Pareto Domain Adaptation

Anonymous Author(s)
Affiliation
Address
email

Abstract

Domain adaptation (DA) attempts to transfer the knowledge from a labeled source domain to an unlabeled target domain that follows different distribution from the source. To achieve this, DA methods include a source classification objective $\mathcal{L}_S$ to extract the source knowledge and a domain alignment objective $\mathcal{L}_D$ to diminish the domain shift, ensuring knowledge transfer. Typically, former DA methods adopt some weight hyper-parameters to linearly combine the training objectives to form an overall objective $\mathcal{L}$. However, the gradient directions of these objectives may conflict with each other due to domain shift. Under such circumstances, the linear optimization scheme might decrease the overall objective value at the expense of damaging one of the training objectives, leading to restricted solutions. In this paper, we rethink the optimization scheme for DA from a gradient-based perspective. We propose a Pareto Domain Adaptation (ParetoDA) approach to control the overall optimization direction, aiming to optimize all training objectives. Furthermore, to reach a desirable solution for the model performance on target domain, we guide the optimization direction by the gradient of a surrogate loss for target classification. Specifically, we integrate domain discrimination results as auxiliary knowledge to refine target predictions based on Bayes' theorem and adopt them to construct the surrogate loss. Extensive experiments on image classification and semantic segmentation benchmarks demonstrate the effectiveness of ParetoDA.

1 Introduction

Domain adaptation (DA) is a well-established paradigm for learning a model on an unlabeled target domain with the assistance of a labeled related source domain, which has attracted a surge of interests in the machine learning community [33, 32, 12]. By bridging the domain gap on the premise of ensuring the performance of source classification task, DA can adapt the model learned from the source domain to the target in the presence of data bias, which solves the dilemma of label scarcity in many real-world applications [54, 46, 10, 44, 2, 56]. Extensive DA approaches have been proposed in recent years, achieving great performances in many areas such as image classification [10, 26, 55], semantic segmentation [48, 51, 15], and object detection [38, 42, 53].

Most DA methods are devoted to aligning the distributions across domains by explicitly minimizing some discrepancy metrics [25, 27, 55] or adopting adversarial learning [10, 26, 40]. However, since the domain alignment objective also depends on the target domain that deviates from the source, the gradient direction of it may conflict with that of classification objective on the source domain. Thus, when the optimization proceeds to a certain stage, one objective will deteriorate if further improving the other one. Under this circumstance, no solution can reach the optimal of each objective at the same time. We can only obtain a set of so-called Pareto optimal solutions instead, where we cannot further decrease all objectives simultaneously [9]. All Pareto optimal solutions in loss space compose Pareto front as shown in Fig. 1.

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

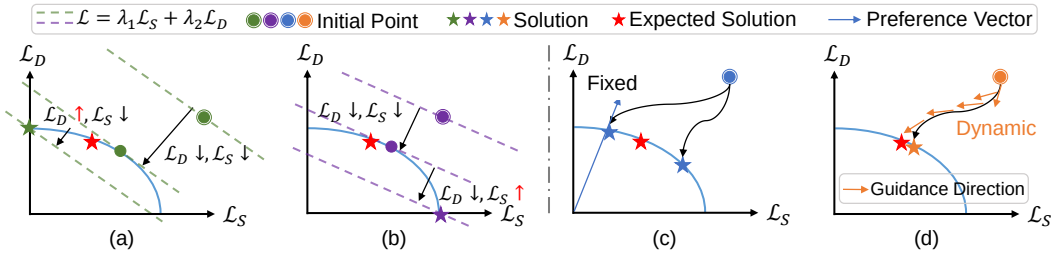


Figure 1: Illustration of different optimization schemes. Blue curve is Pareto front where the region underneath is unaccessible. (a)-(b): Linear scheme that adopts weight hyper-parameters to unify the objectives. The green and purple dash lines represent different hyper-parameters. (c): Previous gradient-based scheme, which reaches Pareto optimal solutions with or without the guidance of preference vector. (d): Our ParetoDA that leverages the gradient direction of the introduced surrogate loss on the target domain to induce the optimization direction, reaching the expected solution.

This dilemma obviously has been overlooked by most former DA methods, which generally adopt some empirical weight hyper-parameters to linearly combine the objectives, constructing an overall objective. This linear weighting scheme has two major concerns. First, as pointed out in [3], it can only obtain solutions on the convex part of the Pareto front while cannot reach the non-convex part, as shown in Fig. 1 (a)-(b). That is because it only considers reducing the overall objective value, which might damage some objectives during optimization (see the red up-arrow). Unfortunately, the Pareto fronts of most typical DA methods are in fact non-convex, as shown in Fig. 2. Second, one could hardly reach the desired solution by tuning the weight hyper-parameters, since the obtained solution will deviate severely when the trade-offs slightly change (see the slopes of the dash lines in Fig. 1 (a)-(b) for reference).

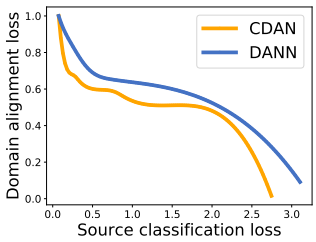


Figure 2: The non-convex Pareto fronts of DANN [10] and CDAN [26] on task W→A (Office-31).

To remedy these issues, we rethink the optimization scheme for DA from a gradient-based perspective. By controlling the overall optimization direction so that no objective deteriorates, one can successfully reach any Pareto optimal solutions on the entire Pareto front [1, 22, 29, 31]. Besides, [29] further controls the optimization direction to move along a fixed preference vector to obtain a specific Pareto optimal solution (see Fig. 1 (c)). However, neither of them suits DA, since which Pareto optimal solution is desirable for model performance on the target domain is uncertain.

In this paper, we propose a Pareto Domain Adaptation (ParetoDA) approach, which dynamically searches the desirable Pareto optimal solution for DA with the guidance of a surrogate loss that mimics the classification objective on the target domain. Specifically, we take domain discrimination information as auxiliary knowledge and embed it into the target prediction based on Bayes' theorem. Then adopt the information maximization loss [16] on the improved prediction to construct the surrogate loss. The key insight is that *if the optimization direction is appropriate for target domain performance, the surrogate loss on a small target validation set (unlabeled) should be minimized*. At each training iteration, we first calculate the gradients of each objective and model the optimization direction as a convex combination of them, and then control the direction by imposing two constraints. The basic constraint is that the optimization direction should not damage any training objectives. The other one is to enforce the optimization direction approximating the gradient direction of the surrogate loss on a target validation set. By searching the optimization direction under the two constraints, we can obtain a desirable Pareto optimal solution that boosts model performance. Our contributions are:

- We rethink the optimization scheme in existing DA methods, which can only reach restricted solutions if the optimization objectives conflict with each other.
- We propose a Pareto Domain Adaptation method that can obtain desired Pareto optimal solution guided by the designed surrogate objective function that mimics the target classification task. As a general technique, ParetoDA can be plugged into various DA methods and enhance their performance.

- We evaluate ParetoDA on two DA applications: image classification and semantic segmentation. Experimental results demonstrate the effectiveness of ParetoDA.

2 Background

Consider m objectives, each with a non-negative loss function $\mathcal{L}_i(\boldsymbol{\theta}) : \mathbb{R}^d \rightarrow \mathbb{R}_+, i \in [m] = \{1, 2, ..., m\}$, where $\boldsymbol{\theta}$ is the shared parameters of the objectives to be optimized. Under such circumstances, there exists no single solution that can reach the optimal solution of each objective at the same time due to they may conflict with each other. What we can obtain instead is a set of so-called Pareto optimal solutions, the related definitions are as follows [57]:

Pareto dominance. Suppose two solutions $\boldsymbol{\theta}_1, \boldsymbol{\theta}_2$ in $\mathbb{R}^d$, we define $\boldsymbol{\theta}_1 \prec \boldsymbol{\theta}_2$ if $\mathcal{L}_i(\boldsymbol{\theta}_1) \leq \mathcal{L}_i(\boldsymbol{\theta}_2), \forall i \in [m]$ and $\mathcal{L}_i(\boldsymbol{\theta}_1) < \mathcal{L}_i(\boldsymbol{\theta}_2), \exists i \in [m]$. We say that $\boldsymbol{\theta}_1$ dominates $\boldsymbol{\theta}_2$ iff $\mathcal{L}(\boldsymbol{\theta}_1) \prec \mathcal{L}(\boldsymbol{\theta}_2)$. Note that $\mathcal{L}(\boldsymbol{\theta}_1) \not\prec \mathcal{L}(\boldsymbol{\theta}_2)$ if $\boldsymbol{\theta}_2$ is not dominated by $\boldsymbol{\theta}_1$.

Pareto optimality. If a solution $\boldsymbol{\theta}_1$ dominates $\boldsymbol{\theta}_2$, then $\boldsymbol{\theta}_1$ is clearly preferable as it perform better or equally on each objective. A solution $\boldsymbol{\theta}^*$ is Pareto optimal if it is not dominated by any other solutions. That is, $\boldsymbol{\theta} \not\prec \boldsymbol{\theta}^*, \forall \boldsymbol{\theta} \in \mathbb{R}^d - \{\boldsymbol{\theta}^*\}$.

Pareto front. The set of all Pareto optimal solutions in loss space is Pareto front, where each point represents a unique solution.

Preference vector. A Pareto optimal solution can be viewed as an intersection of the Pareto front with a specific direction in loss space (a ray). We refer to this direction as the preference vector.

In gradient-based optimization, we can reach a Pareto optimal solution by starting from an arbitrary initialization and iteratively finding the next solution $\boldsymbol{\theta}^{t+1}$ that dominates the previous one θ^t, i.e., $\boldsymbol{\theta}^{t+1} = \boldsymbol{\theta}^t - \eta \boldsymbol{d}_{des}$, where η is the step size and $\boldsymbol{d}_{des}$ is the optimization direction. Let $\boldsymbol{g}_i = \nabla_{\boldsymbol{\theta}} \mathcal{L}_i$ be the gradient of the ith objective function at $\boldsymbol{\theta}$, the descent direction $\boldsymbol{d}_{des}$ is given by $\boldsymbol{d}_{des}^T \boldsymbol{g}_i \geq 0, \forall i \in [m]$. In other words, the descent direction $\boldsymbol{d}_{des}$ should optimize or not damage each objective. Thus, moving along $\boldsymbol{d}_{des}$, starting from $\boldsymbol{\theta}$, amounts to a decrease in the objective values. When $\boldsymbol{d}_{des}^T \boldsymbol{g}_i = 0$, the optimization process ends, implying that no direction can further optimize the all the objectives simultaneously. At this moment, if we still want to improve the performance of a specific objective, other objectives will deteriorate.

This paper focuses on dynamically searching for a desired Pareto optimal solution with the aid of the proposed surrogate loss in DA problems, aiming to enhance the performance on the target domain.

3 Related Work

Domain adaptation aims to transfer knowledge across domains. The key ideology of DA is to diminish the domain gap between two domains, enabling successful transfer. Existing DA methods mainly mitigate domain shift via minimizing some statistic metrics [25, 55, 27, 43] or adversarial training [10, 26, 40, 19, 21]. To name a few, DAN [25] mitigates the layer-wise discrepancy via Maximum Mean Discrepancy (MMD) metric. MMD [55] introduces Margin Disparity Discrepancy with rigorous generalization bounds to measure the discrepancy between domains. On the other hand, inspired by GANs [13], DANN [10] introduces a domain discriminator to enforce the features from two domains to be indistinguishable. CDAN [26] further extends this idea by integrate classification information into domain discriminator, leading to precise domain alignment. However, these methods all directly optimize the linear combination of all objectives, while paying few attention on the potential conflict between them. In such case, the linear scheme is hard to tune and may only access to restricted solutions on the convex part of pareto front [3]. We propose a gradient-based optimization scheme to search the desirable Pareto optimal solution on the entire pareto front without weight hyper-parameters, which is orthogonal to prior DA methods.

Weight-based optimization is an intuitive strategy to optimize multiple objectives at the same time, which unifies all the training objectives using weight hyper-parameters to form an overall objective. Since it is hard to manually assign proper weights in prior, many DA methods conduct Deep Embedded Validation [52] to select weight hyper-parameters. Recently, some methods propose to dynamically weight the objectives leveraging gradient magnitude [5], the rate of change in losses [24], task uncertainty [18], or learning non-linear loss combinations by implicit differentiation [30].

However, these weight-based methods focus on seeking a balanced solution, which is not bound to be optimal for target classification in DA. By contrast, we explore the gradient-based optimization, aiming to find an optimal solution for DA problems.

Gradient-based optimization leverages the gradients of objectives to reach the final Pareto optimal solution. MGDA [1] proposes to simply update the shared network parameters along a descent direction which leads to solutions that dominate the previous one. Nonetheless, MGDA may reach any Pareto optimal solution since it has no extra control for the optimization direction. To ameliorate this problem, PMTL [22] splits the loss space into separated sub-regions based on several preference vectors, and achieve Pareto optimal solutions in each sub-region of the Pareto front. Further, EPO [29] can reach a exact Pareto optimal solution by enforcing decent direction towards a specific preference vector. However, which Pareto optimal solution is the desirable one for target classification is ambiguous in DA methods. In this paper, we propose ParetoDA algorithm to dynamically searching the desirable Pareto optimal solution for target domain guided by a introduced surrogate target loss.

4 Pareto Domain Adaptation

4.1 Decompose DA

Domain adaptation (DA) aims to train a model that performs well on an unlabeled target domain with the assistance of a labeled source domain. Formally, in DA, we can access a source domain $D_s = \{\boldsymbol{x}_i^s, y_i^s\}_{i=1}^{N_s}$ with N_s labeled instances and a target domain $D_t = \{\boldsymbol{x}_i^t\}_{i=1}^{N_t}$ with N_t unlabeled samples, where $y_s^i \in \{1, 2, ..., C\}$ is the corresponding label of source data $\boldsymbol{x}_i^s$. Both source and target domains share one identical label space $\mathcal{Y}_s$ that contains C classes.

Existing DA methods are mainly devoted to establishing domain alignment by explicitly minimizing the domain discrepancy or adversarial training, which have obtained promising results. Discrepancy-based methods attempt to find a metric that can precisely depict the domain discrepancy, then adopt it to minimize the domain gap. While adversarial-based methods play a mini-max game between a feature extractor and a discriminator, where the feature extractor tries to fool the discriminator, leading to domain-invariant features. All in all, these approaches can be decomposed as follows:

$$Discrepancy - based : \min_{\boldsymbol{\theta}, \boldsymbol{\phi}_c} \mathcal{L}_S + \mathcal{L}_D \tag{1}$$

$$Adversarial - based : \min_{\boldsymbol{\theta}, \boldsymbol{\phi}_c} \mathcal{L}_S + \mathcal{L}_D, \ \max_{\boldsymbol{\phi}_d} \mathcal{L}_D, \tag{2}$$

where $\mathcal{L}_S$ is the source classification loss, $\mathcal{L}_D$ is the domain alignment loss, and $\boldsymbol{\theta}, \boldsymbol{\phi}_c, \boldsymbol{\phi}_d$ are the parameters of shared feature extractor, classifier and domain discriminator, respectively.

However, the optimization of domain alignment loss might conflict with the source classification loss due to the domain shift, under which circumstances the de facto optimization scheme can only access restricted Pareto optimal solutions. To search for a desirable Pareto optimal solution for the target domain, we propose a gradient-based optimization method for DA. Concretely, we first design a novel surrogate loss that mimics target classification loss, and then leverage the gradient of it to guide optimization direction, aiming to enhance the performance on the target domain.

4.2 Gradient-based Optimization for DA

4.2.1 Surrogate Loss for Target Domain

The goal of ParetoDA is to coordinately optimize the multiple objectives in DA, aiming to enhance the model performance. Concretely, the ideal optimization direction should optimize all training objectives and reach a solution that is conducive to target classification loss, which is inaccessible due to the absence of target labels. To tackle this issue, we introduce a conditional information maximization loss (CIM) that mimics target classification loss to guide the optimization direction. We first refine target predictions base on Bayes' theorem, then unify them to construct CIM loss.

Former methods generally adopt the outputs of classifier as target predictions, without considering that the domain labels might be beneficial for classification. To maximumly leverage the information at hand, we model the domain information into target prediction as auxiliary knowledge. Formally, the probability of the i-th target sample is $p(y_i^t = c \mid z = d, \boldsymbol{x}, \boldsymbol{\theta}, \boldsymbol{\phi}_c)$ instead of $p(y_i^t = c \mid \boldsymbol{x}_i^t, \boldsymbol{\theta}, \boldsymbol{\phi}_c)$,

where the class $c \in \{1, \ldots, C\}$ and the domain label $d \in \{0, 1\}$, C is the number of classes, 0 and 1 denote the source and target domain respectively. Based on Bayes' theorem, we have:

$$\begin{aligned} p(y = c \mid z = d, \boldsymbol{x}, \boldsymbol{\theta}, \boldsymbol{\phi}_c) &= \frac{p(y = c, z = d \mid \boldsymbol{x}, \boldsymbol{\theta}, \boldsymbol{\phi}_c)}{p(z = d \mid \boldsymbol{x}, \boldsymbol{\theta}, \boldsymbol{\phi}_c)} \\ &= \frac{p(y = c, z = d \mid \boldsymbol{x}, \boldsymbol{\theta}, \boldsymbol{\phi}_c)}{\sum_{c'} p(y = c', z = d \mid \boldsymbol{x}, \boldsymbol{\theta}, \boldsymbol{\phi}_c)} \\ &= \frac{p(z = d \mid y = c, \boldsymbol{x}, \boldsymbol{\theta}, \boldsymbol{\phi}_c) p(y = c \mid \boldsymbol{x}, \boldsymbol{\theta}, \boldsymbol{\phi}_c)}{\sum_{c'} p(z = d \mid y = c', \boldsymbol{x}, \boldsymbol{\theta}, \boldsymbol{\phi}_c) p(y = c' \mid \boldsymbol{x}, \boldsymbol{\theta}, \boldsymbol{\phi}_c)} \\ &= \frac{p(z = d \mid \boldsymbol{x}, \boldsymbol{v_c}) p(y = c \mid \boldsymbol{x}, \boldsymbol{\theta}, \boldsymbol{\phi}_c)}{\sum_{c'} p(z = d \mid \boldsymbol{x}, \boldsymbol{v_{c'}}) p(y = c' \mid \boldsymbol{x}, \boldsymbol{\theta}, \boldsymbol{\phi}_c)} = \rho_{c|d}. \end{aligned} \tag{3}$$

Here, we adopt a multi-layer perception (MLP) network to model $p(z = d \mid \boldsymbol{x}, \boldsymbol{v_c})$, called class-wise discriminator. We train all the class-wise discriminators with:

$$\min_{\boldsymbol{v_1}, \ldots, \boldsymbol{v_C}} \sum_{c=1}^{C} \sum_{d=0}^{1} s_{c|d} I(z = d) \log p(z = d \mid \boldsymbol{x}, \boldsymbol{v_c})| \tag{4}$$

where $s_{c|d} = I(y = c)$ for source domain, and $s_{c|d} = \rho_{c|d}$ for target domain. This objective is from an Expectation-Maximization derivation which is in the supplementary material. Note that the $s_{c|d}$ here only acts as a weighting scalar. Thus, for one specific class-wise discriminator, it can distinguish the target domain from the source well only when the input samples containing the corresponding class information, i.e., belongs to the corresponding category. Otherwise, the probability of predicting samples as the target will be low. Hence, the class-wise domain predictions $p(z = d \mid \boldsymbol{x}, \boldsymbol{v_c})$ can act as a auxiliary to augment or rectify the target predictions by multiplying it with $p(y = c \mid \boldsymbol{x}, \boldsymbol{\theta}, \boldsymbol{\phi}_c)$.

Now we acquire refined predictions ρ for target samples which can substitute for the original predictions. As discussed in [11, 41], the ideal target predictions should be individually certain and globally diverse, which can be obtained by maximizing the mutual information between input samples and the predictions. Thus, we adopt the information maximization (IM) loss [16] on our improved version prediction ρ, the formulation is as follows:

$$\mathcal{L}_T = \sum_{c=1}^{C} \hat{\rho_{c|1}} log \hat{\rho_{c|1}} - \mathbb{E}_{\boldsymbol{x}_i \in X_t} \sum_{c=1}^{C} \rho_{c|1}^{[i]} log \rho_{c|1}^{[i]} \tag{5}$$

where $\rho_{c|1}^{[i]}$ denotes $p(y_i^t = c \mid z = 1, \boldsymbol{x}_i, \boldsymbol{\theta})$, i.e., the probability that the target sample $\boldsymbol{x}_i$ is classified into class c under the prior condition that it is divided into target domain, and $\hat{\rho_{c|1}}$ denotes $\mathbb{E}_{\boldsymbol{x}_i \in X_t} \rho_{c|1}^{[i]}$, represents the mean of c_{th} element in the output of the whole target domain. By minimizing the above objective loss, we can acquire outputs with the same properties as the output obtained by optimizing the real target classification loss.

Hence, we take Eq. 5 as a surrogate for the target classification objective. This loss can be directly added to original DA methods to assist the model adaption to the target domain. More importantly, we leverage it to guide the search for the desirable Pareto optimal solution.

4.2.2 Achieving Desired Pareto Optimal Solution

This section introduces how we establish gradient-based optimization for DA to handle the inconsistency among objectives and how we dynamically guide the optimization direction towards the desired Pareto optimal solution.

Given losses $\mathcal{L}_S$, $\mathcal{L}_D$ and $\mathcal{L}_T$, in each optimization step, we first model the update direction $\boldsymbol{d}$ as a convex combination of gradients of the three losses, i.e., $\boldsymbol{d} = \boldsymbol{G}\boldsymbol{w}$, where $\boldsymbol{w} \in \mathcal{S}^m = \{\boldsymbol{w} \in \mathbb{R}_+^m | \sum_{j=1}^m w_j = 1\}$, $m = 3$, and $\boldsymbol{G} = [\nabla_{\boldsymbol{\theta}} \mathcal{L}_S, \nabla_{\boldsymbol{\theta}} \mathcal{L}_D, \nabla_{\boldsymbol{\theta}} \mathcal{L}_T]$, where $\boldsymbol{\theta}$ is the shared parameters, i.e., the parameters of the shared feature extractor. A fundamental goal of gradient-based optimization is to find a $\boldsymbol{d}$ to simultaneously minimize all the losses. That is, the network can be optimized along a consistent direction and learn the shared knowledge between source and target domains efficiently and stably. Therefore, the direction $\boldsymbol{d}$ first need to satisfy the constraint $\boldsymbol{d}^T \boldsymbol{g}_j \geq 0, \forall j \in [m]$.

Recently, a state-of-art method EPO [29] further controls the direction to reach a Pareto optimal solution with designated trade-off among objectives, using a preference vector fixed apriori. Concretely, EPO guides the optimization direction via minimizing the KL divergence between a normalized loss vector weighted by the preference vector and a uniform probability vector. Since the trade-off among objectives in DA is ambiguous and may vary with methods and training procedures, our unique approach attempt to dynamically guide the optimization direction based on the surrogate loss proposed in Chapter 4.2.1 to search a desired Pareto optimal solution.

Specifically, we argue that if the update direction is appropriate for target domain performance, the surrogate loss on a small target validation set (split from the original target set) should be minimized. Formally, we compute the validation loss on the separated validation set X_t^{Val} as Eq. 5, denoted as $\mathcal{L}_{Val}$, then we can directly obtain the gradient descent direction: $\boldsymbol{g}_v = \nabla_{\boldsymbol{\theta}} \mathcal{L}_{Val}$. Thus we can replace the guidance direction $\boldsymbol{d}_{bal}$ [1] in EPO by $\boldsymbol{g}_v$ as a dynamical guidance of the optimization direction. For clarity, we reformulate the optimization problem as a linear programming (LP) problem:

$$\begin{aligned} \boldsymbol{w}^* = \arg\max_{\boldsymbol{w} \in \mathcal{S}^m} \; & (\boldsymbol{G}\boldsymbol{w})^T (I(\mathcal{L}_{Val} > 0)\boldsymbol{g}_v + I(\mathcal{L}_{Val} = 0) G\mathbf{1}/m), \\ \text{s.t. } & (\boldsymbol{G}\boldsymbol{w})^T \boldsymbol{g}_j \geq I(J \neq \emptyset)(\boldsymbol{g}_v^T \boldsymbol{g}_j), \; \forall j \in \bar{J} - J^*, \\ & (\boldsymbol{G}\boldsymbol{w})^T \boldsymbol{g}_j \geq 0, \; \forall j \in J^*, \end{aligned} \tag{6}$$

where $I(\cdot)$ is an indicator function, $\boldsymbol{g}_j (j \in \{1,2,3\})$ is the jth column of G, $J = \{j | \boldsymbol{g}_v^T \boldsymbol{g}_j > 0\}$, $\bar{J} = \{j | \boldsymbol{g}_v^T \boldsymbol{g}_j \leq 0\}$, and $J^* = \{j | \boldsymbol{g}_v^T \boldsymbol{g}_j = \max_{j'} \boldsymbol{g}_v^T \boldsymbol{g}_{j'}\}$. We first analyze the solutions of Eq. (6) by the following result and then explain its mechanism.

Theorem 1. *Let $\boldsymbol{w}^*$ be the solution of the problem in Eq.* (6)*, and $\boldsymbol{d}^* = \boldsymbol{G}\boldsymbol{w}^*$ be the resulted update direction. If $\mathcal{L}_{Val} = 0$, then the dominating direction $\boldsymbol{d}^*$ becomes a descent direction, i.e.,*

$$(\boldsymbol{d}^*)^T \boldsymbol{g}_j \geq 0, \; \forall j \in \{1,2,3\}. \tag{7}$$

On the other hand, if $\mathcal{L}_{Val} > 0$, Let $\boldsymbol{\gamma}^* = (\boldsymbol{d}^*)^T \boldsymbol{g}_v$ be the objective value in Eq. (6)*. Then,*

$$\begin{aligned} & if \; \boldsymbol{\gamma}^* > 0, \; then \; (\boldsymbol{d}^*)^T \boldsymbol{g}_v > 0; \\ & if \; \boldsymbol{\gamma}^* \leq 0, \; then \; (\boldsymbol{d}^*)^T \boldsymbol{g}_j \geq 0, \; \forall j \in \{1,2,3\}. \end{aligned} \tag{8}$$

The proof of Theorem 1 is deferred to the supplementary material. Now we interpret the learning mechanism in Eq. (6). When $\mathcal{L}_{Val} = 0$, the learning is in the pure descent mode, in which the update direction $\boldsymbol{d}^* = \boldsymbol{G}\boldsymbol{w}^*$ approximates the mean gradient $\boldsymbol{G}\mathbf{1}/m$, and $\boldsymbol{d}^*$ can decrease all the training losses simultaneously. Whereas when $\mathcal{L}_{Val} > 0$, the learning is in the guidance descent mode, in which $\boldsymbol{d}^*$ approximates $\boldsymbol{g}_v$. Since $\boldsymbol{d}^* = \boldsymbol{G}\boldsymbol{w}^*$ is a convex combination of columns in G, then $\boldsymbol{\gamma}^* > 0$ means that $\boldsymbol{g}_v$ is consistent with some columns in G, and $\boldsymbol{d}^*$ is forced to decrease the loss whose gradient is the most consistent with $\boldsymbol{g}_v$—which means the highest contribution to the model performance. On the other hand, $\boldsymbol{\gamma}^* \leq 0$ means none of the columns in G is consistent with $\boldsymbol{g}_v$. In this case, $\boldsymbol{d}^*$ is still forced to decrease all the training losses simultaneously, while approximating $\boldsymbol{g}_v$ to the best. All these solutions are consistent with the spirit of EPO. In summary, $\boldsymbol{g}_v$ can dynamically guide the optimization direction towards the desired Pareto solution when $\mathcal{L}_{Val}$ is not optimized, and we no longer need prior knowledge to set a preference vector. The updating algorithm of every training iteration is presented in Alg. 1. Note that this optimize mode is only applied to the shared parameters while the others are still optimized in the original way.

Algorithm 1 Update Equations for ParetoDA

1: **Input:** Shared parameters $\boldsymbol{\theta}^t \in \mathbb{R}^d$, step size η
2: Calculate objective values: $\mathcal{L}_S, \mathcal{L}_D, \mathcal{L}_T$
3: Calculate guidance loss on target validation set: $\mathcal{L}_{val}$ (using Eq. (5))
4: Calculate the gradients of training objectives: G=$[g_1, g_2, g_3]$ =$[\nabla_{\boldsymbol{\theta}^t} \mathcal{L}_S, \nabla_{\boldsymbol{\theta}^t} \mathcal{L}_D, \nabla_{\boldsymbol{\theta}^t} \mathcal{L}_T]$
5: Calculate the gradient of the guidance loss: $\nabla_{\boldsymbol{\theta}^t} \mathcal{L}_{val}$
6: Obtain $\boldsymbol{w}^*$ by solving LP (Eq. (6))
7: Optimization direction: $\boldsymbol{d}^* = \boldsymbol{G}\boldsymbol{w}^*$
8: **Output:** Updated parameters $\boldsymbol{\theta}^{t+1} = \boldsymbol{\theta}^t - \eta \boldsymbol{d}^*$

Time Complexity. Let $m \in \mathbb{Z}_+$ be the number of objective, and $d \in \mathbb{Z}_+$ the dimension of model parameters. The computation of $(\boldsymbol{G}\boldsymbol{w})^T \boldsymbol{g}_v$ of $(\boldsymbol{G}\boldsymbol{w})^T \boldsymbol{G}$ both has run time $O(m^2 d)$. With the best LP solver [6], our LP (Eq. 6), that has m variables and at most $2m+1$ constraints, has a runtime [2] of $O^*(m^{2.38})$. As in DA networks, we usually have $n \gg m$, our ParetoDA has a total run time of $O(m^2 d + m^{2.38}) = O(d)$.

[1] See [29] for details.

[2] O^* is used to hide $m^{o(1)}$ and $log^{O(1)}(1/\delta)$ factors, δ being the relative accuracy. See [6] for details.

5 Experiment

5.1 Dataset and Setup

We evaluate ParetoDA on three object classification datasets and two semantic segmentation datasets. **Office-31 [39]** is a typical benchmark for cross-domain object classification involving three distinct domain: Amazon (A), DSLR (D) and Webcam (W); **Office-Home [47]** is a more difficult benchmark composed by four image domains: Art (Ar), Clip Art (Cl), Product (Pr) and Real-World (Rw). The images of different domains are more visually dissimilar with each other; **VisDA-2017 [36]** is a large-scale synthetic-to-real dataset, containing over 280K images across 12 categories; **Cityscapes [7]** is a real-world semantic segmentation dataset with 5,000 urban scenes which are divided into training, validation and test splits; **GTA5 [37]** includes 24,966 game screenshots from the game engine of GTA5. Following [45, 19], we establish a transfer semantic segmentation scenario: GTA5 $\rightarrow$ Cityscapes. Specifically, we use the shared 19 categories and report the test results on the validation set of Cityscapes. To evaluate our approach, we apply the proposed ParetoDA on former DA methods, i.e., DANN, CDAN, and MDD, and split 10% data randomly from the original target set as the validation set. All experiments are implemented via PyTorch [35], and we adopt the common-used stochastic gradient descent (SGD) optimizer with momentum 0.9 and weight decay 1e-4 for all experiments. Following the standard protocols in [26, 10, 28], we take the average classification accuracy and mean IoU (mIoU) for image classification and semantic segmentation as evaluation metrics, respectively. More details are presented in the supplementary material due to limited space.

5.2 Experimental Results

Office-31. From the results in Table 1, we can observe that DA methods, such as DANN and CDAN exceed the ResNet model. This validates that mitigating the domain gap between two domains is crucial for enhancing the generalization performance of models. We evaluate ParetoDA on three basic DA methods. The results show that ParetoDA can consistently improve the performance of the basic methods significantly. In particular, ParetoDA improves DANN by 8.0%, revealing that ParetoDA is an extra booster to former DA methods. On the other hand, the encouraging results validate the importance of searching for a desirable Pareto optimal solution.

Table 1: Classification accuracy (%) on Office-31 (ResNet-50).

Method	A:W	D:W	W:D	A:D	D:W	W:A	Avg
ResNet [14]	68.4	96.7	99.3	68.9	62.5	60.7	76.1
DAN [25]	80.5	97.1	99.6	78.6	63.6	62.8	80.4
JADA [21]	90.5	97.5	**100.0**	88.2	70.9	70.6	86.1
BNM [8]	91.5	98.5	**100.0**	90.3	70.9	71.6	87.1
TAT [23]	92.5	99.3	**100.0**	93.2	73.1	72.1	88.4
GSP [50]	92.9	98.7	99.8	94.5	75.9	74.9	89.5
DANN [10]	82.0	96.9	99.1	79.7	68.2	67.4	82.2
+BSP [4]	93.0	98.0	**100.0**	90.0	71.9	73.0	87.7
+MCC [17]	95.6	98.6	99.3	93.8	74.0	75.0	89.4
+ParetoDA	**95.5**	98.7	**100.0**	93.8	76.7	76.3	90.2
CDAN [26]	94.1	98.6	**100.0**	92.9	71.0	69.3	87.7
+BSP [4]	93.3	98.2	**100.0**	93.0	73.6	72.6	88.5
+MCC [17]	94.7	98.6	**100.0**	95.0	73.0	73.6	89.2
+ParetoDA	95.0	**98.9**	**100.0**	**95.4**	**77.6**	75.7	**90.4**
MDD [55]	93.5	98.4	**100.0**	94.5	74.6	72.2	88.9
+ParetoDA	95.4	**98.9**	**100.0**	94.4	76.2	**75.8**	90.1

Office-Home. Compared with Office-31, Office-Home is a more difficult dataset, whose results are shown in Table 2. In this difficult scenario, ParetoDA can still enhance the performance of former DA methods, with 11.8% and 4.8% improvement on DANN and CDAN respectively. Based on these promising results, we can infer that ParetoDA can reach a desirable Pareto optimal solution which is beneficial for the performance of models.

VisDA-2017. To validate the model performance on a large-scale dataset, we carry out experiments on VisDA-2017. The results are presented in Table 3. As a plug-in module, BSP improves the basic models by enhancing the discriminability of feature representations. ParetoDA aims to improve the optimization for DA methods, which is also orthogonal to former DA methods. In addition, ParetoDA achieves greater performance gains, further indicating the effectiveness and versatility of our method.

GTA5 $\rightarrow$ Cityscapes. Compared with classification tasks, segmentation demands more cost for the pixel-level annotation. Hence, it's of vital importance to conduct domain adaptation for semantic segmentation. We evaluate our method on the adaptive semantic segmentation task, GTA5 $\rightarrow$ Cityscapes, aiming to transfer the knowledge from synthesized frames to real scenes. The segmentation results are exhibited in Table 4. We apply ParetoDA on two methods, i.e., AdvEnt and FDA, which boosts them

Table 2: Accuracy(%) on **Office-Home** for unsupervised DA (ResNet-50).

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 [14]	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
DAN [25]	43.6	57.0	67.9	45.8	56.5	60.4	44.0	43.6	67.7	63.1	51.5	74.3	56.3
TAT [23]	51.6	69.5	75.4	59.4	69.5	68.6	59.5	50.5	76.8	70.9	56.6	81.6	65.8
TPN [34]	51.2	71.2	76.0	65.1	72.9	72.8	55.4	48.9	76.5	70.9	53.4	80.4	66.2
BNM [8]	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
MDD [55]	54.9	73.7	77.8	60.0	71.4	71.8	61.2	53.6	78.1	72.5	60.2	82.3	68.1
GSP [50]	**56.8**	75.5	78.9	61.3	69.4	**74.9**	61.3	52.6	79.9	73.3	54.2	83.2	68.4
DANN [10]	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
+BSP [4]	51.4	68.3	75.9	56.0	67.8	68.8	57.0	49.6	75.8	70.4	57.1	80.6	64.9
+MetaAlign [49]	48.6	69.5	76.0	58.1	65.7	68.3	54.9	44.4	75.3	68.5	50.8	80.1	63.3
+ParetoDA	55.2	74.4	79.0	61.9	72.4	72.9	62.1	**55.8**	81.1	74.4	**61.1**	82.4	69.4
CDAN [26]	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 [4]	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
+MetaAlign [49]	55.2	70.5	77.6	61.5	70.0	70.0	58.7	55.7	78.5	73.3	61.0	81.7	67.8
+ParetoDA	**56.8**	**75.9**	**80.5**	**64.4**	**73.5**	73.7	**65.6**	55.2	**81.3**	**75.2**	**61.1**	**83.9**	**70.6**

Table 3: Accuracy(%) on **VisDA-2017** for unsupervised DA (ResNet-101).

Method	plane	bcycl.	bus	car	horse	knife	mcycl.	person	plant	sktbrd	train	truck	Avg.
ResNet [14]	55.1	53.3	61.9	59.1	80.6	17.9	79.7	31.2	81.0	26.5	73.5	8.5	52.4
DAN [25]	87.1	63.0	76.5	42.0	90.3	42.9	85.9	53.1	49.7	36.3	85.8	20.7	61.1
MCD [40]	87.0	60.9	83.7	64.0	88.9	79.6	84.7	76.9	88.6	40.3	83.0	25.8	71.9
JADA [21]	91.9	78.0	81.5	68.7	90.2	84.1	84.0	73.6	88.2	67.2	79.0	38.0	77.0
TPN [34]	93.7	**85.1**	69.2	81.6	93.5	61.9	89.3	81.4	93.5	81.6	84.5	49.9	80.4
DTA [20]	93.7	82.2	**85.6**	**83.8**	93.0	81.0	**90.7**	82.1	**95.1**	78.1	86.4	32.1	81.5
DANN [10]	81.9	77.7	82.8	44.3	81.2	29.5	65.1	28.6	51.9	54.6	82.8	7.8	57.4
+BSP [4]	92.2	72.5	83.8	47.5	87.0	54.0	86.8	72.4	80.6	66.9	84.5	37.1	72.1
+ParetoDA	95.3	82.4	82.9	56.2	92.6	**95.0**	87.1	81.1	90.2	90.8	**87.8**	46.7	82.4
CDAN [26]	85.2	66.9	83.0	50.8	84.2	74.9	88.1	74.5	83.4	76.0	81.9	38.0	73.9
+BSP [4]	92.4	61.0	81.0	57.5	89.0	80.6	90.1	77.0	84.2	77.9	82.1	38.4	75.9
+ParetoDA	**95.9**	82.8	81.3	58.7	**93.9**	93.7	85.9	**83.0**	91.9	**92.0**	87.1	**51.8**	**83.2**

Table 4: Semantic segmentation performance mIoU (%) on **Cityscapes** validation set.

Method	road	side.	buil.	wall	fence	pole	light	sign	veg	terr.	sky	pers.	rider	car	truck	bus	train	mbike	bike	mIoU
ResNet-101	75.8	16.8	77.2	12.5	21.0	25.5	30.1	20.1	81.3	24.6	70.3	53.8	26.4	49.9	17.2	25.9	6.5	25.3	36.0	36.6
AdaSegNet [45]	86.5	36.0	79.9	23.4	23.3	23.9	35.2	14.8	83.4	33.3	75.6	58.5	27.6	73.7	32.5	35.4	3.9	30.1	28.1	42.4
Cycada [15]	86.7	35.6	80.1	19.8	17.5	38.0	**39.9**	**41.5**	82.7	27.9	73.6	**64.9**	19	65.0	12.0	28.6	4.5	31.1	42.0	42.7
CLAN [28]	87.0	27.1	79.6	27.3	23.3	28.3	35.5	24.2	83.6	27.4	74.2	58.6	28.0	76.2	33.1	36.7	6.7	31.9	31.4	43.2
AdvEnt [48]	89.9	36.5	81.6	29.2	25.2	28.5	32.3	22.4	83.9	**34.0**	77.1	57.4	27.9	83.7	29.4	39.1	1.5	28.4	23.3	43.8
+ParetoDA	79.6	29.1	83.4	30.4	22.4	32.6	39.2	26.7	82.5	26.5	**82.2**	63.0	29.8	**85.9**	**36.1**	43.5	9.2	29.9	43.7	46.1
FDA [51]	**90.0**	**40.5**	79.4	25.3	**26.7**	30.6	31.9	29.3	79.4	28.8	76.5	56.4	27.5	81.7	27.7	**45.1**	**17.0**	23.8	29.6	44.6
+ParetoDA	85.4	37.6	**84.0**	**32.0**	23.8	**33.1**	39.6	25.9	**84.0**	30.6	81.6	61.3	**30.9**	85.4	31.9	45.0	7.6	**32.4**	**45.4**	**47.2**

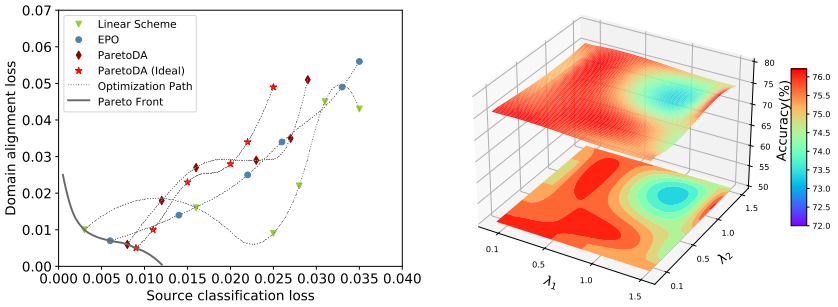
by large margins on average. This manifests that ParetoDA can still reach an appropriate optimization solution for enhancing model performance on this pixel-level transfer scenario.

5.3 Analytical Experiments

Ablation Study. To verify each component of ParetoDA, we establish ablation study on Office-31. The average results are shown in Table 5. CIM performs better than IM, demonstrating that integrating domain labels is beneficial for target predictions. Applying the conventional Pareto optimization method [29] with fixed preference vector on DANN achieves considerable performance gain. This reveals that the gradient-based optimization method is superior to the linear weighting scheme. Further, with our proposed dynamic ParetoDA, DANN obtains the best results, validating the importance of guiding the optimization direction with our introduced loss CIM.

Table 5: Ablation Study of ParetoDA on Office-31 (ResNet-50).

	IM	CIM	EPO	ParetoDA	Accuracy
DANN					82.2
DANN	✓				87.9
DANN		✓			88.4
DANN			✓		87.7
DANN		✓	✓		89.9
DANN		✓		✓	**90.2**



(a) Visualization of the optimization path (b) Objective scale sensitivity analysis

Figure 3: Analytical experiments conducted on W→A (Office-31) based on DANN. (a): Visualization of optimization paths of different optimization schemes. (b): Objective scale sensitivity analysis of ParetoDA by varying the scale factors of objectives.

Optimization Path Visualization. To intuitively verify that ParetoDA can dynamically search the desired Pareto optimal solution under the guidance of our proposed loss, we visualize the optimization paths of different optimization schemes (see Fig. 3(a)). Note that all these schemes start from the same initial point, but we only plot several solutions of the final stage of optimization for better illustration. It is clear that the linear weighting scheme not only deteriorates one of the objectives at some steps, but also reaches a solution on the convex part of the Pareto front, which deviates severely from the ideal one. The conventional gradient-based optimization method EPO (with fixed preference vector $r = [1, 1, 1]$) controls the optimization direction to avoid damaging any of the objectives. However, its solution heavily relies on the pre-defined preference vector for training objectives, which is ambiguous in DA. By contrast, our proposed ParetoDA further dynamically adjusts the optimization direction with the gradient direction of the proposed CIM loss. For comparisons, we illustrate the **ideal** solution guided by the supervised classification loss on target domain. One can observe that the optimization direction of ParetoDA is consistent with the ideal one in general, and reach a final solution closest to the ideal one.

Sensitivity to Different Objective Scales. The scale of the objectives may vary with methods or datasets. To test the adaptivity of ParetoDA to different objective scales, we conduct experiments on a random task W→A (Office-31) based on DANN by varying $\lambda_0, \lambda_1 \in \{0.1, 0.5, 1.0, 1.5\}$, which represent the scale factor of object classification objective and domain alignment objective respectively. Fig. 3(b) shows that ParetoDA is not sensitive to the scale variety of objectives, which indicating that ParetoDA is robust across different methods and datasets.

Adaptivity to Deeper Backbone Networks. In this work, we reach a desirable Pareto optimal solution for the target domain by controlling the optimization direction. Whereas on deeper networks, it might be more difficult to control the optimization direction due to the increased complexity. For a fair comparison, we use ResNet-50 on classification tasks. However, ParetoDA can be easily adapted to deeper backbone networks. Table 6 presents the results of ParetoDA based on DANN with different backbones, demonstrating that ParetoDA can still consistently enhance the performance of the basic method.

Table 6: Test top-1 accuracy (%) on task W→A (Office-31) of methods with different backbone networks.

Networks	Source-only	DANN [10]	+ParetoDA
ResNet-50	60.7	67.4	76.3
ResNet-101	65.8	74.5	77.5
ResNet-152	68.2	76.1	79.0

6 Conclusion

In this paper, we rethink the optimization scheme for DA from a gradient-based optimization perspective. We present a Pareto Domain Adaptation (ParetoDA) to dynamically search a desirable Pareto optimal solution for boosting model performance, which is orthogonal to former DA methods. Extensive experiments on both classification and segmentation tasks validate the effectiveness of ParetoDA. It is worth noting that we mainly concentrate on the traditional domain adaptation setting for clarity, while the generalization to other variants of DA needs to be further explored.

References

[1] Multiple-gradient descent algorithm (mgda) for multiobjective optimization. *Comptes Rendus Mathematique*, 350(5):313–318, 2012.

[2] K. Bousmalis, G. Trigeorgis, N. Silberman, D. Krishnan, and D. Erhan. Domain separation networks. In *NeurIPS*, pages 343–351, 2016.

[3] S. P. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2014.

[4] X. Chen, S. Wang, M. Long, and J. Wang. Transferability vs. discriminability: Batch spectral penalization for adversarial domain adaptation. In *ICML*, pages 1081–1090, 2019.

[5] Z. Chen, V. Badrinarayanan, C. Lee, and A. Rabinovich. Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks. In J. G. Dy and A. Krause, editors, *ICML*, pages 793–802, 2018.

[6] M. B. Cohen, Y. T. Lee, and Z. Song. Solving linear programs in the current matrix multiplication time. In *SOTC*, pages 938–942, 2019.

[7] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. The cityscapes dataset for semantic urban scene understanding. In *CVPR*, pages 3213–3223, 2016.

[8] S. Cui, S. Wang, J. Zhuo, L. Li, Q. Huang, and T. Qi. Towards discriminability and diversity: Batch nuclear-norm maximization under label insufficient situations. *CoRR, abs/2003.12237*, 2020.

[9] M. Ehrgott. *Multicriteria Optimization (2. ed.)*. Springer, 2005.

[10] Y. Ganin and V. Lempitsky. Unsupervised domain adaptation by backpropagation. In *ICML*, pages 1180–1189, 2015.

[11] R. Gomes, A. Krause, and P. Perona. Discriminative clustering by regularized information maximization. In *NeurIPS*, pages 775–783, 2010.

[12] B. Gong, Y. Shi, F. Sha, and K. Grauman. Geodesic flow kernel for unsupervised domain adaptation. In *CVPR*, pages 2066–2073, 2012.

[13] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *NeurIPS*, pages 2672–2680, 2014.

[14] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016.

[15] J. Hoffman, E. Tzeng, T. Park, J.-Y. Zhu, P. Isola, K. Saenko, A. Efros, and T. Darrell. Cycada: Cycle-consistent adversarial domain adaptation. In *ICML*, pages 1994–2003, 2018.

[16] W. Hu, T. Miyato, S. Tokui, E. Matsumoto, and M. Sugiyama. Learning discrete representations via information maximizing self-augmented training. In *ICML*, volume 70, pages 1558–1567. PMLR, 2017.

[17] Y. Jin, X. Wang, M. Long, and J. Wang. Minimum class confusion for versatile domain adaptation. In *ECCV*, volume 12366, pages 464–480, 2020.

[18] A. Kendall, Y. Gal, and R. Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *CVPR*, pages 7482–7491, 2018.

[19] C.-Y. Lee, T. Batra, M. H. Baig, and D. Ulbricht. Sliced wasserstein discrepancy for unsupervised domain adaptation. In *CVPR*, pages 10285–10295, 2019.

[20] S. Lee, D. Kim, N. Kim, and S. Jeong. Drop to adapt: Learning discriminative features for unsupervised domain adaptation. In *ICCV*, pages 91–100, 2019.

[21] S. Li, C. H. Liu, B. Xie, L. Su, Z. Ding, and G. Huang. Joint adversarial domain adaptation. In *ACM MM*, pages 729–737, 2019.

[22] X. Lin, H. Zhen, Z. Li, Q. Zhang, and S. Kwong. Pareto multi-task learning. In *NeurIPS*, pages 12037–12047, 2019.

[23] H. Liu, M. Long, J. Wang, and M. I. Jordan. Transferable adversarial training: A general approach to adapting deep classifiers. In *ICML*, pages 4013–4022, 2019.

[24] S. Liu, E. Johns, and A. J. Davison. End-to-end multi-task learning with attention. In *CVPR*, pages 1871–1880, 2019.

[25] M. Long, Y. Cao, J. Wang, and M. I. Jordan. Learning transferable features with deep adaptation networks. In *ICML*, pages 97–105, 2015.

[26] M. Long, Z. Cao, J. Wang, and M. I. Jordan. Conditional adversarial domain adaptation. In *NeurIPS*, pages 1647–1657, 2018.

[27] M. Long, H. Zhu, J. Wang, and M. I. Jordan. Deep transfer learning with joint adaptation networks. In *ICML*, pages 2208–2217, 2017.

[28] Y. Luo, L. Zheng, T. Guan, J. Yu, and Y. Yang. Taking a closer look at domain shift: Category-level adversaries for semantics consistent domain adaptation. In *CVPR*, pages 2502–2511, 2019.

[29] D. Mahapatra and V. Rajan. Multi-task learning with user preferences: Gradient descent with controlled ascent in pareto optimization. In *ICML*, pages 6597–6607, 2020.

[30] A. Navon, I. Achituve, H. Maron, G. Chechik, and E. Fetaya. Auxiliary learning by implicit differentiation. *CoRR*, abs/2007.02693, 2020.

[31] A. Navon, A. Shamsian, G. Chechik, and E. Fetaya. Learning the pareto front with hypernetworks. In *ICLR*, 2021.

[32] S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang. Domain adaptation via transfer component analysis. *TNNLS*, 22(2):199–210, 2011.

[33] S. J. Pan and Q. Yang. A survey on transfer learning. *TKDE*, 22(10):1345–1359, 2010.

[34] Y. Pan, T. Yao, Y. Li, Y. Wang, C.-W. Ngo, and T. Mei. Transferrable prototypical networks for unsupervised domain adaptation. In *CVPR*, June 2019.

[35] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *NeurIPS*, pages 8024–8035, 2019.

[36] X. Peng, B. Usman, N. Kaushik, J. Hoffman, D. Wang, and K. Saenko. Visda: The visual domain adaptation challenge. *CoRR, abs/1710.06924*, 2017.

[37] S. R. Richter, V. Vineet, S. Roth, and V. Koltun. Playing for data: Ground truth from computer games. In *ECCV*, pages 102–118, 2016.

[38] A. L. Rodriguez and K. Mikolajczyk. Domain adaptation for object detection via style consistency. In *BMVC*, page 232. BMVA Press, 2019.

[39] K. Saenko, B. Kulis, M. Fritz, and T. Darrell. Adapting visual category models to new domains. In *ECCV*, pages 213–226, 2010.

[40] K. Saito, K. Watanabe, Y. Ushiku, and T. Harada. Maximum classifier discrepancy for unsupervised domain adaptation. In *CVPR*, pages 3723–3732, 2018.

[41] Y. Shi and F. Sha. Information-theoretical learning of discriminative clusters for unsupervised domain adaptation. In *ICML*, 2012.

[42] S. Song, H. Yu, Z. Miao, J. Fang, K. Zheng, C. Ma, and S. Wang. Multi-spectral salient object detection by adversarial domain adaptation. In *AAAI*, pages 12023–12030, 2020.

[43] B. Sun, J. Feng, and K. Saenko. Return of frustratingly easy domain adaptation. In *AAAI*, pages 2058–2065, 2016.

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

[45] Y. Tsai, W. Hung, S. Schulter, K. Sohn, M. Yang, and M. Chandraker. Learning to adapt structured output space for semantic segmentation. In *CVPR*, pages 7472–7481, 2018.

[46] E. Tzeng, J. Hoffman, N. Zhang, K. Saenko, and T. Darrell. Deep domain confusion: Maximizing for domain invariance. *CoRR, abs/1412.3474*, 2014.

[47] H. Venkateswara, J. Eusebio, S. Chakraborty, and S. Panchanathan. Deep hashing network for unsupervised domain adaptation. In *CVPR*, pages 5018–5027, 2017.

[48] T. Vu, H. Jain, M. Bucher, M. Cord, and P. Perez. Advent: Adversarial entropy minimization for domain adaptation in semantic segmentation. In *CVPR*, pages 2517–2526, 2019.

[49] G. Wei, C. Lan, W. Zeng, and Z. Chen. Metaalign: Coordinating domain alignment and classification for unsupervised domain adaptation. *CoRR*, abs/2103.13575, 2021.

[50] H. Xia and Z. Ding. Structure preserving generative cross-domain learning. In *CVPR*, pages 4363–4372, 2020.

[51] Y. Yang and S. Soatto. FDA: fourier domain adaptation for semantic segmentation. In *CVPR*, pages 4084–4094, 2020.

[52] K. You, X. Wang, M. Long, and M. Jordan. Towards accurate model selection in deep unsupervised domain adaptation. In *ICML*, pages 7124–7133, 2019.

[53] W. Yun, B. Han, J. Lee, J. Kim, and J. Kim. Target-style-aware unsupervised domain adaptation for object detection. *IEEE Robotics Autom. Lett.*, 6(2):3825–3832, 2021.

[54] K. Zhang, B. Schölkopf, K. Muandet, and Z. Wang. Domain adaptation under target and conditional shift. In *ICML*, pages 819–827, 2013.

[55] Y. Zhang, T. Liu, M. Long, and M. Jordan. Bridging theory and algorithm for domain adaptation. In *ICML*, pages 7404–7413, 2019.

[56] H. Zhao, S. Zhang, G. Wu, J. M. F. Moura, J. P. Costeira, and G. J. Gordon. Adversarial multiple source domain adaptation. In *NeurIPS 2018*, pages 8568–8579, 2018.

[57] E. Zitzler and L. Thiele. Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach. *IEEE Trans. Evol. Comput.*, 3(4):257–271, 1999.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] See section 6
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] See appendix in supplementary materials.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] See appendix in supplementary materials
   (b) Did you include complete proofs of all theoretical results? [Yes] See appendix in supplementary materials
3. If you ran experiments...

(a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] See section 5.1 and the appendix in supplementary materials

(b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See section 5.1 and the appendix in supplementary materials

(c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] See appendix in supplementary materials

(d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See the appendix in supplementary materials

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...

(a) If your work uses existing assets, did you cite the creators? [Yes] See section 5.1

(b) Did you mention the license of the assets? [No] The code and the data are public.

(c) Did you include any new assets either in the supplemental material or as a URL? [No]

(d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [No] The code and the data are public.

(e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [No] The code and the data are public.

5. If you used crowdsourcing or conducted research with human subjects...

(a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]

(b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]

(c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]