Adaptive Boosting for Domain Adaptation: Towards Robust Predictions in Scene Segmentation

Zhedong Zheng and Yi Yang, *Senior Member, IEEE*

***Abstract*—Domain adaptation is to transfer the shared knowledge learned from the source domain to a new environment, *i.e.*, target domain. One common practice is to train the model on both labeled source-domain data and unlabeled target-domain data. Yet the learned models are usually biased due to the strong supervision of the source domain. Most researchers adopt the early-stopping strategy to prevent over-fitting, but when to stop training remains a challenging problem since the lack of the target-domain validation set. In this paper, we propose one efficient bootstrapping method, called Adaboost Student, explicitly learning complementary models during training and liberating users from empirical early stopping. Adaboost Student combines deep model learning with the conventional training strategy, *i.e.*, adaptive boosting, and enables interactions between learned models and the data sampler. We adopt one adaptive data sampler to progressively facilitate learning on hard samples and aggregate "weak" models to prevent over-fitting. Extensive experiments show that (1) Without the need to worry about the stopping time, AdaBoost Student provides one robust solution by efficient complementary model learning during training. (2) AdaBoost Student is orthogonal to most domain adaptation methods, which can be combined with existing approaches to further improve the state-of-the-art performance. We have achieved competitive results on three widely-used scene segmentation domain adaptation benchmarks.**

***Index Terms*—Domain Adaptation, Scene Segmentation.**

I. Introduction

In recent years, deep learning approaches have achieved significant improvement in many computer vision fields, including semantic segmentation [1], [2]. However, improving the scalability of deeply-learned models remains a challenging task. For instance, the segmentation models learned on the data collected on sunny days usually perform terribly in different environments, such as rainy days and foggy days [3], [4]. One straightforward idea is to collect more training data for the target environment and re-train one domain-specific model for inference. However, it is usually unaffordable to annotate large-scale datasets for every target scenario, especially for the tasks demanding pixel-wise annotations, *e.g.*, scene segmentation. Therefore, researchers resort to domain adaptation techniques [4], [5], [6] to "borrow" the common knowledge from homogeneous datasets, *e.g.*, labeled source-domain data. The source-domain data can be the synthetic data generated by game engines, such as GTA5 [6], [7], or the large-scale data collected in other real-world scenarios [8]. In this way, we can largely save the annotation cost as well as the training time to achieve competitive results in the target domain. The common practice is to train the segmentation model on both labeled source-domain data and unlabeled target-domain data. Most existing works focus on domain alignment, minimizing the gap between the source domain and the target domain [9], [10], [11]. Several early attempts leverage generative models to transfer the image style at the pixel level [12], [13], while other works concentrate on narrowing the semantic gap in the feature space [14]. For instance, Tsai *et al.* [15], [16] and Luo *et al.* [17], [18] harness the adversarial loss to encourage the segmentation model to learn domain-invariant features.

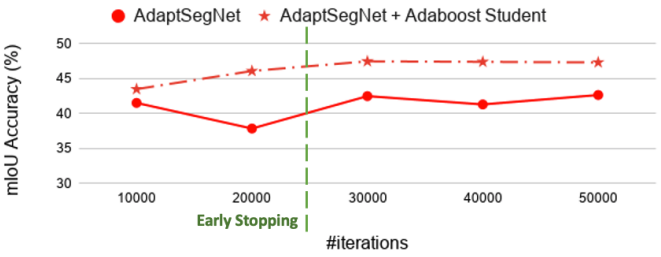


Fig. 1. The sensitivity analysis of the model on the GTA5 [7] → Cityscapes [8] benchmark in the first 50000 iterations on the target-domain test set. We re-implement the widely-used method AdaptSegNet [15], and observe the prediction fluctuation at different training iterations. The proposed Adaboost Student makes the model free from the early-stopping trick and can efficiently converge to one competitive performance on the target domain. For instance, the empirical early-stopping time is usually set at the 25,000-th iteration [19], which is sub-optimal.

However, one inherent problem exists in the scene segmentation domain adaptation. Due to the strong supervision on the source-domain label, we observe that the model is still prone to over-fitting the source domain, leading to the unstable prediction on the target domain (see Figure 1). Although several researchers adopt the early-stopping strategy to prevent over-fitting [17], [18], [19], when to stop the training remains challenging. The unstable prediction not only compromises the method re-implementation, but also harms users to select the final model in the real-world application.

In an attempt to overcome the above-mentioned challenge, this paper adopts the spirit of one conventional training strategy, *i.e.*, adaptive boosting [20], with the deeply-learned model to facilitate learning complementary models and progressively improve the model scalability. In particular, we gradually provide more sampling probability to the "hard" samples that achieve high prediction variance in the current model. In this way, we can efficiently obtain several complementary models during one-time training rather than training

Zhedong Zheng is with the School of Computing, National University of Singapore, Singapore 118404. E-mail: zdzheng@nus.edu.sg

Yi Yang is with the College of Computer Science and Technology, Zhejiang University, China 310027. E-mail: yangyics@zju.edu.cn. He is in part supported by the Fundamental Research Funds for the Central Universities (No. 226-2022-00087).

Yi Yang is the corresponding author.

multiple independent models as the conventional adaptive boosting [20]. Aggregating such model snapshots provide more robust predictions on unlabeled target domain data and, more importantly, prevents the model from over-fitting. As a result, we observe a consistent improvement over other domain adaptation approaches on three prevailing benchmarks. To summarize, our contributions are:

- an efficient bootstrapping method for scene segmentation domain adaptation, AdaBoost Student, which enables interactions between learned models and the data sampler, and aggregates the "weak" model snapshots to prevent over-fitting. AdaBoost Student makes users free from the empirical early stopping trick; and
- an adaptive data sampler to progressively increase the sampling probability of hard samples with ambiguous predictions, facilitating the complementary model learning. The criterion of hard sample selection is based on prediction variance in unlabeled target-domain data; and
- a demonstration that the proposed approach has a consistent improvement over existing methods on two synthetic-to-real benchmarks and one cross-city benchmark.

The rest of this paper is organized as follows. Section II discusses relevant works, including semantic segmentation adaptation, bootstrapping learning, and hard example mining. Section III elaborates on the proposed AdaBoost Student approach followed by experiment results in Section IV. We add further ablation studies and discussion in Section V. Finally, Section VI concludes the paper.

II. Related Work

A. Semantic Segmentation Adaptation

There are two big families of segmentation adaptation. One line of works [15], [16], [17], [21], [22], [23], [10] concentrates on the domain alignment. Some pioneering works [12], [13] leverage the CycleGAN [24] to transfer the image style to the target domain, minimizing the biases from the source domain. Taking one step further, Wu *et al.* [3] and Yue *et al.* [25] transfer the input images into different styles, such as rainy and foggy, to learn domain-invariant features. In contrast, several works do not change the image style at the pixel level but leverage the adversarial loss to disentangle the shared knowledge and minimize the negative impact of the domain-specific information. For instance, Tsai *et al.* [15] and Luo *et al.* [18] demand the generator to fool the discriminator by discriminating the feature between the source domain and target domain. Sankaranarayanan *et al.* [26] leverage the image reconstruction as the self-supervision task to align the two domains, *i.e.*, the real-world environment and synthetic data, yielding a scalable learned model, but it also costs more computation resources for recovering the large-size input image. Another line of works [27], [28], [19], [29], [30], [31] focuses on mining the domain-specific knowledge in the target domain. The main idea is to minimize the prediction entropy in the target domain [32], [33], [34]. Zou *et al.* [27], [28] propose to leverage the pseudo labels with high confidence, and discuss different regularization terms. Taking one step further, Feng *et al.* [35] and Lin *et al.* [36] leverage the group information to acquire pseudo labels with high precision. Zheng *et al.* [19] apply one two-step training strategy, which first conducts the domain alignment to generate better pseudo labels according to multi-level features [37], and then applies the pseudo label learning. Furthermore, Zheng *et al.* [38] rectify the pseudo label learning by explicitly combining the prediction uncertainty into the cross-entropy loss. In this paper, we notice that existing works, to some extent, still suffer from over-fitting and early-stopping tricks, leading to large prediction variance and re-implementation difficulties. We propose an efficient bootstrapping method, which is orthogonal to most existing approaches on minimizing the domain gap. In fact, the proposed method can be fused with existing methods to further improve the model scalability.

B. Bootstrapping Learning

Bootstrapping learning denotes the capability of the model to progressively improve the performance by teaching itself [39]. The teacher-student training policy [40] is first proposed for model distillation, which intends to distill the knowledge from large-scale models to mobile models. It demands the parameter-efficient student model to predict consistent prediction scores with the large teacher model. However, teacher-student policy needs one pre-trained sophisticated network as the teacher model in advance, which is not always available. To address this limitation, Zhang *et al.* [41] introduce deep mutual learning to simultaneously train two models of different structures from scratch and learn from each other. Different from preserving two models during training, Laine *et al.* [42] propose Π model to enable the model learning from the model itself, which demands consistent prediction on the original input and the augmented input. Furthermore, Laine *et al.* also explore the temporal ensembling, which leverages the prediction history to stabilize the training. To address the out-of-the-date prediction in history, Tarvainen *et al.* [43] and Choi *et al.* [44] propose to maintain one mean model by moving average weight [45], which does not need to train one new model again. Similarly, Chen *et al.* [46] and Zheng *et al.* [19] introduce the memory mechanism to help the feature update, facilitating consistency learning. We are mainly different from existing bootstrapping learning approaches in two aspects: 1) We do not introduce more prerequisites, *e.g.*, extra memory modules or an independent teacher model. Instead, we gradually aggregate the complementary model during training. The model can converge efficiently, and achieve competitive performance in the target domain quickly; 2) We explicitly consider the model weakness and adopt one adaptive data sampler, which focuses on the "hard" samples and facilitates learning complementary models in a coherent manner.

C. Hard Example Mining

The neural networks usually underestimate the minority, *i.e.*, hard examples, during the mini-batch optimization [47], leading to over-fitting most "easy" samples. Therefore, several

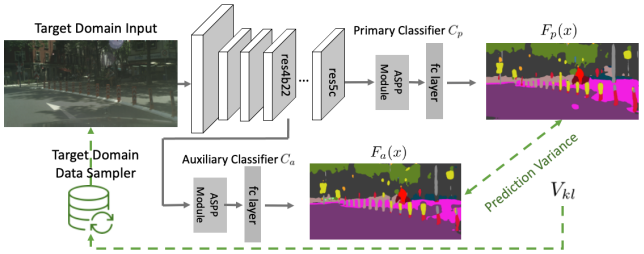


Fig. 2. Illustration of the basic model structure (Gray Parts). We follow most existing works [15], [17], [18], [16], [19], [38] to deploy the modified Deeplab-v2 [2] with ResNet-101 [51] as backbone. The auxiliary classifier is added to help the optimization, preventing gradient vanishment, especially for the lower layers [52], [1]. In this work, we focus on the outer loop (Green Parts). We leverage the discrepancy of the segmentation outputs from the primary classifier C_p and C_a to acquire the prediction variance V_{kl}. The variance is then utilized to update the data sampler. More details can be found in Section III-A.

researchers resort to example mining strategies to add more emphasis on the example with large losses. Most existing works focus on the object detection task. For instance, Shrivastava *et al.* [48] let the network forward twice, called OHEM, to mine hard samples. OHEM searches hard examples from a large pool of 4000 regions in the first round, and then optimize the network by the selected samples, which are the top 128 high-loss regions. To evade the two-stage forward process, Lin *et al.* [47] propose to directly enlarge the punishment on the hard example by a modified cross entropy loss, which also yields performance improvement. In contrast, many metric learning methods [49], [50] do not directly modify the loss, but select the input triplets with the largest loss, such as the farthest positives and the closet negatives to help the model learn the distance (relation) in the semantic space. In this work, we do not have the target-domain label, and could not foreknow the wrong samples. Therefore, we resort to the prediction variance as the hardness indicator. In fact, the proposed method is orthogonal to existing work, including focal loss and OHEM (using cross entropy as the hardness indicator). There are three main differences between our work and OHEM: (1) OHEM needs to feed forward the network twice in every iteration, while the proposed method conducts the inference once. We only update the data sampler after every epoch, which is relatively efficient. (2) OHEM is based on cross-entropy loss with ground-truth labels. In our work, we do not have the target-domain ground-truth labels. Therefore, we resort to the prediction variance, and as shown in Table II, the prediction variance surpasses the prediction entropy. (3) We further leverage the diversity between different epochs to conduct student aggregation. In the experiment, we also add ablation studies on the compatibility with existing methods.

III. Method

Formulation. Given the labeled source-domain dataset $X_m = \{x_m^i\}_{i=1}^M$ and the unlabeled target-domain dataset $X_n = \{x_n^j\}_{j=1}^N$, scene segmentation domain adaptation is to learn the projection function F, which maps the input data X to the segmentation map Y. M and N denote the number of the source-domain images and the target-domain images, respectively. Following the common practice in [15], [17],

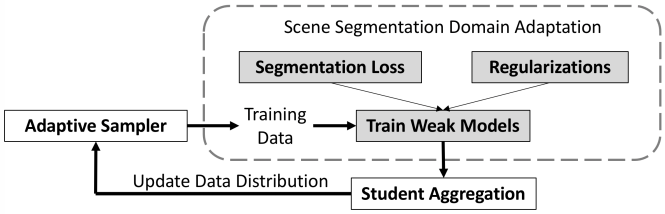


Fig. 3. The brief pipeline of the proposed method. There are two main components, *i.e.*, Adaptive Sampler and Student Aggregation. We modify the training data distribution to learn complementary "weak" models, preventing the model from over-fitting. **Different from existing methods, the pipeline enables interactions between learned models and the data sampler.** The proposed method is orthogonal to most existing scene segmentation domain adaptation approaches (in the rounded rectangle).

[19], we adopt the modified DeepLabv2 [2] as the baseline model, which contains two classifiers, *i.e.*, the primary classifier C_p and the auxiliary classifier C_a. The auxiliary classifier design is to help optimization and prevent the gradient from vanishing [52], [1]. In this work, we also involve the auxiliary prediction into the calculation of the prediction variance V_{kl}. To simplify, we denote the two segmentation functions F_p and F_a, where $F_p(x)$ is the output of the primary classifier, and $F_a(x)$ is the output of the auxiliary classifier (see Figure 2).

Overview. In this work, we do not pursue a sophisticated segmentation structure or optimization losses. We focus on the learning strategy. As shown in Figure 3, we provide one brief pipeline to illustrate the proposed approach, AdaBoost Student. AdaBoost Student contains two primary components, *i.e.*, Adaptive Sampler and Student Aggregation. Adaptive Sampler modifies the input training data distribution, and provides more emphasis on uncertain samples. On the other hand, Student Aggregation is to accumulate the learned "weak" student models and output the prediction uncertainty to update data distribution. It is worth noting that there is one interaction loop from Adaptive Sampler to Student Aggregation, and back to Adaptive Sampler. We adopt one training policy similar to Generative Adversarial Network (GAN) [53]. During training, Student Aggregation updates model parameters when the input data distribution is fixed, and vice versa. We fix the aggregated student model parameter to update the data distribution of Adaptive Sampler. We will illustrate the two components extensively in the following sections.

A. Adaptive Sampler

Most bootstrapping works [39], [42] focus on mining the knowledge from the fixed data distribution but ignore the attention to the "hard" samples. The straightforward solution underpinning the conventional method, *i.e.*, Adaptive boosting [20], is to improve the sampling rate of the wrong-classified samples. In a similar spirit, we adopt one adaptive sampler to progressively update the input data distribution D_t according to the epoch t. The initial data distribution $D_1(j) = 1/N$, and every sample has equal sampling probability. To learn the complementary models for the target domain, we improve the sampling rate of the "hard" target-domain samples with ambiguous predictions during training. It is worth noting that, due to the missing label in the target-domain, we can not

deploy the prediction error to find the "hard" samples. Instead, we deploy the prediction variance as the "hard" indicator. Specifically, we formulate the prediction variance as the KL-divergence between the prediction of the primary classifier and the auxiliary classifier. The prediction variance of one single target-domain image can be calculated as:

$$V_{kl}(x_n^j|\theta) = \mathbb{E}[F_p(x_n^j|\theta)\log(\frac{F_p(x_n^j|\theta)}{F_a(x_n^j|\theta)})]. \quad (1)$$

If two classifiers predict the same class prediction, the value V_{kl} is low. Otherwise, the prediction variance is high, which indicates the model is not confident on such training data. In practice, since the output prediction is the pixel-wise segmentation map, we adopt the mean value of the whole prediction map as the indicator for the input image. To make V_{kl} as one distribution for the whole dataset, we apply the softmax function to normalize the prediction variance and make the sum of the V_{kl} equal to 1. The next data sampling distribution is averaged with the last data distribution and the normalized prediction variance as:

$$D_{t+1}(j) = \frac{1}{2}(D_t(j) + V_{kl}(x_n^j|\theta)). \quad (2)$$

Since $\sum_{j=1}^{N} D_t(j) = 1$ and $\sum_{j=1}^{N} V_{kl} = 1$, we ensure the update distribution $\sum_{j=1}^{N} D_{t+1} = 1$, and $D_{t+1}(j)$ is positive. As a result, the samples obtaining high prediction variance have more chances to be sampled in the next epoch.

Discussion. 1. Why is there a discrepancy between the primary classifier and the auxiliary classifier? It is mostly due to the different receptive fields of the two classifiers. Specifically, the input feature of the primary classifier is "res5c", while the auxiliary classifier learns from the "res4b22" of the ResNet-101 backbone [51]. We leverage the prediction variance to represent the model uncertainty on the input data as [38]. Besides, we insert the dropout function before classifiers, which also enlarges the prediction discrepancy and helps to obtain the accurate uncertainty [54]. **2. What are the advantages of the proposed adaptive sampler?** There are two main points. First, we do not need the target-domain label to estimate the prediction error, which is different from the conventional AdaBoost algorithm in supervised learning. Instead, we leverage the prediction uncertainty to find the "hard" samples, which are more feasible for the domain adaptation problem. Second, existing methods usually ignore the importance of the input data distribution. The "easy" target-domain data, which has high prediction confidence, generally does not provide more domain-specific information about the target domain. The adaptive sampler explicitly introduces the prediction variance into the data distribution update (see Eq. 2). We put more and more emphasis on the data with high uncertainty, facilitating the complementary model learning. The dynamic input distribution via the adaptive sampler encourages the knowledge transfer to the target domain efficiently. **3. Other sampling criterion.** Except for the prediction variance, prediction entropy is another alternative choice. In the ablation study, we also provide the empirical result based on the sampling criterion of the prediction entropy, which can be formulated as: $V_{en}(x_n^j|\theta) = \mathbb{E}[-F_p(x_n^j|\theta)\log(F_p(x_n^j|\theta))]$. A high value indicates that the prediction is close to the uniform distribution, and the model is not confident with the predicted classes given the input data. The samples obtaining high prediction entropy have more chances to be sampled in the next epoch. However, we observe one similar phenomenon with [38] that the prediction entropy is sensitive to the object edges in segmentation maps rather than the "hard" samples with ambiguous predictions, which need to be "seen" again. Therefore, the sampling criterion based on prediction entropy generally performs worse than the prediction variance in Eq. 1, which is verified in Table II.

B. Student Aggregation

In the conventional strategy, *i.e.*, Adaptive Boosting, multiple binary classifiers are sequentially learned with more attentions to the hard samples. Instead of training multiple deeply-learned models sequentially, we leverage the adaptive data sampler to enable learning complementary models in one-time training, which largely saves the training cost. Next, we combine the vote of "weak" model snapshots as one final model. We regard snapshots as student models but do not introduce extra teachers. For the deeply-learned models, it is not efficient to preserve all the model snapshots in the memory, and fuse the prediction of every model, which demands the multiple-time inference. We follow Mean Teacher [43] to apply the weight average moving to keep one weight-averaged model of previous epochs:

$$\Theta_t = \sum \alpha_t \times \theta_t, \quad (3)$$

where the model weight α_t assigns different emphasis to "weak" models, and θ_t denotes the parameters of the student model at the t-th epoch. Θ_t denotes the parameters of the aggregated model. Since we can not foreknow the segmentation error on the target domain, we simply adopt average weights, which can be online updated as:

$$\Theta_t = \frac{t-1}{t}\Theta_{t-1} + \frac{1}{t}\theta_t, \quad (4)$$

where $t \in \{2, ...T_1\}$ and $\Theta_1 = \theta_1$. We do not need to update the mean student model very often. In our practice, we update the mean model every $T_2 = 5000$ iterations. T_1 **is the total training epoch number, while** T_2 **is the iteration number of each epoch.** We find that the weight averaging still can achieve a competitive result.

Discussion. 1. How about computation costs? *Time Cost.* As shown in Eq. 4, only 2 DNN weights need to be considered during every update, and the weight averaging costs about one forward inference time, which can be neglected [55]. *Memory Cost.* The main memory cost is to preserve one mean student model. We note that the mean student model can be moved to CPU memory during idle time. We do not update the mean student model very often. When updating, we just move it back to GPU memory. Since we do not need to calculate the gradient but the mean of model parameters, the update process also costs limited memory. **2. What are the advantages of Student Aggregation?** The advantages are mostly in two aspects: First, instead of training S $(S > 5)$ classifiers in the conventional AdaBoost, we only need to keep

Algorithm 1 Training Procedure of the Proposed Method

Require: The source domain dataset $X_m = \{X_m^i\}_{i=1}^M$; The source domain label $Y_m = \{y_m^i\}_{i=1}^M$; The unlabeled target domain dataset $X_n = \{x_n^j\}_{j=1}^N$;
Require: The source-domain parameter θ_s; The epoch number T_1; The iteration number T_2 in each epoch.
Initialize $\theta = \theta_s$;
Initialize target domain data distribution $D_1(j) = 1/N$;
for *epoch* $t = 1$ to T_1 **do**
  for *iteration* $= 1$ to T_2 **do**
    To be simplify, here we only show the basic cross-entropy segmentation loss on the source domain data:

$$L_{ce} = \mathbb{E}[-p_m^i \log F(x_m^i|\theta)]. \tag{5}$$

    We denotes other regularization losses on the target domain as $R(x_n^j, \theta)$. Update the model weight:

$$L_{total} = L_{ce} + R(x_n^j, \theta). \tag{6}$$

  end for
  $\theta_t = \theta$.
  The mean model parameter is updated:

$$\Theta_t = \frac{t-1}{t}\Theta_{t-1} + \frac{1}{t}\theta_t. \tag{7}$$

  Update the adaptive data distribution D_{t+1}:

$$D_{t+1}(j) = \frac{1}{2}(D_t(j) + V_{kl}(x_n^j|\Theta_t)), \tag{8}$$

  where V_{kl} is the prediction variance defined in Eq. 1, and we apply the current mean model Θ_t to estimate the variance.
end for
return Θ_t.

one mean student model, which can be preserved in CPU memory. Second, the model update is efficient and effective, which does not require gradient calculation. We only need to update the weights of every layer to the mean value of existing snapshots. Meanwhile, student aggregation provides one final model with good generalizability, preventing overfitting as the traditional AdaBoost algorithm. More quantitative results are discussed in Experiment. **3. Correlation with AdaBoost.** The main spirit of the proposed method is following Adaboost [20] to treat "hard" samples differently according to learned model snapshots. However, for domain adaptation, we can not foreknow the error e_m on the unlabeled target-domain data. Therefore, we adopt the trade-off. 1) In this work, we update the distribution based on the prediction variance instead of the error; 2) The binary classifier aggregation in Adaboost can be formulated as: $G(x) = sign(\sum_{m=1}^{M} \alpha_m G_m(x))$, where $G_m()$ is one of weak classifiers and α_m is the adaptive weights based on the error e_m. $\alpha_m = \frac{1}{2} log(\frac{1-e_m}{e_m})$. Since the error e_m in target domain is unavailable, we deploy the temporal ensemble via weight averaging as a trade-off, which equals to set $\alpha_m = \frac{1}{M}$. In ablation studies, we also try different updating strategies, such as momentum, which is inferior to the proposed method. It verifies that the weight averaging trade-off is sub-optimal yet effective.

C. Optimization

We combine the two components, *i.e.*, adaptive sampler and student aggregation, in a coherent manner, and explicitly enable interactions between learned models and the data sampler. As shown in Algorithm 1, the adaptive sampler depends on the mean student model Θ_t to update the data distribution (see Eq. 8). Meanwhile, the adaptive sampler controls the input training data, which, in turn, affects the θ_t and Θ_t (see Eq. 7). Since we do not pursue new segmentation losses or regularization objectives, here we just mention the basic source-domain segmentation loss as:

$$L_{ce} = \mathbb{E}[-p_m^i \log F(x_m^i|\theta)]. \tag{9}$$

where p_m^i is the ground-truth probability vector of the label y_m^i. The value $p_m^i(c)$ equals to 1 if $c == y_m^i$ otherwise 0. Besides, there are multiple feasible regularization terms, including adversarial losses proposed in [15], [17] and consistency loss [19]. Without loss of generality, here we denote the regularization terms on the target domain as $R(x_n^j, \theta)$. Therefore, the optimization objective in every iteration can be formulated as:

$$L_{total} = L_{ce} + R(x_n^j, \theta). \tag{10}$$

It is worth noting that different baseline methods have different regularization terms. We follow the regularization term in the corresponding baseline method for a fair comparison. In this paper, we compare three baseline methods, *i.e.*, AdaptSegNet [15], MRNet [19] and Uncertainty [38]. AdaptSegNet mainly adopts the adversarial loss for regularization, while MRNet adds a consistency-based memory regularization. The Uncertainty follows MRNet, so Uncertainty has the same regularization term as MRNet. For a fair comparison, we follow the corresponding regularization term in different baseline methods. After every T_2 iterations, we update the adaptive sampler and the mean student model, as discussed in Section III-A and Section III-B. The training is stopped after T_1 epochs.

IV. Experiment

A. Implementation Details

Network Architecture. We adopt the Deeplab-v2 [2] as the baseline model, which deploys the ResNet-101 [51] as the backbone. Following most existing works [15], [16], [17], [18], [19], [38], we insert one auxiliary classifier with the same structure as the primary classifier. The classifier consists of one Atrous Spatial Pyramid Pooling (ASPP) module [2] and one fully-connected layer. The auxiliary classifier is added at the $res4b22$ layer. We also insert the dropout layer of 0.2 drop rate before the fully-connected layer.
Training Details. Following existing works [15], [17], the input image is resized to 1280×640, and we randomly crop 1024×512 for training. Random horizontal flipping is also applied. We deploy the SGD optimizer with a mini-batch of 2. The initial learning rate is set as 0.0002. Following [1], [56], [57], we deploy the ploy learning rate policy by multiplying the scale factor $(1 - \frac{current\_iter}{total\_iter})^{0.9}$. The total iteration is set as 50k ($T_1 = 10, T_2 = 5000$). When inference, we follow [19], [38] to combine the prediction of the two classifiers as the final prediction. Our implementation is based on Pytorch [58].

TABLE I
LIST OF CATEGORIES AND NUMBER OF IMAGES OF FOUR DATASETS, *i.e.*, GTA5 [7], SYNTHIA [59], CITYSCAPES [8] AND OXFORD ROBOTCAR [60].

Datasets	GTA5	SYNTHIA	Cityscapes	Oxford RobotCar
#Train Images	24,966	9,400	2,975	894
#Test Images	-	-	500	271
#Category	19	16	19	9
Synthetic	✓	✓	×	×

B. Datasets and Evaluation Metric

Datasets. In this work, we conduct experiments on three scene segmentation domain adaptation benchmarks. To simplify the illustration, we denote the source dataset A and the target dataset B as $A \rightarrow B$. Two widely-used benchmarks are to leverage the synthetic data to learn the shared knowledge, and adapt the knowledge to the real-world scenario. The settings are GTA5 [7] $\rightarrow$ Cityscapes [8] and SYNTHIA [59] $\rightarrow$ Cityscapes [8]. In particular, the GTA5 dataset is collected from one video game, which contains 24,966 labeled images for training. The SYNTHIA dataset is generated from a virtual city engine with pixel-level annotations as well, including 9,400 training images. The target dataset, *i.e.*, Cityscapes, collects the real-world street-view images from 50 different cities, yielding $2,975$ unlabeled training images and 500 images for testing. We also adopt one cross-city benchmark, *i.e.*, Cityscapes [8] $\rightarrow$ Oxford RobotCar [60]. In this setting, we use the annotation of $2,975$ training images of the Cityscapes dataset. In contrast with the Synthetic-to-real setting, the Oxford RobotCar dataset is also collected from the realistic street-view camera. The challenge is in the noisy variants, such as weather and illumination conditions. The Oxford RobotCar dataset is collected on rainy days and cloudy days, while the Cityscapes dataset mostly contains images on sunny days. More details are shown in Table I.

Evaluation Metric. We follow previous works to report the IoU accuracy of each category, and mean IoU (mIoU) for all the classes. For SYNTHIA $\rightarrow$ Cityscapes, some previous works report the mean IoU over 13 classes, while others report both 13 classes and 16 classes. We report both results, and denote the results of 13 classes and 16 classes as mIoU* and mIoU, respectively. For GTA5 $\rightarrow$ Cityscapes and Cityscapes $\rightarrow$ Oxford RobotCar, we report the 19-category and 9-category accuracy, respectively.

C. Ablation Studies

Effect of the Adaptive Sampler. First, we study the adaptive sampler. Since we intend to learn the complementary model snapshots, as shown in Table II, the single "weak" model does not achieve significant improvement, even with the performance drop. The observation is aligned with the conventional AdaBoost algorithm [20], since the current snapshot takes more attention to "hard" samples. Here we give one toy example on a binary classification to illustrate the phenomenon [1]. It is because every weak classifier considers more hard negatives in the last round and may overfit the negative data. Despite the probability of overfitting the negatives, such snapshots are complementary to the snapshot trained in the early stage and help the ensembled model keep performance improvement. Therefore, compared with the baseline method, *i.e.*, MRNet [19], which achieves 45.5% mIoU accuracy, MRNet using adaptive sampler decreases to 45.2%. It is because the hard negatives do not lead to one single optimal student model but the complementary model of the last model snapshot. If we further apply the student aggregation, MRNet + Ours has achieved significant improvement from 45.2% mIoU accuracy to 49.0% mIoU accuracy, which also surpasses MRNet (45.5% mIoU) by a relatively large margin. By default, we apply the sampler based on the prediction variance, and we observe that the alternative sampler based on the prediction entropy leads to one inferior performance of 48.1% mIoU. It is mainly due to the prediction entropy taking more attention to the object edges as shown in [38], which can not nominate "hard" samples with ambiguous predictions accurately.

[1] https://github.com/layumi/AdaBoost_Seg/blob/master/Toy_Example.md

TABLE II
ABLATION STUDY OF THE TWO COMPONENTS IN ADABOOST STUDENT ON GTA5→CITYSCAPES. WE ADOPT MRNET [19] AS BASELINE, AND STUDY THE IMPACT OF THE ADAPTIVE SAMPLER AND THE STUDENT AGGREGATION ALONE. THE RESULTS SUGGEST THAT THE ADAPTIVE SAMPLER DOES NOT IMPROVE THE SINGLE MODEL PERFORMANCE BUT HELPS COMPLEMENTARY MODEL LEARNING, FACILITATING THE FINAL "WEAK" STUDENT MODEL AGGREGATION. OURS (ENTROPY) DENOTES THAT WE APPLY THE PREDICTION ENTROPY TO UPDATE THE ADAPTIVE SAMPLER, WHILE OURS (VARIANCE) IS THE DEFAULT SETTING IF NOT SPECIFIED. PUTBACK SAMPLER DENOTES RANDOM SAMPLING WITH REPLACEMENT.

Method	PutBack Sampler	Adaptive Sampler	Student Aggregation	mIoU
MRNet [19] w Uniform				45.5
w PutBack	✓			43.6
w Random				44.6
w Adaptive Sampler		✓		45.2
w Uniform + Student Aggregation			✓	48.4
w PutBack + Student Aggregation	✓		✓	48.3
w Random + Student Aggregation			✓	48.4
MRNet [19] + Ours (entropy)		✓	✓	48.1
MRNet [19] + Ours (variance)		✓	✓	49.0

Furthermore, we compare other sampling strategies, *i.e.*, sampling with replacement (PutBack) and random sampling (Random), to verify the effectiveness of the proposed Adaptive Sampler. PutBack sampler is different from the original MRNet, which directly adopts the sampling without replacement (Uniform). As shown in Table II, (1) PutBack Sampler does not lead to a better single snapshot. It is because the learned model does not "see" all data in every iteration, which compromises the training process. (2) PutBack Sampler also ensures diverse snapshots. Therefore, PutBack Sampler + Student Aggregation arrives 48.3% mIoU accuracy, which is close to the baseline + Student Aggregation of 48.4%. (3) The proposed Adaptive Sampler (ours) surpasses the PutBack Sampler, since we explicitly add more emphasis on the hard samples. Similarly, we could observe that (1) Random sampling (44.6%) is inferior to Uniform sampling (45.5%). It is because the random sampler, like PutBack sampler, usually does not "see" all data in every epoch. (2) Random sampling leads to diverse snapshots, but it also suffers from a larger performance fluctuation. We observe that some snapshots in

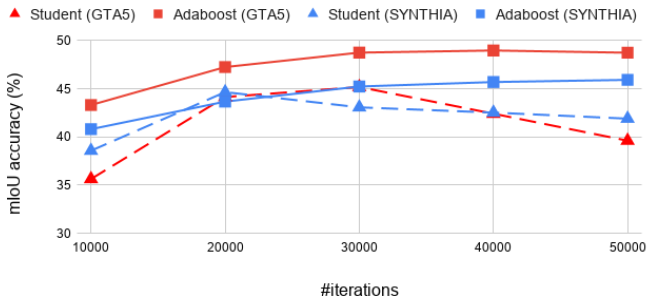


Fig. 4. Convergence of the proposed method. We provide the sensitivity analysis of MRNet+Ours on two benchmarks, *i.e.*, GTA5→Cityscapes (GTA5) and SYNTHIA→Cityscapes (SYNTHIA). The results suggest that the AdaBoost Student model (AdaBoost) quickly converges, while the performance of the student model (Student) is still fluctuating.

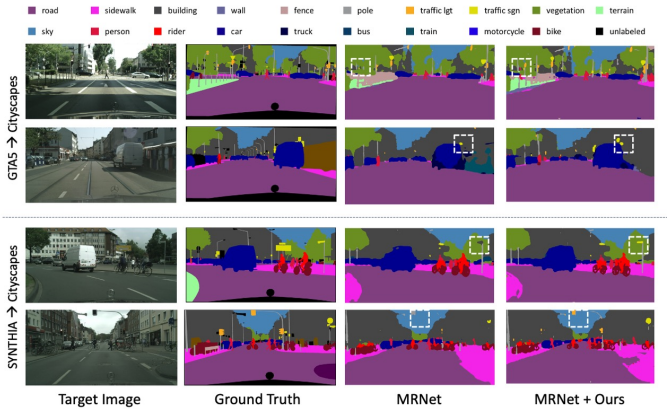


Fig. 5. Qualitative results of semantic segmentation adaptation on GTA5 → Cityscapes and SYNTHIA → Cityscapes. We show the original target-domain image, the ground-truth segmentation, and the output of methods, *i.e.*, MRNet [19], and MRNet [19] + Ours. Our results are in the right column. (Best viewed in *color*). We deploy the white dash boxes to highlight the different predictions. The proposed method is more robust to the minority categories, such as lights, poles, and traffic signs.

our implementation arrive at around 41-42% mIoU, which results in performance convergence. For instance, the snapshot at the 25000th iteration performs 44.6% mIoU, but the model at the 45000th iteration only arrives at 41.8% mIoU. Therefore, Random sampling + student aggregation (averaged model) only converges to 48.4% mIoU, which is also inferior to our method. Due to the random sampling is not stable, we run the experiment twice. Despite the performance fluctuation of the single snapshot, the re-run performance with random sampling also converges to 48.3% mIoU (around 48.4% mIoU). Finally, we could see that no matter uniform, putback, or random sampling method, the aggregated models all converge to around 48.4% compared to the 49.0% of our method. Therefore, it verifies our motivation that the sampling on hard negative samples is non-trivial.

Effect of the Student Aggregation. We also investigate the student aggregation in Adaboost Student (see Table II). There are two main observations. (1) The student aggregation alone can yield performance improvement. It is due to the model discrepancy during training. Especially for the scene segmentation, the extremely small training batch sizes and data shuffle lead to different training orders in every epoch. Therefore, the baseline method, *i.e.*, MRNet, with student aggregation alone, has arrived at 48.4% mIoU accuracy from 45.5% mIoU. (2) Since the input data distribution is fixed in the baseline method, the model snapshots are still homogeneous due to the training data. In contrast, the proposed method explicitly demands learning complementary models by the dynamic input data via the adaptive sampler. The full method with the adaptive sampler, therefore, yields better results at 49.0%, surpassing the student aggregation alone (48.4%). The results also suggest the effectiveness of the adaptive sampler.

Compatibility of the Proposed Method. We argue that the proposed method is orthogonal to most existing methods, and can be fused with existing frameworks to obtain better performance. Therefore, we re-implement three existing approaches, including AdaptSegNet [15], MRNet [19] and Uncertainty [38]. As shown in Table III, we observe one consistent performance improvement on GTA5→Cityscapes for all three methods. For the widely-adopted method, *i.e.*, AdaptSegNet [15], AdaptSegNet + Ours has arrived at 47.3% mIoU, which surpasses the original AdaptSegNet (42.4%) by $+4.9\%$. Furthermore, we observe one similar performance improvement on recent methods. MRNet [19] + Ours has boosted from 45.5% mIoU accuracy to 49.0%, while Uncertainty [38] + Ours obtains $+0.6\%$ from 50.3% to 50.9% mIoU accuracy. As shown in Table IV and Table V, we observe one similar result on both SYNTHIA→Cityscapes and Cityscapes→Oxford RobotCar benchmarks. For the SYNTHIA→Cityscapes benchmark, AdaptSegNet [15], MRNet [19] and Uncertainty [38] have achieved $+1.2\%$, $+2.7\%$ and $+2.6\%$ mIoU*, respectively. On the other hand, AdaptSegNet [15], MRNet [19] and Uncertainty [38] have achieved $+1.8\%$, $+1.2\%$ and $+0.8\%$ mIoU on the Cityscapes→Oxford RobotCar benchmark, respectively. In a summary, the extensive experiment with three methods on three benchmarks verifies that the proposed approach, Adaptive Student, can be fused with existing frameworks to improve the model generalizability in the target domain.

Limitation. We observe that the proposed method shows less improvement on the pseudo label learning based method, *i.e.*, Uncertainty [38] than the domain alignment based method, *i.e.*, AdaptSegNet [15] and MRNet [19]. The main reason is the strong supervised learning of the generated pseudo labels on the target domain, which limits the model discrepancy and compromises learning complementary "weak" models in the proposed AdaBoost Student.

Convergence of the Proposed Method. One advantage of the proposed method is the quick convergence to one robust model after limited iterations, since the complementary models are aggregated during training. Here we provide the sensitivity analysis of convergence time on both GTA5 → Cityscapes and SYNTHIA → Cityscapes. We adopt MRNet as the baseline model. As shown in Figure 4, we observe that the proposed method can efficiently converge at around $30,000$ iterations, while the student models are still jittering during training.

TABLE III
QUANTITATIVE RESULTS ON GTA5 → CITYSCAPES. WE PRESENT PRE-CLASS IOU AND MIOU. THE BEST ACCURACY IN EVERY COLUMN IS IN **BOLD**.

Method	Road	SW	Build	Wall	Fence	Pole	TL	TS	Veg.	Terrain	Sky	PR	Rider	Car	Truck	Bus	Train	Motor	Bike	mIoU
Source	75.8	16.8	77.2	12.5	21.0	25.5	30.1	20.1	81.3	24.6	70.3	53.8	26.4	49.9	17.2	25.9	6.5	25.3	36.0	36.6
AdaptSegNet [15]	86.5	36.0	79.9	23.4	23.3	23.9	35.2	14.8	83.4	33.3	75.6	58.5	27.6	73.7	32.5	35.4	3.9	30.1	28.1	42.4
AdaptSegNet [15] + Ours	89.5	34.1	83.7	31.5	25.0	36.0	40.6	34.6	85.6	**44.6**	77.8	61.4	30.4	85.6	35.9	47.1	1.8	19.4	34.1	47.3
SIBAN [17]	88.5	35.4	79.5	26.3	24.3	28.5	32.5	18.3	81.2	40.0	76.5	58.1	25.8	82.6	30.3	34.4	3.4	21.6	21.5	42.6
CLAN [18]	87.0	27.1	79.6	27.3	23.3	28.3	35.5	24.2	83.6	27.4	74.2	58.6	28.0	76.2	33.1	36.7	6.7	31.9	31.4	43.2
SP-Adv [61]	86.2	38.4	80.8	25.5	20.5	32.8	33.4	28.2	85.5	36.1	80.2	60.3	28.6	78.7	27.3	36.1	4.6	31.6	28.4	44.3
MaxSquare [62]	88.1	27.7	80.8	28.7	19.8	24.9	34.0	17.8	83.6	34.7	76.0	58.6	28.6	84.1	37.8	43.1	7.2	32.3	34.2	44.3
ASA [63]	89.2	27.8	81.3	25.3	22.7	28.7	36.5	19.6	83.8	31.4	77.1	59.2	29.8	84.3	33.2	45.6	16.9	**34.5**	30.8	45.1
APODA [64]	85.6	32.8	79.0	29.5	25.5	26.8	34.6	19.9	83.7	40.6	77.9	59.2	28.3	84.6	34.6	49.2	8.0	32.6	39.6	45.9
PatchAlign [16]	**92.3**	51.9	82.1	29.2	25.1	24.5	33.8	33.0	82.4	32.8	82.2	58.6	27.2	84.3	33.4	46.3	2.2	29.5	32.3	46.5
BL [30]	91.0	44.7	84.2	34.6	27.6	30.2	36.0	36.0	85.0	43.6	83.0	58.6	31.6	83.3	35.3	49.7	3.3	28.8	35.6	48.5
DT [65]	90.6	44.7	84.8	34.3	28.7	31.6	35.0	37.6	84.7	43.3	**85.3**	57.0	31.5	83.8	**42.6**	48.5	1.9	30.4	39.0	49.2
AdvEnt [66]	89.4	33.1	81.0	26.6	26.8	27.2	33.5	24.7	83.9	36.7	78.8	58.7	30.5	84.8	38.5	44.5	1.7	31.6	32.4	45.5
Source	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	29.2
FCAN [67]	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	46.6
Source	71.3	19.2	69.1	18.4	10.0	35.7	27.3	6.8	79.6	24.8	72.1	57.6	19.5	55.5	15.5	15.1	11.7	21.1	12.0	33.8
CBST [27]	91.8	53.5	80.5	32.7	21.0	34.0	28.9	20.4	83.9	34.2	80.9	53.1	24.0	82.7	30.3	35.9	16.0	25.9	42.8	45.9
MRKLD [28]	91.0	**55.4**	80.0	33.7	21.4	37.3	32.9	24.5	85.0	34.1	80.8	57.7	24.6	84.1	27.8	30.1	**26.9**	26.0	42.3	47.1
Source	51.1	18.3	75.8	18.8	16.8	34.7	36.3	27.2	80.0	23.3	64.9	59.2	19.3	74.6	26.7	13.8	0.1	32.4	34.0	37.2
MRNet [19]	89.1	23.9	82.2	19.5	20.1	33.5	42.2	39.1	85.3	33.7	76.4	60.2	33.7	86.0	36.1	43.3	5.9	22.8	30.8	45.5
MRNet [19] + Ours	89.5	36.6	84.8	37.9	**29.8**	37.9	45.1	34.9	85.8	43.6	80.4	**64.7**	35.3	84.9	30.4	45.9	2.7	24.7	35.7	49.0
Uncertainty [38]	90.4	31.2	85.1	36.9	25.6	37.5	48.8	**48.5**	85.3	34.8	81.1	64.4	**36.8**	86.3	34.9	**52.2**	1.7	29.0	44.6	50.3
Uncertainty [38] + Ours	90.7	35.9	**85.7**	**40.1**	27.8	**39.0**	**49.0**	48.4	**85.9**	35.1	85.1	63.1	34.4	**86.8**	38.3	49.5	0.2	26.5	**45.3**	**50.9**

TABLE IV
QUANTITATIVE RESULTS ON SYNTHIA → CITYSCAPES. WE PRESENT PRE-CLASS IOU, MIOU AND MIOU*. MIOU AND MIOU* ARE AVERAGED OVER 16 AND 13 CATEGORIES, RESPECTIVELY. THE BEST ACCURACY IN EVERY COLUMN IS IN **BOLD**.

Method	Road	SW	Build	Wall*	Fence*	Pole*	TL	TS	Veg.	Sky	PR	Rider	Car	Bus	Motor	Bike	mIoU*	mIoU
Source	55.6	23.8	74.6	–	–	–	6.1	12.1	74.8	79.0	55.3	19.1	39.6	23.3	13.7	25.0	38.6	–
MaxSquare [62]	77.4	34.0	78.7	5.6	0.2	27.7	5.8	9.8	80.7	83.2	58.5	20.5	74.1	32.1	11.0	29.9	45.8	39.3
SIBAN [17]	82.5	24.0	79.4	–	–	–	16.5	12.7	79.2	82.8	58.3	18.0	79.3	25.3	17.6	25.9	46.3	–
PatchAlign [16]	82.4	38.0	78.6	8.7	0.6	26.0	3.9	11.1	75.5	84.6	53.5	21.6	71.4	32.6	19.3	31.7	46.5	40.0
AdaptSegNet [15]	84.3	42.7	77.5	–	–	–	4.7	7.0	77.9	82.5	54.3	21.0	72.3	32.2	18.9	32.3	46.7	–
AdaptSegNet [15] + Ours	67.9	30.1	77.9	10.2	1.5	37.2	30.9	22.3	80.8	83.1	52.4	20.9	72.9	25.4	12.7	45.7	47.9	42.0
CLAN [18]	81.3	37.0	80.1	–	–	–	16.1	13.7	78.2	81.5	53.4	21.2	73.0	32.9	22.6	30.7	47.8	–
SP-Adv [61]	84.8	35.8	78.6	–	–	–	6.2	15.6	80.5	82.0	66.5	22.7	74.3	34.1	19.2	27.3	48.3	–
ASA [63]	**91.2**	**48.5**	80.4	3.7	0.3	21.7	5.5	5.2	79.5	83.6	56.4	21.0	80.3	36.2	20.0	32.9	49.3	41.7
DADA [11]	89.2	44.8	81.4	6.8	0.3	26.2	8.6	11.1	81.8	84.0	54.7	19.3	79.7	40.7	14.0	38.8	49.8	42.6
BL [30]	86.0	46.7	80.3	–	–	–	14.1	11.6	79.2	81.3	54.1	27.9	73.7	42.2	25.7	45.3	51.4	–
DT [65]	83.0	44.0	80.3	–	–	–	17.1	15.8	80.5	81.8	59.9	33.1	70.2	37.3	28.5	45.8	52.1	–
CCM [68]	79.6	36.4	80.6	13.3	0.3	25.5	22.4	14.9	81.8	77.4	56.8	25.9	80.7	45.3	29.9	52.0	52.9	45.2
APODA [64]	86.4	41.3	79.3	–	–	–	22.6	17.3	80.3	81.6	56.9	21.0	84.1	**49.1**	24.6	45.7	53.1	–
AdvEnt [66]	85.6	42.2	79.7	8.7	0.4	25.9	5.4	8.1	80.4	84.1	57.9	23.8	73.3	36.4	14.2	33.0	48.0	41.2
Source	64.3	21.3	73.1	2.4	1.1	31.4	7.0	27.7	63.1	67.6	42.2	19.9	73.1	15.3	10.5	38.9	40.3	34.9
CBST [27]	68.0	29.9	76.3	10.8	1.4	33.9	22.8	29.5	77.6	78.3	60.6	28.3	81.6	23.5	18.8	39.8	48.9	42.6
MRKLD [28]	67.7	32.2	73.9	10.7	1.6	37.4	22.2	**31.2**	80.8	80.5	60.8	**29.1**	82.8	25.0	19.4	45.3	50.1	43.8
Source	44.0	19.3	70.9	8.7	0.8	28.2	16.1	16.7	79.8	81.4	57.8	19.2	46.9	17.2	12.0	43.8	40.4	35.2
MRNet [19]	82.0	36.5	80.4	4.2	0.4	33.7	18.0	13.4	81.1	80.8	61.3	21.7	84.4	32.4	14.8	45.7	50.2	43.2
MRNet [19] + Ours	83.6	40.1	81.3	9.6	0.8	36.8	25.8	16.3	84.5	87.1	60.3	23.9	84.9	32.3	19.0	48.4	52.9	45.9
Uncertainty [38]	87.6	41.9	83.1	14.7	**1.7**	36.2	31.3	19.9	81.6	80.6	63.0	21.8	86.2	40.7	23.6	**53.1**	54.9	47.9
Uncertainty [38] + Ours	85.6	43.9	**83.9**	**19.2**	**1.7**	**38.0**	**37.9**	19.6	**85.5**	**88.4**	**64.1**	25.7	**86.6**	43.9	**31.2**	51.3	**57.5**	**50.4**

TABLE V
QUANTITATIVE RESULTS ON THE CROSS-CITY BENCHMARK: CITYSCAPES → OXFORD ROBOTCAR. THE BEST ACCURACY IS IN **BOLD**.

Method	road	sidewalk	building	light	sign	sky	person	automobile	two-wheel	mIoU
Source	79.2	49.3	73.1	55.6	37.3	36.1	54.0	81.3	49.7	61.9
AdaptSegNet [15]	95.1	64.0	75.7	61.3	35.5	63.9	58.1	84.6	57.0	69.5
AdaptSegNet [15] + Ours	96.0	73.0	91.5	65.8	22.9	**94.9**	56.0	89.0	53.0	71.3
PatchAlign [16]	94.4	63.5	82.0	61.3	36.0	76.4	61.0	86.5	58.6	72.0
MRNet [19]	95.9	73.5	86.2	69.3	31.9	87.3	57.9	88.8	**61.5**	72.5
MRNet [19] + Ours	**96.4**	**77.0**	83.4	70.7	39.2	83.0	62.4	**89.9**	61.0	73.7
Uncertainty [38]	95.9	73.7	87.4	**72.8**	**43.1**	88.6	61.7	89.6	57.0	74.4
Uncertainty [38] + Ours	96.1	75.3	**92.1**	72.6	37.0	94.4	**62.7**	89.7	56.9	**75.2**

D. Comparisons with State-of-the-art Methods

Synthetic-to-real. We mainly compare other recent methods on two synthetic-to-real benchmarks, which leverage the synthetic data to learn the shared knowledge. We compare recent methods with reported results and several methods re-implemented by us. For a fair comparison, we compare models with a similar structure, which is based on the Deeplab-v2 [2], and the backbone is ResNet-101 [51]. The competitive methods can be generally classified into two categories according to the usage of the pseudo label. One line of works focuses on the domain alignment, including AdaptSegNet [15], SIBAN [17], CLAN [18], APODA [64], PatchAlign [16], DT [65] and MRNet [19]. Another line of works focuses on mining the target-domain knowledge via noisy labels and deploys the pseudo label learning, such as CBST [27], MRKLD [28] and Uncertainty [38], which generally achieves higher performance. As shown in Table III and Table IV, the proposed method yields consistent improvement in both kinds of works. The competitive performance has been achieved by combining AdaBoost Student with the Uncertainty [38]. Specifically, we have achieved 50.9% and 50.4% mIoU on GTA5→Cityscapes and SYNTHIA→Cityscapes, respectively. Furthermore, the

TABLE VI
ABLATION STUDY ON THE EFFECTIVENESS OF THE PROPOSED METHOD ON A RELATIVELY VANILLA NETWORK, *i.e.*, VGG16 [69] ON GTA5 → CITYSCAPES. *: WE RE-IMPLEMENT MRNET [19] WITH VGG16.

Method	Backbone	mIoU
Curriculum DA [70]	VGG16	28.9
Wang *et al.* [71]	VGG19	33.1
CyCADA [12]	VGG16	35.4
ASA [63]	VGG16	35.6
CBST-SP [27]	VGG16	36.1
DCAN [13]	VGG16	36.2
SP-Adv [61]	VGG16	36.5
LSD [26]	VGG16	37.1
MRNet* [19]	VGG16	25.5
MRNet* [19] + Ours	VGG16	39.5

pre-class IoU results also suggest that the proposed method has achieved relatively good results on several classes of small-scale objectives, including Pole, Wall, and Fence.

Cross-city. We also evaluate the proposed method on one new cross-city benchmarks, *i.e.*, Cityscapes → Oxford RobotCar (see Table V). Both Cityscapes and Oxford RobotCar are collected from the realistic scenarios but in different cities and weather. The proposed method also has achieved the competitive result of 75.2% mIoU accuracy and showed a consistent performance improvement with other existing methods.

Qualitative Results. Furthermore, we provide the segmentation results in Figure 5, which also verifies the effectiveness of the proposed method. Comparing to the segmentation predictions of vanilla MRNet [19], MRNet + Ours generally rectifies the object details, and achieves more robust outputs on the target domain. We apply the white dash boxes to highlight the different predictions. In particular, the proposed method is more robust to the minority categories, such as lights, poles, and traffic signs.

V. FURTHER ANALYSIS AND DISCUSSIONS

Network Structure. We also verify the proposed method on a relatively vanilla network, *i.e.*, VGG16 [69]. Following previous works, we do not introduce BN (Batch Normalization) layers in the backbone network. We re-implement MRNet [19], which has achieved 25.5% mIoU on GTA5 → Cityscapes. As shown in Table VI, we can observe two main points. First, the proposed method is complementary to the existing works, *e.g.*, MRNet [19], and MRNet [19] + Ours achieves +14.0% mIoU improvements. Second, we also achieve one competitive result 39.5% mIoU with other existing methods based on VGG16.

Aggregation with Momentum. Due to the prediction fluctuation on the target domain, we can not foreknow which snapshot is reliable. We also run MRNet+Ours with momentum strategy [72] on GTA5 → Cityscapes. Two momentum values 0.9 and 0.5 are explored to keep old weights. The aggregated model only achieves 46.4% and 45.3%, respectively, which is inferior to the weight averaging. It verifies that the weight averaging trade-off is sub-optimal yet effective. In this paper, we leverage the weight averaging strategy, but it is still open to other aggregation alternatives.

Focal Loss. The focal loss is designed for supervised learning, which is modified from the cross-entropy loss [47]. For domain adaptation, we do not have ground-truth target-domain labels. Therefore, we apply the focal loss on the source-domain data to train the model. We re-implement the focal loss, and apply the default gamma=2 as the original paper suggested. However, the result of Ours+Focal Loss (47.87% mIoU) is not better than Ours. We think that the focal loss makes the model focus on the "hard" samples in the source domain rather than target domain, which compromises the final performance. In the future, we will further study to fuse the focal loss in other formats on the target domain.

TABLE VII
COMPARISON WITH MEAN TEACHER (MT) [43].

Method	mIoU
MRNet [19]	45.5
MRNet [19] + MT (Student)	46.6
MRNet [19] + MT (Teacher)	47.5
MRNet [19] + Ours	49.0

TABLE VIII
ABLATION STUDY OF ADABOOST STUDENT ON GTA5+SYNTHIA→CITYSCAPES. WE ADOPT MRNET [19] AS BASELINE, AND STUDY THE IMPACT OF THE ADAPTIVE SAMPLER AND THE STUDENT AGGREGATION ALONE. THE RESULTS ALSO SUGGEST THAT THE ADAPTIVE SAMPLER DOES NOT IMPROVE THE SINGLE MODEL PERFORMANCE BUT HELPS COMPLEMENTARY MODEL LEARNING, FACILITATING THE FINAL "WEAK" STUDENT MODEL AGGREGATION.

Method	Adaptive Sampler	Student Aggregation	mIoU
MRNet [19]			47.6
w Adaptive Sampler	✓		48.4
w Student Aggregation		✓	50.6
MRNet [19] + Ours	✓	✓	50.8

***vs.* Mean Teacher (MT).** Similarly, the existing method, *i.e.*, Mean Teacher (MT) [43] also preserves one weight-average model during the training process. We apply the same strategy, including iteration numbers, to update the teacher model and re-implement Mean Teacher. The main difference between the proposed method and Mean Teacher (MT) is twofold: First, we adopt the adaptive sampler, which enables interactions between learned models and the data sampler. In this way, the proposed method leads to more complementary snapshots during training for the student aggregation. Second, Mean Teacher needs the teacher prediction in every iteration. It demands extra model inference to calculate the kl-divergence loss (costs extra time and memory) and forces the student model and the weight-average teacher model to predict the same distribution given the input, which also limits learning complementary student models. In contrast, the proposed method does not introduce such an objective and performs more efficiently and effectively with the spirit of AdaBoost [20]. As a result, the proposed method outperforms MT (Student) 46.6% and MT (Teacher) 47.5% mIoU (see Table VII).

Multi-Source Domains. We add a new setting on GTA5 + SYNTHIA→Cityscapes with two source domains to evaluate the proposed method. With multi-source domain data, the model can be trained more robust to the unlabelled target environment as well as more diverse between single snapshots. As shown in Table VIII, we could observe three points: (1) With more source-domain data, the model arrives at a better

TABLE IX
COMPARISON WITH MEAN TEACHER (MT) [43] FROM 10 RUNS ON CIFAR-10 USING 4000 LABELS. *: WE RE-RUN THE OFFICAL CODE OF MEAN TEACHER WITH A SLIGHT BETTER PERFORMANCE.

Method	top1 error (%)
Supervised Learning [73]	2.86
VAT [74]	10.55
CT-GAN [75]	9.98 $\pm$ 0.21
Mean Teacher [43]	6.28 $\pm$ 0.15
Mean Teacher [43]*	6.14 $\pm$ 0.24
Ours	6.05 $\pm$ 0.12

basic result of 47.6% compared to 45.5% (only GTA5). (2) The model with the adaptive sampler (48.4%) is still competitive with the single snapshot on baseline (47.6%). (3) With better single snapshots, our method achieves the best performance of 50.8% mIoU.

Semi-supervised Learning. We follow MeanTeacher [43] and modify the official code on Cifar-10 as our baseline [2]. The MeanTeacher backbone also has two classifiers, one for classification and the other for regressing the teacher prediction. Therefore, the proposed method can directly leverage the difference between two classifiers as the prediction variance to update the data sampler. As shown in Table IX, the proposed method further improves the semi-supervised learning result (10 runs with 4000 labeled data) from 6.28% $\pm$ 0.15 top-1 error (reported in MeanTeacher) to 6.05% $\pm$ 0.12 top-1 error.

VI. CONCLUSION

We identify one practical challenge in deciding the early-stopping time of the scene segmentation domain adaptation task. To address this limitation, we propose an efficient bootstrapping solution, AdaBoost Student, to learn the scalable model by aggregating "weak" models. The main idea underpinning AdaBoost Student is to leverage the model discrepancy during training. Specifically, AdaBoost Student adopts one adaptive data sampler and explicitly facilitates learning complementary models in a coherent manner. Thus, AdaBoost Student can efficiently converge to one robust final model and prevent over-fitting. As a result, we have achieved consistent improvements and competitive results on three benchmarks, including two synthetic-to-real benchmarks and one cross-city benchmark. In the future, we will continue to investigate the usage of Adaboost Student and apply it to other fields, such as medical images, to obtain the model with good generalizability.

REFERENCES

[1] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *CVPR*, 2017.

[2] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 4, pp. 834–848, 2017.

[3] Z. Wu, X. Wang, J. E. Gonzalez, T. Goldstein, and L. S. Davis, "Ace: adapting to changing environments for semantic segmentation," in *ICCV*, 2019.

[4] Q. Wang, J. Gao, W. Lin, and Y. Yuan, "Learning from synthetic data for crowd counting in the wild," in *CVPR*, 2019.

[2] https://github.com/layumi/Cifar10-Adaboost

[5] Y. Pan, T. Yao, Y. Li, Y. Wang, C.-W. Ngo, and T. Mei, "Transferrable prototypical networks for unsupervised domain adaptation," in *CVPR*, 2019.

[6] S. Xiang, Y. Fu, G. You, and T. Liu, "Unsupervised domain adaptation through synthesis for person re-identification," in *ICME*, 2020.

[7] S. R. Richter, V. Vineet, S. Roth, and V. Koltun, "Playing for data: Ground truth from computer games," in *ECCV*, 2016.

[8] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *CVPR*, 2016.

[9] Y. Chen, W. Li, X. Chen, and L. V. Gool, "Learning semantic segmentation from synthetic data: A geometrically guided input-output adaptation approach," in *CVPR*, 2019.

[10] Y.-C. Chen, Y.-Y. Lin, M.-H. Yang, and J.-B. Huang, "Crdoco: Pixel-level domain transfer with cross-domain consistency," in *CVPR*, 2019.

[11] T.-H. Vu, H. Jain, M. Bucher, M. Cord, and P. Pérez, "Dada: Depth-aware domain adaptation in semantic segmentation," in *ICCV*, 2019.

[12] J. Hoffman, E. Tzeng, T. Park, J.-Y. Zhu, P. Isola, K. Saenko, A. A. Efros, and T. Darrell, "Cycada: Cycle-consistent adversarial domain adaptation," in *ICML*, 2018.

[13] Z. Wu, X. Han, Y.-L. Lin, M. Gokhan Uzunbas, T. Goldstein, S. Nam Lim, and L. S. Davis, "Dcan: Dual channel-wise alignment networks for unsupervised scene adaptation," in *ECCV*, 2018.

[14] B. Geng, D. Tao, and C. Xu, "Daml: Domain adaptation metric learning," *IEEE Transactions on Image Processing*, vol. 20, no. 10, pp. 2980–2989, 2011.

[15] Y.-H. Tsai, W.-C. Hung, S. Schulter, K. Sohn, M.-H. Yang, and M. Chandraker, "Learning to adapt structured output space for semantic segmentation," in *CVPR*, 2018.

[16] Y.-H. Tsai, K. Sohn, S. Schulter, and M. Chandraker, "Domain adaptation for structured output via discriminative representations," in *ICCV*, 2019.

[17] Y. Luo, P. Liu, T. Guan, J. Yu, and Y. Yang, "Significance-aware information bottleneck for domain adaptive semantic segmentation," in *ICCV*, 2019.

[18] Y. Luo, L. Zheng, T. Guan, J. Yu, and Y. Yang, "Taking a closer look at domain shift: Category-level adversaries for semantics consistent domain adaptation," in *CVPR*, 2019.

[19] Z. Zheng and Y. Yang, "Unsupervised scene adaptation with memory regularization in vivo," in *IJCAI*, 2020.

[20] Y. Freund, R. E. Schapire *et al.*, "Experiments with a new boosting algorithm," in *ICML*, 1996.

[21] Y. Luo, P. Liu, T. Guan, J. Yu, and Y. Yang, "Adversarial style mining for one-shot unsupervised domain adaptation," in *NeurIPS*, 2020.

[22] R. Gong, W. Li, Y. Chen, and L. V. Gool, "Dlow: Domain flow for adaptation and generalization," in *CVPR*, 2019.

[23] L. Du, J. Tan, H. Yang, J. Feng, X. Xue, Q. Zheng, X. Ye, and X. Zhang, "Ssf-dan: Separated semantic feature based domain adaptation network for semantic segmentation," in *ICCV*, 2019.

[24] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networkss," in *ICCV*, 2017.

[25] X. Yue, Y. Zhang, S. Zhao, A. Sangiovanni-Vincentelli, K. Keutzer, and B. Gong, "Domain randomization and pyramid consistency: Simulation-to-real generalization without accessing target domain data," in *ICCV*, 2019.

[26] S. Sankaranarayanan, Y. Balaji, A. Jain, S. N. Lim, and R. Chellappa, "Learning from synthetic data: Addressing domain shift for semantic segmentation," in *CVPR*, 2018.

[27] Y. Zou, Z. Yu, B. Vijaya Kumar, and J. Wang, "Unsupervised domain adaptation for semantic segmentation via class-balanced self-training," in *ECCV*, 2018.

[28] Y. Zou, Z. Yu, X. Liu, B. Kumar, and J. Wang, "Confidence regularized self-training," in *ICCV*, 2019.

[29] Q. Zhou, Z. Feng, Q. Gu, G. Cheng, X. Lu, J. Shi, and L. Ma, "Uncertainty-aware consistency regularization for cross-domain semantic segmentation," *Computer Vision and Image Understanding*, p. 103448, 2022.

[30] Y. Li, L. Yuan, and N. Vasconcelos, "Bidirectional learning for domain adaptation of semantic segmentation," in *CVPR*, 2019.

[31] P. Zhang, B. Zhang, T. Zhang, D. Chen, Y. Wang, and F. Wen, "Prototypical pseudo label denoising and target structure learning for domain adaptive semantic segmentation," in *CVPR*, 2021.

[32] Y. Huang, J. Xu, Q. Wu, Z. Zheng, Z. Zhang, and J. Zhang, "Multi-pseudo regularized label for generated data in person re-identification," *IEEE Transactions on Image Processing*, vol. 28, no. 3, pp. 1391–1403, 2018.

[33] G. Ding, S. Zhang, S. Khan, Z. Tang, J. Zhang, and F. Porikli, "Feature affinity-based pseudo labeling for semi-supervised person re-identification," *IEEE Transactions on Multimedia*, vol. 21, no. 11, pp. 2891–2902, 2019.

[34] Y. Ding, H. Fan, M. Xu, and Y. Yang, "Adaptive exploration for unsupervised person re-identification," *TOMM*, vol. 16, no. 1, pp. 3:1–3:19, 2020, doi:10.1145/3369393.

[35] H. Feng, M. Chen, J. Hu, D. Shen, H. Liu, and D. Cai, "Complementary pseudo labels for unsupervised domain adaptation on person re-identification," *IEEE Transactions on Image Processing*, vol. 30, pp. 2898–2907, 2021.

[36] Y. Lin, L. Xie, Y. Wu, C. Yan, and Q. Tian, "Unsupervised person re-identification via softened similarity learning," in *CVPR*, 2020.

[37] Y. Yang, Y. Zhuang, and Y. Pan, "Multiple knowledge representation for big data artificial intelligence: framework, applications, and case studies," *Frontiers of Information Technology & Electronic Engineering*, vol. 22, no. 12, pp. 1551–1558, 2021, doi:10.1145/FITEE.2100463.

[38] Z. Zheng and Y. Yang, "Rectifying pseudo label learning via uncertainty estimation for domain adaptive semantic segmentation," *International Journal of Computer Vision (IJCV)*, 2020, doi:10.1007/s11263-020-01395-y.

[39] S. Reed, H. Lee, D. Anguelov, C. Szegedy, D. Erhan, and A. Rabinovich, "Training deep neural networks on noisy labels with bootstrapping," in *ICLR workshop*, 2014.

[40] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," in *NeurIPS*, 2015.

[41] Y. Zhang, T. Xiang, T. M. Hospedales, and H. Lu, "Deep mutual learning," in *CVPR*, 2018.

[42] S. Laine and T. Aila, "Temporal ensembling for semi-supervised learning," in *ICLR*, 2016.

[43] A. Tarvainen and H. Valpola, "Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results," in *NeurIPS*, 2017.

[44] J. Choi, T. Kim, and C. Kim, "Self-ensembling with gan-based data augmentation for domain adaptation in semantic segmentation," in *ICCV*, 2019.

[45] J. Cha, H. Cho, K. Lee, S. Park, Y. Lee, and S. Park, "Domain generalization needs stochastic weight averaging for robustness on domain shifts," *arXiv:2102.08604*, 2021.

[46] Y. Chen, X. Zhu, and S. Gong, "Semi-supervised deep learning with memory," in *ECCV*, 2018.

[47] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *ICCV*, 2017.

[48] A. Shrivastava, A. Gupta, and R. Girshick, "Training region-based object detectors with online hard example mining," in *CVPR*, 2016.

[49] H. Oh Song, Y. Xiang, S. Jegelka, and S. Savarese, "Deep metric learning via lifted structured feature embedding," in *CVPR*, 2016.

[50] A. Hermans, L. Beyer, and B. Leibe, "In defense of the triplet loss for person re-identification," *arXiv:1703.07737*, 2017.

[51] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016.

[52] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *CVPR*, 2015.

[53] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *NeurIPS*, 2014.

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

[55] P. Izmailov, D. Podoprikhin, T. Garipov, D. Vetrov, and A. G. Wilson, "Averaging weights leads to wider optima and better generalization," in *UAI*, 2018.

[56] L. Zhang, X. Li, A. Arnab, K. Yang, Y. Tong, and P. H. Torr, "Dual graph convolutional network for semantic segmentation," in *BMVC*, 2019.

[57] L. Zhang, D. Xu, A. Arnab, and P. H. Torr, "Dynamic graph message passing network," in *CVPR*, 2020.

[58] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in pytorch," 2017.

[59] G. Ros, L. Sellart, J. Materzynska, D. Vazquez, and A. M. Lopez, "The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes," in *CVPR*, 2016.

[60] W. Maddern, G. Pascoe, C. Linegar, and P. Newman, "1 Year, 1000km: The Oxford RobotCar Dataset," *The International Journal of Robotics Research (IJRR)*, vol. 36, no. 1, pp. 3–15, 2017. [Online]. Available: http://dx.doi.org/10.1177/0278364916679498

[61] Y. Shan, C. M. Chew, and W. F. Lu, "Semantic-aware short path adversarial training for cross-domain semantic segmentation," *Neurocomputing*, vol. 380, pp. 125–132, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0925231219315656

[62] M. Chen, H. Xue, and D. Cai, "Domain adaptation for semantic segmentation with maximum squares loss," in *ICCV*, 2019.

[63] W. Zhou, Y. Wang, J. Chu, J. Yang, X. Bai, and Y. Xu, "Affinity space adaptation for semantic segmentation across domains," *IEEE Transactions on Image Processing*, vol. 30, pp. 2549–2561, 2020.

[64] J. Yang, R. Xu, R. Li, X. Qi, X. Shen, G. Li, and L. Lin, "An adversarial perturbation oriented domain adaptation approach for semantic segmentation," in *AAAI*, 2020.

[65] Z. Wang, M. Yu, Y. Wei, R. Feris, J. Xiong, W.-m. Hwu, T. S. Huang, and H. Shi, "Differential treatment for stuff and things: A simple unsupervised domain adaptation method for semantic segmentation," in *CVPR*, 2020.

[66] T.-H. Vu, H. Jain, M. Bucher, M. Cord, and P. Pérez, "Advent: Adversarial entropy minimization for domain adaptation in semantic segmentation," in *CVPR*, 2019.

[67] Y. Zhang, Z. Qiu, T. Yao, D. Liu, and T. Mei, "Fully convolutional adaptation networks for semantic segmentation," in *CVPR*, 2018.

[68] G. Li, G. Kang, W. Liu, Y. Wei, and Y. Yang, "Content-consistent matching for domain adaptive semantic segmentation," in *ECCV*, 2020.

[69] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv:1409.1556*, 2014.

[70] Y. Zhang, P. David, and B. Gong, "Curriculum domain adaptation for semantic segmentation of urban scenes," in *ICCV*, 2017.

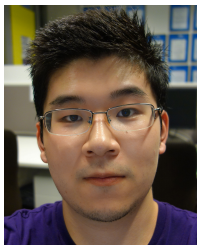
[71] Q. Wang, J. Gao, and X. Li, "Weakly supervised adversarial domain adaptation for semantic segmentation in urban scenes," *IEEE Transactions on Image Processing*, vol. 28, no. 9, pp. 4376–4386, 2019.

[72] I. Sutskever, J. Martens, G. Dahl, and G. Hinton, "On the importance of initialization and momentum in deep learning," in *ICML*, 2013.

[73] X. Gastaldi, "Shake-shake regularization," *arXiv:1705.07485*, 2017.

[74] T. Miyato, S.-i. Maeda, M. Koyama, and S. Ishii, "Virtual adversarial training: a regularization method for supervised and semi-supervised learning," *IEEE transactions on pattern analysis and machine intelligence*, vol. 41, no. 8, pp. 1979–1993, 2018.

[75] X. Wei, B. Gong, Z. Liu, W. Lu, and L. Wang, "Improving the improved training of wasserstein gans: A consistency term and its dual effect," *ICLR*, 2018.



Zhedong Zheng received the Ph.D. degree from the University of Technology Sydney, Australia, in 2021 and the B.S. degree from Fudan University, China, in 2016. He is currently a postdoctoral researcher at NExT++, School of Computing, National University of Singapore. He was an intern at Nvidia Research (2018) and Baidu Research (2020). His research interests include robust learning for image retrieval, generative learning for data augmentation, and unsupervised domain adaptation.

Yi Yang received the Ph.D. degree in computer science from Zhejiang University, China, in 2010. He is currently a professor with the College of Computer Science and Technology, Zhejiang University, China. He was a professor with University of Technology Sydney, Australia. Prior to joining UTS, he was a postdoctoral researcher with the School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, USA. His current research interest includes machine learning and its applications to multimedia content analysis and computer vision, such as multimedia analysis and video semantics understanding.