# Fast Generalized Distillation for Semi-supervised Domain Adaptation

No Author Given

No Institute Given

**Abstract.** Semi-supervised domain adaptation (SDA) is a typical setting when we face the problem of domain adaptation in real applications. How to effectively utilize the unlabeled data is an important issue in SDA. Previous work requires access to the source data to measure the data distribution mismatch, which is ineffective, when the size of the source data is relatively large. In this paper, we propose a new paradigm, called *Generalized Distillation Semi-supervised Domain Adaptation* (GDSDA) that can effectively utilize the unlabeled data to transfer the knowledge from the source models to solve the SDA problem. We demonstrate that the value of the imitation parameter is crucial to the performance of the target model in GDSDA. Therefore, we propose GDSDA-SVM which uses SVM as the base classifier and can efficiently estimate the imitation parameter. Experiment results show that GDSDA-SVM can effectively utilize the unlabeled data to transfer the knowledge between different domains under the SDA setting.

## 1 Introduction

Domain adaptation can be used in many real applications, which addresses the problem of learning a target domain with the help of a different but related source domain. In real applications, it can be very expensive to obtain sufficient labeled examples while there are abundant unlabeled ones. *Semi-supervised domain adaptation* (SDA) tries to exploit the knowledge from the source domain and use a certain amount of unlabeled examples and a few labeled ones from the target domain to learn a target model. Typically, the labeled examples in the target domain are too few to construct a good classifier alone. Therefore, an important issue in SDA is how to effectively utilize the unlabeled examples.

In previous work, many methods have been proposed to leverage the source knowledge with the unlabeled data. Duan et al.[6] proposed a method to force the source models and the target model to agree on the unlabeled data. Daumé et al[3] utilized unlabeled data as a co-regularizer and forced the hypotheses learned from different domains to agree on the unlabeled data. Meanwhile, Yao et al.[20] used the unlabeled target examples to discover the underlying intrinsic information in the target domain. Donahue et al.[5] show that using the smoothness constraints on the classifier scores over the unlabeled data can lead to the improved transfer result. The previous work in SDA requires access to the source data to measure the data distribution mismatch between the source and target domain. However, in some situations, we may not be able to access each of the source examples for many reasons. When we use a large dataset as our source domain, for example, it is ineffective to compare each of the source examples with the target data to estimate the data distribution mismatch.

Recently, a framework called *Generalized Distillation* (**GD**)[13] was proposed, which allows the knowledge to be transferred between different models effectively. GD includes two different models, the teacher model and student model. The student model tries to learn from the teacher model by mimicking the outputs of the teacher model on the training data. Remarkably, in GD, the knowledge can be directly transferred from the teacher model to the student model without accessing the data used to train the teacher. Moreover, GD can be used to exploit the information of the unlabeled data in a semi-supervised scenario[13]. Given that GD has such ability, it is natural to ask the following two questions: (1) Can the GD framework be applied to solve the SDA problem? (2) How can we improve its effectiveness when we apply GD to real SDA applications?

To answer these two questions, in this paper, we first propose a new paradigm, called *Generalized Distillation Semi-supervised Domain Adaptation* (**GDSDA**), to solve the SDA problem. We show that, with GDSDA the knowledge of the source models can be effectively transferred to the target domain using the unlabeled data. Specifically, the target model is trained with the help of the soft labels, i.e. the predictions of the target domain examples given by the source models. Therefore, without accessing each of the source examples, GDSDA is more efficient especially when the source domain is relatively large and the source model is well-trained.

Then we argue that the imitation parameter of GDSDA which controls the amount of knowledge transferred from the source model can greatly affect the performance of the target model. However, according to the previous work[13, 17], the imitation parameter is a hyperparameter and can only be determined by either brute force search or background knowledge. Therefore, we propose a novel imitation parameter estimation method for GDSDA, called GDSDA-SVM, which uses SVM as the base classifier and determines the imitation parameter efficiently. In particular, we use the Mean Square Error loss for GDSDA-SVM and show that the Leave-one-out cross validation (LOOCV) loss can be calculated in a closed form. By minimizing the LOOCV loss on the target data, we can find the optimal imitation parameter. In our experiments, we show that GDSDA-SVM can effectively find the optimal imitation parameter and achieve competitive performance compared to methods using brutal force search but with faster speed.

To summarize, the main contributions of this paper are: (1) We propose the paradigm of GDSDA that can directly transfer the knowledge from the source model with the help of unlabeled data for the SDA problems. (2) We propose the GDSDA-SVM which can effectively find the optimal imitation parameter for real SDA applications.

## 2 Generalized Distillation for Semi-supervised Domain Adaptation

GDSDA is a paradigm using GD for the SDA problem. In this section, we first give a brief review of GD. Then we illustrate the process of GDSDA and demonstrate the reason why GDSDA can work for the SDA problem. Finally, we show the importance of the imitation parameter.

### 2.1 An overview of Generalized Distillation and GDSDA

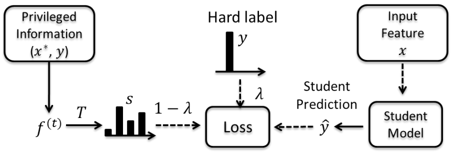


Fig. 1: Illustration of Generalized Distillation training process.

Generalized Distillation can be considered as the hybrid of two famous learning paradigms *Distillation*[10] and *Learning Using Privileged Information*(LUPI)[19]. In GD, the training data can be represented as a collection of triples:

$$\{(x_1, x_1^*, y_1), (x_2, x_2^*, y_2) \dots (x_n, x_n^*, y_n)\}$$

$x^*$ is the privileged information for data $x$, which is only available in the training process and $y$ is the corresponding label. The process of generalized distillation is as follows: in step 1, a teacher model $f^{(t)}$ is trained using the input-output pairs $\{x_i^*, y_i\}_{i=1}^n$. In step 2, $f^{(t)}$ is used to generate the soft label $s_i$ for each training example $x_i$ using the softmax function $\sigma$:

$$s_i = \sigma(f^{(t)}(x_i)/T) \tag{1}$$

where $T$ is a hyperparameter called *temperature* to control the smoothness of the soft label. In step 3, the student $f^{(s)}$ is learned from the pairs $\{(x_i, y_i), (x_i, s_i)\}_{i=1}^n$ using:

$$f^{(s)} = \underset{f^{(s)} \in \mathcal{F}^{(s)}}{\arg\min} \frac{1}{n} \sum_{i=1}^{n} \left[ \lambda \ell\left(y_i, f^{(s)}(x_i)\right) + (1-\lambda)\ell\left(s_i, f^{(s)}(x_i)\right) \right] \tag{2}$$

Here, $\ell(\cdot, \cdot)$ is the loss function and $\lambda$ is the *imitation parameter* to balance the importance between the hard label $y_i$ and the soft label $s_i$. When testing, the student model can predict with the data $x$ alone, without the assistance of the privileged information.

In domain adaptation, when we consider the source model as the teacher and the predictions of the target data given by the source models as the privileged information, GD can be naturally applied to SDA. This leads to *Generalized Distillation Semi-supervised Domain Adaptation* (**GDSDA**). Moreover, in GDSDA, we also consider the multi-source scenario and extend the GD paradigm to fit this scenario. To be consistent with other work in domain adaptation, we use the source model and the target model to denote the teacher model and the student model in the rest of our paper in GDSDA.

An important issue of applying GD to SDA is that, in Eq. (2), each target example is assigned with a hard label $y$ (true label) and a soft label $s$ (class probabilities from the teacher). However, in SDA, we are not able to obtain the hard labels of the unlabeled data. Here we use the “fake label” strategy to label the target data; for the labeled examples, we use *one-hot* strategy to encode their labels while using 0s as the labels of the unlabeled examples. Thus, each example in the target domain is assigned with a label. It is arguable that the “fake label” strategy would introduce extra noise and degrade the performance. However, we will show in our experiment that this noise can be limited by setting a proper value to the imitation parameter and thus, GDSDA can still leverage the source knowledge effectively (See the single source experiment).

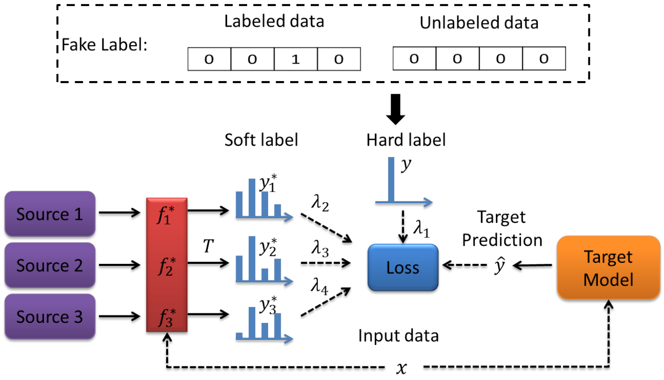


Fig. 2: Illustration of GDSDA training process and the “fake label” strategy.

The process of GDSDA is shown in Figure 2. Suppose we have $M-1$ source domains denoted as $D_s^{(j)} = \{X^{(j)}, Y^{(j)}\}_{j=1}^{M-1}$ and the target domain $D_t = \{X, Y\}$ encoded with the “fake label” strategy. The process of GDSDA is as follows:

1. Train the source models $f_j^*$ for each of the $M-1$ domains with $\{X^{(j)}, Y^{(j)}\}$.
2. For each training example $x_i$ in the target domain, computer the corresponding soft label $y_{ij}^*$ with each of the source model $f_j^*$ and the temperature $T > 0$.
3. Learn the target model $f_t$ with the $(M+1)$-tuples $\{x_i, y_i, y_{i1}^*, \ldots, y_{i(M-1)}^*\}_{i=1}^{L}$ and the imitation parameters $\{\lambda_i\}_{i=1}^{M}$ using (3):

$$f_t(\lambda) = \underset{f_t \in \mathcal{F}}{\arg\min} \frac{1}{L} \sum_{i=1}^{L} \left[ \lambda_1 \ell\left(y_i, f_t(x_i)\right) + \sum_{j=1}^{M-1} \lambda_{j+1} \ell\left(y_{ij}^{*}, f_t(x_i)\right) \right] \tag{3}$$

Compared to other work in SDA which requires using each example of the source domain, by either re-weighting [5, 8] or feature augmentation [3], GDSDA only requires the trained model from the source domain to generate the soft labels. Meanwhile, GDSDA is able to handle the multi-class scenario while some previous work, such as SHFA[7], can only solve the binary classification problem in SDA. Moreover, GDSDA is able to transfer the knowledge from any type of source model that is able to output the soft label (class probabilities) without accessing the source data.

### 2.2 Why does GDSDA work?

In this part, we demonstrate the scenario where GDSDA can work. Before we provide our analysis, we first introduce the two basic assumptions of GDSDA: the *assumption of distillation* and *the assumption of the source model.*

**Assumption of distillation:** *The capacity (VC dimension) of the target model $f_t$ is smaller than the capacity of source model $f^*$.* This assumption is inherited from GD. **Assumption of the source model:** *The source model $f^*$ should work better than a target model $f'_t$ trained only with the hard labels.* This assumption is based on a simple fact that it is more effective to learn from a superior model. This assumption is very common especially in SDA where the labeled examples are often too few to build a good target model. For example, when we only have a single labeled example for each class in the target training set, it is reasonable to assume that the source model trained from another domain can outperform a model trained only with the target training data on the target task.

Suppose the complex source model $f^*$ can generalize well on the target domain. The simple target model $f_t$ that has a similar training error to the source model $f^*$ would typically do better than the source model $f^*$ itself, as well as the target model trained only with hard labels on the target domain (according to the assumption of the source model). This indicates the knowledge can be transferred smoothly between models. Specifically, as it is suggested in [10], the transfer process can be achieved by letting the target model mimick the outputs of the source model (soft labels) on the training set without considering the true labels of the training examples. In another words, the source knowledge can be effectively transferred with the unlabeled data.

As the source models are trained from the source domains, it is necessary to weigh the source knowledge due to the domain shift[11] when we apply it to the target domain. In Eq. (3), we use the imitation parameter to control the relative importance between the soft labels and the hard labels, which in turn reflects the amount of the knowledge transferred from each of the source models. Specifically, the larger the imitation parameter, the more important the soft labels are and more knowledge can be transferred from the source domains. For example, in Figure 2, when we set $\lambda_2 = 0$, we actually ignore the knowledge from source domain 1. As a result, with the proper imitation parameter, GDSDA can effectively transfer the knowledge from each of the source models under the setting of SDA (for more details, please see the experiment section).

How the imitation parameter is chosen is essential for GDSDA. Many previous studies have addressed the importance of knowledge transfer control in domain adaptation[7, 8]. Without carefully controlling the amount of knowledge transferred from the source domain, the target model may suffer from degraded performance or even negative transfer[15]. However, in the previous studies, the imitation parameter can only be determined by either brute force search[13] or background knowledge[17] which scale poorly with the number of available source models and imitation parameters. In this paper, we propose our method, called GDSDA-SVM which can efficiently estimate the transfer parameter.

## 3 GDSDA-SVM

In this section, we propose our method GDSDA-SVM that uses SVM as the base classifier and can effectively estimate the imitation parameter.

### 3.1 Distillation with multiple sources

As we mentioned previously, the imitation parameter is a hyperparameter in GDSDA. A common method to estimate the hyperparameter is to use cross-validation. Here we show that it is possible to obtain the closed form cross-validation error[2] in GDSDA-SVM. As a result, GDSDA-SVM can estimate the imitation parameter effectively with the gradient descent method. Moreover, compared to [6] which also exploits the knowledge from the source model and uses the Maximum Mean Discrepancy to determine the weights of the source models, our method can determine the imitation parameter directly with the source model which is more effective with the relatively large source domains.

In our GDSDA-SVM, instead of using hinge loss, we use Mean Squared Error (MSE) as our loss function to train the GDSDA-SVM for the following two reasons: (1) Many recently studies [1, 14, 16, 18] show that MSE is an efficient measurement for the target model to distill the knowledge from the source model. (2) MSE can provide a closed form cross-validation error estimation, so we can estimate the imitation parameter more effectively.

Suppose we have $L$ examples $\{X, Y\}$ from $N$ classes in the target domain where $X \in R^{L \times d}, Y \in R^{L \times N}$. Meanwhile, there are $M-1$ the source (teacher) models providing the soft labels $Y^* = \{\mathbf{y}^*_{ijn} | i = 1, ..., M-1; j = 1, ..., L; n = 1, ..., N\}$ for each of the $L$ examples. For simplicity, we concatenate the hard label $Y$ and soft label $Y^*$ into a new label matrix $S$, where:

$$S = [Y; Y^*] = [S_1; ...; S_M]; S_i \in R^{L \times N}$$

To solve this $N$-class classification problem, we build $N$ binary SVMs. To learn the $n$th binary SVM, we have to solve the following optimization problem:

$$\min_{w_n} \quad \frac{1}{2}||\mathbf{w}_n||^2 + C\sum_j e_{jn}^2 \quad s.t. \quad e_{jn} = \sum_i \lambda_i s_{ijn} - \mathbf{w}_n \mathbf{x}_j \tag{4}$$

The Lagrangian of above optimization problem:

$$\mathcal{L} =\frac{1}{2}||\mathbf{w}_n||^2 + C\sum_j e_{jn}^2 + \sum_j \eta_{jn}\left(\sum_i \lambda_i s_{ijn} - \mathbf{w}_n\mathbf{x}_j - e_{jn}\right) \tag{5}$$

To find the saddle point,

$$\frac{\partial L}{\partial \mathbf{w}_n} = 0 \rightarrow \mathbf{w}_n = \sum_j \eta_{jn}\mathbf{x}_j; \quad \frac{\partial L}{\partial e_{jn}} = 0 \rightarrow \eta_{jn} = 2Ce_{jn} \tag{6}$$

For each example $\mathbf{x}_j$ and its constraint of label $s_{ijn}$, we have $e_{jn} + \mathbf{w}_n\mathbf{x}_j = \sum_i \lambda_i s_{ijn}$. Replacing $\mathbf{w}_n$ and $e_{jn}$, we have:

$$\mathbf{x}_j \sum_k \eta_{kn}\mathbf{x}_k + \frac{\eta_{jn}}{2C} = \sum_i \lambda_i s_{ijn} \tag{7}$$

Here we use $\varOmega$ to denote the matrix $\varOmega = [K + \frac{\mathbf{I}}{2C}]$ where $K$ is the linear kernel matrix $K = \{\mathbf{x}_i\mathbf{x}_j | i, j \in 1 \ldots L\}$. Let $\varOmega^{-1}$ be the inverse of matrix $\varOmega$ and $\varOmega_{jj}^{-1}$ be the $j$th diagonal element of $\varOmega^{-1}$. We have $\eta = \sum_i \lambda_i \varOmega^{-1} S_i$. For simplification, let $\eta'_i = \varOmega^{-1}S_i \in R^{L\times N}$. According to [2], the Leave-one-out estimation of the example $\mathbf{x}_j$ for the $n$th binary SVM can be written as:

$$\hat{y}_{jn} = \sum_i \lambda_i \left( s_{ijn} - \frac{{\eta'}_{ijn}}{\varOmega_{jj}^{-1}} \right) \tag{8}$$

### 3.2 Cross-entropy loss for imitation parameter estimation

From the previous part, we have already found an effective way to calculate the leave-one-out estimation of the target model. The optimal imitation parameters can be found by minimizing the leave-one-out cross-validation error on the target data by:

$$\min_{\lambda} \quad L_c\left(\lambda\right) = \frac{\beta}{2}\sum_i^M ||\lambda_i||^2 + \frac{1}{L}\sum_{j,n} \ell\left(y_{jn}, \hat{y}_{jn}\left(\lambda\right)\right) \quad s.t. \quad \sum \lambda_i = 1 \tag{9}$$

Here, as the estimation is based on the optimization result on the training set, we use the $\ell$-2 regularization term regularize $\lambda$ so that the target model can achieve good generalization performance even with a small training set. $\beta$ is used to balance the regularizer and loss function. Empirically we found that setting $\beta = 1$ can work well in most situations. For the loss function $\ell(\cdot, \cdot)$, we use the cross-entropy loss function.

$$\ell\left(y_{jn}, \hat{y}_{jn}\left(\lambda\right)\right) = -\, y_{jn}\log(P_{jn}) \quad for \quad P_{jn} = \frac{e^{\hat{y}_{jn}}}{\sum_h e^{\hat{y}_{jh}}} \tag{10}$$

Typically, cross-entropy loss pays less attention to a single incorrect prediction which reduces the affect of the outliers of the training data. Moreover, cross-entropy has its own advantage with our “fake label” strategy. As we mentioned

**Algorithm 1** GDSDA-SVM

```
Input: Input examples $X = \{\mathbf{x}_1, ..., \mathbf{x}_L\}$, number of classes $N$, number of sources $M$,
   3-D label matrix, $S = [Y_1, Y_2, ..., Y_M]$ with size $L \times M \times N$, temperature $T$
Output: Target model $f_t = Wx$
   Compute $\Omega = [K + \frac{\mathbf{I}}{2C}]$
   Compute imitation parameter $\lambda$ with Algorithm 2
   Generate the new label $Y_{new} = \sum_i \lambda_i Y_i$
   Compute $\eta = \Omega^{-1} Y_{new}$
   Compute $w_n = \sum_j \eta_{jn} x_j$
```

**Algorithm 2** $\lambda$ Optimization

```
Input: Input examples $X$, number of classes $N$, size of sources $M$, 3D label matrix $S$,
   optimization iteration $iter$, Kernel matrix $\Omega$
Output: Imitation parameter $\lambda = \{\lambda_1, ..., \lambda_M\}$
   Initialize $\lambda_i = \frac{1}{M}$,
   Let $S_n$ be the label matrix of $S$ for class $n$
   Compute $\eta'_i = \Omega^{-1} S_i$
   for $it \in \{1, ..., iter\}$ do
      Compute $\hat{y}_{jn}$ and $P_{jn}$ with (8) and (10)
      for each $\mathbf{x}_j$ in $X$ do
         $\Delta_\lambda = \Delta_\lambda + \sum_{j,n} \left( s_{ijn} - \frac{\eta'_{ijn}}{\Omega^{-1}_{jj}} \right) (P_{jn} - y_{jn})$
      end for
      $\Delta_\lambda = \Delta_\lambda / L$, $\lambda = \lambda - \frac{1}{it * \beta}(\Delta_\lambda + \beta\lambda)$ ,$\lambda = \lambda / \sum \lambda_i$
   end for
```

previously, we use 0s to encode the unlabeled examples. When we use cross-entropy loss, from Eq.(10) we can see that the loss of the unlabeled data is always 0. Therefore, cross-entropy loss can automatically ignore the loss of the unlabeled examples and reduce the effect of the noise introduced by our "fake label" strategy. The derivative of Eq.(10) can be calculated as:

$$\frac{\partial \ell(\lambda)}{\partial \lambda} = \sum_{j,n} \left( s_{ijn} - \frac{\eta'_{ijn}}{\Omega^{-1}_{jj}} \right) (P_{jn} - y_{jn}) \tag{11}$$

We describe GDSDA-SVM in Algorithm 1. As the optimization problem (9) is strongly convex, we can prove that Algorithm 2 can converge to the optimal $\lambda$ with the rate of $O(\log(t)/t)$ where $t$ is the optimization iteration (We are not able to show our proof here due to the space limit).

## 4 Experiments

In this section, we show the empirical performance of our algorithm GDSDA-SVM on the Office benchmark dataset. Specifically, we provide two scenarios: single source and multi-source transfer scenarios for GDSDA-SVM.

**Dataset:** There are 3 subsets in Office datasets, Webcam (795 examples), Amazon (2817 examples) and DSLR (498 examples), sharing 31 classes. We denote them as W, A and D respectively. In our experiments, we use DSLR and Webcam as the source domains and Amazon as the target domain. We use the features extracted from Alexnet [12] FC7 as the input features for both source and target domain. The source models are trained with multi-layer perception (MLP) on the whole source dataset.

### 4.1 Single Source for Office datasets

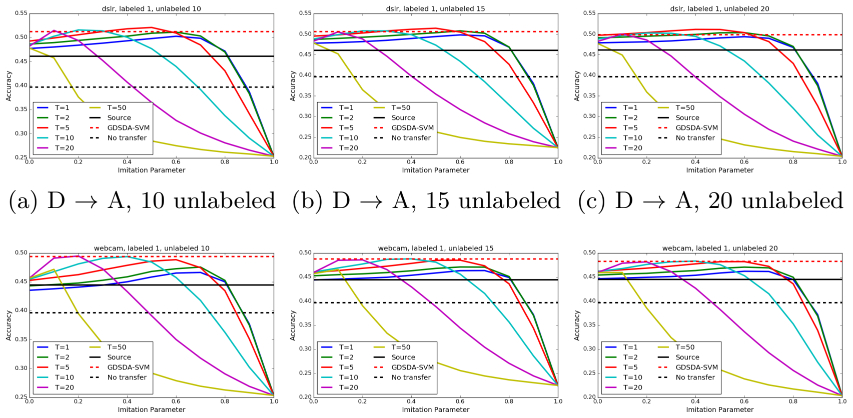
In this experiment, we compare our algorithm under the scenario where the source model is trained from a single source dataset. Specifically, we have two groups of experiments, transferring from Webcam to Amazon and from DSLR to Amazon. As we mentioned, there are significantly fewer labeled examples than unlabeled ones in real SDA applications. Therefore, in each group of experiment, there are only 31 labeled examples (1 per class) and some unlabeled examples (10, 15 and 20 per class) in the target domain used to train the target model.

To demonstrate the effectiveness of GDSDA-SVM, we show the performance of GDSDA using brute force to search the imitation parameter as the baseline. As there are two imitation parameters in this experiment, we use $\lambda_1$ and $1-\lambda_1$ to denote the imitation parameter for hard and soft label respectively. Specifically, we search the imitation parameter $\lambda_1$ in the range $[0, 0.1, ..., 1]$ with different temperature $T$. Meanwhile, we show the performance of the source model (denoted as "Source") and the performance of a target model (denoted as "No transfer" using LIBLINEAR[9]) trained with only labeled examples of the target domain[1] on the target task. We perform each experiment 10 times and report the average result. For GDSDA-SVM, as we are not able to tune the temperature $T$, we empirically set $T = 20$ and $\beta = 1$ for all experiments in this subsection. The experimental results are shown in Figure 3.

From the results of the brutal force search we can see that, the value of imitation parameter can greatly affect the performance of the target model. Also without using any label information of the target data for distillation, i.e. $\lambda_1 = 0$, as we expected, GDSDA can still slightly outperform the source model. This means GDSDA can effectively transfer the knowledge between different domains with the unlabeled data. As we increase the value of imitation parameter, i.e. introducing the hard labels from the target domain, the performance of GDSDA can be further improved. As we mentioned before, even though our "fake label" strategy would introduce extra noise, the noise can be limited by setting a proper value to imitation parameter and the target model can still achieve improved performance compared to the baselines.

Moreover, we can see that GDSDA-SVM can achieve competitive results compared to baselines using brutal force search in D→A experiments. In W→A

[1] We failed to achieve a better performance using semi-supervised learning method [4] on the target data as the no transfer baseline (may due to the size of the initial labeled examples).



(a) D → A, 10 unlabeled (b) D → A, 15 unlabeled (c) D → A, 20 unlabeled

(d) W → A, 10 unlabeled (e) W → A, 15 unlabeled (f) W → A, 20 unlabeled

Fig. 3: Experiment results on DSLR→Amazon and Webcam→Amazon when there are just one labeled examples per class. The results of DSLR→Amazon and Webcam→Amazon are shown in figure (a)-(c) and (d)-(e) respectively. GDSDA-SVM is trained with temperature $T = 20$. The X-axis denotes the imitation parameter of the hard label (i.e. $\lambda_1$ in Fig 2) and the corresponding imitation parameter of the soft label is set to $1 - \lambda_1$.

experiments, it achieves the best performances on all 3 different unlabeled sizes. This indicates that we can efficiently (about 6 times faster than the brutal force search) obtain a good target model with GDSDA-SVM.

### 4.2 Multi-Source for Office datasets

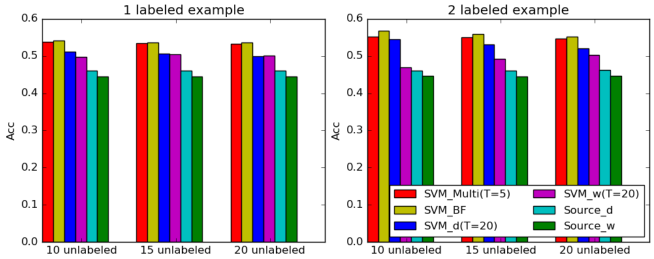


Fig. 4: D+W→A, Multi-source results comparison.

In this experiment, we show the performance of GDSDA-SVM under the multi-source SDA scenario. Specifically, we use Amazon as the target domain and the target domain can leverage the knowledge of two source models trained from Webcam and DSLR. We use the similar settings as our single source experiment and perform 2 groups of experiments using 1 labeled and 2 labeled examples per class respectively. We use temperature $T = 5$ and set and $\beta = 1$. The results of multi-source GDSDA-SVM are denoted as SVM_Multi. Here we also include two single source GDSDA-SVMs obtained from the experiments above (SVM_w and SVM_d trained using Webcam and DSLR as the source respectively) as the baselines. Moreover, we show the best performance of the brutal force search model (SVM_BF). For SVM_BF, we search temperature in range $T = [1, 2, 5, 10, 20, 50]$ and each imitation parameter in range $[0, 0.1, ..., 1]$. The experiment results are shown in Figure 4.

From the results, we can see that, given 2 source models, SVM_Multi can outperform any single source model trained with GDSDA. This indicates GDSDA-SVM can still exploit the knowledge even in the complex multi-source scenario. Even though SVM_Multi performs slightly worse than the best result found by brutal force search in some experiments, considering their time consumption (GDSDA-SVM is around 30 times faster than brutal force search), SVM_Multi still has its advantage in real applications.

## 5 Conclusion

In this paper, we propose a framework called Generalized Distillation Semi-supervised Domain Adaptation that can effectively leverage the knowledge from the source model using the unlabeled data to solve the SDA problem. To make GDSDA more efficient in real applications, we proposed a method called GDSDA-SVM and show that GDSDA-SVM can effectively estimate the imitation parameter for GDSDA. Experiment results show that GDSDA-SVM can effectively leverage the knowledge from one or more source models in real SDA applications.

## References

1. Ba, J., Caruana, R.: Do deep nets really need to be deep? In: Advances in neural information processing systems. pp. 2654–2662 (2014)
2. Cawley, G.C.: Leave-one-out cross-validation based model selection criteria for weighted ls-svms. In: Neural Networks, 2006. IJCNN'06. International Joint Conference on. pp. 1661–1668. IEEE (2006)
3. Daumé III, H., Kumar, A., Saha, A.: Frustratingly easy semi-supervised domain adaptation. In: Proceedings of the 2010 Workshop on Domain Adaptation for Natural Language Processing. pp. 53–59. Association for Computational Linguistics (2010)
4. Delalleau, O., Bengio, Y., Le Roux, N.: Efficient non-parametric function induction in semi-supervised learning. In: AISTATS. vol. 27, p. 100 (2005)
5. Donahue, J., Hoffman, J., Rodner, E., Saenko, K., Darrell, T.: Semi-supervised domain adaptation with instance constraints. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2013)

6. Duan, L., Tsang, I.W., Xu, D., Chua, T.S.: Domain adaptation from multiple sources via auxiliary classifiers. In: Proceedings of the 26th Annual International Conference on Machine Learning. pp. 289–296. ACM (2009)
7. Duan, L., Xu, D., Tsang, I.W.: Learning with augmented features for heterogeneous domain adaptation. In: Proceedings of the International Conference on Machine Learning. pp. 711–718. Omnipress, Edinburgh, Scotland (2012)
8. Duan, L., Xu, D., Tsang, I.W.H., Luo, J.: Visual event recognition in videos by learning from web data. vol. 34, pp. 1667–1680. IEEE (2012)
9. Fan, R.E., Chang, K.W., Hsieh, C.J., Wang, X.R., Lin, C.J.: Liblinear: A library for large linear classification. Journal of machine learning research 9(Aug), 1871–1874 (2008)
10. Hinton, G., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network. NIPS Deep Learning and Representation Learning Workshop (2014)
11. Karl, D., Bidigare, R., Letelier, R.: Long-term changes in plankton community structure and productivity in the north pacific subtropical gyre: The domain shift hypothesis. Deep Sea Research Part II: Topical Studies in Oceanography 48(8), 1449–1470 (2001)
12. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems. pp. 1106–1114 (2012)
13. Lopez-Paz, D., Schölkopf, B., Bottou, L., Vapnik, V.: Unifying distillation and privileged information. In: International Conference on Learning Representations (2016)
14. Luo, P., Zhu, Z., Liu, Z., Wang, X., Tang, X.: Face model compression by distilling knowledge from neurons. In: Thirtieth AAAI Conference on Artificial Intelligence (2016)
15. Pan, S.J., Yang, Q.: A survey on transfer learning. IEEE Transactions on Knowledge and Data Engineering 22(10), 1345–1359 (2010)
16. Romero, A., Ballas, N., Kahou, S.E., Chassang, A., Gatta, C., Bengio, Y.: Fitnets: Hints for thin deep nets. In: In Proceedings of International Conference on Learning Representations (2015)
17. Tzeng, E., Hoffman, J., Darrell, T., Saenko, K.: Simultaneous deep transfer across domains and tasks. In: The IEEE International Conference on Computer Vision (ICCV) (2015)
18. Urban, G., Geras, K.J., Kahou, S.E., Aslan, O., Wang, S., Caruana, R., rahman Mohamed, A., Philipose, M., Richardson, M.: Do deep convolutional nets really need to be deep (or even convolutional)? In: International Conference on Learning Representations (workshop track) (2016)
19. Vapnik, V., Izmailov, R.: Learning using privileged information: Similarity control and knowledge transfer. Journal of Machine Learning Research 16, 2023–2049 (2015)
20. Yao, T., Pan, Y., Ngo, C.W., Li, H., Mei, T.: Semi-supervised domain adaptation with subspace learning for visual recognition. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2142–2150 (2015)