Confidence Calibration for Domain Generalization under Covariate Shift

Yunye Gong[1], Xiao Lin[1], Yi Yao[1], Thomas G. Dietterich[2], Ajay Divakaran[1], and Melinda Gervasio[1]
[1]SRI International, [2]School of Electrical Engineering and Computer Science, Oregon State University
[1]first.last@sri.com, [2]tgd@oregonstate.edu

Abstract

Existing calibration algorithms address the problem of covariate shift via unsupervised domain adaptation. However, these methods suffer from the following limitations: 1) they require unlabeled data from the target domain, which may not be available at the stage of calibration in real-world applications and 2) their performance depends heavily on the disparity between the distributions of the source and target domains. To address these two limitations, we present novel calibration solutions via domain generalization. Our core idea is to leverage multiple calibration domains to reduce the effective distribution disparity between the target and calibration domains for improved calibration transfer without needing any data from the target domain. We provide theoretical justification and empirical experimental results to demonstrate the effectiveness of our proposed algorithms. Compared against state-of-the-art calibration methods designed for domain adaptation, we observe a decrease of 8.86 *percentage points in expected calibration error or, equivalently, an increase of* 35 *percentage points in improvement ratio for multi-class classification on the Office-Home dataset.*

1. Introduction

Deep neural networks (DNNs) have demonstrated high accuracy for tasks such as classification and detection given adequate data and supervision [34, 29]. However, for real-world applications, the ability to indicate how much users should trust model predictions can be even more crucial than just having an accurate but unpredictable model [2, 12, 28]. While discriminative networks provide confidence scores that can be used as a heuristic measure of the probability of correct classification, such scores are not guaranteed to match the true probabilities of correct classification [9]. A recent development, referred to as model calibration, addresses this problem directly [24, 9].

A classifier is calibrated with respect to a distribution (or a dataset sampled from that distribution) if its predicted probability of being correct matches its true probability. If the distribution changes, calibration is usually lost, and this has been demonstrated empirically [20]. Recent work has begun to investigate the problem of calibration in the context of transfer learning, specifically in an unsupervised domain adaptation scenario under the assumption of covariate shift [22, 32, 20]. However, these methods need at least unlabeled data from the target domain, which may not be available at the stages of training and calibration in real-world applications. Furthermore, as these methods are designed to handle a single source domain, there may be an undesired disparity between selected source and target domains either due to the limited availability of sources or uncertainties (e.g., extreme weather/unexplored terrain) in the target. Fig. 1(a) notionally depicts such a scenario. This disparity results in a large variance of the density ratio defined as P^T/P^S, which significantly degrades the accuracy of calibration [20, 4].

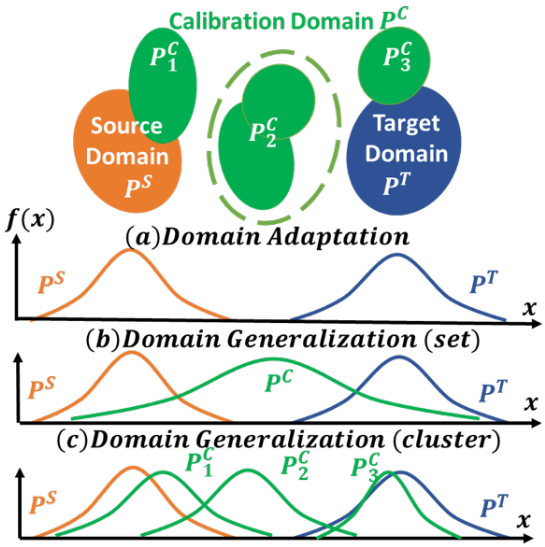

Figure 1: Calibration for domain adaptation with a single source domain may suffer from a large variance of the density ratio (i.e., P^T/P^S) caused by disjoint P^S and P^T and, therefore, a large calibration error as shown in (a). Our proposed calibration algorithms for domain generalization leverage multiple calibration domains to reduce the disparity between P^C and P^T for a decreased variance of the density ratio P^T/P^C and, in turn, improved calibration performance, as shown in (b) and (c).

To tackle the aforementioned limitations of calibration transfer via domain adaptation, we focus instead on calibration for domain generalization. Our key idea is to use multiple source domains and cluster their labeled data into groups. We then fit post-hoc calibration parameters to each group. The class probability of a test example is calibrated using the calibration parameters of the group that is nearest (in Euclidean distance) to the test example. By using many calibration domains, we increase the likelihood of overlap in the distributions, which in theory will improve the effectiveness of cross-domain calibration [20, 4]. By learning calibration parameters separately for each group, we increase the likelihood that each test query will be adjusted by the best calibration correction.

We study two calibration methods within each cluster. Both are based on temperature scaling [9]. The first computes a fixed scaling temperature for each group (Fig. 1(c)). The second fits a regression model to these fixed temperatures to enable extrapolating the temperature to points outside the clusters. We compare these two methods against a baseline that computes one temperature for scaling based on the union of all the calibration data (Fig. 1(b)). We refer to our methods as *cluster-level* and the baseline as *set-level*. Notably, while Fig. 1 depicts a notional scenario where the source and target domains have no overlap and the calibration domains bridge the gap, our methods will also work well in cases where the source is closer to the target, as long as at least one of the calibration domains is also close to the target.

Our major contributions include the following:

1) We propose novel solutions to calibrate a classification model for domain generalization. Our proposed algorithms are trained to produce accurate confidence predictions without needing any data from the target domain.

2) We provide theoretical error bounds for our proposed calibration methods and demonstrate the advantage of our methods in maximizing the overlap between the supports of the target and calibration distributions, a critical factor that determines the generalization performance of calibration.

3) We justify the proposed algorithms with experimental results on real-world data. A decrease of 8.86 percentage points in expected calibration error or, equivalently, an increase of 35 percentage points in improvement ratio is achieved on the Office-Home dataset [30] compared against state-of-the-art (SOTA) calibration methods designed for domain adaptation.

2. Related Work

Calibration. Existing calibration methods provide post-hoc correction for classification models so that their confidence scores better match the true probabilities of correct classification [35, 36, 19, 24, 9]. Among those, Platt Scaling [24] provides a parametric solution for binary classification. It learns a logistic regression model with two scalar parameters that maps the initial predicted probabilities to calibrated probabilities. It is trained on a holdout validation set with respect to the negative log-likelihood (NLL) loss. Matrix scaling and Vector scaling [9] are two extensions of Platt scaling to multi-class classification problems, where a linear transformation is applied to the logit vectors before the softmax operation. Given a classification model, the additional linear layer is finetuned on the validation set with respect to NLL. In this case, classification accuracy is affected by calibration. Temperature scaling is another special case of Platt scaling. Here, a scalar temperature parameter is applied to scale the logit vectors without changing class predictions. The temperature is optimized on the validation set with respect to NLL and can be interpreted as the solution of a constrained entropy maximization. Alexandari et al. [1] investigated variants of vector and temperature scaling including no-bias vector scaling and bias-corrected temperature scaling in the context of domain adaptation under label shift. Our proposed algorithms are all based on temperature scaling [9].

Domain generalization vs. domain adaptation. Transfer learning is generally challenging for deep learning, as models trained on one domain (source) can suffer performance drops when evaluated on test data from a different domain (target). One type of transfer learning is domain adaptation [31, 5], which seeks to improve target domain performance by leveraging data from both source and target domains. Specifically, unsupervised domain adaptation (UDA) [33] addresses the problem when only unlabeled data is available from the target domain. Multiple UDA methods have been developed based on strategies such as learning domain invariant features [18, 13, 27, 21] and learning mappings between domains [11, 26].

An alternative to domain adaptation is domain generalization, which aims at robust transfer without any data, either labeled or unlabeled, from the target domain at the training stage, by leveraging information from multiple related source domains. Ghifary et al. [8] propose a multi-task autoencoder (MTAE) that jointly reconstructs analogous views of a source image over multiple domains to acquire robust features for generalization in the context of object recognition. Li et al. [15] minimize maximum mean discrepancy (MMD) to align distributions from different domains. Several recent studies adopt model-agnostic meta-learning (MAML) originally proposed for few-shot learning [7]. For instance, Li et al. [14] propose meta-learning for domain generalization (MLDG) using model-agnostic optimization across domains instead of across tasks. Balaji et al. [3] apply meta-learning to learn a generalizable regularizer for the classification layers instead of the full network. Dou et al. [6] introduce complementary losses to encourage class alignment across domains and improve compactness of class-specific clusters.

All of this work is devoted to learning models to improve generalization with respect to *classification accuracy*. In comparison, our proposed methods focus on calibrating a classifier using multiple related domains to improve confidence scores so that they are better calibrated in the unseen target domain (i.e., *fidelity of confidence scores*).

Calibration for domain adaptation. Several recent papers investigate the problem of calibration in the context of transfer learning, specifically in an unsupervised domain adaptation scenario under the assumption of covariate shift [22, 32, 20]. These studies adopt similar frameworks based on estimating importance weights that describe the density ratio between the source and target distributions. The weights are estimated by learning a discriminator distinguishing source samples from target samples. The calibration loss in the target domain can then be formulated as a weighted version of the original loss in the source domain. Calibration loss is quantified using Brier Score [22], NLL [20], and expected calibration error (ECE) [31].

While these recent efforts are the most relevant to our work, we address an arguably more challenging problem. Instead of calibrating classifiers using unlabeled target data, we calibrate classifiers without any data, either labeled or unlabeled, from the target domain.

3. Background

Calibration. Let x, y denote the data and label drawn from a joint distribution $P(x, y)$. Let $\phi(\cdot)$ be a learned multi(K)-class classification model that projects each sample x_i to a logit vector z_i with K dimensions. The class prediction $\hat{y}_i$ and confidence prediction $\hat{p}_i$ can be expressed as

$$\hat{p}_i = \max_k \sigma(z_i)^{(k)} \quad \hat{y}_i = \operatorname*{argmax}_k \sigma(z_i)^{(k)}, \tag{1}$$

where σ denotes the the softmax function:

$$\sigma(z^{(k)}) = \frac{\exp(z^{(k)})}{\sum_{j=1}^{K} \exp(z^{(j)})}. \tag{2}$$

Miscalibration refers to the problem where confidence predictions $\hat{p}_i$ do not match the true probabilities of correct classification. The goal of calibration is to adjust the confidence so that $\mathbb{P}(\hat{y} = y|\hat{p} = p) = p, \forall p \in [0, 1]$ [9].

Temperature scaling [9]. A scalar $t > 0$ is applied to adjust the confidence prediction:

$$\hat{p}_i = \max_k \sigma(z_i/t)^{(k)}. \tag{3}$$

The value of t is optimized over a small validation set with respect to the same NLL loss used in classification training:

$$t^* = \operatorname*{argmin}_t \mathbb{E}_{x,y\sim P(x,y)} \mathcal{L}(\phi(x), y, t), \tag{4}$$

where $\mathcal{L}$ denotes the NLL loss. Note that temperature scaling does not affect the overall classification accuracy, as the same t is applied to all classes.

Expected calibration error (ECE) [19]. To measure calibration accuracy, we employ the ECE metric. Given a set of class predictions and corresponding confidence predictions, ECE is computed by grouping test samples into M bins of equal width based on confidence values. Let B_m denote the set of indices where $B_m = \{i|\hat{p}_i \in (\frac{m-1}{M}, \frac{m}{M}]\}$. The classification accuracy and average confidence for each bin are computed as

$$acc(B_m) = \frac{1}{|B_m|} \sum_{i\in B_m} 1(\hat{y}_i = y_i) \tag{5}$$

$$conf(B_m) = \frac{1}{|B_m|} \sum_{i\in B_m} \hat{p}_i. \tag{6}$$

A well-calibrated model should reduce the mismatch between classification accuracy and confidence prediction. Therefore, ECE is computed as the weighted sum of the mismatch over bins:

$$ECE = \sum_{m=1}^{M} \frac{|B_m|}{N} |acc(B_m) - conf(B_m)|, \tag{7}$$

where N is the total number of the samples. In the case of $M = 1$, ECE reduces to the absolute error between the average confidence prediction and the classification accuracy over the entire test set.

Calibration for domain adaptation. Let $P^S(x, y)$ and $P^T(x, y)$ denote the source and target distributions, respectively. Covariate shift between distributions refers to the assumption that $P^T(x) \neq P^S(x)$ while $P^T(y|x) = P^S(y|x)$. Following a formulation similar to those in domain adaptation approaches (Theorem 4.1 in [20]), the desired calibration loss can be expressed as

$$\begin{aligned}
&\mathbb{E}_{x,y\sim P^T(x,y)} \mathcal{L}(\phi(x), y, t) \\
&= \int_x \int_y \mathcal{L}(\phi(x), y, t) P^T(x, y)\, dx\, dy \\
&= \int_x \int_y \mathcal{L}(\phi(x), y, t) \frac{P^T(x)P^T(y|x)}{P^S(x)P^S(y|x)} P^S(x, y)\, dx\, dy \\
&= \mathbb{E}_{x,y\sim P^S(x,y)} w_S(x) \mathcal{L}(\phi(x), y, t)
\end{aligned} \tag{8}$$

for $\{x|P^T(x) > 0\} \subseteq \{x|P^S(x) > 0\}$, where the importance weight $w_S(x) = \frac{P^T(x)}{P^S(x)}$ is the density ratio.

4. Method

In contrast to confidence calibration under the single-source single-target unsupervised domain adaptation scenario, we consider domain generalization with multiple source domains that are related but different from the holdout target domain. In this case, we use S (source) to denote the group of domains used for training the classifier, C (calibration) to denote the group of domains used for calibrating the given classifier, and T (target) to denote the group of holdout test domains that are completely unseen at both the

classifier training and calibration stages. Accordingly, the desired calibration loss is given by

$$\begin{aligned}
&\mathbb{E}_{x,y\sim P^T(x,y)}\mathcal{L}(\phi(x),y,t)\\
&=\int_x\int_y \mathcal{L}(\phi(x),y,t)P^T(x,y)\,dx\,dy\\
&=\int_x\int_y \mathcal{L}(\phi(x),y,t)\frac{P^T(x)P^T(y|x)}{P^C(x)P^C(y|x)}P^C(x,y)\,dx\,dy\\
&=\mathbb{E}_{x,y\sim P^C(x,y)}w_C(x)\mathcal{L}(\phi(x),y,t), \qquad (9)
\end{aligned}$$

for $\{x|P^T(x)>0\}\subseteq\{x|P^C(x>0\}$, where $w_C(x)=\frac{P^T(x)}{P^C(x)}$ denotes the density ratio between the target and calibration domains.

Following [20, 32], we derive the gap between the calibration loss and the oracle loss using the true target distribution $P^T(x,y)$ to show that the variance of the density ratio $w_C(x)$ or equivalently the divergence between $P^T(x)$ and $P^C(x)$ is critical for calibration transfer. The same observations apply to calibration for domain adaptation, where the variance of $w_S(x)$ or the divergence between $P^T(x)$ and $P^S(x)$ is critical. For simplicity, we use $w(x)$ to denote $w_C(x)$ or $w_S(x)$ when these two are interchangeable. The gap is given by

$$\begin{aligned}
&\left|\mathbb{E}_{P^C(x,y)}\mathcal{L}(\phi(x),y,t)-\mathbb{E}_{P^T(x,y)}\mathcal{L}(\phi(x),y,t)\right|\\
=&\left|\int_x\int_y(1-w_C(x))\mathcal{L}(\phi(x),y,t)P^C(x,y)\,dx\,dy\right|\\
=&\left|\mathbb{E}_{P^C(x,y)}\left[(1-w_C(x))\mathcal{L}(\phi(x),y,t)\right]\right| \qquad (10)\\
\leq&\sqrt{\mathbb{E}_{P^C(x)}\left[(1-w_C(x))^2\right]\mathbb{E}_{P^C(x,y)}\left[\mathcal{L}(\phi(x),y,t)^2\right]} \qquad (11)\\
\leq&\frac{1}{2}(\mathbb{E}_{P^C(x)}\left[(1-w_C(x))^2\right]+\mathbb{E}_{P^C(x,y)}\left[\mathcal{L}(\phi(x),y,t)^2\right]), \qquad (12)
\end{aligned}$$

where the inequality in Eq. 11 follows from the Cauchy-Schwarz Inequality and the inequality in Eq. 12 follows from the inequality of arithmetic and geometric means. This formulation can also be interpreted as the bound of the bias of the estimated loss given

$$\hat{w_C}(x)=\mathbb{E}_{P^C}[w_C(x)]=1 \qquad (13)$$

as an estimator of $w_C(x)$. We exploit this property to design calibration algorithms bypassing the direct computation of $w_C(x)$ due to the lack of target data at the classifier training and calibration stages.

Given a fixed classification model ϕ, the second term in Eq. 12 is computed based on calibration data. Therefore, only the first term is affected by the shift between the calibration and target domains. Following Cortes et al., [4], the first term in Eq. 12 can be expressed as

$$\begin{aligned}
\mathbb{E}_{P^C}\left[(w_C(x)-1)^2\right]&=\mathbb{E}_{P^C}\left[(w_C(x)-\mathbb{E}_{P^C}[w_C(x)])^2\right]\\
&=\text{Var}(w_C(x))\\
&=d_2\big(P^T(x)||P^C(x)\big)-1, \qquad (14)
\end{aligned}$$

where $d_\alpha(P||Q)=\left[\sum_x\frac{P^\alpha(x)}{Q^{\alpha-1}(x)}\right]^{\frac{1}{\alpha-1}}$ with $\alpha>0$ is the exponential in base 2 of the Renyi-divergence [25] between distributions P and Q.

The calibration errors are dominated by the variance of $w_C(x)$ given by $\text{Var}(w_C(x))=d_2\big(P^T(x)||P^C(x)\big)-1$. Similarly, for domain adaptation, we have $\text{Var}(w_S(x))=d_2\big(P^T(x)||P^S(x)\big)-1$. Intuitively we seek to reduce the variance of $w(x)$, or equivalently, the divergence between target and source distributions for domain adaptation or target and calibration distributions for domain generalization. If there exist large shifts between the source and target domains, the density ratio is unbounded over $\{x|P^T(x)\neq 0, P^S(x)=0\}$ leading to large variance of the density ratio. Let $\{x|P(x)>0\}$ be the support of a distribution $P(x)$. Reducing the variance of $w(x)$ requires larger overlap between the supports of $P^T(x)$ and $P^S(x)$ for domain adaptation or between $P^T(x)$ and $P^C(x)$ for domain generalization.

For domain adaptation with fixed source and target domains [22, 32, 20], there is limited room for calibration to adjust such overlap. In contrast, for domain generalization, we can manipulate the learning of calibration models over different calibration domains to maximize such overlap. Motivated by these theoretical advantages of using multiple sources, we propose calibration algorithms for domain generalization (Fig. 2) including set- (Sec. 4.1) and cluster-level approaches (Sec. 4.2). Note that while maximizing the overlap between $P^T(x)$ and $P^S(x)$ (or $P^C(x)$) can be achieved via feature alignment, commonly used for domain adaptation, that is out of the scope of this paper.

4.1. Set-level calibration

Set-level calibration is our baseline method. We learn the temperature t using multiple calibration domains C. Only a small set of data from C is required, and temperature scaling is applied post-hoc to the classification model ϕ trained on the source domain S. The temperature is learned via

$$t^*=\underset{t}{\text{argmin}}\,\mathbb{E}_{x,y\sim P^C(x,y)}\mathcal{L}(\phi(x),y,t), \qquad (15)$$

where $P^C(x,y)$ is the joint distribution over all calibration domains, meaning that each calibration domain is treated equally. We refer to this algorithm as set-level calibration (Fig. 2 (a)), since a single temperature is learned with respect to all calibration data and applied to all test data.

By leveraging multiple related domains, P^C is more likely to be better aligned with P^T especially for scenarios where P^S and P^T are distant as shown in Fig. 1(b). This leads to a density ratio $w_C(x)$ with fewer unbounded values and, thus, a smaller variance. As a result, better calibration transfer can be achieved.

4.2. Cluster-level calibration

Learning the temperature at the set level assumes the same optimal scaling for all samples. Considering cali-

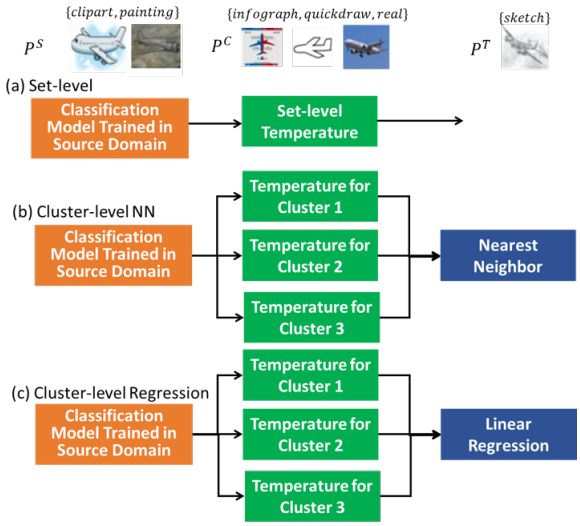

Figure 2: Block diagram of proposed calibration algorithms for domain generalization: (a) set-level, (b) cluster-level NN, and (c) cluster-level regression.

bration data from multiple domains, it would be natural to relax this constraint so that different samples can have different optimal scaling. Several classic calibration algorithms [19, 35, 36] perform calibration based on binning the data according to their uncalibrated confidence scores. Motivated by these successes, we propose to group the calibration samples by the similarity of image features to correlate the optimal temperature scaling to feature distributions. Then, during testing we can predict the most appropriate temperature given only a single test image feature from an unknown distribution. Specifically, we perform K-means clustering [16] on the image features (at the penultimate layer of ResNet18 feature extractors) using calibration data. The centroids of clusters are determined by minimizing the within-cluster sum of squares. For each cluster, we perform a standard temperature scaling. At the testing stage, we exploit two alternative methods to determine the most appropriate temperature for each test sample.

Nearest neighbor (NN). In the first method, we simply assign a given sample from the test domain to the cluster whose centroid is the closest to the sample feature in Euclidean distance. Then we apply the corresponding optimal temperature for that cluster to calibrate the test sample (Fig. 2 (b)). Intuitively, this procedure facilitates the alignment of the calibration and target distributions at the cluster level. Let N denote the number of clusters, P_j^C denote the distribution of calibration domain features that are grouped into cluster j, and P_j^T denote the unknown distribution of all test domain features that are assigned to the cluster j. The optimal temperature for each cluster j is learned with respect to the following objective:

$$t_j^* = \underset{t}{\operatorname{argmin}}\, \mathbb{E}_{P_j^C}\big[\mathcal{L}(\phi(x), y, t)\big]. \tag{16}$$

Following a similar formulation in Eq. 9-Eq. 14, the oracle objective for test samples from P_j^T can be expressed as

$$\mathbb{E}_{P_j^T}\big[\mathcal{L}(\phi(x), y, t)\big] = \mathbb{E}_{P_j^C}\big[w_{C,j}(x, y)\mathcal{L}(\phi(x), y, t)\big], \tag{17}$$

for $\{x|P_j^T(x) > 0\} \subseteq \{x|P_j^C(x) > 0\}$. Here, $w_{C,j}(x) = \frac{P_j^T(x)}{P_j^C(x)}$, and the generalization of learned temperatures depends on the divergence $d_2\big(P_j^T(x)||P_j^C(x)\big)$. Since P_j^C is chosen as the closest cluster to the test samples, chances are high that P_j^T and P_j^C are better aligned than the set-level distributions P^T and P^C, resulting in a density ratio $w_{C,j}$ with a smaller variance (e.g., Fig. 1(c) with $j = 3$). Therefore, cluster-level calibration holds the promise of further improved calibration transfer.

Regression-based prediction. Nearest neighbor can be considered as a special case of linear regression where the j^{th} weight is set to 1 while the others are set to 0. We further investigate cluster-level calibration using learned weights as a more generalized scheme (Fig. 2 (c)). Specifically, we train a regression model that maps the mean feature of each cluster to its corresponding cluster-level optimal temperature. We can thus apply the learned regression model to any test feature to predict a proper temperature specific to the test instance. Let R_θ denote the regression model R parameterized by θ, R_θ is determined by minimizing the following mean-squared error:

$$\theta^* = \underset{\theta}{\operatorname{argmin}} \frac{1}{N}\sum_{j=1}^{N}\big(R_\theta(\mathbb{E}_{P_j^C}(x)) - t_j^*\big)^2. \tag{18}$$

Essentially, we learn a function capturing the underlying mapping from features to the proper temperature for calibration and transfer it to the unknown target domain, instead of directly transferring the temperatures learned on calibration domains.

5. Experiments

5.1. Datasets

Office-Home [30] contains images of 65 classes across four domains corresponding to different rendering styles: Clipart (4365 images), Art (2427 images), Product (4439 images), and Real (4357 images). We split these four domains into three subsets: one domain as the source for training the classifier, two domains for post-hoc calibration of the classifier, and one holdout domain as the target for evaluating the calibrated classifier. We perform experiments for all 12 possible splits of domains, including the combinations where the source is relatively similar to the target judging from the image realism of the domains (e.g., Art as source, Clipart as target, Product and Real as calibration) and combinations where the source is relatively distant to the target (e.g., Clipart as source, Real as target, Art and Product as calibration). We randomly divide data from each domain into a *Large* subset (80%) and a *Small* subset (20%). We use the *Large* subset for either training the classifier or evaluating the calibration performance and we use the *Small* subset

for either tuning the hyperparameters of classification training or calibrating the classifier. For each source domain, we train a ResNet18 [10] initialized with parameters pretrained on ILSVRC-1000. We extract image features at the penultimate layer of the network for clustering. For each domain split, we perform 1000 evaluations each with 1500 randomly selected samples from the target domain to estimate confidence intervals.

DomainNet [23] contains images of 345 classes across six domains corresponding to different rendering styles: Quickdraw (172500 images), Infograph (51605 images), Sketch (69128 images), Clipart (48129 images), Painting (72266 images), and Real (172947 images). We split these six domains into three subsets: two domains as the source, three domains for the calibration, and one holdout domain as the target. We perform experiments for all 60 possible splits of domains, including the combinations where the source is relatively similar to the target judging from the image realism of the domains (e.g., Quickdraw and Sketch as source, Infograph as target, Clipart, Painting and Real as calibration) and combinations where the source is relatively distant to the target (e.g., Quickdraw and Sketch as source, Real as target, Clipart, Painting and Infograph as calibration). Following the train/test splits from [23], we use the train split for training the classifier, a *Small* subset (10%) of the test split for calibration, and a *Large* subset (90%) of the test split for evaluation. We use a ResNet18 pretrained on ILSVRC-1000 as the feature extractor and train an MLP classifier. For each domain split, we conduct 1000 evaluations each with 10000 randomly selected samples from the target domain to estimate confidence intervals.

5.2. Experimental settings

Source-only calibration. We split the source domain as described in Sec 5.1, using the *Large* subset to learn the classifier and the *Small* subset to calibrate it. We directly evaluate the calibrated model on the holdout target domain. This experiment serves as a reference without calibration transfer.

Target-only (oracle) calibration. Given a classifier trained on the source domain, we calibrate it using the *Small* subset of the target domain data and evaluate the calibration on the *Large* subset. This is an oracle experiment, since it uses the ground-truth labels and data from the target domain which are not available for the domain generalization setting. Consequently, this experiment sets the target performance for calibration transfer.

Cross-domain calibration. Given a classifier trained on the source domain, we calibrate it via our algorithms described in Sec. 4 using the *Small* subsets from the calibration domains. We also average the logit outputs from these three methods as an additional ensemble-based approach. We evaluate the calibrated model on the *Large* subset of the target domain which is unseen at both the classifier training and calibration stages. For cluster-level calibration, we use eight clusters for Office-Home and nine clusters for DomainNet. Our experiments suggest that the numbers of clusters have minor effects on calibration performance. For the cluster-level regression method, we choose a linear regression model considering the high dimensional feature space and availability of samples from calibration domains.

5.3. Results and Discussions

Experimental results are summarized in Tables 1-3. We report the mean and standard deviation of ECE scores (%). Each column of Table 1 (for Office-Home) and Table 3 (for DomainNet) lists the ECE scores for a specific target domain averaged over different domain splits across source and calibration domains. The last column lists the ECE scores averaged over different target domains. Table 2 lists the performance on Office-Home for each domain split (i.e., one combination of the source and target domains). ECE scores for the domain adaptation baselines [20, 32] are based on the reported results (using the CDAN [17] method) from the original papers.

The standard deviation of ECE, σ_{ECE}, is mainly affected by two factors: sample variations within a domain split and domain variations. Each σ_{ECE} in Table 2 indicates the effect of sample variations within a fixed domain split. For all of the tested domain splits, we observe that $\sigma_{ECE} < 1\%$. Results for each domain split on DomainNet are included in the supplementary materials to save space, where we observe that $\sigma_{ECE} < 0.4\%$. We see a much larger σ_{ECE} across domain splits in Table 1 and Table 3, which measures the combined variations from samples and domains. It is clear that domain variations dominate the variance of the ECE scores.

To evaluate the effectiveness of calibration transfer, we define improvement ratio (IR) as

$$IR = \frac{ECE_S - ECE}{ECE_S - ECE_T}, \tag{19}$$

where ECE_S and ECE_T refer to the averaged ECE scores obtained via source-only calibration and target-only calibration, respectively. Without calibration transfer, we start from the performance of source-only calibration. Using calibration transfer, we want to approach the performance of target-only calibration. The IR metric evaluates where the performance of a calibration transfer method is located with respect to these starting and ending points. Table 5 lists the IRs evaluated on Office-Home and DomainNet.[1]

Comparison against source-only calibration. Without calibration transfer, directly using the temperature learned from the source distribution fails for both datasets. It can even potentially lead to larger errors in comparison to uncalibrated models (comparing the first and second rows in Table 1). Compared to uncalibrated and source-only, our algo-

[1] Additional results including alternative metrics and confidence intervals can be found in the supplementary materials.

Methods	Clipart	Art	Product	Real	**Average**
Uncalibrated	14.74±2.23	9.31±2.33	5.66±1.23	4.92±0.94	8.66±4.38
Source-only	18.02±1.81	10.79±4.46	6.90±2.78	6.09±1.05	10.45±5.50
Target-only (oracle)	4.10±0.72	3.56±0.59	4.10±1.10	4.01±0.68	3.94±0.83
TransCal [32]	21.37	20.60	11.37	8.23	15.39
WTS [20]	18.97	8.60	3.90	7.63	9.78
Set-level	9.96±1.78	4.21±0.59	5.57±2.12	7.47±3.00	6.80±2.99
Cluster-level NN	11.43±1.83	4.90±1.65	5.10±1.36	6.49±2.06	6.98±3.17
Cluster-level Regression	12.00±1.41	4.61±0.80	5.11±1.12	6.03±1.82	6.94±3.26
Ensemble	11.31±1.75	4.14±0.94	4.69±1.23	5.98±1.82	6.53±3.20

Table 1: Calibration performance (ECE %) on Office-Home averaged by target domain.

	Uncalibrated	Source-only	Target-only (oracle)	TransCal [32]	WTS [20]	Set-level	Cluster -level NN	Cluster-level Regression	**Ensemble**
A→ C	11.84±0.76	16.95±0.77	4.30±0.70	22.9	12.8	10.98±0.76	12.54±0.81	13.10±0.81	12.53±0.76
P→ C	15.81±0.82	20.30±0.84	4.24±0.68	40.4	26.8	7.71±0.76	9.12±0.84	10.43±0.84	9.10±0.79
R→ C	16.58±0.86	16.82±0.86	3.76±0.66	4.5	17.3	11.19±0.84	12.64±0.84	12.48±0.83	12.28±0.84
C→ A	7.61±0.53	7.37±0.52	4.08±0.48	21.7	6.9	4.50±0.45	4.72±0.48	4.48±0.48	5.02±0.47
P→ A	12.52±0.53	17.05±0.53	3.43±0.46	18.5	8.5	3.68±0.46	6.92±0.53	5.46±0.50	4.36±0.47
R→ A	7.80±0.48	7.96±0.48	3.16±0.41	21.6	10.4	4.44±0.46	3.05±0.43	3.90±0.42	3.04±0.39
C→ P	5.78±0.78	5.57±0.76	3.32±0.61	14	6.4	3.22±0.59	3.50±0.59	3.95±0.63	3.25±0.57
A→ P	6.81±0.77	10.64±0.82	5.35±0.71	9.3	1.5	5.39±0.78	6.26±0.79	6.08±0.76	5.56±0.75
R→ P	4.38±0.62	4.50±0.64	3.63±0.59	15.6	3.8	8.1±0.74	5.54±0.68	5.30±0.67	5.27±0.68
C→ R	5.86±0.69	5.75±0.68	3.67±0.62	6.4	5.7	3.73±0.63	4.00±0.63	3.95±0.64	3.82±0.66
A→ R	4.31±0.63	5.34±0.70	4.19±0.62	5.1	6.4	7.86±0.79	6.75±0.73	6.08±0.71	6.25±0.74
P→ R	4.59±0.64	7.18±0.78	4.18±0.67	13.9	10.8	10.83±0.77	8.72±0.76	8.05±0.78	7.88±0.75

Table 2: Calibration performance (ECE %) on Office-Home.

Methods	Quickdraw	Infograph	Sketch	Clipart	Painting	Real	**Average**
Uncalibrated	21.42±3.33	23.99±3.94	16.65±2.06	11.63±2.68	16.43±3.91	11.82±2.09	16.99±5.51
Source-only	20.24±2.57	24.58±4.52	17.06±2.18	11.67±3.26	16.83±3.71	11.24±3.21	16.93±5.72
Target-only (oracle)	0.68±0.28	1.81±0.33	2.01±0.86	2.51±0.58	2.68±0.72	2.33±0.37	2.00±0.87
Set-level	10.01±2.25	7.39±3.30	3.52±2.58	6.83±4.68	5.87±4.43	13.38±4.37	7.83±4.87
Cluster-level NN	8.10±1.95	7.91±2.24	3.04±1.31	6.06±2.74	4.17±1.77	9.35±2.65	6.44±3.12
Cluster-level Regr.	11.72±5.81	11.93±6.74	7.29±5.19	8.49±3.74	7.08±5.51	9.55±5.39	9.34±5.80
Ensemble	9.81±2.54	9.51±2.99	3.12±1.59	5.71±2.50	3.66±2.54	8.15±2.74	6.66±3.67

Table 3: Calibration performance (ECE %) on DomainNet averaged for each target domain.

$\times\sigma_{ECE}$	< -3	$(-3,-2)$	$(-2,0)$	$(0,2)$	$(2,3)$	> 3
TransCal [32]	1	0	0	1	0	10
WTS [20]	1	1	0	1	1	8

Table 4: Number of Office-home domain splits with reduction in ECE achieved by our methods, with 95% confidence ($2\sigma_{ECE}$) and 99% confidence ($3\sigma_{ECE}$).

	Office-Home	DomainNet
TransCal [32]	0.25	-
WTS [20]	-0.03	-
Set-level	0.56	0.61
Cluster-level NN	0.53	0.70
Cluster-level Regression	0.54	0.51
Ensemble	0.60	0.69

Table 5: Improvement ratio based on averaged ECE scores.

rithms can improve the generalization performance of calibration and achieve lower ECEs for both Office-Home (a reduction of 3.92 percentage points in ECE from the source-only calibration in Table 1) and DomainNet (a reduction of 10.27 percentage points in ECE from the source-only calibration in Table 3).

Comparison against domain adaptation methods. For Office-Home, we compare the performance of our methods against two recent calibration transfer methods designed for domain adaptation: TransCal [32] and Weighted Temperature Scaling (WTS) [20]. On average, we achieve a reduction of 8.86 percentage points in ECE against TransCal and a reduction of 3.25 percentage points in ECE against WTS (Table 1). While σ_{ECE} across domain splits is relatively high, the reduction in ECE achieved by our algorithms is still significant, considering that $\sigma_{ECE} < 1\%$ for each domain split. Comparing to TransCal, the reduction in ECE is larger than $3\sigma_{ECE}$ for 10 out of 12 domain splits (i.e., with a confidence $> 99\%$ for 83% tested cases in Table 4). For WTS, the reduction in ECE is larger than $2\sigma_{ECE}$ for 9 out of the 12 domain splits (i.e., with a confidence $> 95\%$ for 75% tested cases in Table 4).

As for the IRs in Table 5, TransCal is able to compensate for about one fourth (IR=25%) of the difference between the ECE scores of the target-only and source-only calibration, whereas WTS turns out to perform only comparably with the source-only calibration (IR=−3%)[2]. As expected, our methods yield a higher IR (IR=60%, an improvement of 35 percentage points over TransCal), compensating for slightly more than half of the differences between the ECE scores of the source-only and the target-only calibration.

Comparison among our methods. As expected, our ensemble method performs the best (on Office-Home) or on par with the best performance (on DomainNet). Comparing set-level and cluster-level methods, they achieve better performance with respect to different domain splits. On Office-Home (Table 1), the set-level method achieves an averaged ECE of 6.80% and IR of 56% whereas the cluster-level NN method yields an averaged ECE of 6.98% and IR of 53%. On DomainNet, the cluster-level NN method achieves better performance over several different target domains (Table 3). It produces an averaged ECE of 6.44% and IR of 70% (compensating for 70% of the calibration errors caused by domain shifts). This verifies that the strategy of learning multiple calibration models at the cluster level and using the nearest neighbor algorithm to select the most proper temperature for each test sample can effectively improve calibration performance.

The cluster-level regression-based method produces slightly higher errors than the other two methods on average. Conceptually, learning a regression model that captures the underlying mapping from features to corresponding optimal temperatures can allow run-time extrapolation such that, instead of selecting from temperatures learned for clusters, one can directly predict a proper temperature for the specific test instance. In practice, its performance is more sensitive to the accuracy and robustness of the learned regression model, clusters and features used. More parameters also need to be estimated, compared to temperature scaling that estimates only a single parameter.

Comparison across different domains. In Table 1 and Table 3, domains are arranged from left to right with increasing image realism. For both datasets, the lowest calibration errors are achieved for domains that reside in the middle of the spectrum. For example, Art has the lowest ECE on Office-Home, whereas Sketch has the lowest ECE on DomainNet. These observations agree with our theoretical analysis, which states that the overlap of the data distributions between the target and calibration domains determines the calibration error. For domains at the ends of the spectrum, chances of obtaining good alignment using the remaining domains decrease. This directly leads to the observed U shape of ECE scores across the domain spectrum (i.e., low in the middle and high at both ends).

In comparison to domain adaptation, we assume the availability of multiple source domains. Our assumption is more realistic for applications such as recognition using new sensor platforms or autonomous driving under extreme weather/unexplored terrains. If unlabeled target data is available at the calibration stage, we can use it to estimate the density ratio. In this setting, our method reduces to calibration transfer via domain adaptation but with the capability to choose the right calibration domain or portion of it to optimize the transfer. It is also worth noting that we bypass the step of optimizing feature alignment, a commonly used method for domain generalization. Instead, we focus on improving the confidence prediction to better match the classification accuracy, given any classifiers whether or not optimized to maintain accuracy across domains. Our calibration methods can be applied on top of a feature space that is aligned across domains to further reduce the remaining misalignment from the calibration point of view. We include additional experiments and discussion in the supplementary materials.

6. Conclusions

In this work, we addressed the problem of confidence calibration for domain generalization, a more challenging problem than calibration for domain adaption as no data from the target domain is used. Our key idea is to exploit multiple calibration domains with covariate shifts against the source domain used for training the classification model and between each other. We compared the proposed solutions under the same theoretical framework against calibration methods based on domain adaptation. We showed that introducing multiple calibration domains can effectively reduce the variance of the density ratio, the main factor that determines the upper bound of the calibration error against the oracle. Encouraged by our theoretical study, we proposed three alternative algorithms based on temperature scaling, namely set-level, cluster-level with nearest neighbor, and cluster-level with linear regression. Through experiments using the Office-Home and DomainNet datasets, we demonstrated that our methods can outperform calibration methods via domain adaptation with statistically significant (with a confidence $> 95\%$) improvement for at least 75% of the tested scenarios.

7. Acknowledgments

This material is based upon work supported by the Defense Advanced Research Projects Agency (DARPA) under Contract No. HR001119C0112. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the DARPA.

[2]As TransCal and WTS reported different ECE scores for source-only and target-only calibration, we use their respective numbers to compute the improvement ratios for fair comparison.

References

[1] Amr Alexandari, Anshul Kundaje, and Avanti Shrikumarn. Maximum likelihood with bias-corrected calibration is hard-to-beat at label shift adaptation. In *International Conference on Machine Learning*, 2020. 2

[2] Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. Concrete problems in ai safety. In *arXiv:1606.06565*, 2016. 1

[3] Yogesh Balaji, Swami Sankaranarayanan, and Rama Chellappa. Metareg: Towards domain generalization using meta-regularization. In *Adv. Neural Inform. Process. Syst.*, 2018. 2

[4] Corinna Cortes, Yishay Mansour, and Mehryar Mohri. Learning bounds for importance weighting. In *Adv. Neural Inform. Process. Syst.*, 2010. 1, 2, 4

[5] Gabriela Csurka. Domain adaptation for visual applications: A comprehensive survey. In *Domain Adaptation in Computer Vision Applications*, 2017. 2

[6] Qi Dou, Daniel C. Castro, Konstantinos Kamnitsas, and Ben Glocker. Domain generalization via model-agnostic learning of semantic features. In *Adv. Neural Inform. Process. Syst.*, 2019. 2

[7] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning*, 2017. 2

[8] Muhammad Ghifary, W. Bastiaan Kleijn, Mengjie Zhang, and David Balduzzi. Domain generalization for object recognition with multi-task autoencoders. In *Int. Conf. Comput. Vis.*, 2015. 2

[9] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In *International Conference on Machine Learning*, 2017. 1, 2, 3

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 6

[11] Judy Hoffman, Eric Tzeng, Taesung Park, Jun-Yan Zhu, Phillip Isola, Kate Saenko, Alexei Efros, and Trevor Darrell. Cycada: Cycle-consistent adversarial domain adaptation. In *International Conference on Machine Learning*, 2018. 2

[12] Bhavya Kailkhura, Brian Gallagher, Sookyung Kim, Anna Hiszpanski, and T. Yong-Jin Han. Reliable and explainable machine-learning methods for accelerated material discovery. In *npj Comput. Mater.*, 2019. 1

[13] Guoliang Kang, Lu Jiang, Yi Yang, and Alexander G. Hauptmanng. Contrastive adaptation network for unsupervised domain adaptation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019. 2

[14] Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M. Hospedales. Learning to generalize: Meta-learning for domain generalization. In *AAAI*, 2018. 2

[15] Haoliang Li, Sinno Jialin Pan, Shiqi Wang, and Alex C. Kot. Domain generalization with adversarial feature learning. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018. 2

[16] Stuart P Lloyd. Least squares quantization in PCM. In *IEEE Transactions on Information Theory*, 1982. 5

[17] Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I. Jordan. Conditional adversarial domain adaptation. In *Adv. Neural Inform. Process. Syst.*, 2018. 6

[18] Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I. Jordan. Deep transfer learning with joint adaptation networks. In *International Conference on Machine Learning*, 2017. 2

[19] Mahdi Pakdaman Naeini, Gregory F Cooper, and Milos Hauskrecht. Obtaining well calibrated probabilities using Bayesian binning. In *AAAI*, 2015. 2, 3, 5

[20] Anusri Pampari and Stefano Ermon. Unsupervised calibration under covariate shift. In *arXiv:2006.16405*, 2020. 1, 2, 3, 4, 6, 7

[21] Yingwei Pan, Ting Yao, Yehao Li, Yu Wang, Chong-Wah Ngo, and Tao Mei. Transferrable prototypical networks for unsupervised domain adaptation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019. 2

[22] Sangdon Park, Osbert Bastani, James Weimer, and Insup Leen. Calibrated prediction with covariate shift via unsupervised domain adaptation. In *AISTATS*, 2020. 1, 3, 4

[23] Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *Int. Conf. Comput. Vis.*, 2019. 6

[24] John C. Platt. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In *Advances in Large Margin Classifiers*, 1999. 1, 2

[25] Alfréd Rényi. On measures of information and entropy. In *Proceedings of the 4th Berkeley Symposium on Mathematics, Statistics and Probability*, 1960. 4

[26] Ashish Shrivastava, Tomas Pfister, Oncel Tuzel, Josh Susskind, Wenda Wang, and Russ Webb. Learning from simulated and unsupervised images through adversarial training. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2017. 2

[27] Yu Sun, Eric Tzeng, Trevor Darrell, and Alexei A. Efros. Unsupervised domain adaptation through self-supervision. In *arXiv:1909.11825*, 2019. 2

[28] Ehsan Toreini, Mhairi Aitken, Kovila P. L. Coopamootoo, Karen Elliott, Vladimiro Gonzalez Zelaya, Paolo Missier, Magdalene Ng, and Aad van Moorsel. Technologies for trustworthy machine learning: A survey in a socio-technical context. In *arXiv:2007.08911*, 2020. 1

[29] Hugo Touvron, Andrea Vedaldi, Matthijs Douze, and Hervé Jégou. Fixing the train-test resolution discrepancy. In *Adv. Neural Inform. Process. Syst.*, 2019. 1

[30] Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. Deep hashing network for unsupervised domain adaptation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2017. 2, 5

[31] Mei Wang and Weihong Deng. Deep visual domain adaptation: A survey. In *Neurocomputing*, 2018. 2, 3

[32] Ximei Wang, Mingsheng Long, Jianmin Wang, and Michael I. Jordan. Transferable calibration with lower bias and variance in domain adaptation. In *NeurIPS*, 2020. 1, 3, 4, 6, 7

[33] Garrett Wilson and Diane J. Cook. A survey of unsupervised deep domain adaptation. In *ACM Transactions on Intelligent Systems and Technology*, 2020. 2

[34] Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V. Le1. Self-training with noisy student improves imagenet classification. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2020. 1

[35] Bianca Zadrozny and Charles Elkan. Obtaining calibrated probability estimates from decision trees and naive Bayesian classifiers. In *International Conference on Machine Learning*, 2001. 2, 5

[36] Bianca Zadrozny and Charles Elkan. Transforming classifier scores into accurate multiclass probability estimates. In *KDD*, 2002. 2, 5