbol{g}_t^{src} - \eta_t\boldsymbol{g}_t^{tgt} + N_t; D_S|D_T, \boldsymbol{P}_{t-1}) && (21)\\
&= \sum_{t=1}^{\mathcal{T}} I(-\boldsymbol{g}_t^{src} - \boldsymbol{g}_t^{tgt} + N_t/\eta_t; D_S|D_T, \boldsymbol{P}_{t-1}) && (22)\\
&\leq \sum_{t=1}^{\mathcal{T}} I\left(-\boldsymbol{g}_t^{src} + \frac{N_t}{2\eta_t}, -\boldsymbol{g}_t^{tgt} + \frac{N_t}{2\eta_t}; D_S|D_T, \boldsymbol{P}_{t-1}\right) && (23)\\
&= \sum_{t=1}^{\mathcal{T}} I\left(-\boldsymbol{g}_t^{src} + \frac{N_t}{2\eta_t}; D_S|D_T, \boldsymbol{P}_{t-1}\right)\\
&\quad + I\left(-\boldsymbol{g}_t^{tgt} + \frac{N_t}{2\eta_t}; D_S|D_T, \boldsymbol{P}_{t-1}, -\boldsymbol{g}_t^{src} + \frac{N_t}{2\eta_t}\right) && (24)
\end{aligned}
$$

Eq. 21 is due to the assumption of independence of $\boldsymbol{P}_0$ w.r.t D_S and D_T, and Eq. 22 is because mutual information is scale-invariant.

Consider the first term in Eq. 24, for all t, inspired by the proof of Lemma 3 in [118], we have

$$
\begin{aligned}
&I\left(-\boldsymbol{g}_t^{src} + \frac{N_t}{2\eta_t}; D_S|D_T, \boldsymbol{P}_{t-1}\right)\\
&= \mathbb{E}_{D_S, D_T, \boldsymbol{P}_{t-1}}\left[\mathrm{KL}\left(P_{-\boldsymbol{g}_t^{src}+\frac{N_t}{2\eta_t}|D_T, \boldsymbol{P}_{t-1}, D_S} || P_{-\boldsymbol{g}_t^{src}+\frac{N_t}{2\eta_t}|D_T, \boldsymbol{P}_{t-1}}\right)\right] && (25)\\
&= \mathbb{E}_{D_S, D_T, \boldsymbol{P}_{t-1}}\left[\mathrm{KL}\left(P_{-\boldsymbol{g}_t^{src}+\frac{N_t}{2\eta_t}|D_T, \boldsymbol{P}_{t-1}, D_S} || P_{\tilde{\boldsymbol{G}}_t|D_T, \boldsymbol{P}_{t-1}}\right) - \mathrm{KL}\left(P_{-\boldsymbol{g}_t^{src}+\frac{N_t}{2\eta_t}|D_T, \boldsymbol{P}_{t-1}} || P_{\tilde{\boldsymbol{G}}_t|D_T, \boldsymbol{P}_{t-1}}\right)\right] && (26)\\
&\leq \mathbb{E}_{D_S, D_T, \boldsymbol{P}_{t-1}}\left[\mathrm{KL}\left(P_{-\boldsymbol{g}_t^{src}+\frac{N_t}{2\eta_t}|D_T, \boldsymbol{P}_{t-1}, D_S} || P_{\tilde{\boldsymbol{G}}_t|D_T, \boldsymbol{P}_{t-1}}\right)\right], && (27)
\end{aligned}
$$

where $P_{\tilde{\boldsymbol{G}}_t|D_T, \boldsymbol{P}_{t-1}}$ is some random distribution, every choice of which results in a upper bound for the MI, and the equality holds when $P_{\tilde{\boldsymbol{G}}_t|D_T, \boldsymbol{P}_{t-1}} = P_{-\boldsymbol{g}_t^{src}+\frac{N_t}{2\eta_t}|D_T, \boldsymbol{P}_{t-1}}$.

Therefore, if we choose $P_{\tilde{\boldsymbol{G}}_t|D_T, \boldsymbol{P}_{t-1}} = \mathcal{N}(\boldsymbol{0}, \frac{\sigma_t^2}{4\eta_t^2}\boldsymbol{I})$, the R.H.S of Eq. 27 will be upper-bounded by $\frac{2\eta_t^2}{\sigma_t^2}\mathbb{E}_{D_S, D_T, \boldsymbol{P}_{t-1}}\left[||\boldsymbol{g}_t^{src}||^2\right]$, which is derived from the KL-divergence between two Gaussian distributions.

Similarly for the second term in Eq. 24, choosing $P_{\tilde{\boldsymbol{G}}_t|D_T, \boldsymbol{P}_{t-1}, -\boldsymbol{g}_t^{src}+\frac{N_t}{2\eta_t}} = \mathcal{N}(\boldsymbol{0}, \frac{\sigma_t^2}{4\eta_t^2}\boldsymbol{I})$ gives us the upper bound $\frac{2\eta_t^2}{\sigma_t^2}\mathbb{E}_{D_S, D_T, \boldsymbol{P}_{t-1}}\left[||\boldsymbol{g}_t^{tgt}||^2\right]$. Furthermore, letting $P_{\tilde{\boldsymbol{G}}_t|D_T, \boldsymbol{P}_{t-1}, -\boldsymbol{g}_t^{src}+\frac{N_t}{2\eta_t}} = P_{-\boldsymbol{g}_t^{src}+\frac{N_t}{2\eta_t}}$, which is also a Gaussian distribution due to the effect of the added noise, we reach the gradient matching term, $\frac{2\eta_t^2}{\sigma_t^2}\mathbb{E}_{D_S, D_T, \boldsymbol{P}_{t-1}}\left[||\boldsymbol{g}_t^{src} - \boldsymbol{g}_t^{tgt}||^2\right]$.

Note that in Eq. 15, we suppose the source-weight term $\lambda = 1$ to simplify proof. However, if one wishes to keep the impact of λ, they can change $\boldsymbol{g}_t^{src} = [\lambda\boldsymbol{g}_t^{sh,src}, \boldsymbol{g}_t^S, \boldsymbol{0}]$. In this case, the terms under the expectation in the bound will become: $(\lambda^2-1)||\boldsymbol{g}_t^{sh,src}||^2 + ||\boldsymbol{g}_t^{src}||^2 + ||\boldsymbol{g}_t^{tgt}||^2 + ||\boldsymbol{g}_t^{tgt} - \lambda\boldsymbol{g}_t^{src}||^2$.

Combining everything together, the proof is done. □

https://doi.org/10.52202/079017-1447

B Algorithm

B.1 Final objectives

As we cast UDA as a MOO problem, the ideal final objectives, in the case of single-source UDA, would be

$$[\mathcal{L}_S^{\text{PGA}}(\boldsymbol{P}), \mathcal{L}_T^{\text{PGA}}(\boldsymbol{P})],$$

where

$$\mathcal{L}_T^{\text{PGA}}(\boldsymbol{P}) := \mathcal{L}_T(\boldsymbol{P}_{sh} - \rho_{ga}\frac{\boldsymbol{g}_{sh,S}}{\|\boldsymbol{g}_{sh,S}\|.\|\boldsymbol{g}_{sh,T}\|} + \rho_{gn}\frac{\boldsymbol{g}_{sh,T}}{\|\boldsymbol{g}_{sh,T}\|}, \boldsymbol{P}_T + \rho_{gn}\frac{\boldsymbol{g}_T}{\|\boldsymbol{g}_T\|}),$$
$$\mathcal{L}_S^{\text{PGA}}(\boldsymbol{P}) := \mathcal{L}_S(\boldsymbol{P}_{sh} - \rho_{ga}\frac{\boldsymbol{g}_{sh,T}}{\|\boldsymbol{g}_{sh,S}\|.\|\boldsymbol{g}_{sh,T}\|} + \rho_{gn}\frac{\boldsymbol{g}_{sh,S}}{\|\boldsymbol{g}_{sh,S}\|}, \boldsymbol{P}_S + \rho_{gn}\frac{\boldsymbol{g}_S}{\|\boldsymbol{g}_S\|}).$$

As aforementioned, we use scalarization method, i.e. reweighting loss functions with λ put on the PGA source objective. As a result, the PGA gradient updates for prompts are

$$\begin{aligned}
\boldsymbol{g}_{sh,T}^{\text{PGA}}, \boldsymbol{g}_T^{\text{PGA}} &:= \nabla_{\boldsymbol{P}}\mathcal{L}_T^{\text{PGA}}(\boldsymbol{P}), \quad \boldsymbol{g}_{sh,S}^{\text{PGA}}, \boldsymbol{g}_S^{\text{PGA}} := \nabla_{\boldsymbol{P}}\mathcal{L}_S^{\text{PGA}}(\boldsymbol{P}), \\
\boldsymbol{P}_S &= \boldsymbol{P}_S - \eta\boldsymbol{g}_S^{\text{PGA}}, \qquad \boldsymbol{P}_T = \boldsymbol{P}_T - \eta\boldsymbol{g}_T^{\text{PGA}}, \\
\boldsymbol{P}_{sh} &= \boldsymbol{P}_{sh} - \eta(\boldsymbol{g}_{sh,T}^{\text{PGA}} + \lambda\boldsymbol{g}_{sh,S}^{\text{PGA}}).
\end{aligned}$$

However, computing these PGA gradients will trigger the computation of the Hessian matrix. Hence, we approximate them with a practical version:

$$\begin{aligned}
\boldsymbol{g}_{sh,T}^{\text{PGA}}, \boldsymbol{g}_T^{\text{PGA}} &:= \nabla_{\boldsymbol{P}}\mathcal{L}_T^{\text{PGA}}(\boldsymbol{P}) \\
&\approx \nabla_{\boldsymbol{P}}\mathcal{L}_T(\boldsymbol{P}_{sh}, \boldsymbol{P}_T)|_{\boldsymbol{P}_{sh}=\boldsymbol{P}_{sh}-\rho_{ga}\frac{\boldsymbol{g}_{sh,S}}{\|\boldsymbol{g}_{sh,S}\|.\|\boldsymbol{g}_{sh,T}\|}+\rho_{gn}\frac{\boldsymbol{g}_{sh,T}}{||\boldsymbol{g}_{sh,T}||}, \boldsymbol{P}_T=\boldsymbol{P}_T+\rho_{gn}\frac{\boldsymbol{g}_T}{||\boldsymbol{g}_T||}}, \\
\boldsymbol{g}_{sh,S}^{\text{PGA}}, \boldsymbol{g}_S^{\text{PGA}} &:= \nabla_{\boldsymbol{P}}\mathcal{L}_S^{\text{PGA}}(\boldsymbol{P}) \\
&\approx \nabla_{\boldsymbol{P}}\mathcal{L}_S(\boldsymbol{P}_{sh}, \boldsymbol{P}_S)|_{\boldsymbol{P}_{sh}=\boldsymbol{P}_{sh}-\rho_{ga}\frac{\boldsymbol{g}_{sh,T}}{\|\boldsymbol{g}_{sh,S}\|.\|\boldsymbol{g}_{sh,T}\|}+\rho_{gn}\frac{\boldsymbol{g}_{sh,S}}{||\boldsymbol{g}_{sh,S}||}, \boldsymbol{P}_S=\boldsymbol{P}_S+\rho_{gn}\frac{\boldsymbol{g}_S}{||\boldsymbol{g}_S||}}.
\end{aligned}$$

B.2 Extension to Multi-source UDA

Our method can be easily extended to work with multi-source domains by noting that the target gradient is aligned with each of the source gradients.

$$\begin{aligned}
\boldsymbol{g}_{sh,T}^{\text{PGA}}, \boldsymbol{g}_T^{\text{PGA}} &:= \nabla_{\boldsymbol{P}}\mathcal{L}_T^{\text{PGA}}(\boldsymbol{P}), \\
\boldsymbol{g}_{sh,i}^{\text{PGA}}, \boldsymbol{g}_{S,i}^{\text{PGA}} &:= \nabla_{\boldsymbol{P}}\mathcal{L}_{S,i}^{\text{PGA}}(\boldsymbol{P}), \forall i = 1 \to N \\
\boldsymbol{P}_{S,i} &= \boldsymbol{P}_{S,i} - \eta\boldsymbol{g}_{S,i}^{\text{PGA}}, \forall i = 1 \to N \\
\boldsymbol{P}_T &= \boldsymbol{P}_T - \eta\boldsymbol{g}_T^{\text{PGA}}, \\
\boldsymbol{P}_{sh} &= \boldsymbol{P}_{sh} - \eta(\boldsymbol{g}_{sh,T}^{\text{PGA}} + \lambda\sum_i \boldsymbol{g}_{sh,i}^{\text{PGA}}), \\
\boldsymbol{g}_{sh,T}^{\text{PGA}}, \boldsymbol{g}_T^{\text{PGA}} &\approx \nabla_{\boldsymbol{P}}\mathcal{L}_T(\boldsymbol{P}_{sh}, \boldsymbol{P}_T)|_{\boldsymbol{P}_{sh}=\boldsymbol{P}_{sh}-\rho_{ga}\sum_i\frac{\boldsymbol{g}_{sh,i}}{\|\boldsymbol{g}_{sh,i}\|.\|\boldsymbol{g}_{sh,T}\|}+\rho_{gn}\frac{\boldsymbol{g}_{sh,T}}{||\boldsymbol{g}_{sh,T}||}, \boldsymbol{P}_T=\boldsymbol{P}_T+\rho_{gn}\frac{\boldsymbol{g}_T}{||\boldsymbol{g}_T||}}, \qquad (28) \\
\boldsymbol{g}_{sh,i}^{\text{PGA}}, \boldsymbol{g}_{S,i}^{\text{PGA}} &\approx \nabla_{\boldsymbol{P}}\mathcal{L}_{S,i}(\boldsymbol{P}_{sh}, \boldsymbol{P}_{S,i})|_{\boldsymbol{P}_{sh}=\boldsymbol{P}_{sh}-\rho_{ga}\frac{\boldsymbol{g}_{sh,T}}{\|\boldsymbol{g}_{sh,i}\|.\|\boldsymbol{g}_{sh,T}\|}+\rho_{gn}\frac{\boldsymbol{g}_{sh,i}}{||\boldsymbol{g}_{sh,i}||}, \boldsymbol{P}_{S,i}=\boldsymbol{P}_{S,i}+\rho_{gn}\frac{\boldsymbol{g}_{S,i}}{||\boldsymbol{g}_{S,i}||}}. \qquad (29)
\end{aligned}$$

The details of our proposed method for the general case of N source domains are presented in Algorithm 1. When $N = 1$, our method degrades to PGA.

C Experimental details

In this section, we provide additional information for our experimental settings in Section C.1 and C.2 then include detailed ablation studies and other empirical results in Section C.3.

Algorithm 1 Prompt gradient alignment for unsupervised domain adaptation

Input: Prompt $\boldsymbol{P} = [\boldsymbol{P}_{sh}, \{\boldsymbol{P}_{S,i}\}_{i=1}^{N}, \boldsymbol{P}_T]$, gradient norm penalization trade-off ρ_{gn}, alignment strength ρ_{ga}, source-gradient trade-off λ, learning rate η.
Output: Updated prompt $\boldsymbol{P}^*$

1: Compute target loss $\mathcal{L}_T(\boldsymbol{P}_{sh}, \boldsymbol{P}_T)$ as in Eq. 3
2: Compute gradients of shared and target-specific prompts w.r.t target loss
$\boldsymbol{g}_{sh,T}, \boldsymbol{g}_T \leftarrow \nabla_{\boldsymbol{P}} \mathcal{L}_T(\boldsymbol{P}_{sh}, \boldsymbol{P}_T)$
3: Compute source losses $\mathcal{L}_{S,i}(\boldsymbol{P}_{sh}, \boldsymbol{P}_{S,i})$ as in Eq. 2
4: Compute gradient of shared and source-specific prompts w.r.t each source loss
$\boldsymbol{g}_{sh,i}, \boldsymbol{g}_{S,i} \leftarrow \nabla_{\boldsymbol{P}} \mathcal{L}_{S,i}(\boldsymbol{P}_{sh}, \boldsymbol{P}_{S,i}), \forall i = 1 \rightarrow N$
5: Compute $\boldsymbol{g}_{sh,T}^{\text{PGA}}, \boldsymbol{g}_T^{\text{PGA}}$ as in Eq. 28
6: Compute $\boldsymbol{g}_{sh,i}^{\text{PGA}}, \boldsymbol{g}_{S,i}^{\text{PGA}}$ as in Eq. 29 $\forall i = 1 \rightarrow N$
7: Compute combined gradient of shared prompt $\boldsymbol{g}_{sh}^{\text{PGA}} = \boldsymbol{g}_{sh,T}^{\text{PGA}} + \lambda \sum_i \boldsymbol{g}_{sh,i}^{\text{PGA}}$
8: Update prompt
$$\boldsymbol{P^*} = [\boldsymbol{P}_{sh}, \{\boldsymbol{P}_{S,i}\}_{i=1}^{N}, \boldsymbol{P}_T] - \eta[\boldsymbol{g}_{sh}^{\text{PGA}}, \{\boldsymbol{g}_{S,i}^{\text{PGA}}\}_{i=1}^{N}, \boldsymbol{g}_T^{\text{PGA}}]$$

C.1 Datasets

ImageCLEF is a small-scaled dataset with 1,800 images across 12 object categories from three domains: ImageNet ILSVRC 2012 (I), Pascal VOC 2012 (P), and Caltech-256 (C). Office-Home is a medium-scaled dataset containing approximately 15,500 images from 65 categories in four domains: Art, Clipart, Product, and Real World. DomainNet is the largest dataset, comprising around 600,000 images from 345 categories across six domains: Clipart, Infograph, Painting, Quickdraw, Real, and Sketch.

C.2 Implementation details

For fair comparisons, we use a ResNet50 as our backbone on Image-CLEF and Office-Home and a ResNet101 on DomainNet. Their weights are taken from pretrained-CLIP and kept frozen during training. Prompts are trained with the mini-batch SGD optimizer with a learning rate of 0.003 and 0.005. We use a batch size of 32 and adopt a cosine learning rate scheduler. For hyper-parameters, token lengths M_1 and M_2 are both set to 16. Pseudo-label threshold τ is set to 0.4 for producing reliable labels. ρ_{gn}, ρ_{ga} and λ are found using grid-search. Details are provided in the public source code.

During inference, we average the prediction of both source $\boldsymbol{P}_S$ and target $\boldsymbol{P}_T$ prompts, which empirically yield the best performance. Please note that the inference cost remains almost the same as using a pretrained CLIP as computing class embeddings is an one-time-cost. The complexity grows linearly with the number of prompts during training ($= 2$ with PGA and $N + 1$ in the case of MPGA), which is typically not a big issue in practice since the model training can quickly converge by fine-tuning under intrinsic dimension [119]. We further confirm this in the computation complexity ablation study below.

C.3 Additional experiments

C.3.1 Illustrative example

We run a small multi-objective-optimization problem on the ZDT-1 problem [120]. The ZDT-1 problems have a 30-dimensional variable and two differentiable objective functions f_1, f_2:

$$\begin{aligned} &\min f_1(x) \\ &\min f_2(x) = g(x) h\left(f_1(x), g(x)\right) \end{aligned}$$

The function $g(x)$ can be considered as the function for convergence, their formulas are given by:

https://doi.org/10.52202/079017-1447

$$
\begin{aligned}
f_1(x) &= x_1 \\
g(x) &= 1 + \frac{9}{n-1}\sum_{i=2}^{n} x_i \\
h\left(f_1, g\right) &= 1 - \sqrt{f_1/g} \\
0 \leq x_i &\leq 1 \quad i = 1, \ldots, n
\end{aligned}
$$

with the Pareto solutions are given by:

$$
0 \leq x_1^* \leq 1 \quad \text{and} \quad x_i^* = 0 \text{ for } i = 2, \ldots, n
$$

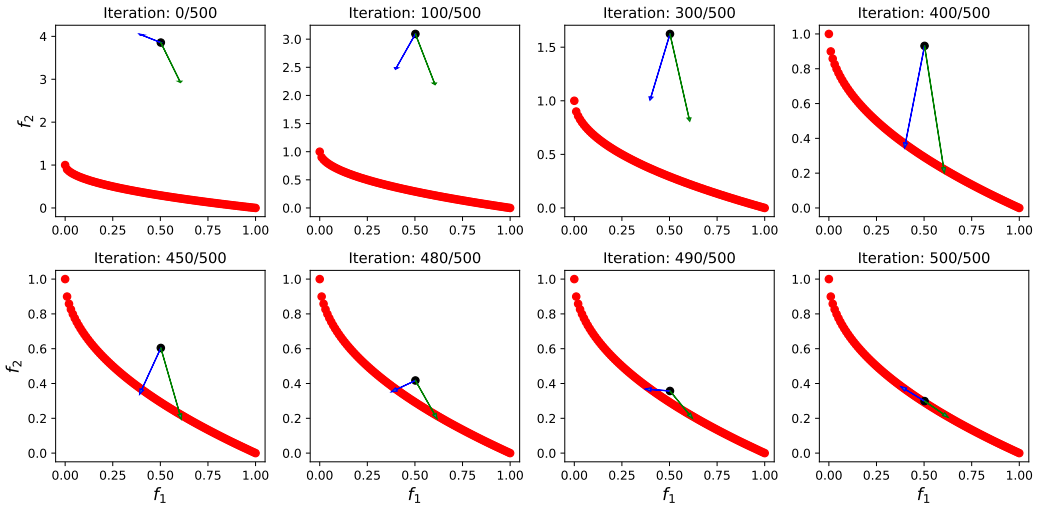

Figure 4: ZDT-1 task-specific gradient directions at different iterations. Red curve represents the Pareto front while the blue and green arrows indicate the updating directions for minimizing f_1 and f_2, respectively.

As can be seen from Figure 4, the cosine similarity increases at the beginning of the training and then decreases when the obtained solution reach the region near the Pareto front. This behavior aligns with the gradient similarity evolution experiment in the main paper.

C.3.2 Large-scale single-source unsupervised domain adaptation

Apart from those experiments in the main paper, we expand the single-source unsupervised domain adaptation setup by including the empirical results on two large-scale synthetic-to-real benchmark for classification adaptation S2RDA-49 and S2RDA-MS-39 [121]. For each task, synthetic samples are created by rendering 3D models from ShapeNet, matching the label space of the real/target domain, with 12K RGB images per class. The S2RDA-49 real domain contains 60,535 images across 49 classes from various sources including the ImageNet validation set. The S2RDA-MS-39 real domain includes 41,735 natural images for 39 classes from MetaShift, featuring complex contexts like object co-occurrence and attributes, which adds to the task's difficulty.

Table 6: Unsupervised domain adaptation results on S2RDA. The best accuracy is indicated in **bold**.

Transfer Task	No Adaptation		DANN		MCD		RCA		SRDC		DisClusterDA		CLIP		DAPL		PGA (Ours)	
	Acc.	Mean	Acc.	Mean	Acc.	Mean	Acc.	Mean	Acc.	Mean	Acc.	Mean	Acc.	Mean	Acc.	Mean	Acc.	Mean
S2RDA-49	51.9	42.2	47.1	47.6	42.5	47.8	47.1	48.5	61.5	53.0	53.0	52.3	69.9	65.7	71.5	66.5	**74.1**	**67.8**
S2RDA-MS-39	22.0	20.5	22.8	22.2	22.1	22.2	23.3	22.5	25.8	24.6	27.1	25.3	36.4	35.8	36.9	35.7	**38.0**	**36.9**

Table 6 illustrates accuracy and mean score over classes, where utilizing pretrained vision-language models still shows their impressive performance. Using pretrained CLIP standalone outperforms other traditional DA methods and PGA further boosts the performance by large margins, 4% on S2RDA-49 and 1.5% on S2RDA-MS-39, respectively.

C.3.3 Ablation studies

Similar to previous work on CLIP adaptation[25, 28], we vary the pseudo label threshold τ value to study its sensitivity. As can be seen in Figure 7, both PGA and MPGA's performance is relatively stable across different values of τ, indicating that our methods are not sensitive to τ, and the best result is obtained at a reasonable trade-off between the quantity and quality of pseudo data.

Table 7: Accuracy (%) of different threshold τ on ImageCLEF.

	0.1	**0.2**	**0.3**	**0.4**	**0.5**	**0.6**	**0.8**	**0.9**
PGA	91.5	92.0	92.1	92.4	92.3	92.1	92.0	92.0
MPGA	92.4	92.6	92.9	92.7	92.7	92.7	92.6	92.5

In Figure 5, we provide the complexity for some comparative baselines. Accuracy curve (left): While DANN and CDAN obtain their best performance at approximately 77% after more than 1000s, PGA and MPGA achieve 84% within 100s. Besides, the first stage of pairwise source-target training of MPA takes 159s, followed by 35s for the second stage to actually train the final model. Number of Trainable Parameters (middle): PGA and MPGA, with fewer than 140k parameters, require significantly fewer parameters than MPA, DANN and CDAN, which have around 1M, 48.9M and 51.7M parameters, respectively. GPU Memory Usage (right) PGA, MPGA, and MPA exhibit substantially lower memory footprints, around 1300MB compared to 7000MB of DANN and CDAN throughout training.

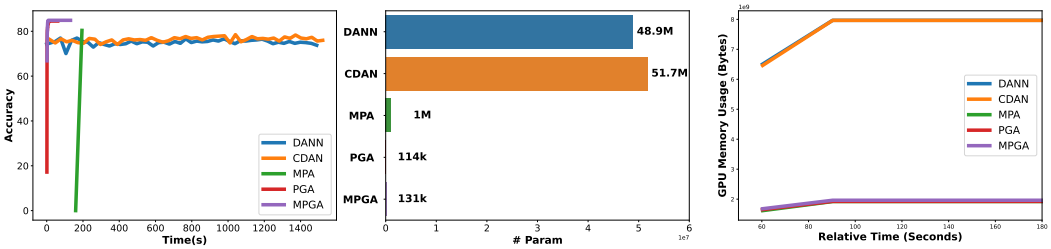

Figure 5: Computational complexity: accuracy curve (left), number of trainable parameters (middle), and GPU memory (right).

Figure 6 shows that PGA is generally not sensitive to ρ_{ga} and ρ_{gn} within their acceptable range, i.e. 1e-2 to 10 for ρ_{ga} and 1e-5 to 0.1 for ρ_{gn}. Specifically, (i) a too large value of ρ_{gn} is less effective than smaller ones; (ii) ImageCLEF prefers larger values of ρ_{ga} while OfficeHome prefers smaller ones, suggesting that source and target domains in the former dataset may be more similar than those in the latter, hence over-matching gradients in the latter dataset may be adverse.

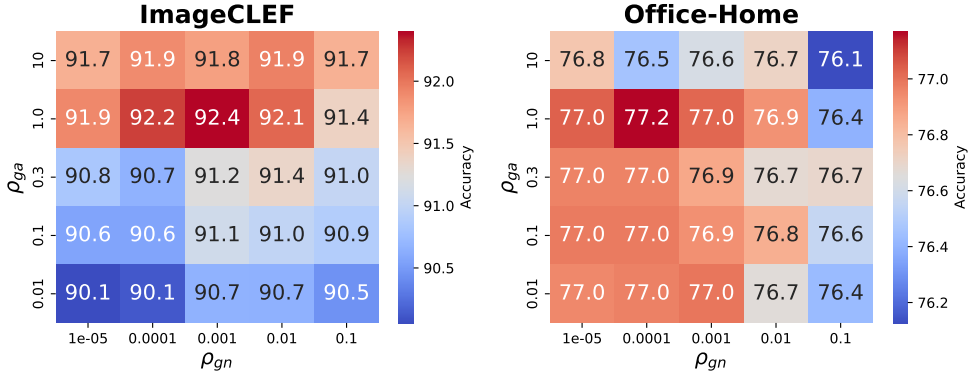

Figure 6: Parameter sensitivity analysis on ρ_{gn} and ρ_{ga} of PGA on ImageCLEF and Office-Home with CLIP-RN50 backbone.

https://doi.org/10.52202/079017-1447

We present results of our methods using ViT-B/16, ViT-L/14 backbones on OfficeHome in Tables 8 and 9, following experimental setups in [122, 123]. We can observe the superiority of our methods among all baselines while finetuning a small portion of the backbones using prompt tuning. Especially, PGA outperforms the second-best method on ViT-B/16 backbone by $\approx 1\%$ accuracy score.

Table 8: Accuracy (%) on Office-Home of ViT-based vision encoder CLIP backbones (except CDTrans* uses DeiT). The overall best accuracy and best within per backbone are indicated in **bold** and underscore, respectively.

Method	Backbone	Ar→Cl	Ar→Pr	Ar→Rw	Cl→Ar	Cl→Pr	Cl→Rw	Pr→Ar	Pr→Cl	Pr→Rw	Rw→Ar	Rw→Cl	Rw→Pr	Avg
CDTrans*		68.8	85.0	86.9	81.5	87.1	87.3	79.6	63.3	88.2	82.0	66.0	90.6	80.5
TVT		74.9	86.8	89.5	82.8	88.0	88.3	79.8	71.9	90.1	85.5	74.6	90.6	83.6
linear probe CLIP		60.1	73.7	80.9	66.4	76.4	76.3	63.4	61.0	82.3	74.7	64.8	83.3	72.4
CoOp		70.0	90.8	90.9	83.2	90.9	89.2	82.0	71.8	90.5	83.8	71.5	92.0	83.9
CoCoOp		70.4	91.4	90.4	83.5	91.8	90.3	83.4	70.9	91.0	83.4	71.6	91.7	84.1
VPT-shallow		66.9	89.1	89.1	81.7	89.0	89.2	81.6	70.0	89.1	81.7	66.9	89.0	81.7
VPT-deep	ViT-B/16	71.6	89.9	90.3	82.8	91.0	89.7	82.0	71.5	90.3	84.6	71.7	91.6	83.9
IVLP		71.4	91.7	90.8	83.6	90.2	89.3	82.2	72.4	90.4	84.1	72.1	92.0	84.2
MaPLe		72.2	91.6	90.3	82.6	90.9	89.8	82.4	71.6	90.0	85.1	72.0	92.1	84.2
CLIP		67.8	89.0	89.8	82.9	89.0	89.8	82.9	67.8	89.8	82.9	67.8	89.0	82.4
DAPL		70.6	90.2	91.0	84.9	89.2	90.9	84.8	70.5	90.6	84.8	70.1	90.8	84.0
PGA (Ours)		71.8	91.5	91.0	84.8	91.6	90.9	84.9	71.5	91.1	85.9	72.1	92.4	85.1
CLIP		74.2	93.1	93.3	87.3	93.1	93.3	87.3	74.2	93.3	87.3	74.2	93.1	87.0
DAPL	ViT-L/14	77.3	94.6	**94.3**	88.6	94.6	94.0	88.8	76.8	94.0	**89.0**	77.8	94.4	88.7
PGA (Ours)		**79.0**	**95.1**	**94.3**	**88.9**	**95.1**	**94.2**	**88.9**	**78.8**	**94.2**	88.9	**79.0**	**95.3**	**89.4**

Following a different protocol, Table 9 provides the results of ViT-L/14 backbones on Office-Home but with three source domains per category on Art, Clipart, Realworld and Product domain. In this setup, MPGA and PGA still consistently yield the best and second-best scores among all categories.

Table 9: Three-source domain adaptation of the Office-Home dataset on ViT-L/14.

Method	→ Ar	→ Rw	→ Pr	Avg
CLIP ZS(G)	84.97	91.94	90.96	89.29
CLIP ZS(A)	86.34	92.10	87.73	88.73
CLIP LP	87.02	92.55	92.70	90.76
LADS	87.71	93.86	93.00	91.52
LanDA	88.83	94.09	93.22	92.05
PGA (Ours)	89.17	95.37	94.34	92.96
MPGA (Ours)	**89.88**	**95.49**	**94.97**	**93.45**

C.3.4 Domain adaptation with label shift

This section is to study how does the method performs when there are extreme label distribution shifts between source and target domains. We test PGA on the setting of label shift following [124], where the source or target domains are down-sampled with only 30% of data from the first-half of the classes are taken (indicated by s- prefix).

Table 10: Accuracy (%) on the sub-sampled Office-Home for unsupervised domain adaptation. The prefix s- denotes the domain where we sample only 30% of the images from the first half of its classes, following the label shift setting from prior work.

Method	sAr→Cl	sAr→Pr	sAr→Rw	sCl→Ar	sCl→Pr	sCl→Rw	sPr→Ar	sPr→Cl	sPr→Rw	sRw→Ar	sRw→Cl	sRw→Pr	Avg
ResNet-50	35.7	54.7	62.6	43.7	52.5	56.6	44.3	33.0	65.2	57.1	40.5	70.0	51.4
DANN	36.1	54.2	61.7	44.3	52.6	56.4	44.6	37.1	65.2	56.7	43.2	69.9	51.8
JAN	34.5	56.9	64.5	46.2	56.8	59.0	50.6	37.2	70.0	58.7	40.6	72.0	53.9
CDAN	38.9	56.8	64.8	48.0	60.0	61.2	49.7	41.4	70.2	62.4	47.0	74.7	56.3
IWDANN	39.8	63.0	68.7	47.4	61.1	60.4	50.4	41.6	72.5	61.0	49.4	76.1	57.6
IWJAN	36.2	61.0	66.3	48.7	59.9	61.9	52.9	37.7	70.9	60.3	41.5	73.3	55.9
IWCDAN	43.0	65.0	71.3	52.9	64.7	66.5	54.9	44.8	75.9	67.0	50.5	78.6	61.2
PCT	51.9	69.7	76.5	63.3	70.8	71.1	66.0	49.9	82.0	73.1	**58.6**	83.2	67.8
PGA (Ours)	**54.7**	**85.4**	**85.4**	**75.3**	**84.4**	**85.2**	**75.4**	**54.9**	**85.7**	**75.6**	54.3	**85.7**	**75.2**

Label-shift results presented in Table 10 and 11 below and Table 5 in the main text show the effectiveness of PGA on different levels of label shift. PGA consistently yields superior performance on every sub-experiment under these two setups.

Table 11: Accuracy (%) on the sub-sampled (target) Office-Home for unsupervised domain adaptation.

Method	Ar→sCl	Ar→sPr	Ar→sRw	Cl→sAr	Cl→sPr	Cl→sRw	Pr→sAr	Pr→sCl	Pr→sRw	Rw→sAr	Rw→sCl	Rw→sPr	Avg
ResNet-50	41.5	65.8	73.6	52.2	59.5	63.6	51.5	36.4	71.3	65.2	42.8	75.4	58.2
DANN	47.8	55.9	66.0	45.3	54.8	56.8	49.4	48.0	70.2	65.4	55.5	72.7	58.3
JAN	45.8	69.7	74.9	53.9	63.2	65.0	56	42.5	74	65.9	47.4	78.8	61.4
CDAN	51.1	69.7	74.6	56.9	60.4	64.6	57.2	45.5	75.6	68.5	52.7	79.8	63.0
IWDANN	48.7	62.0	71.6	50.4	57.0	60.3	51.4	41.1	69.9	62.6	51.0	77.2	58.6
IWJAN	44.0	71.9	75.1	55.2	65.0	67.7	57.1	42.4	74.9	66.1	46.1	78.5	62.0
IWCDAN	52.3	72.2	76.3	56.9	67.3	67.7	57.2	44.8	77.8	67.3	53.3	80.6	64.6
PCT-Uniform	55.8	77.6	80.4	65.1	72.3	74.7	67.0	50.9	81.1	72.6	57.0	84.0	69.8
PCT-Learnable	**57.5**	78.2	80.5	66.7	74.3	75.4	64.6	50.7	81.3	72.9	57.3	83.5	70.2
PGA (Ours)	57.4	**84.8**	**86.4**	**76.0**	**84.6**	**85.6**	**74.5**	**57.1**	**86.1**	**75.9**	**57.4**	**85.3**	**75.9**

C.4 Training Resources

All experiments are run on Intel(R) Xeon(R) Platinum 8358 CPU @ 2.60GHz and NVIDIA A100-SXM4-80GB GPU.

D Additional related work

Another work sharing the same intuition of gradient alignment is ProGrad [125], which manipulates gradient of the fine-tuned loss to preserve general knowledge of the pretrained model. Similar to other gradient-based MTL methods [34, 35], it attempts to remove conflicts between per-objective gradients at each time step, thus is orthogonal to our approach. In contrast, we aim to stimulate their inherent consensus throughout training by encouraging the same training trajectory for both domains, hence, the model can find commonly good regions for them. Another concept that relates to gradient alignment is meta-learning. This has been introduced to Domain generalization in [126, 127]. Their intuition is a training procedure that enables the model to achieve low loss on a subset of training domains after having learned the other ones, and they work on the full model space. In a recent work about Vision-Language Models [128], meta-learning was used to deal with the problem of few-shot prompt learning by meta-learning prompt initialization. The gradient of the inner loop is modified with a learnable regulating function, and data for the support and query sets are found by hierarchically clustering an auxiliary large-scale image-text dataset. This method also has the impact of aligning gradient between support and query data as a result of meta-learning. However, its computation and space complexity is rather large as it requires the computation of Hessian matrix, web-scale of image-text pairs, and meta-learns the soft initialization for prompts.

E Limitations and Future works

First, our work relies on pretrained-CLIP, meaning that if UDA data is too different from pretrained knowledge, our method may fail to learn adequately. Therefore adapting our method to scratch-training scenarios without heavy computation and space complexity should be investigated. Second, the derived bound can be potentially loose as the number of training iterations increases. Thus studying other types of bounds could be an interesting work. Finally, as we mentioned, a strategy to explicitly align feature distribution across domains is worth looking into.

https://doi.org/10.52202/079017-1447

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes] , [No] , or [NA] .
- [NA] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for NA).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will be asked to also include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While "[Yes] " is generally preferable to "[No] ", it is perfectly acceptable to answer "[No] " provided a proper justification is given (e.g., "error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering "[No] " or "[NA] " is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS paper checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The main claims made in the abstract and introduction accurately reflect the paper's contributions and scope.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

Justification: We reported our computational complexity and also have a discussion section in Appendix.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: We provided full assumption and proof of our theory in appendix.

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We reported detailed descriptions and hyperparameters for all experiments.

   Guidelines:

https://doi.org/10.52202/079017-1447

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: Our reproducible codebase is published in `https://github.com/VietHoang1512/PGA`.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.

- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We provide detailed settings and hyperparameters for all experiments.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We conducted the illustrative experiment ten times independently with different random seeds and reported the mean and standard derivation of the result. The other experiment setups follow the protocol of prior well-known work.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
   - If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We mention about compute resources in Appendix.

   Guidelines:

https://doi.org/10.52202/079017-1447

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We follow the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: This paper has no negative social impact.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

Justification: All datasets used in our paper are public.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All datasets used in our paper are public.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: There is no new assets introduced in the paper.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

https://doi.org/10.52202/079017-1447

Answer: [NA]

Justification: There are no crowdsourcing experiments and research with human subjects in this paper.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: There is no potential risks incurred by study participants.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.