GIIDA: Unsupervised Domain Adaptation via Gradual Interpolation Intermediate Domain Auxiliary

First Author[1,2*], Second Author[2,3] and Third Author[1,2]

[1*]Department, Organization, Street, City, 100190, State, Country.
[2]Department, Organization, Street, City, 10587, State, Country.
[3]Department, Organization, Street, City, 610101, State, Country.

*Corresponding author(s). E-mail(s): iauthor@gmail.com;
Contributing authors: iiauthor@gmail.com; iiiauthor@gmail.com;

Abstract

Unsupervised domain adaptation (UDA) aims to improve the model's generalization ability by learning domain invariant representation, which has made significant progress. However, it becomes more challenging as domain discrepancies between source and target domains increased. To overcome this issue, we propose a GIIDA (Gradual Interpolation Intermediate Domain Auxiliary) method. Different from the traditional directly adaptation, our method constructs a series of intermediate domains in the principles of category consistency matching and confidence-based sample mixing up. Thereby learning rich interdomain information through self-training. Concretely, we introduce a clean classifier and noise classifier to estimate noise transition matrix. The clean classifier is used to assign pseudo-labels for interdomain samples. Then, the intermediate domain samples provide the optimal parameters for the noise classifier in the form of closed solution. In the end, we designed an uncertainty guided regularization to alleviate the overconfidence issue. Extensive experimental results on three common benchmarks show the effectiveness of our method while bringing remarkable improvements against the baseline. Codes are available at https://github.com/Tinel46/GIIDA.

Keywords: Unsupervised domain adaptation, Noise transition matrix, Regularization, Mixup

1 Introduction

Deep convolutional neural networks (CNNs) have made great progress in various tasks such as object recognition, semantic segmentation. Its success deeply relies on large-scale tagging datasets. However, collecting and annotating samples accurately is costly and time consuming. For this, Unsupervised domain adaptation (UDA) methods have been proposed to alleviate the dilemma. It aims to learn a model from easily accessible labeled source data and adaptative to unlabeled target data without degrade the model's performance. Considering the correlations between different domains, the main challenge is to mitigate the discrepancy between the two domains while reduce the negative effects on model's generalization ability.

Typically, some methods [1–3]learn domain invariant representations in an adversarial way. Other methods [4, 5] explore a variety of distance measurement to align the two domains at feature-level. Although these two mainstream methods show competitive performance in small domain discrepancy. However, the model's performance is degraded significantly in the case of large domain shift. Adapting directly from the source domain to the target domain is obviously difficult. Therefore, recent gradual domain adaptation methods [6–11] have been proposed. Among these methods, some intermediate domains are given and served as bridge auxiliary domain adaptation. For example, Fixbi [11] handles large domain discrepancies via a fixed ratio-based mixup strategy and confidence-based learning methodologies. Therefore, our main concern is focus on constructing intermediate domain auxiliary from labeled source data and unlabeled target data.

In addition, previous studies [12] indicate that some semi-supervised learning methods [13–15] can often achieve excellent results when applied in UDAs. Especially, self-training is widely and effectively used. However, self-training-based methods deeply rely on pseudo labels. To improve the quality of pseudo-labels, some strategies such as high threshold filtering and weighting-based sample selection are often exploited. Then, suffered from domain discrepancy, the pseudo-labels are still noised. At the same time, the noise pseudo-label also further leads to error accumulation during training.

In this paper, we propose a gradual interpolation intermediate domain auxiliary method, which is to achieve domain adaptation from source domain to target domain indirectly with the aid of intermediate domains. As shown in Fig. 1, we firstly use the baseline model Fixmatch [13] to assign the pseudo-labels for target domain samples. With the progressive fixed proportion mixing up at pixels-level, a series of intermediate domains and corresponding pseudo labels are constructed. Besides, to ensure the validity of the intermediate domains, we adopt the principles of category consistency matching and confidence-based sample mixing up. Considering the facts that the pseudo label of intermediate domain sample is partly depend on the target data, it is inevitably containing noise. Therefore, directly using the noise data for self-training may lead to error accumulation. For this problem, we reduce the

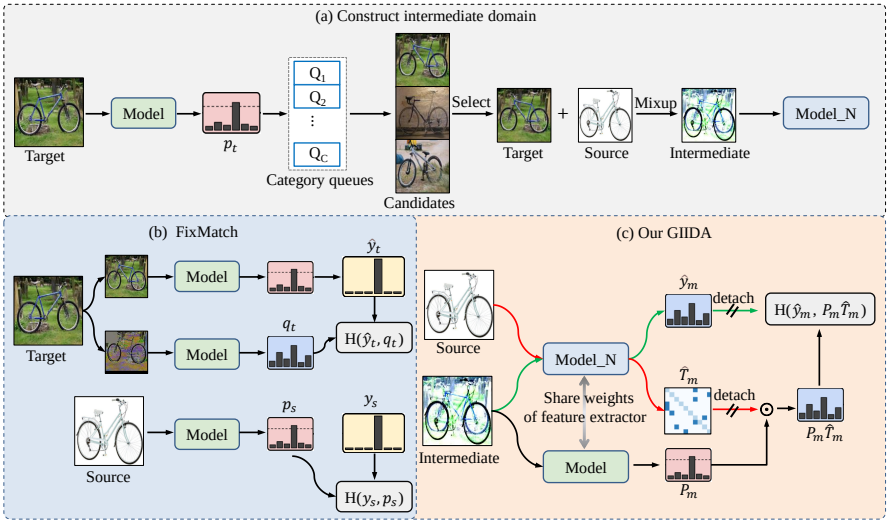


Fig. 1 Overview of the proposed GIIDA method. Where Model and Model_N share the weights of feature extractor, which is denoted as clean and noisy model respectively. We aim to train the clean model that perform well on both two domains. In concrete implementation, we choose FixMatch as the clean model. The intermediate domain is constructed to update the noise classifier of Model_N. Building upon the baseline FixMatch, we estimate the noise transition matrix via noise model and train our method with intermediate domain. Compare to the baseline, our method fully utilizes intermediate domain information.

negative impact of noise labels via a noise transition matrix. On the implementation side, we design a clean model and a noise model, which consist of feature extractor and classification head. Notably, both of them share the same feature extractor. Furthermore, the classification heads are denoted as clean classifier and noise classifier respectively. We set the noise classifier as a linear model. It has closed form solution that can be estimated by intermediate domain samples. To estimate the noise transition matrix, one can regard the labeled source data as anchor points and then calculate the noisy class posterior probability. However, due to the domain discrepancy, it can't be applied to target data especially when the domain discrepancy is large. Hence, we exploit the intermediate domain samples and estimate the noise transition matrix progressively. In experiments, we find that most of the model predictions are sharply. It is adversely to loss correction. Hence, we design a novel uncertainty guided regularization to alleviate the problem of over-confident mistaken prediction. Different from traditional regularization, the proposed method takes the uncertainty of target sample into account and shows better generate smooth outputs. In summary, the main contributions of this paper are listed as follows:

1. We propose a progressive interpolation intermediate domains auxiliary UDA method, which can effectively learn from the noise target data by virtue of the intermediate domains.

2. We design two principles: category consistency matching and confidence-based sample mixing up. It is used to construct more effective and reliable intermediate domains.
3. We conduct extensive experiments on common UDA benchmarks. The experimental results show that our method achieves significant improvements over state-of-the-art UDA methods.

2 Related Work

Unsupervised Domain Adaptation. UDA aims to transfer the knowledge learned from the labeled source domain to unlabeled target domain. One kind of method is based on adversarial training [1–3]. For example, DANN [1] learn the invariant representations through minimax optimization between feature extractor and domain discriminator. CDAN [2] further utilizes the category information expressed in classifier prediction and trained in an adversarial way. ALDA [3] introduced confusion matrix to improve the class discrimination of the network. Another approach is discrepancy-based [4, 5]. To reduce the domain gap, [4] propose to minimize the Maximum Mean Discrepancy (MMD). While a novel metric Margin Disparity Discrepancy (MDD) is used in [5]. At the same time, recent studies [12] show that some traditional semi-supervised methods also perform well in UDA, such as Fixmatch [13], Entropy mini [14], and MixMatch [15]. In addition, the self-training-based method is also widely applied in UDA [16–18]. In this study, we construct a series of intermediate domains. It provides more information in domain discrepancy to assist the adaptation from source domain to target domain.

Intermediate domains-based domain adaptation. It is difficult to perform domain adaptation directly, especially in the case of large domain gap. Recent studies [6–11, 19, 20] have made great efforts to utilize the intermediate domain as a bridge auxiliary domain adaptation. Kumar et.al [6] propose to adapt the model gradually with a series of intermediate domains. Under the strict theoretical analysis, gradual domain adaptation is better than iteratively self-training strategy when some assumptions are satisfied. Furthermore, in [7], a stepwise training domain discriminator and cycle consistency method is proposed to solve the problem of lacking sequence indexes in intermediate domains. However, above methods depend on a given set of intermediate domains, which are difficult to obtain in many practical tasks. Hence, some studies try to construct the intermediate domain across two domains. [19] propose to build gradually vanishing bridge layer for domain specific features learning, which implicitly connect two domains to intermediate domain. Besides, mixup strategy is widely applied in constructing intermediate domains. For example, [8] propose to construct intermediate domain by high-level feature interpolation. To select reliable samples for mixing, [9] design an auxiliary model and then generate intermediate samples with different ratio. Recently, [10] construct intermediate domain by adaptively interpolation strategy at specific feature layer, which is optimized with the shortest geodesic

distance and sample diversity constraint. Similarly, [11] adopt fixed ratio-based mixup and bidirectional matching for model adaptation. Different with above methods, we set the interpolation coefficient incrementally and select reliable sample for the intermediate domain construction.

Noisy label learning. To alleviate the negative effects of noise labels, a bulk of work resort to novel strategies such as loss correction [21], loss reweighting [22], label transition matrix estimation [23], and important sample selection [24]. In this paper, we focus on learning the noise transition matrix from the noise labels. GLC [25] propose to construct the noise transfer matrix by selecting the anchor points. However, the anchor-based method is heavily dependence on the quality of anchor points and the noise class posterior probability estimation. At the same time, the estimation accuracy is also affected by the number of anchors. In this regard, [26] introduce an intermediate class to decompose the transformation matrix. On the other hand, some works are to explore transition matrix estimation without anchors. For example, [27] use the meta-learning method to help constructing the transition matrix adaptively. [28] adopt features cluster and multi-order consistency check instead of anchors for transition matrix estimation. To learn the transition matrix online, [29] introduce volume regularization, anchor guidance, and convex guarantee to ensure the quality of the transition matrix. While [30] extend the transition matrix estimation from category level to instance level, and using the confidence to estimate separately for each instance. In this article, we estimate the transition matrix from the class target source data.

3 The gradual interpolation intermediate domain auxiliary method

For unsupervised domain adaptation, given the labeled source domain $X_s = \{(x_i^s, y_i^s)\}_{i=1}^{N_s}$ and unlabeled target domain $X_t = \{x_j^t\}_{j=1}^{N_t}$, where N_s and N_t denote the number of samples of X_s and X_t, respectively. We aim to transfer the model learned from source domain and can also perform well in the target domain. In this section, we will introduce our GIIDA method and its details.

3.1 Construct intermediate domain

To construct the intermediate domain, some studies directly mixing two domains at the feature level [10] or pixel level [11]. Different from the above methods, we propose an efficient and reliable progressive pixel-level interpolation method, which mixing two domains in the principles of category consistency matching and confidence-based sample mixing up. In the implementation side, it is necessary to choose the high confidence samples. However, the pseudo label of target domain sample is not reliable. We cannot guarantee that the source samples are matched with target samples of the same category in each mini-batch. Hence, we introduce category queues, which used to store each category target samples. As shown in Fig. 1, we initialize it as empty and set the number of queues to classes. For every mini-batch iteration, we

firstly use the baseline model to assign pseudo label to each target sample. Then, the target samples are enqueue into the corresponding category queue. For each labeled source sample, we query the corresponding target samples in the category queue and set it as a candidate sample to be matched. Notably, the confidence of all candidate samples is normalized, and the target domain samples are selected to match the source domain samples according to the probability.

Given a pair of source sample (x_i^s, y_i^s) and target sample $(x_i^t, \hat{y}_i^t)$, we gradual interpolation the intermediate domains as:

$$\begin{aligned} \widetilde{x}_i^m &= \lambda x_i^s + (1-\lambda) x_i^t \\ \widetilde{y}_i^m &= \lambda y_i^s + (1-\lambda) \hat{y}_i^t \end{aligned} \tag{1}$$

where λ is the interpolation coefficient. During training, λ is increased from 0 to 1 and set interval at 0.1. Formally, it seems have no difference with traditional mixup strategies. However, exist mixup methods does not require the categories of paired samples are matching. In UDA, domain differences between source and target domains are not negligible. There will be both domain and category information differences between source samples and target samples. It is not independent of each other. Therefore, the category consistency match is required, that is, the matched samples meet the category consistency $(y_i^s = \hat{y}_i^t)$, which enables it to focus on acquiring intermediate domain information and reduce the interference of category information.

On the other hand, for the classification task, the learning difficulty of each category is different, which resulting in higher sample confidence in some categories and lower sample confidence in others. It is also the defect of the traditional high threshold filtering method. While our category consistency matching only needs to consider the confidence differences of the same category samples. Intuitively, the higher confidence in the same category, the more reliable the label. As a result, the proposed confidence-based sample mixing up strategy increases the probability of selecting samples with high confidence under the same category, which ensure the reliability of the intermediate domain.

3.2 Construct noise transition matrix

The standard self-training loss of target domain is usually defined as follows:

$$L_{st}(X_T, \hat{Y}_T) = -\sum_{x_t \in X_T} \hat{y}_t \log P_t \tag{2}$$

where P_t is the output probability of target sample. $\hat{y}_t$ is the corresponding pseudo label. The quality of the pseudo-label determines the self-training effect. Commonly, the pseudo-label cannot guarantee of exact correctness. That is, the pseudo label always contain noise. As training progresses, errors would accumulate and result in degrading the model performance. In this

paper, we aim to introduce the noise transition matrix $T \in R^{C*C}$ to reduce the negative impact of noise pseudo labels. Where C represents the number of categories, $T_{i,j}$ represents the probability of transferring the ground truth label i to noisy label j.

The noise transition matrix [25, 26] is widely applied in deep vision tasks. In these tasks, the training data usually satisfy the same distribution. Assuming that $\hat{y}$ and y given x is conditional independence. Integrating over all x, we have:

$$P(\hat{y} \mid y) = P(\hat{y} \mid y) \int P(x \mid \hat{y}, y)\, dx = \int P(\hat{y} \mid y, x)\, P(x \mid y)\, dx. \tag{3}$$

Empirically, we can approximate the integral $\int P(\hat{y} \mid y, x)\, P(x \mid y)\, dx$ with the expectation of $P(\hat{y} \mid y, x)$ over distribution $P(x \mid y)$. In terms of implementation, we just have to estimate $P(\hat{y} \mid x)$, where $P(\hat{y} \mid y, x) = P(\hat{y} \mid x)$. Therefore, we design two classifiers, denoted as clean classifier and noisy classifier. The first classifier predicts the truth label of the sample. The second one predicts the noise class posterior probability, which is used to estimate the noise transition matrix. Intuitively, it is tempting to regard the source domain as clean dataset while the target domain with noise pseudo label as the noise dataset.

However, in UDA, the conditional independence assumption is hard to meet. Hence, it is inappropriate to rectify the self-training loss with source domain samples directly. Correspondingly, we use the intermediate domain instead of source domain. Considering that the intermediate domain is adaptive from the source domain to the target domain gradually. The conditional independence assumption can be approximately satisfied to a certain extent. Assuming that the feature extractor F and two classifiers are parameterized by $\varnothing$, θ_1, and θ_2 respectively. In the concrete implementation, the second classifier is a single linear layer. Given the input feature matrix $\mathcal{F} \in R^{B*d}$, where B is batch-size, d is the feature dimension. y is the ground truth label. The optimal parameter of the second classifier is taken as closed form and written as:

$$\hat{\theta}_2 = (\mathcal{F}^T \mathcal{F})^{-1} \mathcal{F}^T y \tag{4}$$

Each element of the estimated noise transition matrix can be written as:

$$\hat{T}_{i,j} = \frac{1}{\mid D_i \mid} \sum_{(x,y) \in D_i} P_{\phi, \hat{\theta}_2}(\hat{y} = j \mid y = i, x) \approx P(\hat{y} = j \mid y = i), \tag{5}$$

where $D_i = \{(\tilde{x}, \tilde{y}) \mid \tilde{x} \in \tilde{X}_m, \tilde{y} = i\}$ is a set of samples in intermediate domain with label i. At the same time, our gradual interpolation domain adaptation also accords with the idea of course learning. When the intermediate domain is constructed, the self-training loss $L_{seg}^M = -\sum_{x_m \in X_M} \hat{y}_m \log P_m$ can

be modified as:

$$L_{seg}^{M+T}(X_M, \hat{Y}_M) = -\sum_{\tilde{x}_m \in X_M} \hat{y}_m \log P_m \hat{T}_m \tag{6}$$

3.3 Uncertainty guided regularization

During training, the high confidence outputs are not guaranteed to be completely correct. If the output of the clean classifier is too sharp, the model is easily overfit to the noise pseudo label and generate overconfident mistaken. Even if the noise transition matrix estimation is correct, the clean classifier is less possible to update. Intuitively, due to the noise transition matrix is diagonally dominant, that is, the diagonal entry is larger than non-diagonal elements. if the model output as sharp as hard pseudo label, the clean class posterior probabilities flipped to the noise class posterior by noise transition matrix is almost keep the same. Therefore, we need to regularize the output of the model to make the pseudo-labels more smoothness so that the clean classifier has a chance to update the parameters. For simplification, we only perform regularization on target domain. The proposed uncertainty guided regularization is written as:

$$L_{reg}(P_t) = \frac{1}{\mid B_t \mid} \sum_{x_t \in B_t} (1 - w_t \sum_{k=1}^{C} P(k \mid x_t)^2) \tag{7}$$

As shown in Eq.(7), we need to estimate the uncertainty of each target example. Specifically, we model the uncertainty by entropy measurement, that $H(P_t) = -P_t log P_t$. Similar to [31, 32], we also exploit the same transformation scheme. Hence, the uncertainty weights w_t is represented as:

$$w_t = \frac{\mid B_t \mid (1 + exp(-H(P_t)))}{\sum_{i=1}^{B_t} (1 + exp(-H(P_i)))}, \tag{8}$$

where B_t is the batch size. w_t is used to quantify the importance of each target example for guiding regularization. It encourages the clean classifier output smoothness.

3.4 Overall formulation of GIIDA

We adopt FixMatch as a strong baseline. Given strongly and weakly augmented target images, the baseline loss function contains two parts: the standard cross-entropy loss $H(.)$ on source image and consistency loss between two augmented target views. We denote $\alpha(\cdot)$ and $\mathcal{A}(\cdot)$ as weakly-augmented and strongly-augmented respectively. First, we compute the clean class posterior probability of weakly-augmented view, being abbreviated to $P_{\phi,\theta_1}(x)$. Then we use $\hat{y} =$

$argmaxP_{\phi,\theta_1}(x_t)$ as the pseudo label, the loss term is written as:

$$\begin{aligned} l_{fixmatch} &= \frac{1}{\mid B_s \mid} \sum_{x_s \in B_s} H\left(y_s, P_{\phi,\theta_1}(x_s)\right) + \lambda_1 l_{st} \\ l_{st} &= \frac{1}{\mid \mu B_t \mid} \sum_{x_t \in \mu B_t} \mathbb{1}\left(maxP_{\phi,\theta_1}(x_t) \geq \tau\right) H\left(\hat{y}_t, P_{\phi,\theta_1}\left(\mathcal{A}(x_t)\right)\right), \end{aligned} \tag{9}$$

where μ is the ratio of labeled source data to unlabeled target data in a mini-batch. τ is the threshold and default set as 0.97. Given the estimated noise transition matrix $\hat{T}_m$, it is rational for using the modified self-training loss in Eq.(6). Together with regularization loss in Eq.(7), the overall loss for our GIIDA is formulated as:

$$L_{total} = l_{fixmatch} + \lambda_2 L_{seg}^{M+T} + \lambda_3 L_{reg}, \tag{10}$$

where λ_2 and λ_3 are used to balance each loss term.

4 Experiment

In this section, we evaluate the proposed method on three public benchmarks and report comparative results of recent state-of-the-art domain adaptation methods. In addition, we also conduct extensive ablation studies to validate the contribution of the proposed method.

4.1 Datasets

Office-31 [33] is a popular but small-scale benchmark for cross-domain adaptation, which contains three distinct domains, namely Amazon (A) domain, DSLR (D) domain and Webcam (W) domain. Each domain includes 2817, 498, and 795 images and shares 31 categories. Following the common settings, all methods are evaluated on the six transfer task pairs A→W, A→D, D→W, D→A, W→A, and W→D.

Office-Home [34] is a larger and more challenging domain adaptation dataset than the Office-31. It consists of 15,500 images with 65 object categories collected in the office and home scenes. Similarly, all images are divided into four different domains: Artistic images (Ar), Clip Art (Cl), Product images (Pr), and Real-World images (Rw). We conducted comparative experiments in all 12 transfer tasks.

VisDA-2017 [35] is a large-scale simulation-to-real dataset for visual domain adaptation, which includes over 280K images shared by 12 categories. Generally, the source domain includes 152,397 synthetic images while the target domain contains 55,388 real images. We evaluate all methods on the Synthetic→Real task.

Table 1 Accuracy (%)on Office-31 for UDA (ResNet-50).The best accuracy is indicated in bold. * Reproduced by [12]

Method	A → W	D → W	W→D	A→D	D→A	W→A	Avg.
ResNet-50 [36]	68.4	96.7	99.3	68.9	62.5	60.7	76.1
MCC [32]	95.5	98.6	**100.0**	94.4	72.9	74.9	89.4
DSAN [37]	93.6	98.3	**100.0**	90.2	73.5	74.8	88.4
GDCAN [38]	94.8	98.2	**100.0**	93.6	76.9	74.4	89.7
ATDOC [39]	94.6	98.1	99.7	95.4	77.5	77.0	90.4
BCDM [40]	95.4	98.6	**100.0**	93.8	73.1	73.2	89.0
TSA [41]	96.0	98.7	**100.0**	95.4	76.7	76.8	90.6
SCDA [42]	94.2	98.7	99.8	95.2	75.7	76.2	90.0
UDA [43]	93.6	98.6	**100.0**	95.6	73.5	74.2	89.2
FixBi [11]	96.1	**99.3**	**100.0**	95.0	78.7	**79.4**	91.4
CST [31]	90.3	98.9	**100.0**	93.4	76.6	77.1	89.4
FixMatch* [13]	92.6	98.9	**100.0**	93.6	73.5	71.6	88.3
FixMatch(μ=2)	92.6	98.1	**100.0**	95.3	66.7	70.3	87.2
GIIDA(ours)	**97.2**	98.4	**100.0**	**96.8**	**78.9**	77.4	**91.5**

4.2 Experimental settings

Following common unsupervised domain adaption setting, we train the model on labeled source data and unlabeled target data. As for data pre-processing, we adopt weakly-augmentation and strongly-augmentation, which have been adopted in some semi-supervised methods [13]. In terms of backbone network, ResNet-50 [36] pre-trained on ImageNet [44] is used for Office-31 and Office-Home datasets while ResNet-101 pre-trained on ImageNet for VisDA-2017 dataset. We set hyperparameters $\mu = 1$ for VisDA-2017 and $\mu = 2$ for other datasets. In training, we adopt SGD optimizer with momentum 0.9, initial learning rate 0.01, and weight decay 0.001. we set batch size 32 and follow the same scheme for adjusting learning rate as in [31]. As for the interpolation coefficient λ, which is initialized to 0.1 and ranged from 0 to 1. It is increased 0.1 for every training epoch. Other hyperparameters λ_1, λ_2, and λ_3 are set to 1.0, 1.0, 0.5 for VisDA-2017 and $\lambda_1 = 0.5$ for other datasets, respectively. All experiments are implemented by Pytorch framework with an NVIDIA TESLA V100 GPU.

4.3 Experimental Results

Results on Office-31. Table 1 shows the comparison results of our method on the OFFCE-31 dataset. We adopt Fixmatch as the baseline and its effect is largely affected by hyperparameter μ. Where μ is the ratio of labeled source data to unlabeled target data in mini-batch. In the original setting [13], μ is set to 7. Compared to baseline, our method sets μ as 2 and brings average 4.3% improvement. It is greatly reduced the consumption of memory without compromising the performance of the model. In addition, it is worth noting that our method achieved significant improvement over the state-of-the-art

Table 2 Accuracy (%) on OfficeHome for UDA (ResNet-50). The best accuracy is indicated in bold. * Reproduced by [31].

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 [36]	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
MCC [32]	55.1	75.2	79.5	63.3	73.2	75.8	66.1	52.1	76.9	73.8	58.4	83.6	69.4
DSAN [37]	54.4	70.8	75.4	60.4	67.8	68.0	62.6	55.9	78.5	73.8	60.6	83.1	67.6
GDCAN [38]	57.3	75.7	83.1	68.6	73.2	77.3	66.7	56.4	82.2	74.1	60.7	83.0	71.5
ATDOC [39]	60.2	77.8	82.2	68.5	78.6	77.9	68.4	58.4	**83.1**	74.8	61.5	**87.2**	73.2
TSA [41]	53.6	75.1	78.3	64.4	73.7	72.5	62.3	49.4	77.5	72.2	58.8	82.1	68.3
SCDA [42]	**60.7**	76.4	82.8	69.8	77.5	78.4	68.9	59.0	82.7	74.9	61.8	84.5	73.1
TCM [45]	58.6	74.4	79.6	64.5	74.0	75.1	64.6	56.2	80.9	74.6	60.7	84.7	70.7
ToAlign [46]	57.9	76.9	80.8	66.7	75.6	77.0	67.8	57.0	82.5	75.1	60.0	84.9	72.0
FixBi [11]	58.1	77.3	80.4	67.7	**79.5**	78.1	65.8	57.9	81.7	76.4	**62.9**	86.7	72.7
CST [31]	59.0	79.6	**83.4**	68.4	77.1	76.7	68.9	56.4	83.0	75.3	62.2	85.1	73.0
FixMatch* [13]	51.8	74.2	80.1	63.5	73.8	61.3	64.7	51.4	80.0	73.3	56.8	81.7	67.7
GIIDA(ours)	58.0	**79.7**	83.3	**70.9**	77.7	**79.6**	**72.6**	**59.2**	83.0	**77.8**	62.5	85.7	**74.2**

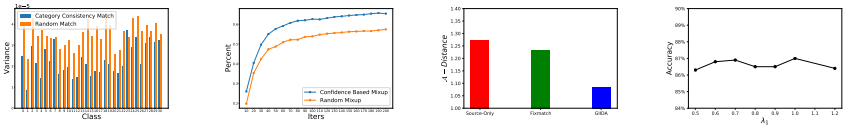
Table 3 Accuracy (%) on VisDA2017 for UDA (ResNet-101). The best accuracy is indicated in bold. * Reproduced by [31].

Method	aero	bicycle	bus	car	horse	knife	motor	person	plant	skate	train	truck	Avg.
ResNet-101 [36]	67.7	27.4	50.0	61.7	69.5	13.7	85.9	11.5	64.4	34.4	84.2	19.2	49.1
CDAN [2]	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
MCC [32]	92.2	82.9	76.8	66.6	90.9	78.5	87.9	73.8	90.1	76.1	87.1	41.0	78.7
DSAN [37]	90.9	66.9	75.7	62.4	88.9	77.0	**93.7**	75.1	92.8	67.6	89.1	39.4	76.6
BCDM [40]	95.1	**87.2**	81.2	73.2	92.7	95.4	86.9	**82.5**	**95.1**	84.8	88.1	39.5	83.4
ATDOC [39]	95.3	84.7	82.4	75.6	95.8	**97.7**	88.7	76.6	94.0	91.7	**91.5**	61.9	86.3
DWL [47]	90.7	80.2	**86.1**	67.6	92.4	81.5	86.8	78.0	90.6	57.1	85.6	28.7	77.1
TSA [41]	-	-	-	-	-	-	-	-	-	-	-	-	82.0
CGDM [48]	93.4	82.7	73.2	68.4	92.9	94.5	88.7	82.1	93.4	82.5	86.8	49.2	82.3
CST [31]	96.1	86.3	83.2	79.9	94.1	97.6	87.2	77.9	94.6	90.1	85.8	**62.2**	86.3
FixMatch* [13]	-	-	-	-	-	-	-	-	-	-	-	-	79.5
GIIDA(ours)	**96.6**	86.2	84.8	**80.1**	**96.7**	96.4	90.5	82.0	**95.1**	**92.7**	88.9	53.4	**87.0**

methods, which can be attributed to the intermediate domain providing a variety of data information. Particularly, the encouraging results on tasks D→A and W→A further show the proposed uncertainty guided regularization is beneficial to avoid the overfitting to noisy pseudo-labels.

Results on Office-Home. We show the average classification accuracy in Table 2. Overall, our method achieves large improvement in all tasks, which surpasses the baseline by a large margin of 6.5%. Compared with other state-of-art domain adaptation methods, our method shows the best average accuracy of 74.2%. Even in difficult transfer tasks such as Cl→Rw and Pr→Ar, we achieve 7.9% and 18.3% improvements. The promising results indicate that gradual interpolation intermediate domain can effectively assist the domain adaptation task, especially in large domain difference.

Results on VisDA-2017. Table 3 shows per-class accuracy results on VisDA-2017 dataset. Obviously, we obtain the highest mean accuracy of 87.0% and show superior performance over other excellent UDA method. These considerable results show the GIIDA can effectively improve the adaptation ability.



(a) Variance analysis (b) Accuracy analysis (c) $\mathcal{A}$-distance (d) Parameter sensitivity

Fig. 2 (a): Variance (%) of each category for different matching strategy. (b): Percent (%) of the correct pseudo-label in intermediate domainon for different mixup strategy. Taking the task D→A on Office-31 as example, Best viewed in color. (c): $\mathcal{A}$-distance results on task Pr→Ar of Office-Home dataset. (d): Parameter sensitivity analysis on VisDA-2017.

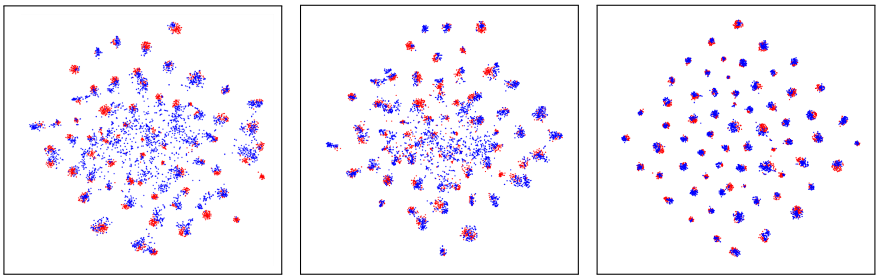
Table 4 Ablation results (%) of investigating the effects of our components on Office-31.

FixMatch	L_{seg}^{M}	L_{seg}^{M+T}	L_{reg}	A → W	D → W	W→D	A→D	D→A	W→A	Avg.
√				92.6	98.1	100.0	95.3	66.7	70.3	87.2
√	√			95.2	97.9	100.0	95.0	73.1	74.2	89.2
√	√	√		96.7	98.1	100.0	96.0	72.1	76.3	89.9
√	√	√	√	97.2	98.4	100	96.8	78.9	77.4	91.5

4.4 Analysis

Effects of the components of our GIIDA. To validate the effectiveness of each component, we conduct ablation studies on Office-31 dataset. L_{seg}^{M} means standard self-training while L_{seg}^{M+T} is modified with noise transition matrix. L_{reg} means the proposed regularization. As shown in Table 4, the constructed intermediate domain is effectively, which has gain 2% improvement over baseline. When applied noise transition matrix to modify the model output, it has further 0.7% gains. This shows that the noise transition matrix is effective to reduce the negative impact of noise pseudo labels. In addition, our uncertainty guided regularization helps to alleviate the problem of overfitting to noise pseudo label and improve performance. Overall, GIIDA improves the baseline by an average of 4.3%. Each component is effective in improving performance.

The Mixup strategy in GIIDA. Mixup is an effective image enhancement technique. Typically, it randomly mixes two samples at pixel level and the interpolation ratio is randomly sampled from distribution $Beta(\alpha, \alpha)$. Different from traditional Mixup, we take the domain discrepancy into account and adopt a progressive fixed interpolation ratio. Furthermore, our Mixup strategy need to meet the principles of category consistency matching and confidence-based sample mixing up. To validate the effectiveness of these two principles, we compare our category consistency matching with random ways. Firstly, we train a domain discriminator and set the mixing ratio λ to 0.5. According to the principle of category consistency matching and random category matching, we construct the intermediate domain respectively. Then, we put the intermediate domain samples into discriminator and calculate the variance of each category



(a) Source-Only (b) Fixmatch (c) GIIDA

Fig. 3 The t-SNE visualization of features on the task Pr (red) → Ar (blue) of Office-Home.

respectively. As shown in Fig. 2(a), almost all categories, the variance of random category matching is significantly higher than our category consistency matching. It indicates that the domain information and category information of samples are not independent of each other, and it is more reasonable and effective to use category consistency matching. In addition, we also count the ratio of correct pseudo labels under confidence-based sample mixing up and random sample mixing up. For brevity, we take the task D→A of office-31 as an example. As illustrated in Fig. 2(b), the proportion of correct pseudo-labels is increasing as training. Obviously, our confidence-based sample mixing up show higher proportion. It shows that the pseudo-labels are more reliable, verifying the effectiveness the principle of confidence-based sample mixing up.

Feature visualization. Taking task Pr→Ar of Office-Home as an example, we show the visualize result of features learned by different methods with t-SNE in Fig. 3. The feature distribution of Source-Only method is very scattered while the features for FixMatch and our GIIDA are more compact cluster. In further comparison, although the feature distribution of Fixmatch can form relatively independent clusters across two domains, most target features cannot be well aligned with source features. By contrast, our GIIDA method is well aligned and show more compact cluster features. Besides, we also computer the a-distance between the compared approaches. As shown in Fig. 2(c), the a-distance of our method is significantly smaller than the baseline. It demonstrates that our method is more effective to reduce the domain divergence.

Parameter Sensitivity. Our training object contains three hyper-parameters λ_1, λ_2, and λ_3. When λ_2 and λ_3 is fix as 1.0, 0.5 respectively, our GIIDA method performs well on VisDA-2017. Considering that the modified self-training loss of intermediate domain and the regularization loss term constitute main contribution in this paper, we only adjust λ_1 from 0.5 to 1.2. The parameter sensitivity result is shown in Fig. 2(d). On the whole, the mean accuracy is not sensitive to λ_1(with $\lambda_1 = 1.0$ working best), which reflects the stability of our method.

5 Conclusion

In this paper, we studied the issue that domain adaptation is difficult in large domain discrepancy. We proposed gradual interpolation intermediate domain auxiliary method, which exploited the noise transition matrix to improve the self-training effect, and introduced the uncertainty guided regularization to reduce overfitting to noise pseudo labels. Comprehensive experimental results demonstrated the effectiveness of our method, and the proposed method achieved competitive performance with the state-of-the-art methods. However, the mixing ratio updating strategy needs to be manually selected. In future research, we expect to select the mixing ratio adaptively under the shortest geodesic constraint in manifold space.

Acknowledgments. This work was supported in part by the National Science Fund of China no.61871170; Key Research and Development Plan of Zhejiang: No.2021C03131.

References

[1] Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M., Lempitsky, V.: Domain-adversarial training of neural networks. The journal of machine learning research **17**(1), 2096–2030 (2016)

[2] Long, M., Cao, Z., Wang, J., Jordan, M.I.: Conditional adversarial domain adaptation. Advances in neural information processing systems **31** (2018)

[3] Chen, M., Zhao, S., Liu, H., Cai, D.: Adversarial-learned loss for domain adaptation. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, pp. 3521–3528 (2020)

[4] Long, M., Cao, Y., Wang, J., Jordan, M.: Learning transferable features with deep adaptation networks. In: International Conference on Machine Learning, pp. 97–105 (2015). PMLR

[5] Zhang, Y., Liu, T., Long, M., Jordan, M.: Bridging theory and algorithm for domain adaptation. In: International Conference on Machine Learning, pp. 7404–7413 (2019). PMLR

[6] Kumar, A., Ma, T., Liang, P.: Understanding self-training for gradual domain adaptation. In: International Conference on Machine Learning, pp. 5468–5479 (2020). PMLR

[7] Chen, H.-Y., Chao, W.-L.: Gradual domain adaptation without indexed intermediate domains. Advances in Neural Information Processing Systems **34**, 8201–8214 (2021)

[8] Abnar, S., Berg, R.v.d., Ghiasi, G., Dehghani, M., Kalchbrenner, N., Sedghi, H.: Gradual domain adaptation in the wild: When intermediate distributions are absent. arXiv preprint arXiv:2106.06080 (2021)

[9] Zhang, Y., Deng, B., Jia, K., Zhang, L.: Gradual domain adaptation via self-training of auxiliary models. arXiv preprint arXiv:2106.09890 (2021)

[10] Dai, Y., Sun, Y., Liu, J., Tong, Z., Yang, Y., Duan, L.-Y.: Bridging the source-to-target gap for cross-domain person re-identification with intermediate domains. arXiv preprint arXiv:2203.01682 (2022)

[11] Na, J., Jung, H., Chang, H.J., Hwang, W.: Fixbi: Bridging domain spaces for unsupervised domain adaptation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 1094–1103 (2021)

[12] Zhang, Y., Zhang, H., Deng, B., Li, S., Jia, K., Zhang, L.: Semi-supervised models are strong unsupervised domain adaptation learners. arXiv preprint arXiv:2106.00417 (2021)

[13] Sohn, K., Berthelot, D., Carlini, N., Zhang, Z., Zhang, H., Raffel, C.A., Cubuk, E.D., Kurakin, A., Li, C.-L.: Fixmatch: Simplifying semi-supervised learning with consistency and confidence. Advances in neural information processing systems **33**, 596–608 (2020)

[14] Grandvalet, Y., Bengio, Y.: Semi-supervised learning by entropy minimization. Advances in neural information processing systems **17** (2004)

[15] Berthelot, D., Carlini, N., Goodfellow, I., Papernot, N., Oliver, A., Raffel, C.A.: Mixmatch: A holistic approach to semi-supervised learning. Advances in neural information processing systems **32** (2019)

[16] Zou, Y., Yu, Z., Kumar, B., Wang, J.: Unsupervised domain adaptation for semantic segmentation via class-balanced self-training. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 289–305 (2018)

[17] Akkaya, I.B., Altinel, F., Halici, U.: Self-training guided adversarial domain adaptation for thermal imagery. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 4322–4331 (2021)

[18] Mei, K., Zhu, C., Zou, J., Zhang, S.: Instance adaptive self-training for unsupervised domain adaptation. In: European Conference on Computer Vision, pp. 415–430 (2020). Springer

[19] Cui, S., Wang, S., Zhuo, J., Su, C., Huang, Q., Tian, Q.: Gradually vanishing bridge for adversarial domain adaptation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 12455–12464 (2020)

[20] Hsu, H.-K., Yao, C.-H., Tsai, Y.-H., Hung, W.-C., Tseng, H.-Y., Singh, M., Yang, M.-H.: Progressive domain adaptation for object detection. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pp. 749–757 (2020)

[21] Patrini, G., Rozza, A., Krishna Menon, A., Nock, R., Qu, L.: Making deep neural networks robust to label noise: A loss correction approach. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1944–1952 (2017)

[22] Liu, T., Tao, D.: Classification with noisy labels by importance reweighting. IEEE Transactions on pattern analysis and machine intelligence **38**(3), 447–461 (2015)

[23] Wang, Z., Hu, G., Hu, Q.: Training noise-robust deep neural networks via meta-learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 4524–4533 (2020)

[24] Wei, H., Feng, L., Chen, X., An, B.: Combating noisy labels by agreement: A joint training method with co-regularization. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 13726–13735 (2020)

[25] Hendrycks, D., Mazeika, M., Wilson, D., Gimpel, K.: Using trusted data to train deep networks on labels corrupted by severe noise. Advances in neural information processing systems **31** (2018)

[26] Yao, Y., Liu, T., Han, B., Gong, M., Deng, J., Niu, G., Sugiyama, M.: Dual t: Reducing estimation error for transition matrix in label-noise learning. Advances in neural information processing systems **33**, 7260–7271 (2020)

[27] Shu, J., Zhao, Q., Xu, Z., Meng, D.: Meta transition adaptation for robust deep learning with noisy labels. arXiv preprint arXiv:2006.05697 (2020)

[28] Zhu, Z., Song, Y., Liu, Y.: Clusterability as an alternative to anchor points when learning with noisy labels. In: International Conference on Machine Learning, pp. 12912–12923 (2021). PMLR

[29] Guo, X., Liu, J., Liu, T., Yuan, Y.: Simt: Handling open-set noise for domain adaptive semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 7032–7041 (2022)

[30] Berthon, A., Han, B., Niu, G., Liu, T., Sugiyama, M.: Confidence scores make instance-dependent label-noise learning possible. In: International Conference on Machine Learning, pp. 825–836 (2021). PMLR

[31] Liu, H., Wang, J., Long, M.: Cycle self-training for domain adaptation. Advances in Neural Information Processing Systems **34**, 22968–22981 (2021)

[32] Jin, Y., Wang, X., Long, M., Wang, J.: Minimum class confusion for versatile domain adaptation. In: European Conference on Computer Vision, pp. 464–480 (2020). Springer

[33] Saenko, K., Kulis, B., Fritz, M., Darrell, T.: Adapting visual category models to new domains. In: European Conference on Computer Vision, pp. 213–226 (2010). Springer

[34] Venkateswara, H., Eusebio, J., Chakraborty, S., Panchanathan, S.: Deep hashing network for unsupervised domain adaptation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5018–5027 (2017)

[35] Peng, X., Usman, B., Kaushik, N., Hoffman, J., Wang, D., Saenko, K.: Visda: The visual domain adaptation challenge. arXiv preprint arXiv:1710.06924 (2017)

[36] He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)

[37] Zhu, Y., Zhuang, F., Wang, J., Ke, G., Chen, J., Bian, J., Xiong, H., He, Q.: Deep subdomain adaptation network for image classification. IEEE transactions on neural networks and learning systems **32**(4), 1713–1722 (2020)

[38] Li, S., Xie, B., Lin, Q., Liu, C.H., Huang, G., Wang, G.: Generalized domain conditioned adaptation network. IEEE Transactions on Pattern Analysis and Machine Intelligence (2021)

[39] Liang, J., Hu, D., Feng, J.: Domain adaptation with auxiliary target domain-oriented classifier. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 16632–16642 (2021)

[40] Li, S., Lv, F., Xie, B., Liu, C.H., Liang, J., Qin, C.: Bi-classifier determinacy maximization for unsupervised domain adaptation. In: AAAI, vol. 2, p. 5 (2021)

[41] Li, S., Xie, M., Gong, K., Liu, C.H., Wang, Y., Li, W.: Transferable

semantic augmentation for domain adaptation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 11516–11525 (2021)

[42] Li, S., Xie, M., Lv, F., Liu, C.H., Liang, J., Qin, C., Li, W.: Semantic concentration for domain adaptation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 9102–9111 (2021)

[43] Xie, Q., Dai, Z., Hovy, E., Luong, T., Le, Q.: Unsupervised data augmentation for consistency training. Advances in Neural Information Processing Systems **33**, 6256–6268 (2020)

[44] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., *et al.*: Imagenet large scale visual recognition challenge. International journal of computer vision **115**(3), 211–252 (2015)

[45] Yue, Z., Sun, Q., Hua, X.-S., Zhang, H.: Transporting causal mechanisms for unsupervised domain adaptation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 8599–8608 (2021)

[46] Wei, G., Lan, C., Zeng, W., Zhang, Z., Chen, Z.: Toalign: Task-oriented alignment for unsupervised domain adaptation. Advances in Neural Information Processing Systems **34**, 13834–13846 (2021)

[47] Xiao, N., Zhang, L.: Dynamic weighted learning for unsupervised domain adaptation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 15242–15251 (2021)

[48] Saunders, D.: Domain adaptation and multi-domain adaptation for neural machine translation: A survey. arXiv preprint arXiv:2104.06951 (2021)