MULTI-SOURCE FULLY TEST-TIME ADAPTATION

Yuntao Du, Siqi Luo, Yi Xin, MingCai Chen, Shuai Feng, Mujie Zhang, Chongjun Wang*
Department of computer science and technology, Nanjing University
duyuntao@smail.nju.edu.cn, siqiluo647@163.com
{duyuntao,xinyi,chenmc,shuaifeng,mujiezhang}@smail.nju.edu.cn
chjwang@nju.edu.cn

ABSTRACT

Deep neural networks often generalizes poorly when the distribution of test samples varies from that of the training samples. Recently, some fully test-time adaptation methods have been proposed to adapt the trained model with the unlabeled test samples before prediction. Despite achieving remarkable results, these methods only involve one trained model, which could only provide certain side information for the test samples. In real-world scenarios, there could be multiple available trained models that are beneficial to the test samples and these models are complementary to each other. Consequently, to better utilize these trained models, in this paper, we propose the problem of multi-source fully test-time adaptation to adapt **multiple trained models** to the test samples. To achieve this, we introduce a simple yet effective method utilizing a weighted aggregation scheme and introduce two unsupervised losses. The former could adaptively assign a higher weight to a more relevant model, while the latter could jointly adapt models with online unlabeled samples. Extensive experiments on three image classification datasets show that the proposed method achieves better results than baseline methods.

1 INTRODUCTION

Deep neural networks often suffer from severe performance degradation when applied to new environments due to the distribution shift Pan & Yang (2010); Wang et al. (2021); Xin et al. (2023). They are sensitive to the test samples with natural variations or corruptions, such as changing weather conditions and sensor degradation noises Hendrycks & Dietterich (2019); Koh et al. (2021). Thus, it is necessary to adapt the models to different data distributions.

Recently, several works have proposed to handle distribution shifts by online adapting the model at test time, known as test-time adaptation (TTA). TENT Wang et al. (2021) adopts entropy minimization loss Grandvalet & Bengio (2004) as the auxiliary loss to adapt the model. MEMO Zhang et al. (2021) proposes to augment one sample multiple times and average the predictions to perform entropy minimization. Besides, the following methods adopt normalization calibration Mirza et al. (2021); Zhao et al. (2023) or pseudo-label-based strategy Jang & Chung (2023); Chen et al. (2022) to deal with changing distribution. The former investigates the effects of different normalization methods at the test time for distribution matching while the latter generates pseudo labels for model adaptation.

Generally, existing fully test-time adaptation methods focused on adapting one single trained model to a new domain. While in real-world applications, there are often multiple source domains or source models available. The single model usually provides certain side transfer information, which limits their usage in real-world scenarios. On the contrary, a more practical and challenging scenario is that *a bag of trained models* are available for the test samples. Each model could provide complementary information to each other, thus it is more likely to achieve better adaptation performance than adaptation with a single trained model Ahmed et al. (2021).

Based on the above analysis, in this paper, we propose a novel problem of *multi-source fully test-time adaptation* (**MS-FTTA**) to adapt multiple trained models to test samples. Comparison with similar problem settings is shown in Table 5 and the detailed analysis is shown in Appendix. We empirically find that simply ensembling multiple trained models achieves limited improvements compared to the non-adapted model(see Table 1). Thus, in such cases, adaptation is performed not only to incorporate the combined prior knowledge from multiple models but also to prevent potential negative transfer.

*Corresponding author

To achieve this, we propose a simple yet effective method named ***M**ulti-source f**U**lly **T**est-tim**E** adaptation* (**MUTE**). Firstly, considering that different models contribute differently to the test samples, we assign a weight to each model and adopt the weight aggregation strategy Hoffman et al. (2018) to combine these methods. The weight is adaptively estimated from the perspective of *distribution shift*, aiming to assign higher weights to the models with smaller distribution shifts between training and test samples. Secondly, for model adaptation, considering that all the test samples are unlabeled, we introduce two auxiliary unsupervised losses with the estimated model weight, i.e., weighted-combined entropy loss and weighed-combined consistency loss. Both losses are chosen as they have a strong correlation to classification loss and previous work Liu & et al. (2021) has proved that a closer auxiliary task could yield better accuracy on the main task. Then, all models are jointly adapted together with adaptive weight estimation.

To sum up, the contribution of this work could be summarized as follows, **i) New problem:** We propose the problem of *multi-source fully test-time adaptation* (MS-FTTA) where multiple trained models are available for adaptation, which could utilize complementary information within multiple models to achieve better adaptation. **ii) Novel method:** We propose a novel method MUTE via a weighted aggregation scheme and two novel unsupervised losses related to the main task. **iii) Extensive evaluation:** Extensive experiments show that the proposed method consistently gets better results than baseline methods. For example, MUTE achieves an improvement of 4.2%, 1.8%, and 2.6% on the Digit-five, Office-Home, and Office31 datasets, respectively.

2 Related Work

2.1 Fully Test-Time Adaptation

TENT Wang et al. (2021) proposes to adapt the models by minimizing the entropy of model predictions, which is independent of the source data given the model parameters. Also with entropy minimization, MEMO Zhang et al. (2021) augments a sample multiple times and then minimizes the entropy of average predictions for better robustnes. To deal with distribution shift, following methods adopt batch normalization calibrationMirza et al. (2021); Zhao et al. (2023); Burns & Steinhardt (2021); Zhao et al. (2023), which investigates the effects of different normalization layers under the test-time adaptation setting, or pseudo-label based strategyJang & Chung (2023); Han et al. (2023), where pseudo labels are generated at test time for model updates. Following works explore test-time training under different scenarios, such as sample-efficient adaptation Niu & rt al. (2022), and continue challenging environment Wang et al. (2022); Gong et al. (2022); Gan et al. (2023).

2.2 Source-Free Domain Adaptation

In this setting, a trained source model, instead of the source samples, is given to the target domain and the goal is to adapt the trained model to the target domain while keep source privacy. SFDA methods consist of pseudo-label-based strategy and a generative strategy. The former firstly generates the pseudo-labels for target samples, and then adapts the model with pseudo-labels by classification loss in a self-training mannerLiang et al. (2020); Yi et al. (2023); Du et al. (2023). The latter aims to generate more training samples by generative adversarial nets with either source-like samples Du et al. (2023); Li et al. (2020) or target-like samples Kurmi et al. (2021). Considering that multi-source domains could be available, multi-source source-free domain adaptation is proposed Ahmed et al. (2021); Li et al. (2023) with a weight-aggregation scheme and carefully designed unsupervised loss.

3 Problem Definition

In this paper, we address the problem of multi-source fully test-time adaptation. We are given a bag of pre-trained models $\{F_m\}_{m=1}^{M}$, where the m^{th} model $F_m : \mathcal{X} \to \mathbb{R}^K$ is a base model trained on m^{th} labeled source training dataset $\mathcal{D}_m = \{x_i^m, y_i^m\}_{i=1}^{N_m}$ and $y_i^m \in \{1, ..., K\}$. Each source dataset is sampled from different distribution, namely $P_1(x, y), P_2(x, y), ..., P_M(x, y)$. During the inference time, due to possible distribution shift, many test samples whose distribution (denoted as $Q(x, y)$) is different from the source ones may arrive, i.e., $P_1(x, y) \neq P_2(x, y) \neq, ..., \neq P_M(x, y) \neq Q(x, y)$. We aim to boost the performance during inference by doing model adaptation only on test samples. Specially, we focus on online settings, where a batch of samples sampled from $Q(x)$, denoted as $\mathcal{B} = \{x_i\}_{i=1}^{n}$, arrived in each time. One needs to adapt the models with these unlabeled samples and then make predictions on these samples before the next batch of samples arrives.

4 Method

4.1 Estimation of the Model Weights

The overall framework is shown in Figure 1. Denote $\{\alpha_m\}_{m=1}^M$ as the weight of each trained model, and it is obvious that $\alpha_m \geq 0$ and $\sum_{m=1}^M \alpha_m = 1$. Previous work Ben-David et al. (2009); Zhang et al. (2019) has shown that if the distribution shift across domains is small, the model learned in one domain could generalize well to samples of new domains. Based on this insight, we propose to estimate the model weight from the perspective of *distribution shifts* where the smaller distribution shift between the training and the test samples, the higher weight the models get. Thus, the crucial problems are the choice of distance function and the calculation of distribution shifts. In this work, we adopt the first-order and the second-order measurement, namely the mean and covariance as the distance function, as they are simple yet effective Long et al. (2015); Sun et al. (2016); Liu & et al. (2021).

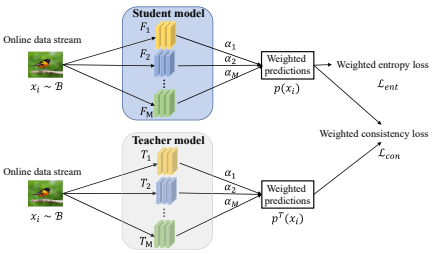


Figure 1: The model structure of the proposed method.

To measure distribution shift, following previous work Liu & et al. (2021), we assume that during the training of available models, not only source models are saved, but also some extra information, e.g., the mean $\{\mu_m\}_{m=1}^M$ and the covariance matrix $\{\Sigma_m\}_{m=1}^M$ of the training samples are also saved. To be specific, once training completes, one offline feature summarization step is performed to characterize the distribution of feature vectors of the training samples. For m^{th} source model, the mean is calculated by $\mu_m = \sum_{i=1}^{|\mathcal{D}_m|} z_i^m$ and covariance matrix is calculated by $\Sigma_m = \frac{1}{|\mathcal{D}_m|}(Z_m^T Z_m - (I^T Z_m)^T (I^T Z_m))$, where z_i^m is the feature vectors in the m^{th} training domain and $Z_m = \{z_1^m, ..., z_{|\mathcal{D}_m|}^m\}$.

During adaptation, we calculate the statistical information of test samples in an online manner such that the distribution shift could be online measured. Taking the m^{th} model as an example, at t^{th} batch, the test samples are firstly fed to this trained model, then the corresponding mean μ_m^t and covariance matrix Σ_m^t of test samples could also be calculated. As we are under the online scenario, the mean and covariance matrix calculated in different batches may vary greatly. To overcome this challenge, we maintain a global mean $\hat{\mu}_m$ and covariance matrix $\hat{\Sigma}_m$ for each model and we update the global mean and covariance matrix via an EMA scheme. At t^{th} batch, they are updated by,

$$\hat{\mu}_m = \lambda \hat{\mu}_m + (1-\lambda)\mu_m^t, \hat{\Sigma}_m = \lambda \hat{\Sigma}_m + (1-\lambda)\Sigma_m^t \tag{1}$$

where λ is the update parameter. Then, the distribution shifts across domains could be estimated as,

$$d_m = ||\hat{\mu}_m - \mu_m||^2 + ||\hat{\Sigma}_m - \Sigma_m||^2 \tag{2}$$

As the model weight considered to be inversely proportional to the distribution shifts Ben-David et al. (2009), then the weight of each model is estimated by normalizing the estimated distribution shifts and keeping the sum to be 1 as $\alpha_m = \frac{\exp^{-d_m}}{\sum_{i=1}^M \exp^{-d_i}}$.

4.2 Adaptation Objectives

4.2.1 Weighted-Combined Entropy Loss

Even though the test samples are fully unlabeled, previous methods have shown that unsupervised loss could effectively boost the adaptation performance. Entropy minimization Grandvalet & Bengio (2004) is the widely used strategy, which stems from the *cluster assumption* van Engelen & Hoos (2019) in semi-supervised learning. Such an objective could push the samples away from the decision boundary, thus it could help the model learn discriminative features for the test samples. However, existing FTTA methods pay equal importance to each model and does not consider the model's relevance to the test samples. Thus, we propose *weighted-combined entropy loss*.

Specially, given a test sample x_i, we make predictions by $F_m(x_i) = f_m(g_m(x_i))$ for m^{th} model. We combine the predictions of trained models by a weighted aggregation scheme via $\{\alpha_m\}_{m=1}^M$. And we get the weighted prediction by $p(x_i) = \sum_{m=1}^M \alpha_m F_m(x_i)$.

Lastly, denote $p(x_i)_k$ as the k-th dimension of $p(x_i)$, the weighted entropy loss is represented as,

$$\mathcal{L}_{ent}(x_i) = -\mathbb{E}_{x_i \sim \mathcal{B}} \left[\sum_{k=1}^K p(x_i)_k \log p(x_i)_k \right] \tag{3}$$

Source	Method	MT,UP,SVN,SYN → MM	MM,UP,SVN,SYN → MT	MM,MT,SVN,SYN → UP	MM,MT,UP,SYN → SVN	MM,MT,UP,SVN → SYN	Avg.
Multiple(w)	DAN	63.7	96.3	94.2	62.5	85.4	80.4
	DANN	71.3	97.6	92.3	63.5	85.3	82.0
	MCD	72.5	96.2	95.3	78.9	87.5	86.1
	CORAL	62.5	97.2	93.4	64.4	82.7	80.1
	ADDA	71.6	97.9	92.8	75.5	86.5	84.8
	M^{3}SDA-β	72.8	98.4	**96.1**	**81.3**	89.6	87.6
Single(w/o)	Source-best	62.6	**99.5**	88.9	77.8	91.8	84.1
	Source-worst	34.1	69.3	62.1	12.0	20.1	39.5
	MEMO-best	63.6	99.3	94.4	79.4	91.7	85.7
	MEMO-worst	46.4	71.5	77.8	12.0	31.5	47.8
	TENT-best	64.1	**99.5**	**96.1**	80.6	**91.9**	86.4
	TENT-worst	47.1	79.0	80.1	16.3	36.2	51.7
Multiple(w/o)	Source-only-Ens	67.0	95.5	92.9	70.7	76.6	80.5
	MEMO-Ens	67.0	97.7	92.3	73.2	79.2	81.9
	TENT-Ens	68.2	98.9	92.9	78.0	80.5	83.7
	MUTE	**73.4**	**99.5**	96.0	79.2	91.6	**87.9**

Table 1: The results on digit recognition.

4.2.2 WEIGHTED-COMBINED CONSISTENCY LOSS

For a well-performed model, we hope it has such a property that the model would produce consistent predictions with its past model which stores the already learned knowledge. To achieve this goal, following previous method Tarvainen & Valpola (2017), we maintain a weight-average teacher model for each trained model and constrain the weighted-combined predictions of the teacher models to be consistent with that of the student models.

During adaptation, for each trained model F_m, we regard it as a student model and maintain a teacher model T_m for training. The teacher model is initialized to be the same as the student model and at t^{th} batch its parameters are updated by exponential moving average on that of the student model:$\theta^T_{m,t} = \omega\theta^T_{m,t-1} + (1-\omega)\theta_{m,t}$, where $\theta^T_{m,t}$ is the parameters of m^{th} teacher model at t^{th} batch, and $\theta_{m,t}$ is the parameters of m^{th} student model. At t^{th} batch, the weight-combined prediction of teacher models is: $p^T(x_i) = \sum_{m=1}^{M} \alpha_m T_m(x_i)$. Then, we force the predictions of student models to be consistent with that of teacher models by:

$$\mathcal{L}_{con}(x_i) = \mathbb{E}_{x_i \sim \mathcal{B}} \mathcal{L}_{ce}(\hat{y}(x_i), p(x_i)) \tag{4}$$

where $\mathcal{L}_{ce}(\cdot,\cdot)$ is cross-entropy loss and $\hat{y}(x_i) = \arg\max p^T(x_i)$ is pseudo-label by teacher models.

4.3 OPTIMIZATION

To sum up, given a batch of unlabeled samples $\mathcal{B} = \{x_i\}_{i=1}^n$, the total training losses are,

$$\min_{\theta_1,\ldots,\theta_m} \mathbb{E}_{x_i \sim \mathcal{B}} \mathcal{L}(x_i) = \mathcal{L}_{ent}(x_i) + \beta \mathcal{L}_{con}(x_i) \tag{5}$$

where β is the trade-off parameter. In each step, we first update the statistics information and model weight on a batch of data. Then, all learnable parameters are updated by the gradient of the total losses $\nabla\mathcal{L}(x_i)$, during the backward pass. Finally, the updated models and model weights are used for the prediction for the current batch. For online adaptation, no termination is necessary, and iteration continues as long as there is test data. At inference, the combination of student models is used to make predictions and the evaluation is performed online.

5 EXPERIMENTS

Datasets Existing TTA methods concentrate on a single-source scenario. For these methods, some classical datasets such as CIFAR-10, CIFAR-100, CIFAR-10-C, and CIFAR100-C are used for evaluation Wang et al. (2021). Typical adaptation tasks are CIFAR-10→CIFAR-10-C and CIFAR-100→CIFAR-100-C. However, these datasets could not be used for evaluation in our work as there are only two domains. Thus, we are inspired by multi-source domain adaptation (MSDA) methods Peng et al. (2019), and adopt the widely used datasets in MSDA methods, e.g. Office-Home, and Digit-Five.

Baselines We compare our method with a wide range of baselines. The first one is the source-only method, where no adaptation is performed and the trained models are directly applied to the test samples. We also compare against the **source-best** and the **source-worst** method. The second one is the single-source test-time adaptation methods, including TENT Wang et al. (2021), and MEMO Zhang et al. (2021). As these methods only utilize one source domain for adaptation, We compare against the best adapted model and the worst one, denoted as **TENT-best**, **TENT-worst**, **MEMO-best**, and **MEMO-worst**. We also compare against a multi-source extension of these methods via ensembling with equal weight. We name these methods as **MEMO-Ens**, and **TENT-Ens**. We also extend the source-only method with the same strategy, denoted as **source-only-Ens**. Lastly, we compare **MUTE** against multi-source domain adaptation methods, including M^{3}SDA-β Peng et al. (2019), DAN Long et al. (2015), DANN Ganin et al. (2016), MCD Saito et al. (2018), CORAL Sun et al. (2016), ADDA Tzeng et al. (2017), and DCTN Xu et al. (2018). Note that the MSDA methods need to access source data during adaptation.

	Office-31	Office-Home
MUTE-Ens	86.8	66.5
MUTE	**88.2**	**67.5**

Table 3: Ablation of model weights.

$\mathcal{L}_{ent}$	$\mathcal{L}_{con}$	Office-31	Office-Home
×	×	84.9	64.8
✓	×	86.5	65.6
✓	✓	**88.2**	**67.5**

Table 4: Ablation of losses.

Results The results on **Digital** dataset are shown in Table 1. Firstly, the test performance of different trained models varies greatly. For example, on the SVHN dataset as the test samples, the source-best model outperforms the source-worst model by 65.8%, which verifies that different models contribute differently to the test samples. Secondly, among single-source test-time adaptation baselines, MEMO-best and TENT-best achieve better results than source-best by 1.6% and 2.8% respectively, which shows that adapting the models with unlabeled samples could effectively boost the test performance. Thirdly, for multi-source test-time adaptation methods, Combing single-source test-time adaptation methods via uniform ensembling (MEMO-Ens, TENT-Ens) underperforms the corresponding best-performed source. The proposed method could avoid this by weighted aggregation and jointly training and outperforms MEMO-Ens and TENT-Ens by 6.0% and 1.2%, respectively. And the average increase across all digit tasks over MEMO-Ens and TENT-Ens is 6.0% and 4.2%, respectively. Besides, MUTE also achieves competitive results at par with the best-adapted source and even achieves better in MNIST-M dataset. Fourthly, MUTE outperforms some multi-source unsupervised domain adaptation methods that not only access the source samples but perform offline training, e.g., MUTE outperforms M^{3}SDA-β by 0.3%.

The results on **Office-Home** dataset is shown in Table 2. Similar results with digits datasets are also observed. The proposed method achieves the best results and the increase over MEMO-Ens and TENT-Ens are 2.2% and 1.8%. The lower performance increase than that of the digital dataset could be attributed to the relatively small performance gap between the best and worst unadapted sources.

Source	Method	Ar,Cl,Pr → Rw	Ar,Cl,Rw → Pr	Ar,Pr,Rw → Cl	Cl,Pr,Rw → Ar	Avg.
Single	Source-best	72.7	75.6	46.7	65.0	65.0
	Source-worst	64.7	62.4	40.0	51.7	54.7
	MEMO-best	73.7	75.8	47.5	65.2	65.6
	MEMO-worst	62.8	62.6	42.2	52.1	54.9
	TENT-best	74.6	**76.3**	51.0	**65.7**	66.9
	TENT-worst	61.5	63.6	45.4	53.7	56.1
Multiple	Source-only-Ens	75.9	72.1	47.0	64.2	64.8
	MEMO-Ens	75.7	72.4	48.5	64.4	65.3
	TENT-ens	75.3	72.8	50.0	64.8	65.7
	MUTE	**77.8**	75.1	**51.3**	**65.7**	**67.5**

Table 2: Results on Office-Home.

5.1 Insight Analysis

Ablation study of model weights To show the effectiveness of the weighted aggregation scheme and the weight estimation strategy, we design a variant of MUTE, denoted as MUTE-Ens, where each trained model is assigned the same weight and trained with two introduced losses. The comparison results on the Office-31 and Office-Home datasets are shown in Table 3. Without a weighted scheme, the performance dropped by 1.4% and 1.0% on the Office-31 and Office-Home datasets, respectively. The degeneration reveals that combing the models with proper weight could benefit the test samples and the estimation strategy could effectively reveal the relevance of each model.

Ablation study of losses During adaptation, two introduced unsupervised losses are used. To evaluate the importance of each loss, we compare our method with variants trained by different combinations of losses. The results of the Office-31 and Office-Home datasets are shown in Table 4. The first line represents the results of source-only-Ens. After adding weighted entropy loss, the performance is increased by 1.6% and 0.8% on digit and Office-Home, respectively. And the combination of all losses achieves the best results. As the adopted losses are very related to the main classification task, thus they could effectively adapt the model with only unlabeled samples.

6 Conclusion

In this work, we propose the problem of multi-source fully test-time adaptation (MS-FTTA), where many trained models are adapted to the test samples. Moreover, to solve this problem, a weighted aggregation scheme is adopted to combine these source models with different weights and two weighted unsupervised losses are proposed to jointly adapt the models. We conducted experiments on three image datasets, and the results show the effectiveness of the proposed method. In the future, we would extend to foundation models by parameter-efficient fine-tuning Xin et al. (2024).

REFERENCES

Sk. Miraj Ahmed, Dripta S. Raychaudhuri, S. Paul, Samet Oymak, and A. Roy-Chowdhury. Unsupervised multi-source domain adaptation without access to source data. *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

Shai Ben-David, John Blitzer, K. Crammer, A. Kulesza, Fernando C Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Machine Learning*, 79:151–175, 2009.

Collin Burns and Jacob Steinhardt. Limitations of post-hoc feature alignment for robustness. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2525–2533, 2021. URL https://api.semanticscholar.org/CorpusID:232170131.

Dian Chen, Dequan Wang, Trevor Darrell, and Sayna Ebrahimi. Contrastive test-time adaptation. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 295–305, 2022.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, K. Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

Yuntao Du, Haiyang Yang, Mingcai Chen, Hongtao Luo, Juan Jiang, Yi Xin, and Chongjun Wang. Generation, augmentation, and alignment: A pseudo-source domain based method for source-free domain adaptation. *Machine Learning*, pp. 1–21, 2023.

Yulu Gan, Xianzheng Ma, Yihang Lou, Yan Bai, Renrui Zhang, Nian Shi, and Lin Luo. Decorate the newcomers: Visual domain prompt for continual test time adaptation. *AAAI*, 2023.

Yaroslav Ganin and V. Lempitsky. Unsupervised domain adaptation by backpropagation. In *ICML*, 2015.

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor S. Lempitsky. Domain-adversarial training of neural networks. *JMLR*, 2016.

Taesik Gong, Jongheon Jeong, Taewon Kim, Yewon Kim, Jinwoo Shin, and Sung-Ju Lee. Note: Robust continual test-time adaptation against temporal correlation. In *Neural Information Processing Systems*, 2022. URL https://api.semanticscholar.org/CorpusID:252846066.

Yves Grandvalet and Yoshua Bengio. Semi-supervised learning by entropy minimization. In *Advances in Neural Information Processing Systems*, 2004.

Jiayi Han, Longbin Zeng, Liang Du, Weiyang Ding, and Jianfeng Feng. Rethinking precision of pseudo label: Test-time adaptation via complementary learning. *ArXiv*, abs/2301.06013, 2023.

Kaiming He, Xiangyu Zhang, et al. Deep residual learning for image recognition. *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

Dan Hendrycks and Thomas G. Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *ICLR*, 2019.

Judy Hoffman, Mehryar Mohri, and Ningshan Zhang. Algorithms and theory for multiple-source adaptation. In *NeurIPS*, 2018.

M-U Jang and Sae-Young Chung. Test-time adaptation via self-training with nearest neighbor information. *ICLR*, 2023.

Pang Wei Koh, Shiori Sagawa, Henrik Marklund, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard L. Phillips, Sara Beery, Jure Leskovec, Anshul Kundaje, Emma Pierson, Sergey Levine, Chelsea Finn, and Percy Liang. Wilds: A benchmark of in-the-wild distribution shifts. In *ICML*, 2021.

V. Kurmi, Venkatesh K. Subramanian, and Vinay P. Namboodiri. Domain impression: A source data free domain adaptation method. *WACV*, pp. 615–625, 2021.

Keqiuyin Li, Jie Lu, Hua Zuo, and Guangquan Zhang. Source-free multi-domain adaptation with fuzzy rule-based deep neural networks. *IEEE Transactions on Fuzzy Systems*, 2023.

Rui Li, Qianfen Jiao, et al. Model adaptation: Unsupervised domain adaptation without source data. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

Jian Liang, Dapeng Hu, et al. Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation. In *International Conference on Machine Learning (ICML)*, 2020.

Yuejiang Liu and et al. Ttt++: When does self-supervised test-time training fail or thrive? In *NeurIPS*, 2021.

Mingsheng Long, Yue Cao, Jianmin Wang, and Michael I. Jordan. Learning transferable features with deep adaptation networks. *International Conference on Machine Learning (ICML)*, 2015.

Mingsheng Long, Zhangjie Cao, et al. Conditional adversarial domain adaptation. In *Advances in Neural Information Processing Systems*, 2018.

M. Jehanzeb Mirza, Jakub Micorek, Horst Possegger, and Horst Bischof. The norm must go on: Dynamic unsupervised domain adaptation by normalization. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14745–14755, 2021. URL `https://api.semanticscholar.org/CorpusID:244773006`.

Shuaicheng Niu and rt al. Efficient test-time model adaptation without forgetting. In *ICML*, 2022.

Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *TKDE*, 22:1345–1359, 2010.

Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. *Proceedings of the IEEE/CVF international conference on computer vision (ICCV)*, 2019.

Kate Saenko, Brian Kulis, et al. Adapting visual category models to new domains. In *European Conference on Computer Vision (ECCV)*, 2010.

Kuniaki Saito, Kohei Watanabe, et al. Maximum classifier discrepancy for unsupervised domain adaptation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

Baochen Sun, Jiashi Feng, and Kate Saenko. Return of frustratingly easy domain adaptation. In *AAAI conference on artificial intelligence*, 2016.

Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *NeurIPS*, 2017.

Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

Jesper E. van Engelen and Holger H. Hoos. A survey on semi-supervised learning. *Machine Learning*, 109:373–440, 2019.

Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno A. Olshausen, and Trevor Darrell. Tent: Fully test-time adaptation by entropy minimization. In *ICLR*, 2021.

Qin Wang, Olga Fink, Luc Van Gool, and Dengxin Dai. Continual test-time domain adaptation. *CVPR*, 2022.

Yi Xin, Siqi Luo, Pengsheng Jin, Yuntao Du, and Chongjun Wang. Self-training with label-feature-consistency for domain adaptation. In *International Conference on Database Systems for Advanced Applications*, 2023.

Yi Xin, Siqi Luo, Haodi Zhou, Junlong Du, Xiaohong Liu, Yue Fan, Qing Li, and Yuntao Du. Parameter-efficient fine-tuning for pre-trained vision models: A survey. *arXiv preprint arXiv:2402.02242*, 2024.

Ruijia Xu, Ziliang Chen, Wangmeng Zuo, Junjie Yan, and Liang Lin. Deep cocktail network: Multi-source unsupervised domain adaptation with category shift. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3964–3973, 2018.

Li Yi, Gezheng Xu, Pengcheng Xu, Jiaqi Li, Ruizhi Pu, Charles Ling, Allan Mcleod, and Boyu Wang. When source-free domain adaptation meets learning with noisy labels. *ICLR*, 2023.

Marvin Zhang, Sergey Levine, and Chelsea Finn. Memo: Test time robustness via adaptation and augmentation. *ArXiv*, abs/2110.09506, 2021.

Yuchen Zhang, Tianle Liu, et al. Bridging theory and algorithm for domain adaptation. In *International Conference on Machine Learning (ICML)*, 2019.

Bowen Zhao, Chen Chen, and Shutao Xia. Delta: degradation-free fully test-time adaptation. *ICLR*, 2023.

A RELATED WORK

Comparison with related problem settings is shown in Table 5 and the following are detailed analysis:s

- Compared with **fine-tuning**, MS-FTTA and other problem settings only need unlabeled target samples to train or adapt the models, without the cost of labeling samples.
- Compared with conventional **single-source or multi-source unsupervised domain adaptation and test-time training**, MS-FTTA does not need to access the source samples or modify the training process. Thus it protects data safety and privacy.
- Comparison with **multi-source source-free domain adaptation** (MS-SFDA): MS-SFDA methods perform offline training, where target samples are given in advance and we could train the network with multiple epcohs. On the contrary, MS-FTTA needs to perform online adaptation, where the models need to update and predict immediately when a batch of test samples arrives, which is more difficult than offline adaptation and prediction.
- Compared with **fully test-time adaptation**, multiple trained models are available in MS-FTTA and more useful information could be used for the test samples by combing the models. Besides, MS-FTTA does not assume to know which one is the best model.

Table 5: Comparison with different problem settings that adapt a trained model to a potentially shifted test domain. 'Online' means that adaptation can predict a batch of incoming test samples immediately. M is the number of domains/trained models.

Setting	Source Data	Number of Source Domains	Target Data	Training Loss	Testing Loss	Online
Fine-tuning	×	1	x^t, y^t	$\mathcal{L}(x^t, y^t)$	–	×
Unsupervised domain adaptation	x^s, y^s	1	x^t	$\mathcal{L}(x^s, y^s) + \mathcal{L}(x^s, x^t)$	–	×
Multi-source domain adaptation	x^s, y^s	M	x^t	$\mathcal{L}(x^s, y^s) + \mathcal{L}(x^s, x^t)$	–	×
Test-time training	x^s, y^s	1	x^t	$\mathcal{L}(x^s, y^s) + \mathcal{L}(x^s)$	$\mathcal{L}(x^t)$	✓
Multi-source source-free domain adaptation	×	M	x^t	$\mathcal{L}(x^t)$	–	×
Fully test-time adaptation	×	1	x^t	×	$\mathcal{L}(x^t)$	✓
Multi-source fully test-time adaptation (ours)	×	M	x^t	×	$\mathcal{L}(x^t)$	✓

B EXPERIMENTS

Setup Our method is implemented by PyTorch with A100. For source training, we use DTN Long et al. (2018) architecture for digits and ResNet-50 He et al. (2016) pre-trained on ImageNet Deng et al. (2009) as the feature extractor for Office-31 and Office-Home datasets. The classifier is made up of randomly initialized fully connected layers. During adaptation, we use SGD as the optimizer, with a momentum of 0.9. The batch size and the learning rate are set to be 128,64,48 and 0.005,0.005,0.00025 for Digits, Office-31, and Office-Home. The update parameter λ in Equ. 2 is chosen from $\{0.1, 0.2, 0.5\}$ and set to be 0.1 for digits and Office-31 and 0.5 for Office-Home. The trade-off parameter β in Equ. 5 is chosen from $\{0.5, 1.0, 2.0\}$ and set to be 1.0 for all datasets. Note that we focus on online settings, where the samples are adapted and evaluated in each batch. The pseudo-code is shown in Algorithm 1

- **Digits:** Digital dataset contains five datasets of 10 categories: **USPS** contains 7,438 images. **MNIST** is composed of 55,000 images, and **MNIST-M** Ganin & Lempitsky (2015) also consists of 55,000 images. **SVHN** is composed of 73,257 images and **SynthDigits Digits** Ganin & Lempitsky (2015) consists of 25,000 images. The images in USPS, MNIST, and MNIST-M are gray while the images in SVHN and DIGITS are in color. We construct five tasks on this dataset.
- **Office-31 Saenko et al. (2010):** Office-31[1] is a classical real-world dataset for domain adaptation. It has 4110 images with 31 classes shared with three domains: Amazon (**A**), Webcam (**W**), and DSLR (**D**). In this dataset, we contrust three adaptation tasks, i.e., A,D →W, A,W→D and D,W→A.
- **Office-Home Peng et al. (2019):** Office-Home[2] is a larger dataset, which consists of four domains: Artistic (**Ar**), Clipart (**Cl**), Product (**Pr**), and Real-World (**Rw**), containing 15500 images with 65 classes. Four transfer tasks are constructed, i.e., Ar,Cl,Pr →Rw, ..., and Cl,Pr,Rw →Ar.

[1] https://faculty.cc.gatech.edu/~judy/domainadapt/
[2] https://www.hemanthdv.org/officeHomeDataset.html

Algorithm 1 MUTE.

Input: Online batch samples $\mathcal{B}$; Trained source models $\{F_m\}_{m=1}^M$; Source statistics $\{\mu_m\}_{m=1}^M$ and $\{\Sigma_m\}_{m=1}^M$.
Output: Online prediction results $\hat{y}$;

▷ Calculate online batch data statistic
$\mu_m^t \leftarrow \sum_{i=1}^{|\mathcal{B}_t|} z_i^m$
$\Sigma_m^t \leftarrow \frac{1}{|\mathcal{B}_t|}(Z_m^T Z_m - (I^T Z_m)^T (I^T Z_m))$
▷ Estimate model weights
for m $\leftarrow$ 1 to M **do**
  $d_m \leftarrow ||\mu_m - \mu_m^t||^2 + ||\Sigma_m - \Sigma_m^t||^2$
end for
for m $\leftarrow$ 1 to M **do**
  $\alpha_m = \frac{\exp^{-d_m}}{\sum_{i=1}^M \exp^{-d_i}}$
end for
▷ Model prediction
for x_i in $\mathcal{B}$ **do**
  for m $\leftarrow$ 1 to M **do**
    $p(x_i) = \sum_{m=1}^M \alpha_m F_m(x_i)$ # student Model
    $p^T(x_i) = \sum_{m=1}^M \alpha_m T_m(x_i)$ # teacher Model
  end for
end for
▷ Update model and global statistics
$\{F_m\}_{m=1}^M \leftarrow \min_{\theta_1,\ldots,\theta_m} \mathbb{E}_{x_i \sim \mathcal{B}}(\mathcal{L}_{ent}(x_i) + \beta \mathcal{L}_{con}(x_i))$
$\mu_m = \lambda \mu_m + (1-\lambda)\mu_m^t$
$\Sigma_m = \lambda \Sigma_m + (1-\lambda)\Sigma_m^t$
▷ Inference
for x_i in $\mathcal{B}$ **do**
  for m $\leftarrow$ 1 to M **do**
    $p(x_i) = \sum_{m=1}^M \alpha_m F_m(x_i)$ # student Model
    $\hat{y} = \arg\max p(x_i)$
  end for
end for
Return Online prediction results $\hat{y}$.

Source	Method	A,D → W	A,W → D	D,W → A	Avg.
Single	Source-best	97.5	99.8	66.2	87.8
	Source-worst	78.1	81.9	64.0	74.7
	MEMO-best	97.5	99.3	66.4	87.7
	MEMO-worst	78.9	82.7	61.8	74.5
	TENT-best	**97.9**	**100**	66.0	88.0
	TENT-worst	79.4	83.7	62.7	75.3
Multiple	Source-only-Ens	95.6	96.4	62.6	84.9
	MEMO-Ens	95.9	96.7	62.8	85.1
	TENT-Ens	96.0	96.7	64.0	85.6
	MUTE	97.5	99.1	**67.9**	**88.2**

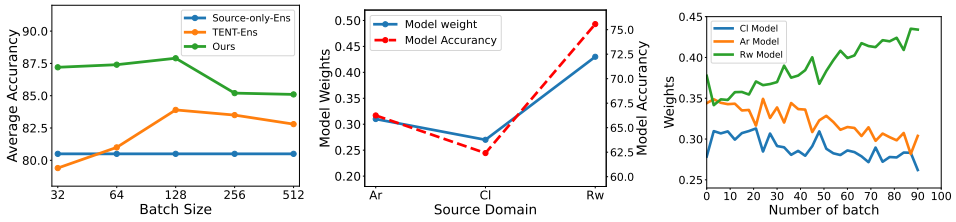
Table 6: **The results on Office-31**

C MORE RESULTS

Results on Office-31 Table 6 shows the results on **Office-31** dataset. As can see, among the single-source test-time adaptation method, the average performance of MEMO-best is slightly worse than source-best, which implies the instability of this method. It is also noticed that on task A,W→D, source-best achieves 99.8% and there is nearly no room for improvement. Although TENT-best achieves the best results in two tasks, but it is difficult to determine which is the best performed model in real-world scenario. On the contrary, our method achieves the best results on average and outperforms MEMO-Ens and TENT-Ens by 3.1% and 2.6%. Our method also achieves competitive results at par with the best-adapted model.

Performance under different number of samples in a batch As we focus on online adaptation, the number of samples in a batch (characterized by batch size) could also affect the adaptation performance. To evaluate the proposed method, the results with different batch size on digital datasets for different methods are shown in Figure 2a. We can see that the performance of all methods except source-only-Ens varies with different batch size and our method always achieves the best results under different numbers of samples.

Visualization of model weights To show the effectiveness of estimated model weights, the final model weights and the accuracy of the unadapted model (source-only) on task Ar,Cl,Rw $\rightarrow$ Rr are drawn together in Figure 2b. As we can see, the model with better-updated model accuracy has a higher weight eventually, which shows that the proposed strategy could effectively recognize the relevant model. And this strategy may be a possible method to select models. Moreover, the change of model weights during the adaptation is shown in Figure 2c. From the overall trend, the weight of the relevant model is increased and the weight of the irrelevant model is decreased with the increase of test samples.



(a) Effect of the number of samples in a batch.

(b) Relation between model weight and unadapted model accuracy.

(c) Changes of weight with the increase of batches.

Figure 2: Insight analysis of the proposed method.

Analysis of adaptation modules In TENTWang et al. (2021), only the parameters of BN layers are adapted. We argue that TENT has a strong assumption on the parameters and we instead adapt all parameters. We compare both strategies and the results on the Office-31 and Office-Home datasets are shown in Table 7. The results show that adapting all parameters achieves better than that of only adapting BN layers as the former could offer a larger adaptation space and model capacitance.

	Office-31	Office-Home
Only BN	86.9	66.3
All networks	**88.2**	**67.5**

Table 7: Analysis of adaptation modules.

Analysis of Computational Efficiency Table 8 shows the computational efficiency, we compare the number of forward and backward passes. The pass number is calculated on Office-Home datasets (15,588 samples and 3 given models). As we can see, Source-only-Ens requires the least pass numbers and could be approximatively seen as a lower bound. The forward pass number of MUTE is slightly more than TENT-Ens(both teacher and student are updated) but they have the same backward pass number(only the student model is updated). Compared with MEMO-Ens, the computational efficiency of MUTE is significantly improved.

	#forward	#backward
Source-only-Ens	15,588×3	0
MEMO-Ens	15,588×3×16 (Augmentation times)	15,588×3×16
TENT-ens	15,588×3	15,588×3
MUTE	15,588×3×2 (Teacher and Student)	15,588×3 (Only student)

Table 8: Analysis of computational efficiency on Office-Home datasets.