Unsupervised Accuracy Estimation of Deep Visual Models using Domain-Adaptive Adversarial Perturbation without Source Samples

JoonHo Lee[1], Jae Oh Woo[2], Hankyu Moon[2] and Kwonho Lee[1]
[1]Samsung SDS, [2]Samsung SDS America
{joonholee,jaeoh.w,hankyu.m,kwonho81.lee}@samsung.com

A. Experimental Details

A.1. Datasets

We evaluate the effectiveness of our proposed framework on several benchmark datasets including Digits, Office-31, Office-Home, and VisDA for natural distribution shift scenarios. Additionally, we conduct experiments on synthetic distribution shift scenarios by employing CIFAR-10-C and CIFAR-100-C datasets.

Digits. We conducted experiments on smaller Digits datasets: MNIST [19], USPS [15], and SVHN [24]. These datasets comprise labeled images of digits ranging from 0 to 9. MNIST dataset comprises 60K training images and 10K test images, while USPS dataset contains 7,291 training images and 2,007 test images. SVHN dataset comprises 73,257 training images and 26,032 test images. All images are resized to 32×32 during training and testing. We considered all six cross-domain tasks, including the challenging MNIST→SVHN and MNIST→USPS scenarios.

Office-31. We also evaluate Office-31 [26] datasets. The Office-31 dataset contains 4,652 images, similar to ImageNet [4], distributed across 31 objects from three distinct domains, namely, Amazon (A), DSLR (D), and Webcam (W), with 2,817, 498, and 795 images, respectively. For training and testing, we resized each domain set to 224×224 after splitting it into a development set (90%) and a holdout set (10%). We evaluated all six source-target combinations.

Office-Home. The Office-Home [27] dataset consists of 15,500 high-resolution images belonging to 65 categories from four distinct domains: Artistic images (Ar), Clip Art (Cl), Product images (Pr), and Real-World images (Rw). In this study, we considered all possible 12 UDA tasks.

VisDA. The VisDA [25] dataset contains a large-scale collection of complex images from 12 categories that include 152K synthetically rendered images from different angles and lighting configurations and 55K real-world images that are sampled from MSCOCO [22]. These images are resized to 224×224 for training and testing. Due to its more practical sim-to-real UDA problem, this dataset is suitable for more challenging cross-domain benchmarks.

CIFAR-10/CIFAR-10-C and CIFAR-100/CIFAR-100-C. CIFAR-10 and CIFAR-100 datasets [18] consist of 50K training images and 10K test images with 32×32 dimensions per image, and ten and hundred mutually exclusive classes, respectively. CIFAR-10-C and CIFAR-100-C datasets [14] introduce 19 types of artificial corruptions such as blur, noise, weather, and digital categories, creating corresponding corrupted subsets. These CIFAR-10 to CIFAR-10-C and CIFAR-100 to CIFAR-100-C settings are quite suitable for measuring the UAE performance against diverse synthetic distribution shift scenarios.

A.2. Network Architectures

For Digits experiments, we used a LeNet variant for Digits with a convolutional kernel size of 5×5 and applied ReLU function after each Batch Normalization (BN) layer. The entire architecture consists of: Conv(3, 64) - BN(64) - Maxpool(2) - Conv(64, 64) - BN(64) - Maxpool(2) - Conv(64, 128) - BN(128) - Linear(8192, 3072) - BN(3072) - Linear(3072, 2048) - BN(2048) - Linear(2048, 10).

For CIFAR-10 / CIFAR-10-C and CIFAR-100 / CIFAR-100-C datasets, we employed the ResNet18 backbone, while we used the ResNet50 backbone for Office-31 and Office-Home datasets. The ResNet101 backbone was applied for VisDA. The complete network comprises each backbone (i.e. from the initial layer to the global average pooling layer) and two additional fully connected layers.

A.3. Baselines

We consider Proxy Risk (Proxy) [2], Difference of Confidence (DoC) [11], Random Initialization (RI) [1], Representation Matching (RM) [1], and Generalization Disagreement Equality (GDE) [16] as our baselines. We reimplement all baselines and conducted a comparative analysis under identical experimental conditions. It is worth noting that while these baselines necessitate access to source samples, our framework only requires unlabeled target samples when estimating the target risk of the source model.

A.4. Training Configurations

Source Model Training. During the source model training, we train the models from the scratch for the smaller networks such as the LeNet variant (Digits) and ResNet18 (CIFAR-10 and CIFAR-100). We leverage ImageNet pretrained initialization for ResNet50 (Office-31 and Office-Home) and ResNet101 (VisDA) to compensate for the small dataset size and to reduce time to converge, respectively.

Referring to the *random initialization* strategy of [16] combined by additional hyperparameter selection, we train ten source models for each source dataset independently, by applying different augmentations out of $\{$weak augmentation, strong augmentation$\}$ to input, different initial seeds for random number generation selected from $\{$2021, 2022, 2023, 2024, 2025$\}$, and different learning rates if necessary. Note that MNIST and USPS are trained only with strong augmentation as the challenging M→S and U→S UDA on source models trained with weak augmentation often lead to collapsed results. Specifically, standard random cropping, rotating, flipping, and color jittering are applied for weak augmentation (on Digits, flipping is not used) whereas we employ RandAugment [3] with Cutout [7] for strong augmentation.

Each source model of Digits is trained for 50 epochs with a mini-batch size of 500 while the learning rate is initially set to one of $\{$0.1, 0.05$\}$ and steps down by $\times 0.1$ after 40-th epoch. For CIFAR-10 and CIFAR-100, we train from scratch for 300 epochs with a mini-batch size of 200 while the learning rate is initially set to 0.1 and steps down by $\times 0.1$ every 80-th epoch. We fine-tune the source models of Office-31, Office-Home, and VisDA using ImageNet pretrained backbone with learning rate schedules ranging from 0.001 to 0.00001 on cosine annealing for 1200, 1200, and 400 iterations, respectively, with reference to a recent model training tricks [13]. We set the mini-batch size to 192, 192, and 132 for Office-31, Office-Home, and VisDA, respectively. For CIFAR-10 and CIFAR-100, we apply Adam [17] optimizer whereas SGD with momentum 0.9 and weight decay 0.0001 is applied for the other models.

During investigation on straightforward pseudo-labeling (Sec 3 in the main manuscript), we trained 192 ResNet50 source models by combining six different learning rates, two input augmentation options, two optimizers among $\{$Adam, SGD with momentum$\}$, two label smoothing options and four different training epochs.

Source-Free UDA. In the proposed SF-DAP framework, we first adapt the given source model to the target distribution by a source-free UDA method such as SHOT [21], FAUST [20], or the proposed PAFA. We apply the default hyperparameters that are used by each method unless otherwise stated. The same random seeds are applied that are used for the source model training. Two types of augmentations are employed to enable perturbation in PAFA: standard random crop-rotate-flip and color jittering for weak augmentation (no flipping is used on Digits), and RandAugment [3] with Cutout [7] for strong augmentation. In PAFA, we simply set α to 0.5 without any intensive tuning effort since this value leads to a viable performance.

In all source-free UDA methods, the common training configurations are as follows: For Digits and Office-31 settings, we apply a fixed learning rate of 0.0002. VisDA is trained by a learning rate from 0.0005 to 0.00005 scheduled with cosine annealing, whereas a learning rate from 0.001 to 0.0001 scheduled with cosine annealing is applied to CIFAR-10→CIFAR-10-C and CIFAR-100→CIFAR-100-C scenarios. Mini-batch size is set to 500 in Digits, CIFAR-10-C, and CIFAR-100-C target domains while 92 in Office-31 and 64 in VisDA are applied. The determination of the number of UDA epochs is based on the point of loss saturation within 60 epochs for Digits, Office-31, and Office-Home datasets, and 4, 20, and 20 epochs for VisDA, CIFAR-10-C, and CIFAR-100-C datasets, respectively.

A.5. Evaluation Protocol

We repreat experiments on ten source models trained independently and present the mean and standard deviation of the mean absolute errors (MAE) for 63 scenarios distributed across six groups. Given the variability in the number of tasks within each group, we computed two types of averages to provide a comprehensive evaluation of the estimation performance across benchmark groups. The first average, referred to as the *micro average*, represents the overall average. The second average, presented as the *macro average*, considers the mean of the average MAE values within each group, allowing for a more balanced assessment of the estimation performance.

We evaluate each baseline according to the definition of the risk estimator as reported in each paper. We utilize $\mathbb{E}_{x\sim\mathcal{D}_S}[\max_{k\in\mathcal{Y}} h_S^k(x)] - \mathbb{E}_{x\sim\mathcal{D}_T}[\max_{k\in\mathcal{Y}} h_S^k(x)]$ for DoC [11], whereas $\max_{h'\in\mathcal{P}}\ \varepsilon_T(h_S, h')$ is used for Proxy [2] where $\mathcal{P}$ is a set of check models. For GDE [16], we apply $\mathbb{E}_{h_S,h'_S\in\S}[\varepsilon_T(h_S(x), h'_S(x))]$ where § is a set of sibling source models that includes the source model of interest for evaluation. For RI and RM [1], iterative self-trained ensemble of models $\{h_i\}_{i=1}^N$ as a pseudo-label is used to estimate $\varepsilon_T(h_S)$. For a fair comparison, five pairs of sibling source models are used for GDE experiments and five independently-adapted check models are used for

Table 1. Full UAE benchmark results on natural distribution shift scenarios (MAE, %). **Bold** for the best and ***bold-italic*** for the next best.

			source access approach					source-free approach (ours)	
datasets	source	target	DoC [11]	Proxy [2]	RI [1]	RM [1]	GDE [16]	SF-DAP (ADV)	SF-DAP (AAP)
Digits	MNIST	USPS	0.63±0.02	0.45±0.34	3.87±2.45	0.55±0.25	1.45±0.02	1.17±0.05	**0.05±0.04**
	MNIST	SVHN	29.14±0.21	20.08±2.64	7.31±1.29	28.07±1.70	58.46±0.17	**3.02±0.26**	5.39±0.21
	USPS	MNIST	12.83±0.12	6.70±2.87	29.10±2.60	10.16±4.03	33.65±0.10	**0.40±0.14**	1.25±0.29
	USPS	SVHN	32.80±0.17	18.97±2.96	9.59±2.04	21.97±3.01	65.20±0.10	6.16±0.21	**3.84±0.28**
	SVHN	MNIST	4.08±2.21	6.22±1.93	5.20±3.72	3.28±1.29	2.77±1.95	1.38±0.50	**0.99±0.54**
	SVHN	USPS	8.25±1.13	5.35±1.76	1.95±1.67	3.93±0.81	5.24±1.00	**0.96±0.72**	1.41±0.66
Digits average			14.62±0.80	9.63±1.44	9.50±1.51	11.33±1.36	27.80±0.75	***2.18±0.56***	**2.15±0.58**
Office-31	Amazon	DSLR	9.50±0.96	1.99±1.60	2.69±2.63	1.49±1.38	8.71±1.40	1.85±1.22	**0.96±0.87**
	Amazon	Webcam	8.81±0.86	**1.42±0.64**	7.82±4.60	2.19±1.35	9.70±1.69	4.43±2.10	1.62±0.98
	DSLR	Amazon	5.24±3.20	10.47±1.81	11.57±1.93	4.21±2.76	18.80±1.91	9.05±1.78	**3.28±1.97**
	DSLR	Webcam	1.82±1.24	0.74±0.88	7.38±1.33	**0.62±0.38**	1.62±0.36	2.97±0.52	3.08±0.89
	Webcam	Amazon	**3.07±2.26**	11.98±3.05	13.29±3.19	7.39±3.63	19.71±1.71	9.56±1.77	5.23±2.03
	Webcam	DSLR	1.58±1.00	1.22±1.09	9.40±2.03	0.50±0.26	0.22±0.15	**0.46±0.24**	0.88±0.49
Office-31 average			5.00±1.26	4.64±1.23	8.69±1.62	***2.73±1.28***	9.79±1.10	4.72±1.13	**2.51±1.10**
Office-Home	Art	Clipart	33.63±0.17	10.03±2.21	17.80±1.86	3.88±2.07	53.08±0.15	8.79±0.24	**3.41±0.43**
	Art	Product	28.17±0.31	6.91±1.31	8.85±1.88	2.80±1.37	35.99±0.25	3.83±0.38	**0.86±0.24**
	Art	Real-World	25.80±0.23	6.77±2.01	6.60±1.47	1.91±1.13	26.54±0.23	5.75±0.63	**1.26±0.57**
	Clipart	Art	31.84±0.37	11.38±2.49	10.06±6.08	1.85±0.99	41.33±0.45	9.32±0.63	**6.02±0.74**
	Clipart	Product	30.50±0.31	9.75±2.91	11.54±2.17	**2.93±0.94**	35.88±0.32	12.59±0.34	7.59±0.38
	Clipart	Real-World	31.02±0.30	9.94±3.13	12.50±1.15	**2.30±1.19**	34.22±0.26	11.94±0.41	7.20±0.58
	Product	Art	18.79±0.28	12.51±2.41	**6.56±4.04**	6.94±2.23	42.09±0.33	10.51±0.83	8.98±0.65
	Product	Clipart	25.27±0.35	12.15±2.56	17.79±3.83	**7.00±3.17**	52.68±0.25	17.12±0.23	11.98±0.43
	Product	Real-World	14.27±0.19	6.33±1.58	10.80±1.10	**2.67±1.06**	25.62±0.15	7.00±0.31	3.85±0.63
	Real-World	Art	19.91±0.07	6.59±0.84	13.04±2.91	4.98±1.50	29.61±0.14	5.44±0.42	**3.26±0.41**
	Real-World	Clipart	28.97±0.23	8.80±0.98	20.50±2.07	**4.04±2.30**	50.49±0.33	8.96±0.38	4.11±0.32
	Real-World	Product	19.23±0.23	3.79±1.28	9.57±1.51	2.61±0.73	22.69±0.25	3.54±0.32	**1.28±0.15**
Office-Home average			25.62±0.50	8.75±1.41	12.13±1.58	**3.66±1.25**	37.52±0.51	8.73±0.65	***4.98±0.68***
VisDA (Syn-to-Real)			15.72±4.35	8.90±1.71	7.50±4.70	***4.41±2.52***	29.31±3.73	***4.41±1.10***	**1.73±0.93**

Proxy Risk experiments. We did our best to achieve the reported results, but the higher MAE numbers, if any, may be partially due to our lack of a proper recipe for adjusting hyperparameters when running baseline approaches.

A.6. Detailed Results

All results of 63 scenarios are presented in Table 1 for the natural distribution shift scenarios and in Table 2 for the synthetic distribution shift scenarios.

A.7. Computing Infrastructure

We conduct all experiments using PyTorch and NVIDIA A100 GPUs.

B. Review on Existing UAE Methods

B.1. Regression-based Approaches

To estimate the accuracy of an already trained model to unknown unlabeled data, one of the first approaches [6] draw on the (negative) correlation between the differences in data distribution and model accuracy, and builds a regression model from the Frechet distance between a collection of artificially augmented from the source data and the model accuracy on these augmented datasets. [5] established the relation between the rotation estimation task and the classification task. When the network is trained to optimize both criteria, they observed a strong correlation between rotation estimation accuracy and classification accuracy. This observation led to a simple regression-based classification accuracy estimation approach by fitting a linear mapping between both accuracies from a collection of datasets synthetically generated from standard datasets [6].

B.2. Confidence-based Approaches

Guillory *et al.* [11] approaches the problem of accuracy estimation from the per-sample prediction confidence that is robust to distributional shifts. They came up with a simple prediction confidence-based measure called AC (Average Confidence) and extended it to DoC (Difference of Confidence) measure, which shows improved estimation accuracy. In a similar manner, [10] developed a 'score function' based accuracy estimation that is essentially per-sample confidence measures from either maximum class probability or negative entropy of the class probability. They first

Table 2. Full UAE benchmark results on synthetic distribution shift scenarios (MAE, %).

		source access approach					source-free approach (ours)	
datasets	corruptions	DoC [11]	Proxy [2]	RI [1]	RM [1]	GDE [16]	SF-DAP (ADV)	SF-DAP (AAP)
CIFAR-10	Gaussian noise	22.33±3.33	11.20±0.31	3.79±1.95	**0.78±0.50**	4.86±1.48	1.87±0.43	3.08±2.33
to	Shot noise	20.42±2.81	10.95±0.31	3.08±1.19	**0.52±0.30**	3.77±1.44	1.95±0.27	3.31±2.48
CIFAR-10-C	Impulse noise	27.94±5.23	12.25±1.04	4.97±2.35	2.81±1.00	8.01±2.03	**1.14±0.57**	3.37±2.05
	Speckle noise	20.49±2.81	10.93±0.23	3.12±1.14	**0.68±0.39**	3.68±1.41	1.79±0.31	3.04±2.21
	Defocus blur	15.79±1.42	9.22±0.82	2.35±0.49	2.48±1.95	**1.64±1.20**	1.65±0.52	2.86±2.06
	Glass blur	19.26±0.51	11.96±1.19	**1.72±1.05**	2.19±1.69	2.62±1.97	2.20±0.43	3.18±2.66
	Motion blur	18.96±1.62	10.93±0.83	2.70±0.29	2.58±1.88	2.16±1.36	**1.27±0.32**	3.38±1.68
	Zoom blur	15.73±1.24	8.88±0.89	2.31±0.69	2.79±1.63	1.45±0.92	**1.43±0.81**	2.60±1.89
	Gaussian blur	16.66±1.57	9.42±0.82	2.64±0.45	2.74±2.24	1.83±1.40	**1.55±0.45**	2.94±2.08
	Snow	19.56±1.71	11.37±0.48	2.31±0.74	**1.39±0.55**	2.54±0.40	1.71±0.99	3.71±2.19
	Fog	19.73±2.88	11.80±1.00	3.92±0.67	2.70±1.65	3.27±2.28	3.14±2.68	**2.56±2.35**
	Frost	18.09±2.01	9.80±0.56	2.81±0.43	2.02±0.76	2.35±0.71	**1.40±0.83**	2.34±1.85
	Brightness	15.17±2.08	8.69±0.49	2.53±0.95	2.26±1.88	1.75±1.17	**1.29±0.40**	2.79±1.93
	Spatter	19.78±3.08	11.48±0.31	3.16±0.61	**1.35±1.05**	2.66±1.62	2.21±0.85	3.44±2.76
	Contrast	23.34±6.17	9.70±1.09	8.86±1.72	4.30±2.72	6.65±4.98	11.66±5.42	**2.90±0.59**
	Elastic transform	16.60±1.30	10.63±1.28	2.11±0.26	2.78±1.33	**1.33±0.91**	1.87±0.53	2.96±2.44
	Pixelate	15.34±1.25	9.16±0.73	1.92±0.34	2.34±1.04	**1.19±0.60**	1.77±0.36	2.82±2.10
	JPEG compression	16.82±1.77	10.29±0.37	2.60±0.25	**1.78±1.52**	2.02±1.48	2.09±0.96	3.34±2.32
	Saturate	17.30±3.34	10.34±0.33	3.83±1.25	2.70±2.39	3.06±1.90	**2.39±2.19**	3.56±2.55
CIFAR-10 to CIFAR-10-C average		18.91±1.56	10.47±0.83	3.20±0.94	**2.17±1.18**	2.99±1.24	***2.34±1.47***	3.06±1.46
CIFAR-100	Gaussian noise	52.05±6.07	22.92±13.56	7.38±4.02	**0.70±0.50**	7.97±3.11	2.98±1.69	2.48±1.92
to	Shot noise	50.05±5.24	19.20±2.71	6.52±2.89	**1.17±0.24**	6.74±3.08	4.30±2.17	2.88±2.51
CIFAR-100-C	Impulse noise	55.92±9.18	39.79±27.47	9.39±5.13	**0.55±0.39**	10.17±4.00	0.70±0.38	1.52±0.31
	Speckle noise	50.30±5.30	19.49±2.58	6.54±2.80	**1.16±0.32**	6.73±3.11	4.32±2.31	2.70±2.20
	Defocus blur	45.08±2.52	30.47±6.06	4.93±1.40	**2.61±1.25**	4.87±3.13	9.95±3.01	3.06±1.71
	Glass blur	48.00±1.40	24.54±0.79	3.50±3.10	**0.91±0.40**	5.61±1.47	7.07±4.21	2.86±0.88
	Motion blur	47.36±2.35	30.12±7.08	4.73±2.31	**2.17±0.51**	5.56±2.54	9.27±2.32	3.83±2.58
	Zoom blur	44.92±2.34	29.26±6.25	4.80±1.23	**2.50±1.55**	4.45±2.89	8.80±2.46	2.89±2.54
	Gaussian blur	46.09±2.53	30.44±6.48	5.04±1.76	**2.60±0.88**	5.22±3.25	10.12±2.56	3.17±2.93
	Snow	49.93±3.06	22.15±0.48	4.60±3.65	**1.02±0.33**	6.39±1.16	6.32±2.22	1.76±1.06
	Fog	51.15±3.62	31.95±9.00	5.58±3.96	2.49±0.73	7.48±3.29	6.51±2.77	**1.81±1.03**
	Frost	48.58±3.36	24.23±10.16	4.77±3.19	**1.70±0.28**	6.26±1.54	4.03±1.92	**1.70±1.16**
	Brightness	45.06±3.61	23.15±1.78	5.27±1.52	**2.47±1.12**	4.78±3.13	7.86±2.65	3.71±2.92
	Spatter	50.44±4.65	20.59±1.32	5.98±3.33	**1.43±0.22**	6.65±3.45	5.53±2.33	2.89±2.05
	Contrast	52.17±5.73	24.20±13.17	10.12±4.47	3.03±0.52	9.51±6.13	**2.18±0.92**	11.03±2.70
	Elastic transform	45.63±2.60	29.04±1.92	4.57±1.99	**2.10±1.51**	4.74±2.34	9.61±3.82	2.97±0.49
	Pixelate	44.42±2.66	25.03±1.54	4.42±1.67	**1.99±1.39**	4.58±2.29	8.43±3.69	2.87±1.10
	JPEG compression	46.85±3.15	26.38±0.75	5.05±1.87	2.41±1.03	5.19±3.15	8.62±3.34	**1.93±1.22**
	Saturate	50.20±5.78	27.58±1.91	7.68±2.63	**3.17±0.22**	7.29±4.97	12.80±3.36	6.31±3.47
CIFAR-100 to CIFAR-100-C average		48.64±1.99	26.34±2.46	5.84±1.67	**1.90±0.84**	6.33±1.75	6.81±1.59	***3.28±1.35***

calibrate the probability outputs so that they match the accuracy using temperature scaling [12]. Then the key to their approach is to identify the score function threshold from the source data that match the source accuracy and apply the same threshold to the target score function leading to the target accuracy estimation.

B.3. Disagreement-based Approaches

Nakkiran and Bansal [23] first found that if one train two networks of identical architecture on two independently sampled subsets of a dataset, the disagreement rate on test data linearly correlates with the network's test accuracy. Jiang *et al*. [16] further extends the behavior to two identical networks trained on the same dataset but with different random initialization. They verified that this observed cor-

Table 3. Ablation study on various UAE tasks by comparing SF-DAP (ADV), SF-DAP (AAP) and their intermediate configuration. We report average MAEs(%) in each benchmark group for simplicity. The notation $C_{adj\_unc}C_{cls}$ refers to an intermediate configuration that is identical to SF-DAP (AAP) except that the data volume density factor C_{den} is not used. Micro average computes the mean of all 63 cross-domain scenarios, whereas macro average represents the mean of average MAE values within the six benchmark groups.

Method	Digits	Office-31	Office-Home	VisDA	CIFAR-10-C	CIFAR-100-C	micro avg.	macro avg.
SF-DAP (ADV)	2.18±0.56	4.72±1.13	8.73±0.65	4.41±1.10	2.34±1.01	6.81±1.59	5.15±1.15	4.86±1.00
$C_{adj\_unc}C_{cls}$	**2.05±0.61**	3.05±1.07	6.44±1.75	3.05±0.86	4.07±1.47	3.83±1.33	4.15±1.39	3.75±1.09
SF-DAP (AAP)	2.15±0.58	**2.51±1.10**	**4.98±0.68**	**1.73±0.93**	**3.06±1.46**	**3.28±1.35**	**3.33±1.20**	**2.95±1.01**

Table 4. Running time comparison of various methods that require additional training. For GDE and RI, the time for additional training of the source models is excluded.

Setting	GDE [16]	RI [1]	Proxy [2]	RM [1]	SF-DAP (ADV)	SF-DAP (AAP)
Amazon→DSLR	27s	1m 24s	19m 3s	18m 58s	5m 54s	6m 21s
Amazon→Webcam	48s	2m 35s	30m 25s	30m 50s	9m 58s	10m 49s
CIFAR-10 (average)	13s	1m 20s	6m 43s	5m 30s	4m 40s	4m 55s

relation leads to a *disagreement* based accuracy estimation for unlabeled datasets, called GDE. They further established the necessary conditions for the calibration of the prediction outputs for the method to work.

B.4. Iterative Ensemble-based Approaches

Chen *et al.* [1] also extends the disagreement-based approach with the flavor of the UDA approach. It employs model ensembles of more than two, where the generated ensemble serves as the *check model* (by a majority vote) for the given source model to be evaluated. The ensembles are generated by either (1) random initialization (RI), or (2) random checkpoints when the models are trained to match the source and target features (RM), as in the well-known domain adaptation training [9]. The approach also improves accuracy by using pseudo labels generated from the disagreement to be fed back to the training. A slightly earlier work [2] also uses DIR training to find a check model that shows maximum disagreement while minimizing the DIR loss. Then the disagreement rate (called Proxy Risk) approximates the error of the given source model on the target data.

C. Further Analysis

C.1. Ablation Study

We conducted an ablation study on various datasets by comparing SF-DAP (AAP), SF-DAP (ADV), and their intermediate configuration as shown in Table 3, where the magnitude of VAP (ϵ) is computed by $\epsilon_0 C_{adj\_unc}C_{cls}$ for the intermediate configuration. Table 3 demonstrates the gradual improvement of estimation performance from SF-DAP (ADV) to SF-DAP (AAP).

C.2. Estimation Time

We compare the runtime of various methods that require additional training as shown in Table 4. SF-DAP shows comparable or superior running time to other existing methods, particularly when compared with RM and Proxy Risk, which also perform UDA during estimation. Once the target model adaptation is completed, the inference time of SF-DAP should be almost the same as that of RI. Note that these results are measured with A100 GPUs.

Table 5. UAE performance comparison when different UDA methods are employed. *SHOT-IM is applied without the network augmentation. **Epistemic uncertainty loss is not applied.

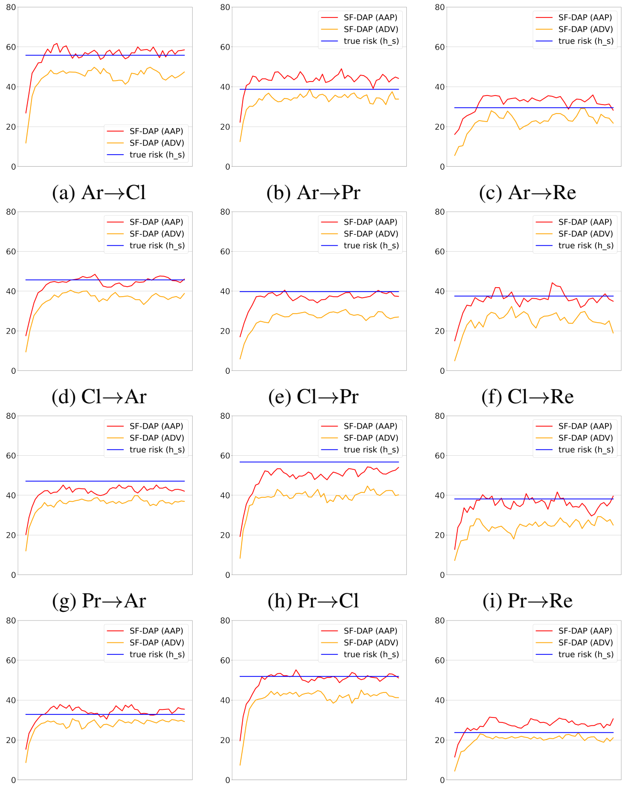
datasets	PAFA	SHOT*	FAUST**
Digits	**2.15±0.58**	3.33±1.63	2.95±1.18
Office-31	**2.51±1.10**	3.85±1.60	3.79±1.54
Office-Home	**4.98±0.68**	5.67±1.72	5.75±1.66
VisDA	1.73±0.93	**1.10±1.01**	1.16±0.95
CIFAR-10	**3.06±1.46**	4.02±1.45	3.95±1.46
CIFAR-100	**3.28±1.35**	3.34±1.31	3.39±1.33
micro average	**3.33±1.20**	4.00±1.49	3.97±1.44
macro average	**2.95±1.01**	3.55±1.21	3.50±1.16

C.3. Performance with Different UDA Methods

We evaluate some other source-free UDA methods such as SHOT-IM [21] and FAUST [20] within the proposed framework. As shown in Table 5, there are no significant differences between their results, while PAFA shows the most preferable performance in our SF-DAP framework under both natural and synthetic distribution shift scenarios. Because of these superior results, we consider PAFA as our UDA recommendation for the proposed framework. However, the potential benefits of more diverse source-free UDA methods are worth exploring in future research that jointly tune the feature generator with the head classifier.

C.4. Uncertainty Measurement

Recent studies have shown the dropout during inference, known as Monte Carlo (MC) dropout sampling, is equivalent to an approximation of a deep Gaussian process [8]. In this work, we estimate the predictive uncertainty by using the standard deviation of multiple (n=10) stochastic for-



(a) Ar→Cl (b) Ar→Pr (c) Ar→Re

(d) Cl→Ar (e) Cl→Pr (f) Cl→Re

(g) Pr→Ar (h) Pr→Cl (i) Pr→Re

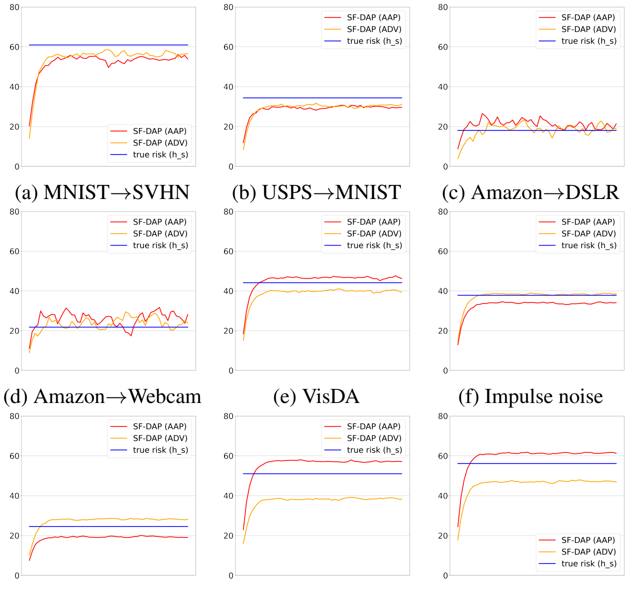
(j) Re→Ar (k) Re→Cl (l) Re→Pr

Figure 1. (Best viewed in color) Performance trends of estimation as UDA progresses are presented for Office-Home benchmarks. The true risk of the source model on the target data as well as the risk estimated by SF-DAP (AAP) and SF-DAP (ADV) are represented by blue, red and orange lines, respectively.

Table 6. UAE performance comparison. RND, ADV, and AAP denote the perturbation method of the SF-DAP framework. RND (ens.) uses the ensemble of the five independent RND estimates.

datasets	RND	RND (ens.)	ADV	AAP
Digits	4.00±0.51	**0.94±0.50**	2.18±0.56	2.15±0.58
Office-31	5.93±1.64	3.58±1.14	4.72±1.13	**2.51±1.10**
Office-Home	**2.19±0.70**	9.03±0.66	8.73±0.65	4.98±0.68
VisDA	**0.85±0.64**	7.14±1.25	4.41±1.10	1.73±0.93
CIFAR-10	20.83±2.58	4.32±1.90	**2.34±1.01**	3.06±1.46
CIFAR-100	17.57±2.09	3.97±1.39	6.81±1.59	**3.28±1.35**
micro average	12.96±1.92	4.77±1.38	5.15±1.15	**3.33±1.20**
macro average	8.56±1.17	4.83±1.07	4.86±1.00	**2.95±1.01**

ward passes that leverage MC dropout. To enhance the accuracy estimation further, future research may explore the utilization of newly developed uncertainty measures, such as the balanced entropy, which captures the information balance between the model and the class label suggested by [28].



(a) MNIST→SVHN (b) USPS→MNIST (c) Amazon→DSLR

(d) Amazon→Webcam (e) VisDA (f) Impulse noise

(g) Fog (h) Glass blur (i) Snow

Figure 2. (Best viewed in color) Performance trends of estimation as UDA progresses are presented for some of Digits, Office-31, CIFAR-10-C, CIFAR-100-C and VisDA benchmarks. (f) and (g) depict some results in 19 CIFAR-10 → CIFAR-10-C experiments, whereas (h) and (i) display outcomes in CIFAR-100 → CIFAR-100-C experiments.

C.5. Ensemble Effect on SF-DAP (RND)

Applying an ensemble to SF-DAP (RND) yields estimation performance similar to or better than that of SF-DAP (ADV) as presented in Table 6.

As we have shown from RI and RM methods in Table 1 and Table 2, the ensemble approach notably improves the accuracy estimation performance in general which is comparable with the intermediate configuration $C_{adj\_unc}C_{cls}$ shown in Table 3. We interpret that the ensemble approach can equate to the role of capturing model uncertainty. However, our consolidated SF-DAP (AAP) shows superior performance and requires fewer computational resources compared to the ensemble method.

C.6. UAE Performance during UDA

We track the performance trend of accuracy estimation as the proposed source-free UDA, PAFA, progresses to each target domain. Our proposed framework, SF-DAP, starts producing accurate estimates surprisingly early and remains steady throughout the rest of the UDA iterations as illustrated in Fig 1 and Fig 2.

References

[1] Jiefeng Chen, Frederick Liu, Besim Avci, Xi Wu, Yingyu Liang, and Somesh Jha. Detecting errors and estimating accuracy on unlabeled data with self-training ensembles. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *NeurIPS*, volume 34, pages 14980–14992. Curran Associates, Inc., 2021. 1, 2, 3, 4, 5

[2] Ching-Yao Chuang, Antonio Torralba, and Stefanie Jegelka. Estimating generalization under distribution shifts via domain-invariant representations. *ICML*, 2020. 1, 2, 3, 4, 5

[3] E. D. Cubuk, B. Zoph, J. Shlens, and Q. V. Le. Randaugment: Practical automated data augmentation with a reduced search space. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 3008–3017, 2020. 2

[4] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. A large-scale hierarchical image database. In *CVPR*, 2009. 1

[5] Weijian Deng, Stephen Gould, and Liang Zheng. What does rotation prediction tell us about classifier accuracy under varying testing environments? In *ICML*. 3

[6] Weijian Deng and Liang Zheng. Are labels always necessary for classifier accuracy evaluation? In *CVPR*, pages 15069–15078, June 2021. 3

[7] Terrance DeVries and Graham Taylor. Improved regularization of convolutional neural networks with cutout. *arXiv:1708.04552*, 08 2017. 2

[8] Y. Gal and Z. Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *ICML*, 2016. 5

[9] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky. Domain adversarial training of neural networks. *JMLR*, 17(59):1–35, 2016. 5

[10] Saurabh Garg, Sivaraman Balakrishnan, Zachary Chase Lipton, Behnam Neyshabur, and Hanie Sedghi. Leveraging unlabeled data to predict out-of-distribution performance. In *ICLR*, 2022. 3

[11] Devin Guillory, Vaishaal Shankar, Sayna Ebrahimi, Trevor Darrell, and Ludwig Schmidt. Predicting with confidence on unseen distributions. In *ICCV*, pages 1134–1144, October 2021. 1, 2, 3, 4

[12] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In Doina Precup and Yee Whye Teh, editors, *ICML*, volume 70 of *Proceedings of Machine Learning Research*, pages 1321–1330. PMLR, 06–11 Aug 2017. 4

[13] Tong He, Zhi Zhang, Hang Zhang, Zhongyue Zhang, Junyuan Xie, and Mu Li. Bag of tricks for image classification with convolutional neural networks. In *CVPR*, June 2019. 2

[14] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In *ICLR*, 2019. 1

[15] J. J. Hull. A database for handwritten text recognition research. *PAMI*, pages 550–554, 1994. 1

[16] Yiding Jiang, Vaishnavh Nagarajan, Christina Baek, and J Zico Kolter. Assessing generalization of SGD via disagreement. In *ICLR*, 2022. 1, 2, 3, 4, 5

[17] D. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv:1412.6980*, 2014. 2

[18] Alex Krizhevsky. Learning multiple layers of features from tiny images. pages 32–33, 2009. 1

[19] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient based learning applied to document recognition. volume 86(11), pages 2278–2324, 1998. 1

[20] JoonHo Lee and Gyemin Lee. Feature alignment by uncertainty and self-training for source-free unsupervised domain adaptation. *Neural Networks*, 161:682–692, 2023. 2, 5

[21] Jian Liang, Dapeng Hu, and Jiashi Feng. Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation. In *ICML*, pages 6028–6039, July 13–18 2020. 2, 5

[22] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollar, and C. L. Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014. 1

[23] Preetum Nakkiran and Yamini Bansal. Distributional generalization: A new kind of generalization. *CoRR*, abs/2009.08092, 2020. 4

[24] Y. Netzer, T.Wang, A. Coates, A. Bissacco, B.Wu, and A. Y. Ng. Reading digits in natural images with unsupervised feature learning. volume 2011, page 5, 2011. 1

[25] X. Peng, B. Usman, N. Kaushik, J. Hoffman, D. Wang, and K. Saenko. Visda: The visual domain adaptation challenge. In *CVPR*, 2016. 1

[26] K. Saenko, B. Kulis, M. Fritz, and T. Darrell. Adapting visual category models to new domain. In *ECCV*, 2010. 1

[27] Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. Deep hashing network for unsupervised domain adaptation. In *CVPR*, pages 5018–5027, 2017. 1

[28] Jae Oh Woo. Active learning in bayesian neural networks with balanced entropy learning principle. In *ICLR*, 2023. 6