Uncertainty-Aware Source-Free Domain Adaptive Semantic Segmentation

Uncertainty-aware Source-free Domain Adaptive Semantic Segmentation

Zhihe Lu, Da Li, Yi-Zhe Song, *Senior Member, IEEE*, Tao Xiang, Timothy M. Hospedales, *Senior Member, IEEE*

***Abstract*—Source-Free Domain Adaptation (SFDA) is becoming topical to address the challenge of distribution shift between training and deployment data, while also relaxing the requirement of source data availability during target domain adaptation. In this paper, we focus on SFDA for semantic segmentation, in which pseudo labeling based target domain self-training is a common solution. However, pseudo labels generated by the source models are particularly unreliable on the target domain data due to the domain shift issue. Therefore, we propose to use Bayesian Neural Network (BNN) to improve the target self-training by better estimating and exploiting pseudo-label uncertainty. With the uncertainty estimation of BNNs, we introduce two novel self-training based components: Uncertainty-aware Online Teacher-Student Learning (UOTSL) and Uncertainty-aware FeatureMix (UFM). Extensive experiments on two popular benchmarks, GTA5 $\rightarrow$ Cityscapes and SYNTHIA $\rightarrow$ Cityscapes, show the superiority of our proposed method with mIoU gains of 3.6% and 5.7% over the state-of-the-art respectively.**

***Index Terms*—Source-free Domain Adaptation, Semantic Segmentation, Self-training, Bayesian Neural Network, Uncertainty Estimation.**

I. Introduction

Semantic segmentation is a fundamental task in computer vision that has been widely studied for decades [1]. With the advance of deep learning in the past years, semantic segmentation performance has been improved dramatically [2]. However, such strong performance relies heavily on large-scale annotated training data, which is extremely expensive in dense pixel-wise annotation for segmentation masks. For example, the popular COCO [3] segmentation dataset with only 80 commonly-seen object classes has taken over 70,000 worker hours. Meanwhile, in line with other applications in machine learning and computer vision [4], [5], segmentation performance degrades rapidly under domain shift that inevitably occurs between training data and deployment conditions [6]. This is a severe problem for semantic segmentation in practice, as it is infeasible to repeat dense annotation for each deployment domain.

Unsupervised domain adaptation (UDA) has thus gained significant interest as a route to tackling this problem by adapting models trained on labeled source domain data to unlabeled target domain data [6]–[8], [8], [9], [9]–[14]. The majority of UDA methods assume the source and target domain data is available jointly such that the trained models can adapt to the target domain by methods such as adversarial training [8], [9] and feature alignment [15], [16]. However, the requirement to jointly process both labeled source and unlabeled target domain data is prohibitive in reality. *e.g.*, the source data has privacy or copyright constraints, or is simply too large to be re-distributed.

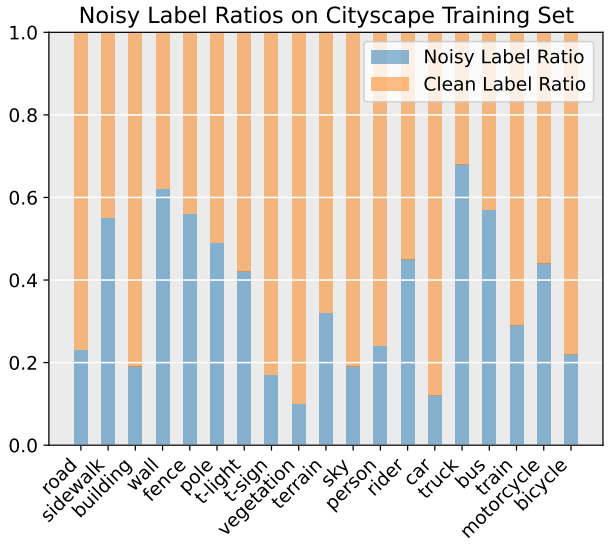


Fig. 1: blue Directly deploying the source model [17] on target domain leads to large Pseudo-Label (PL) noise. The top-5 wrongly predicted classes per category are further given in Table I.

For this reason, Source-Free Domain Adaptation (SFDA) is becoming topical [17], [18]. In this setting, only the source trained model and unlabeled target domain data are available during target domain adaptation. blue Most approaches to SFDA are based on self-training in the target domain using some kind of pseudo labels generated from the source model [17], [18]. This is because pseudo labeling is the most effective technique for a SFDA model to achieve good performance, thus the focus of this work. However, pseudo labels generated from the source model are not always accurate in the target domain data due to distribution-shift (blue see noisy label ratio in Figure 1 and top-5 wrongly predicted classes per category in Table I), or there would be no need for adaptation. blue These methods therefore attempt to make use of prediction confidence thresholds to select reliable pseudo labels, which are less affected by noise and enjoy high confidence (above

Manuscript received April 19, 2021; revised August 16, 2021.

Zhihe Lu (e-mail: zhihelu.academic@gmail.com), Yi-Zhe Song and Tao Xiang are with the Centre for Vision Speech and Signal Processing (CVSSP), University of Surrey, and also with iFlyTek-Surrey Joint Research Center on Artificial Intelligence, University of Surrey, Guildford GU2 7XH, United Kingdom.

Da Li and Timothy M. Hospedales are with Samsung AI center, Cambridge CB1 2JH, United Kingdom.

TABLE I: blue Top-5 wrongly predicted classes per category. The results indicate that the model trained on source domain tends to mis-classify the classes sharing similar appearances, thereby leading to label noise when deploying the model on target data for pseudo label generation.

class/top-k	road	sidewalk	building	wall	fence	pole	t-light	t-sign	vegetation
1	car	road	vegetation	building	building	building	building	building	building
2	sidewalk	building	sky	vegetation	vegetation	vegetation	vegetation	vegetation	terrain
3	building	terrain	fence	fence	wall	fence	pole	pole	sky
4	terrain	car	wall	terrain	sidewalk	sidewalk	sky	t-light	fence
5	fence	wall	pole	sidewalk	terrain	sky	fence	fence	pole

class/top-k	terrain	sky	person	rider	car	truck	bus	train	motorcycle	bicycle
1	sidewalk	building	building	person	truck	building	truck	building	car	building
2	road	vegetation	vegetation	building	building	car	building	bus	building	vegetation
3	vegetation	terrain	sidewalk	vegetation	road	vegetation	car	truck	vegetation	road
4	building	pole	car	car	vegetation	road	vegetation	car	rider	car
5	car	t-light	road	bicycle	bus	wall	road	fence	road	person

a certain threshold) predicted by a trained model. However, the adaptation process then hinges on model uncertainty estimation: can prediction uncertainty be used to differentiate between reliable pseudo labels that have been correctly estimated and should be used for learning; *vs.* unreliable pseudo labels that were not correctly estimated and should not be used for adaptation? This question has not been studied well in source-free domain adaptive semantic segmentation, and unfortunately conventional neural networks used in modern segmentation systems are well known to suffer from poor uncertainty estimation [19].

To address this issue, our first contribution is to introduce a Bayesian Neural Network (BNN) based adaptation approach to SFDA. BNNs are widely considered the gold standard for providing proper uncertainty estimation [20], [21], but are not often used in domain adaptation. In particular, for the neural network blocks to be adapted from source to target, we use the source pre-trained model to define a prior distribution over neural network weights. Then during the adaptation process, we learn a posterior distribution over the target domain adapted weights. Learning this posterior BNN enables us to better estimate both target domain confidence and uncertainty (noise ratio descends/ascends wrt confidence and uncertainty respectively in Figure 2) and thus develop an improved adaptation pipeline.

Building on our BNN, we then introduce two novel self-training based components that exploit uncertainty estimation: Uncertainty-aware Online Teacher-Student Learning (UOTSL) and Uncertainty-aware FeatureMix (UFM). blue First, we construct a teacher-student learning pipeline, in which predictions averaged by multiple Monte Carlo (MC) samples of our BNN after $argmax$ operation are regarded as teacher pseudo labels to guide the (student) predictions. However, as with any teacher-student setup, incorrect teacher predictions can be detrimental for the student. Therefore, we use the uncertainty estimation of our BNN to weight each teacher supervision of the student. That is, a pixel-wise uncertainty map is employed to weight the supervision to emphasis the effect of high certainty pseudo labels. Second, we take inspiration from the semi-supervised learner ClassMix [22], and propose a new feature-space extension called FeatureMix, that enjoys better performance and faster speed. blue Vanilla ClassMix essentially performs data augmentation by sampling pseudo-labeled class masks from two images and generating a new synthetic image for training. As with any pseudo-label setup, this introduces detrimental noise if pseudo labels are incorrect. Therefore we introduce uncertainty-awareness into FeatureMix such that class-wise masks with high certainty are preferred for data generation. Concretely, the class-wise uncertainty is estimated by our BNN based MC sampling, which then can be used as guidance for class selection, *i.e.*, classes with higher certainty are more likely to be chosen. Together these uncertainty aware training objectives enable effective adaptation with reduced influence of incorrect self-training supervision (a consistent lower noisy label ratio of our BNN shown in Figure 2 (left)), thus leading to the state-of-the-art SFDA for semantic segmentation.

We summarize our **contributions** as follows:

- We provide the first analysis of the pseudo label noise issue in self-training of SFDA for semantic segmentation, and the first solution by upgrading the network backbone to a Bayesian Neural Network (BNN) which enables uncertainty estimation for improved self-training.
- We introduce an Uncertainty-aware Online Teacher-Student Learning (UOTSL) pipeline that adapts to the unlabeled target domain by using a multiple Monte Carlo sample teacher to guide a student network; and ensure its reliability by exploiting prediction uncertainty.
- Furthermore we propose Uncertainty-aware FeatureMix (UFM), a faster feature-level extension of ClassMix [22] with uncertainty-awareness. UFM mask generation exploits BNN prediction uncertainty to ensure high-quality images and labels are synthesized for training.
- Extensive experiments on two popular benchmarks demonstrate that our method outperforms the existing state of the art significantly.

II. Related Work

A. UDA for Semantic Segmentation

UDA for semantic segmentation has been widely studied in the literature [6]–[9]. These methods can be categorized into three groups: generative model based [8]–[10], [12]–[14], feature-alignment based [15], [16], [23], [24] and self-training

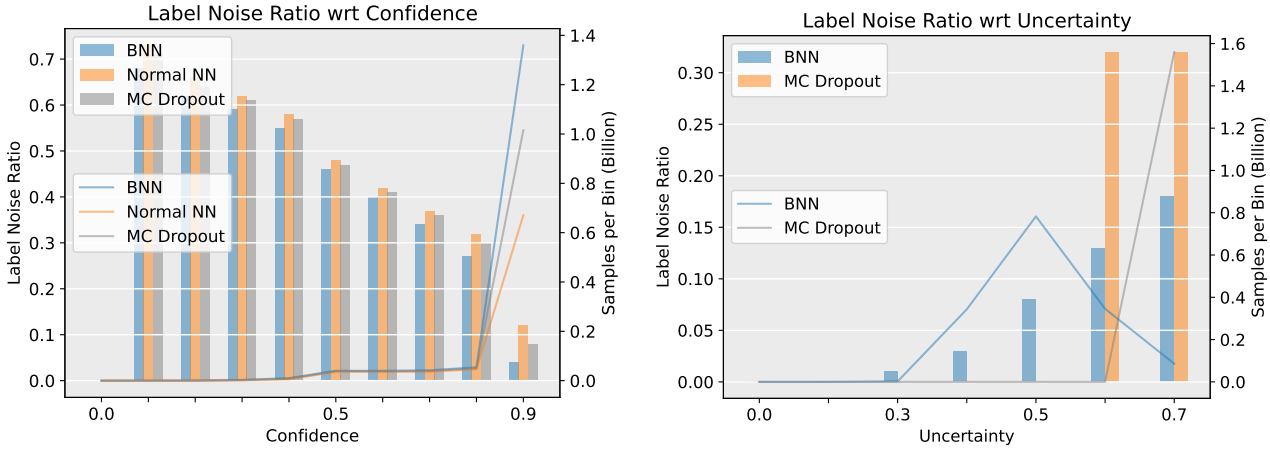


Fig. 2: blue **Left**: Histogram of noisy PL by prediction confidence. Lower label noise ratio at high confidence means that detecting noisy PLs by conventional confidence thresholding works better in our BNN than both the normal NN baseline [17] and the MC dropout based method. **Right**: Our BNNs uniquely provides a second way to detect noisy PLs by computing prediction uncertainty over Monte Carlo samples of network weights. High uncertainty predictions are more likely noisy PLs in our BNN, but the noise ratio is not increasing with the higher uncertainty for MC dropout.

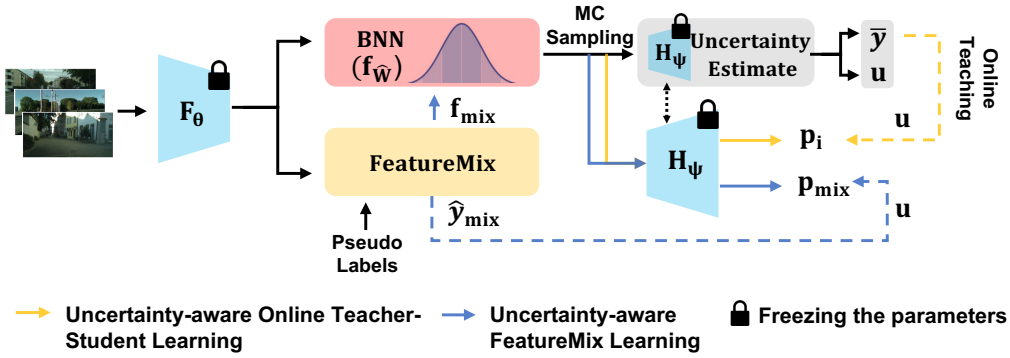


Fig. 3: blue Our method consists of three branches: self-training baseline, Uncertainty-aware Online Teacher-Student Learning (UOTSL) and Uncertainty-aware FeatureMix learning (UFM). BNN adapter ($f_{\hat{\mathbf{W}}}$) is the only trainable part during target adaptation. $\mathbf{F}_\theta$: feature extractor. $\mathbf{H}_\psi$: segmentation head. $\mathbf{p}$: predictions. $(\mathbf{f}_{\text{mix}}, \hat{\mathbf{y}}_{\text{mix}})$: mixed features and labels via the class-wise guidance (pseudo labels). $\mathbf{u}$: estimated uncertainty (Ep. 4). $\bar{\mathbf{y}}$: teacher pseudo labels (Ep. 6). The basic self-training branch is omitted for simplicity.

based [25], [26]. Due to GANs' [27] success of generating high-fidelity images, [8], [9] proposed to transfer the source domain data into the target domain style such that the target-style source data can be leveraged to train the model directly with labels. Feature-alignment based methods often resorted to attention mechanisms [15], [16] or used an extra domain-adversarial discriminator [23], [24]. Self-training approaches based on pseudo labels [28] have also been adapted to UDA for semantic segmentation [26]. As with semi-supervised learning the key issue with self-training and Pseudo Labels (PLs) is to prevent noisy PLs from causing model drift. Thus studies focused on techniques such as regularisation [25] and class-balancing [26] to improve confidence-based detection of valid PLs for training. Our approach is also based on self-training and PLs but we extend to address the source-free case, and propose a BNN-based adaptation framework that enables both confidence- and uncertainty-based rejection of poor PLs.

B. Source-free Domain Adaptation

Source-Free Domain Adaptation (SFDA) assumes that the source data is inaccessible during the target domain adaptation. Due to its practical significance, SFDA for image classification [29], [30] and semantic segmentation [17], [18], [31], [32] have obtained increasing attention recently. For image classification, SHOT [29] proposed to learn a target-specific feature extractor with a frozen source-domain-trained classifier by using information maximization learning and pseudo-labeling based self-training. 3C-GAN [30] leveraged the collaboration between a conditional GAN and the frozen prediction model to synthesize target-style training samples, which can be used for model adaptation. In the semantic segmentation applications, [31] used class-ratio as a prior to guide the entropy minimization for adapting model across different MRI modalities. [18] introduced data-free knowledge distillation and patch-level self-supervision to improve SFDA

for semantic segmentation. [32] proposed to minimize the uncertainty between the feature-corrupted and feature-intact branches. In contrast, [17] focused on training a domain-generalizable source model by mimicking multiple domains with various augmentations before conventional PL-based self-training. In this work, we emphasize the negative impact of PL noise and propose an uncertainty-aware BNN framework to detect noisy PLs. We then show how to use this to enhance both a teacher-student self-training branch and a feature augmentation based self-training branch.

C. Uncertainty Estimation

Uncertainty and confidence estimation in deep learning [20], [21], [33] are increasingly studied due to their significance for trustworthy and explainable AI. This is also important in the context of PL and self-training based semi-supervised learning and domain adaptation, where reliable labels should be selected for learning [17], [32]. However, conventional deep networks are overconfident [19] – assigning high probabilities to incorrect labels; and do not provide any epistemic uncertainty estimates. This undermines existing schemes to select good PLs for training [17], [32]. While Bayesian Neural Networks can in principle provide both confidence (aleatoric uncertainty) and uncertainty (epistemic uncertainty) estimates [20], these have not been widely used in practical semi-supervised or domain-adaptive learning. A few initial attempts have used Monte Carlo dropout as a means to estimate epistemic uncertainty for use with PL weighting [34], based on the notion that it can approximate BNN inference [35]. However, this corresponds to inference in an *arbitrary* BNN. blue In contrast, Ciosek *et al*. [36] proposed to fit randomly initialized prior distributions for uncertainty estimate while Deep Ensembles [37] used model ensemble to against the impact of uncertainty. blue The similar ensemble technique, *e.g*., multi-rater agreement/consensus modeling, was adopted in [38], [39]. blue Moreover, some theoretical research [40] for examining the limitations of common variational methods, *i.e*., mean-field variational inference (MFVI) (used in our BNNs) and Monte Carlo dropout (MCDO) [35], have been proposed. However, their examination was conducted under the low-dimensional and small data regime with mainly 2-hidden layer BNNs, which limits its applicability to scenarios that exist a large amount of data and use multi-layer BNNs, *e.g*., in our set-up. In this paper, we perform variational inference to learn a posterior BNN after adaptation to the target domain, given the source domain model as a prior. We further show how to use the uncertainty estimates of this BNN to improve adaptation performance.

III. Methodology

A. Task Definition and Overview

SFDA for semantic segmentation has two stages: 1) source model training and 2) target domain adaptation. In source model training, a set of source data image-segmentation map pairs $\mathcal{D}_s := \{\mathbf{X}_s, \mathbf{y}_s\}$ are used to train a source model. During target domain adaptation, only the trained source model and unlabeled target domain data $\mathcal{D}_t := \{\mathbf{X}_t, null\}$ can be leveraged. We follow the same setting as [17] for the source model training and focus on improving the target domain adaptation in this paper. Our method consists of ResNet feature extraction followed by three head branches: basic self-training (reused from [17], omitted for simplicity), Uncertainty-aware Online Teacher-Student Learning (UOTSL) and Uncertainty-aware FeatureMix (UFM), as shown in Figure 3. We follow [17]'s choice of feature extraction modules to update, but differently learn a BNN posterior for the adapted module, rather than a point estimate for it. We denote the frozen feature extractor and segmentation head as f_θ and f_ψ, and the adaptive BNN modules or adapter as $f_{\hat{\mathbf{W}}}$, which will be detailed in the following section.

B. Bayesian Neural Networks

Given a pretrained source model $\mathbf{w}_s$, one typically adapts it on $\mathcal{D}_t$ using some unsupervised objective, such as self-training. In our framework, we use source model $\mathbf{w}_s$ to define a fixed-variance prior $p(\mathbf{W}) = \mathcal{N}(\mathbf{w}_s, \sigma_0^2)$, which we adapt to a BNN posterior model $p(\mathbf{W}|\mathcal{D}_t) = \mathcal{N}(\mathbf{w}_t, \sigma_t^2)$ using the target data $\mathcal{D}_t$, by learning new mean $\mathbf{w}_t$ and variance σ_t^2 parameters. Specifically, given the training samples $\mathbf{X}_t$ from $\mathcal{D}_t$ and the current pseudo labels $\hat{\mathbf{Y}}$ as $\hat{\mathcal{D}}_t := \{\mathbf{X}_t, \hat{\mathbf{Y}}\}$, the goal is to estimate a posterior distribution $p(\mathbf{W}|\hat{\mathcal{D}}_t)$. However, directly computing $p(\mathbf{W}|\hat{\mathcal{D}}_t) = p(\mathbf{W}, \hat{\mathcal{D}}_t)/p(\hat{\mathcal{D}}_t)$ is intractable, due to the unknown marginal $p(\hat{\mathcal{D}}_t)$. Therefore, we use variational inference [41], [42] and define a variational distribution $q(\mathbf{W})$ to approximate the posterior $p(\mathbf{W}|\hat{\mathcal{D}}_t)$ by minimizing their KL divergence

$$\begin{aligned} q(\mathbf{W}) &= \operatorname{argmin} \ \mathrm{KL}(q(\mathbf{W})||p(\mathbf{W}|\hat{\mathcal{D}}_t)) \\ &= \operatorname{argmin} \ \mathbb{E}[\log q(\mathbf{W})] - \mathbb{E}[\log p(\mathbf{W}, \hat{\mathcal{D}}_t)] + \log p(\hat{\mathcal{D}}_t) \\ &= \operatorname{argmin} \ -\mathrm{ELBO}(\mathbf{W}, \hat{\mathcal{D}}_t) + \log p(\hat{\mathcal{D}}_t). \end{aligned} \tag{1}$$

where ELBO is the evidence lower bound objective,

$$\begin{aligned} &\mathrm{ELBO}(\mathbf{W}, \hat{\mathcal{D}}_t) \\ &= \mathbb{E}[\log p(\mathbf{W}, \hat{\mathcal{D}}_t)] - \mathbb{E}[\log q(\mathbf{W})] \\ &= \mathbb{E}[\log p(\mathbf{W})] + \mathbb{E}[\log p(\hat{\mathcal{D}}_t|\mathbf{W})] - \mathbb{E}[\log q(\mathbf{W})] \\ &= \mathbb{E}[\log p(\hat{\mathcal{D}}_t|\mathbf{W})] - \mathrm{KL}(q(\mathbf{W})||p(\mathbf{W})) \\ &= \mathbb{E}[\log p(\hat{\mathbf{Y}}|\mathbf{X}_t, \mathbf{W})] - \mathrm{KL}(q(\mathbf{W})||p(\mathbf{W})) \end{aligned} \tag{2}$$

Maximizing ELBO is equivalent to minimizing the KL divergence in Eq. 1. The first term in Eq. 2 corresponds to the standard pseudo-label self-training objective $\mathcal{L}_p$, and the second to prior regularization $\mathcal{L}_{kld}$. Both can be estimated by Monte Carlo samples of $\mathbf{W}$ [41], [42]. Rather than using a standard Gaussian, $q(\mathbf{W})$ is set as $p(\mathbf{W})$ in our case.

1) Uncertainty Estimation with BNNs: Standard NNs estimate prediction confidence via the label posterior $p(\mathbf{y}|\mathbf{x}, \hat{\mathbf{W}})$ where $\hat{\mathbf{W}}$ are a set of learned weights using $\hat{\mathcal{D}}_t$. This enables, *e.g*., rejecting of pseudo labels $\hat{\mathbf{y}}$ as unreliable if their confidence is below a threshold $p(\hat{\mathbf{y}}|\mathbf{x}, \hat{\mathbf{W}}) < \tau$ [17], [32]. BNNs provide the opportunity to marginalize over the weights as $p(\hat{\mathbf{y}}|\mathbf{x}, \hat{\mathcal{D}}_t) = \int_{\hat{\mathbf{W}}} p(\hat{\mathbf{y}}|\mathbf{x}, \hat{\mathbf{W}}) p(\hat{\mathbf{W}}|\hat{\mathcal{D}}_t)$, thus accounting for weight uncertainty in inference. This has two effects: (i) It improves standard *confidence* calibration [21], as illustrated in

Figure 2 (left). This makes pseudo-label filtering $p(\hat{\mathbf{y}}|\mathbf{x}, \hat{\mathcal{D}}_t) < \tau$ more reliable. (ii) blue It provides the ability to estimate *uncertainty* of predictions based on multiple sets of sampled weights, which conventional NNs cannot provide [20], as illustrated in Figure 2 (right).

With our BNN, we can draw N Monte Carlo samples from the weight posterior $p(\hat{\mathbf{W}}|\hat{\mathcal{D}}_t)$ and compute prediction uncertainty w.r.t. these samples. Specifically, given T predictions for each of B input images, we have a prediction tensor $\mathbf{P} = [\mathbf{p}_1, \ldots, \mathbf{p}_T] \in \mathbb{R}^{T\times B\times C\times H\times W}$, we estimate uncertainty by calculating the standard deviation $\boldsymbol{\sigma}$ as,

$$\boldsymbol{\mu} = \frac{1}{T}\sum_{i}^{T}\mathbf{P}_i \tag{3}$$

$$\boldsymbol{\sigma} = \sqrt{\frac{1}{T}\sum_{i}^{T}(\mathbf{p}_i - \boldsymbol{\mu})^2}. \tag{4}$$

Finally, we average $\boldsymbol{\sigma}$ along dimension C to get pixel-wise standard deviation (STD) $\boldsymbol{\sigma}^* \in \mathbb{R}^{B\times H\times W}$. We then normalize each element of $\boldsymbol{\sigma}^*$ to $(0, 1]$ as

$$\mathbf{u} = e^{-\boldsymbol{\sigma}^*} \in \mathbb{R}^{B\times H\times W}. \tag{5}$$

$\mathbf{u}$ now provides a pixel-wise uncertainty measure for each image, which we will use to guide self-training based domain adaptation, *e.g.*, by down-weighting high uncertainty pixel pseudo labels.

blue

a) The Advantage of Using Uncertainty: The optimization on deterministic weights used in normal neural networks often fails into a sub-optimal solution when noisy labels exist, while the estimated uncertainty of predictions from multiple sets of weights by our BNN can be leveraged to downplay the low certain predictions and stress the high certain ones, thereby alleviating the noisy impact and approaching to the optimal solution.

C. Uncertainty-aware Online Teacher-Student Learning

We introduce here our first new self-training based component, uncertainty-aware online teacher-student learning. Given an input $\mathbf{x}$ we can use Monte Carlo sampling of a BNN model to get T different corresponding predictions $[\mathbf{p}_1, \ldots, \mathbf{p}_T]$. Then, for the teacher branch, we average the multiple predictions and generate a pseudo label as

$$\bar{\mathbf{y}} = \text{argmax}(\bar{\mathbf{p}}) = \text{argmax}(\text{mean}[\mathbf{p}_1, \ldots, \mathbf{p}_T]) \tag{6}$$

, and use it to guide a single (student) prediction $\mathbf{p}_i$. Importantly, since pseudo labels provided by the teacher supervisions vary in reliability, we use our uncertainty estimator $\mathbf{u}$ (Sec III-B, Eq. 5) to weight the loss, leading to

$$\mathcal{L}_{\text{uotsl}} = \frac{1}{|\mathcal{D}_t|}\sum_{\mathcal{D}_t}\frac{1}{HW}\sum_{h,w}^{H,W}\mathbf{u}^{h,w}\ell_{ce}(\bar{\mathbf{y}}^{h,w}, \mathbf{p}_i^{h,w}), \tag{7}$$

which automatically down-weights noisy teacher pseudo labels.

D. Uncertainty-aware FeatureMix

1) Revisting ClassMix.: ClassMix [22] is a strong augmentation mechanism, shown successful in semi-supervised semantic segmentation. A network's predicted pseudo-label masks are used to mix two initially unlabeled samples to a new synthesized labeled sample. Given two images $\mathbf{x}_A$ and $\mathbf{x}_B$ and their pseudo-label masks $\hat{\mathbf{y}}_A, \hat{\mathbf{y}}_B$, ClassMix produces a new image-mask pair $\mathbf{x}_C, \hat{\mathbf{y}}_C$ for learning by pixel-wise binary mixing of both input images and labels. Specifically, ClassMix randomly selects half of classes shown in $\mathbf{x}_A$ as foreground and the remaining classes as the background, defining a pixel-wise binary mask $\mathbf{m}$. Applying this mask, a new mixed image and pseudo mask are generated as follows:

$$\mathbf{x}_C \leftarrow \mathbf{m}\odot\mathbf{x}_A + (1-\mathbf{m})\odot\mathbf{x}_B$$
$$\hat{\mathbf{y}}_C \leftarrow \mathbf{m}\odot\hat{\mathbf{y}}_A + (1-\mathbf{m})\odot\hat{\mathbf{y}}_B$$

where $\odot$ is element-wise multiplication, $\mathbf{x}_C$ the new mixed image and $\hat{\mathbf{y}}_C$ the corresponding mixed pseudo mask. The newly generated images and pseudo-label masks can then be added to enrich the training data. However, through visualizing the mixed image, we found arbitrarily selecting half of classes shown in $\mathbf{x}_A$ often generates some unrealistic images (as shown in top two rows of Figure 4), which do not benefit training. This is because the learned model is needed to work for the real data.

2) Uncertainty-aware FeatureMix.: To address the aforementioned problem in ClassMix, we propose an uncertainty-aware sampling where classes with higher certainty have higher probability to be sampled. In addition, we do the class-mixing in the feature level. The implementation details are shown in Algorithm 1.

a) Uncertainty-aware Sampling: Given an input image $\mathbf{x}_A$, we can obtain its predicted segmentation probability map using the training model, and maximize over classes to obtain a pseudo mask $\hat{\mathbf{y}}_A$. Now we define a class-wise sampling probability $\mathbf{q}$ taking into account the prediction uncertainty $\mathbf{u}$ from Sec III-B Eq. 5 as

$$\mathbf{q}^k = \frac{1}{HW}\sum_{h,w}^{H,W} p(\hat{\mathbf{y}}_A^{h,w} = k)\odot\mathbf{u}^{h,w} \tag{8}$$

where k is the class index. Now sampling foreground classes from $\mathbf{q}^k$, rather than uniformly as in [22], leads to high-certainty classes being preferentially sampled.

b) FeatureMix: SFDA aims to adapt the source-domain trained model to the unlabeled target domain data. No adequate and accurate supervision during model adaptation makes training the whole model error-prone. Existing methods empirically found adapting part of the source model, *e.g.*, feature extractor [29] and $Block3$ of ResNet-101 [17], performs well. We follow [17] such that the early layers before $Block3$ are frozen during target adaptation. Denoting $\mathbf{f}_A = f_\theta(\mathbf{x}_A)$ as the feature encoding of image $\mathbf{x}_A$ prior to the adaptation block (resp. $\mathbf{f}_B = f_\theta(\mathbf{x}_B)$ and $\mathbf{x}_B$), feature mixing computes

$$\mathbf{f}_C \leftarrow \mathbf{m}\odot\mathbf{f}_A + (1-\mathbf{m})\odot\mathbf{f}_B$$
$$\hat{\mathbf{y}}_C \leftarrow \mathbf{m}\odot\hat{\mathbf{y}}_A + (1-\mathbf{m})\odot\hat{\mathbf{y}}_B$$

Fig. 4: Top two rows are from ClassMix while bottom two rows are from ClassMix with uncertainty-aware sampling. From left to right: image A, image B, binary mask generated from image A's pseudo mask and mixed image. Mixed images from ClassMix are obviously artificial, *e.g.*, the red-circle regions. Zoom in for better observation

Algorithm 1: Uncertainty-aware FeatureMix

Inputs: Two unlabeled images $\mathbf{x}_A$ and $\mathbf{x}_B$;
Model: Fixed feature encoder f_θ and segmentation head f_ψ; a learnable BNN adapter $f_{\hat{\mathbf{W}}}$;
Uncertainty Map: $\mathbf{u}_A$ of $\mathbf{x}_A$ via Eq. 4 and Eq. 5;
$(\mathbf{f}_A, \mathbf{f}_B) \leftarrow (f_\theta(\mathbf{x}_A), f_\theta(\mathbf{x}_B))$;
$\mathbf{p}_A \leftarrow \mathrm{SF}((f_\psi \circ f_{\hat{\mathbf{W}}})(\mathbf{f}_A))$ ▷ SF is softmax;
$\mathbf{p}_B \leftarrow \mathrm{SF}((f_\psi \circ f_{\hat{\mathbf{W}}})(\mathbf{f}_B))$;
$\hat{\mathbf{y}}_A \leftarrow \mathrm{argmax}(\mathbf{p}_A, \mathrm{dim} = 1)$ ▷ Pseudo mask;
$\hat{\mathbf{y}}_B \leftarrow \mathrm{argmax}(\mathbf{p}_B, \mathrm{dim} = 1)$;
$\mathcal{C} \leftarrow$ Set of the classes present in $\hat{\mathbf{y}}_A$;
Get uncertainty-aware ($\mathbf{u}_A$) class-wise sampling probability $\mathbf{q}$ as Eq. 8;
$\{c\} \leftarrow$ Sample $\frac{|\mathcal{C}|}{2}$ classes according to $\mathbf{q}$;
$\mathbf{m}(i,j)$=1 if $\hat{\mathbf{y}}_A(i,j) \in \{c\}$ else 0 ▷ Binary mask;
$\mathbf{f}_C \leftarrow \mathbf{m} \odot \mathbf{f}_A + (1 - \mathbf{m}) \odot \mathbf{f}_B$ ▷ Mixed feature;
$\hat{\mathbf{y}}_C \leftarrow \mathbf{m} \odot \hat{\mathbf{y}}_A + (1 - \mathbf{m}) \odot \hat{\mathbf{y}}_B$ ▷ Mixed mask;
return $\mathbf{f}_C, \hat{\mathbf{y}}_C$;

We now have a synthetic example $\{\mathbf{f}_C, \hat{\mathbf{y}}_C\}$ for which we can supervise the network's prediction $\mathbf{p}_C$.

$$\mathcal{L}_{\text{ufm}} = \frac{1}{|\mathcal{D}_t|} \sum_{\mathbf{x}_A, \mathbf{x}_B \sim \mathcal{D}_t} \frac{1}{HW} \sum_{h,w}^{H,W} \mathbf{u}^{h,w} \cdot \ell_{ce}(\hat{\mathbf{y}}_C^{h,w}, \hat{\mathbf{y}}_{pred}) \quad (9)$$

where $\hat{\mathbf{y}}_{pred} = f_\psi(f_{\hat{\mathbf{W}}}(\mathbf{f}_C))^{h,w}$ and the uncertainty map again weights supervisions according to their reliability as estimated by our uncertainty estimator $\mathbf{u}$ as in Eq. 7, resulting in an uncertainty-aware self-training.

E. blue Loss Function

In summary, our framework as shown in Figure 3 performs BNN adaptation of a source model to the target domain. This is driven by three heads/objectives: The standard pseudo-label loss $\mathcal{L}_p$ and KL regularizer from from Eq. 2, the uncertainty-aware teacher-student loss $\mathcal{L}_{\text{uotsl}}$ from Eq. 7 and the uncertainty-aware FeatureMix loss $\mathcal{L}_{\text{ufm}}$ from Eq. 9. The complete loss function is

$$\mathcal{L} = \mathcal{L}_{\text{p}} + \mathcal{L}_{\text{uotsl}} + \mathcal{L}_{\text{ufm}} + \lambda \mathcal{L}_{\text{kld}}. \quad (10)$$

IV. Experiments

A. Datasets and Evaluation Metric

a) Datasets: We evaluate the proposed method on two commonly-used synthetic-to-real benchmarks: GTA5 [52] → Cityscapes [53] and SYNTHIA [54] → Cityscapes. Specifically, the GTA5 dataset is collected from a popular video game GTA5 with 24,966 images. We follow [17] exactly to split it into a training set (24,500 images) and a validation set (466 images). For SYNTHIA, a dataset rendered from a virtual city scene, it has 9,400 images with 9,000 images used for training and 400 images for validation. The target dataset, *i.e.*, Cityscapes, contains 2,975/500 training/validation images from real street-view scenes in 50 different cities. As per [17], [55], GTA5 images are resized to 1280×720 then randomly cropped to 1024×512; SYNTHIA images are resized to 1280×760 before random crop to 1024×512; Cityscapes images are directly resized to 1024×512.

TABLE II: Quantitative evaluation on GTA5→Cityscapes. Architecture: DeepLab-V2 with ResNet-101. SF: source-free setting. †: reproduced by us. Best results are in **bold**.

Method	SF	road	sidewalk	building	wall	fence	pole	t-light	t-sign	vegetation	terrain	sky	person	rider	car	truck	bus	train	motorcycle	bicycle	mIoU
STAR [11]	×	88.4	27.9	80.8	27.3	25.6	26.9	31.6	20.8	83.5	34.1	76.6	60.5	27.2	84.2	32.9	38.2	1.0	30.2	31.2	43.6
CBST [26]	×	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
PLCA [7]	×	84.0	30.4	82.4	35.3	24.8	32.2	36.8	24.5	85.5	37.2	78.6	66.9	32.8	85.5	40.4	48.0	8.8	29.8	41.8	47.7
CrCDA [43]	×	92.4	55.3	82.3	31.2	29.1	32.5	33.2	35.6	83.5	34.8	84.2	58.9	32.2	84.7	40.6	46.1	2.1	31.1	32.7	48.6
PIT [44]	×	87.5	43.4	78.8	31.2	30.2	36.3	39.9	42.0	79.2	37.1	79.3	65.4	37.5	83.2	46.0	45.6	25.7	23.5	49.9	50.6
TPLD [45]	×	94.2	60.5	82.8	36.6	16.6	39.3	29.0	25.5	85.6	44.9	84.4	60.6	27.4	84.1	37.0	47.0	31.2	36.1	50.3	51.2
RPT [46]	×	89.7	44.8	86.4	44.2	30.6	41.4	51.7	33.0	87.8	39.4	86.3	65.6	24.5	89.0	36.2	46.8	17.6	39.1	58.3	53.2
FADA [47]	×	91.0	50.6	86.0	43.4	29.8	36.8	43.4	25.0	86.8	38.3	87.4	64.0	38.0	85.2	31.6	46.1	6.5	25.4	37.1	50.1
IAST [48]	×	94.1	58.8	85.4	39.7	29.2	25.1	43.1	34.2	84.8	34.6	88.7	62.7	30.3	87.6	42.3	50.3	24.7	35.2	40.2	52.2
DACS [49]	×	89.9	39.7	87.9	30.7	39.5	38.5	46.4	52.8	88.0	44.0	88.8	67.2	35.8	84.5	45.7	50.2	0.0	27.3	34.0	52.1
CorDA [50]	×	94.7	63.1	87.6	30.7	40.6	40.2	47.8	51.6	87.6	47.0	89.7	66.7	35.9	90.2	48.9	57.5	0.0	39.8	56.0	56.6
ProDA [51]	×	87.8	56.0	79.7	46.3	44.8	45.6	53.5	53.5	88.6	45.2	82.1	70.7	39.2	88.8	45.5	59.4	1.0	48.9	56.4	57.5
URMA [32]	√	92.3	55.2	81.6	30.8	18.8	37.1	17.7	12.1	84.2	35.9	83.8	57.7	24.1	81.7	27.5	44.3	**6.9**	24.1	40.4	45.1
SRDA [31]	√	90.5	47.1	82.8	32.8	28.0	29.9	35.9	34.8	83.3	39.7	76.1	57.3	23.6	79.5	30.7	40.2	0.0	26.6	30.9	45.8
GTA [17]	√	90.9	48.6	85.5	35.3	**31.7**	36.9	34.7	34.8	86.2	**47.8**	88.5	61.7	**32.6**	**85.9**	**46.9**	**50.4**	0.0	38.9	52.4	51.6
GTA† [17]	√	90.1	46.3	83.8	33.6	28.6	34.8	35.3	29.0	85.1	46.4	86.1	61.0	31.8	85.3	41.9	46.9	0.0	**39.7**	45.3	50.1
Ours	√	**94.4**	**64.5**	**86.6**	**42.3**	28.4	**38.0**	**43.8**	**45.3**	**86.7**	47.0	**90.0**	**62.6**	32.2	84.9	36.1	46.0	0.0	38.1	**53.8**	**53.7**

b) Evaluation Metric: We report IoU per class and mIoU over all classes. For GTA5 → Cityscapes, IoU for 19 classes are reported. For SYNTHIA → Cityscapes, we report both 13-class and 16-class IoU results [17]. We test our model on the standard validation set as per [17], [55]. We also adopt multi-scale evaluation as the final results following [17], [26], [47], [48].

B. Implementation Details

a) Network Architecture: We use DeepLab-V2 [56] with backbone ResNet-101 [57] in this work as per [17], [34], [55], [58].

b) Source Training: We adopt five augmentations following [17]: FDA [55], style [59], AdaIN [60], weather [61], cartoon [62], to train one global head H_g and five leave-one-out heads $H_i, i \in \{1, 5\}$. We use SGD optimizer (momentum 0.9, weight decay $5e-4$), initial learning rate $2.5e-4$ with a polynomial learning rate decay powered by 0.9, batch size 4, and training iterations $50k$.

c) Target Adaptation: Self-training with pseudo labels is our baseline method for target adaptation as in [17]. Concretely, pseudo labels are generated by averaging the predictions over all heads with class-wise confidence thresholds as per [17]. To initialize the target training, the optimal head is chosen by the lowest averaged entropy on the target training set. When training, only $Block3$ is trainable while other modules are frozen. We replace the standard Conv layers in $Block3$ with Bayesian Neural Network (BNN) layers. The means of posterior and prior Gaussian distributions in BNN are initialized by weights of the trained source model while the diagonal variance matrix is randomly initialized. As per [17], [55], we use SGD optimizer with batch size 4, momentum 0.9, weight decay $5e-4$. The learning rate is initialized as $2.5e-4$ scheduled with a polynomial learning rate decay powered by 0.9. Each round of self-training has $50k$ iterations. We do 3-round self-training as in [17]. λ is set as 0 which is the best as we found. We also adopt entropy minimization learning as per [17]. During inference, only the mean of the BNN posterior distribution is used without including extra cost beyond with a typical deterministic model.

C. Comparison with State-of-the-art

a) GTA5 → Cityscapes Results: Table II compares our results against existing domain adaptive semantic segmentation methods with/without source-free setting. Overall, our proposed method achieves a new state-of-the-art performance for most cases, often outperforming other competitors by a large margin. Specifically, our method surpasses the best prior art [17] by **2.1%** (comparing reported) / **3.6%** (comparing reproduced). We also have two interesting observations from experiments. First, the proposed source-free method can even beat the majority of non-source-free competitors. This is because we address a key issue, *i.e.*, pseudo label noise, and design a targeted solution to deal with it, thereby gaining even better performance than many UDA methods. Second, comparing with source-free methods, our method has an evident superiority over class-wise IoU. For example, we gain by 9.3% on "sidewalk" class, 7.0% on "wall" class, 7.9% on "traffic light" class, etc. blue Despite the high performance on most classes, we found our method under-performs in some semantic classes, *e.g.*, "car", "truck" and "bus". We further investigate the top-1 wrongly predicted classes for these classes and observe that "car"/"truck"/"bus" is often mis-classified as "truck"/"car"/"truck". This indicates that our method tends to make mistakes when the given classes share similar appearances. In the qualitative evaluation (see Figure 5), our method shows clearly better segmentation results than prior state of the art [17], which is consistent with our quantitative observations.

b) SYNTHIA → Cityscapes Results: We further show the comparison on SYNTHIA → Cityscapes in Table III. Overall, our method again outperforms all other alternatives with larger margins. Concretely, we exceed the best competitor by **5.7%** on 13-class mIoU and **4.9%** on 16-class mIoU. More interestingly, the performance of our method is much closer to that of the methods leveraging source data for adaptation (only 1.6% gap in 13-class mIoU). Furthermore, our method wins on

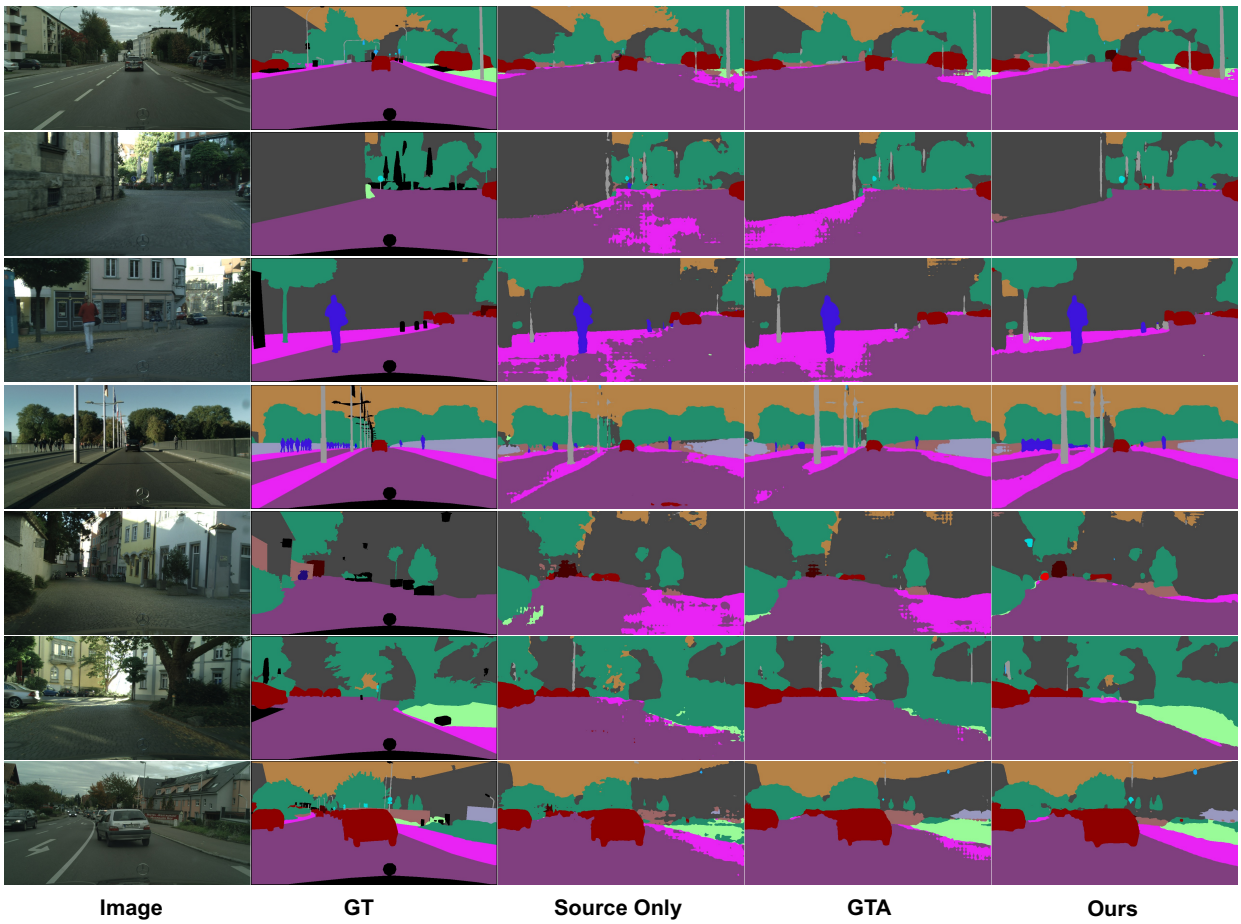


Fig. 5: Qualitative evaluation. From left to right: images, ground-truth labels, source only results, GTA [17] results and our results. Task: GTA5 → Cityscapes. Architecture: DeepLab-V2 with ResNet-101. Note that the black pixels in GT are void and ignored in loss computation.

most class-wise IoUs except four classes where performance is comparable.

D. Further Analysis

We conduct comprehensive further analysis to verify the contributions of each component in our proposed method. Note that we do all experiments on task GTA5 → Cityscapes with DeepLab-V2 using ResNet-101 backbone, and the reported mIoU is obtained by 3-round self-training under multi-scale evaluation.

a) Effect of Different Components: blackTable IV shows the quantitative results of Uncertainty-aware Online Teacher-Student Learning (UOTSL), Uncertainty-aware FeatureMix (UFM) and the combination of they two. Note that the baseline is the source only model and simple pseudo-label self-training system of GTA [17]. Improving on this with our framework, UOTSL alone boosts performance by 1.6% while UFM alone gains 2.8% improvement. The combined version, *i.e.*, our proposed method, outperforms baseline by 3.6%. This demonstrates the efficacy of our proposed components. We also show some qualitative results in Figure 6. Both our UOTSL and UFM show better segmentation results than source-only, and our full model performs best, which is consistent to the observations in the previous quantitative results.

b) Ablation Study of UFM: We compare our FeatureMix with vanilla ClassMix [22] and further explore the effect of Uncertainty-aware Sampling (US) of Eq. 8 and Uncertainty-aware Pseudo-labeling (UP) of Eq. 9 used in our UFM. The results can be seen in Table V. blackWe can see that FeatureMix achieves slightly better performance than ClassMix, although it runs with significantly less (149.68) GFLOPs. Further, the proposed US and UP each improves the vanilla FeatureMix separately and gives the best boost when used in combination.

c) Effect of Uncertainty-awareness: blackWe first compare the Uncertainty-aware Pseudo-labeling (UP) with the widely-used alternative hard thresholds in Table VI upon our Uncertainty-aware FeatureMix (UFM) component. From the results, we can see UP outperforms the ones using hard thresholds by 0.5%. Furthermore, we investigate UP's effect on the Uncertainty-aware Online Teacher-Student Learning (UOTSL) component. Interestingly, we can see UP also outperforms the one using the best threshold by 0.5% mIoU. More importantly, UP offers an automatic weighting scheme without the need of tuning thresholds for different tasks.

TABLE III: Quantitative evaluation on SYNTHIA → Cityscapes. Architecture: DeepLab-V2 with ResNet-101. SF: source-free setting. †: reproduced by us. ⋆ is the 13-class mIoU. ‡: excluded classes for 13-class mIoU. The best results are in **bold**.

Method	SF	road	sidewalk	building	wall‡	fence‡	pole‡	t-light	t-sign	vegetation	sky	person	rider	car	bus	motorcycle	bicycle	mIoU	mIoU⋆
STAR [11]	×	82.6	36.2	81.1	-	-	-	12.2	8.7	78.4	82.2	59.0	22.5	76.3	33.6	11.9	40.8	-	48.1
CAG [63]	×	84.8	41.7	85.5	-	-	-	13.7	23.0	86.5	78.1	66.3	28.1	81.8	21.8	22.9	49.0	-	52.6
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
CBST [26]	×	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	42.6	48.9
PyCDA [65]	×	75.5	30.9	83.3	20.8	0.7	32.7	27.3	33.5	84.7	85.0	64.1	25.4	85.0	45.2	21.2	32.0	46.7	53.3
TPLD [45]	×	80.9	44.3	82.2	19.9	0.3	40.6	20.5	30.1	77.2	80.9	60.6	25.5	84.8	41.1	24.7	43.7	47.3	53.5
USAMR [66]	×	83.1	38.2	81.7	9.3	1.0	35.1	30.3	19.9	82.0	80.1	62.8	21.1	84.4	37.8	24.5	53.3	46.5	53.8
RPL [34]	×	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	47.9	54.9
DACS [49]	×	80.6	25.1	81.9	21.5	2.9	37.2	22.7	24.0	83.7	90.8	67.6	38.3	82.9	38.9	28.5	47.6	48.3	54.8
IAST [48]	×	81.9	41.5	83.3	17.7	4.6	32.3	30.9	28.8	83.4	85.0	65.5	30.8	86.5	38.2	33.1	52.7	49.8	57.0
RPT [46]	×	89.1	47.3	84.6	14.5	0.4	39.4	39.9	30.3	86.1	86.3	60.8	25.7	88.7	49.0	28.4	57.5	51.7	59.5
CorDA [50]	×	93.3	61.6	85.3	19.6	5.1	37.8	36.6	42.8	84.9	90.4	69.7	41.8	85.6	38.4	32.6	53.9	55.0	62.8
ProDA [51]	×	87.8	45.7	84.6	37.1	0.6	44.0	54.6	37.0	88.1	84.4	74.2	24.3	88.2	51.1	40.5	45.6	55.5	62.0
URMA [32]	√	59.3	24.6	77.0	14.0	**1.8**	31.5	18.3	32.0	83.1	80.4	46.3	17.8	76.7	17.0	18.5	34.6	39.6	45.0
GTA [17]	√	89.0	44.6	80.1	7.8	0.7	34.4	22.0	22.9	82.0	86.5	**65.4**	33.2	**84.8**	**45.8**	38.4	31.7	48.1	55.5
GTA† [17]	√	85.7	42.0	80.2	12.7	0.0	34.1	24.1	25.8	81.6	85.7	63.1	30.0	76.7	39.5	36.6	49.8	48.0	55.5
Ours	√	**93.0**	**59.0**	**83.8**	**16.7**	1.2	**35.1**	**36.9**	**35.2**	**84.2**	**89.1**	64.9	**34.3**	82.3	38.8	**41.4**	**52.9**	**53.0**	**61.2**

TABLE IV: The effectiveness of different components. UFM: Uncertainty-aware FeatureMix. UOTSL: Uncertainty-aware Online Teacher-Student Learning. Task: GTA5 → Cityscapes. Archtecture: DeepLab-V2 with ResNet-101.

UFM	UOTSL	mIoU
		50.1
√		52.9
	√	51.7
√	√	**53.7**

TABLE V: Ablation study of Uncertainty-aware FeatureMix. US: Uncertainty-aware Sampling. UP: Uncertainty-aware Pseudo-labeling. Task: GTA5 → Cityscapes. Archtecture: DeepLab-V2 with ResNet-101.

ClassMix	FeatureMix	US	UP	mIoU
√				51.7
	√			51.8
	√	√		52.4
	√		√	52.2
	√	√	√	**52.9**

TABLE VI: Effect of Uncertainty-awareness upon Uncertainty-aware FeatureMix (UFM). UP: Pncertainty-aware Pseudo-labeling. Task: GTA5 → Cityscapes. Archtecture: DeepLab-V2 with ResNet-101.

FeatureMix	UP/Threshold	mIoU
√	$\tau = 0.90$	52.4
√	$\tau = 0.93$	52.4
√	$\tau = 0.95$	52.3
√	UP	**52.9**

TABLE VII: Effect of Uncertainty-awareness upon Uncertainty-aware Online Teacher-Student Learning (UOTSL). UP: Pncertainty-aware Pseudo-labeling. Task: GTA5 → Cityscapes. Archtecture: DeepLab-V2 with ResNet-101.

OTSL	UP/Threshold	mIoU
√	$\tau = 0.90$	51.0
√	$\tau = 0.93$	51.3
√	$\tau = 0.95$	51.2
√	UP	**51.8**

d) Effect of Monte Carlo Sampling Number: The MC sampling number influences the uncertainty estimation and the teacher prediction of Online Teacher-Student Learning (OTSL). We analyze the performance variation by varying the MC sampling number as shown in Table VIII. From the results, we can see more MC samples induce better model performance, which is expected.

e) blue BNN based vs. Other Uncertainty Estimate Methods: Dropout enables a simple way to estimate uncertainty in neural networks [32], [35] where uncertainty is estimated over predictions corresponding to different sampled dropout masks. It is worth noting that the uncertainty here is governed by the dropout rate and is usually not learned. In contrast, our variational inference framework learns the correct posterior distribution for each neuron end-to-end simply using our objective (Eq. 10). Table IX shows the results of our BNN and the conventional Monte-Carlo dropout. Our BNN outperforms MC dropout by a margin of **4.6%** and **4.4%** with 3 and 5 MC samples, respectively. This comparison shows the better value of end-to-end learning of a Bayesian neural network than a Dropout based approximation when accurate uncertainty estimation is needed, such as using it against pseudo-label noise.

blue We also compare Deep Ensembles [37], which leverages the model ensemble to obtain more reliable predictions. Specifically, to implement [37] in SFDASS, we manually set the seed for network initialization from 1 to 5. Following the training pipeline of the baseline method [17], 5 models are obtained after training. We then use these 5 models for deep ensemble [37], *i.e.*, the 5 predictions from these 5 models are averaged as the final prediction for evaluation. From Table IX, we found that despite the promising performance gain over the baseline method [17], *i.e.*, 51.2%/52.0% *vs.* 50.1, deep ensemble [37] still performs worse than our method.

f) Initialization of BNN Posterior: We tried two different types of initialization of our BNN posterior, including a diagonal (element-wise variance) and a spherical (shared single variance) Gaussian distribution. We can see that the

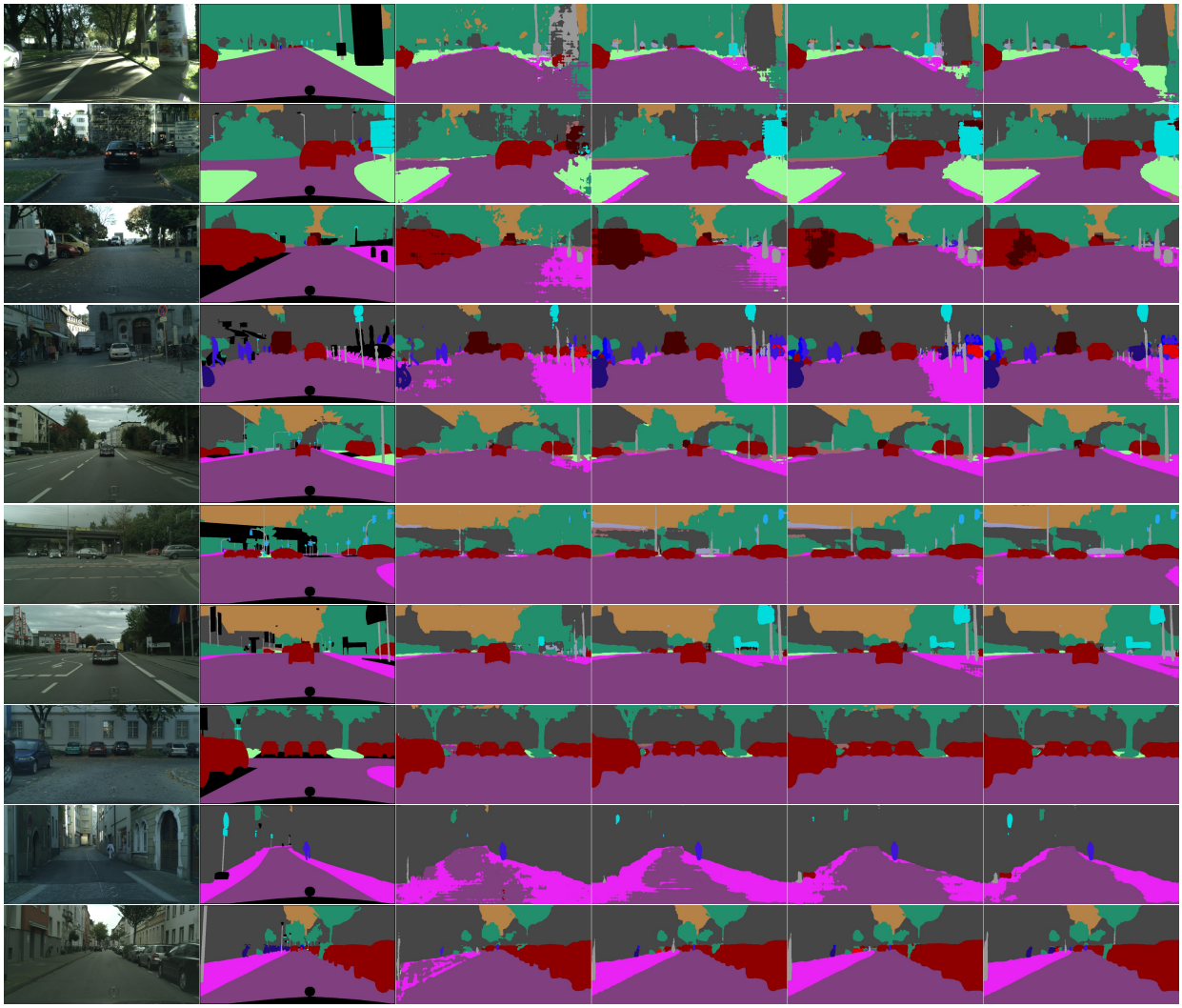


Fig. 6: Qualitative evaluation with different components. UOTSL: Uncertainty-aware Online Teacher-Student Learning. UFM: Uncertainty-aware FeatureMix. From left to right: images, ground-truth labels, source only results, UOTSL only results, UFM only results and UOTSL+UFM (our full model) results. Task: GTA5 $\rightarrow$ Cityscapes. Architecture: DeepLab-V2 with ResNet-101. Note that the black pixels in GT are void and ignored in loss computation.

TABLE VIII: The Effect of Monte Carlo Sampling Number. FM: FeatureMix. OTSL: Online Teacher-Student Learning. US: Uncertainty-aware Sampling. UP: Uncertainty-aware Pseudo-labeling. MC: Monte Carlo sampling number. Task: GTA5 $\rightarrow$ Cityscapes. Archtecture: DeepLab-V2 with ResNet-101.

FM	OTSL	US	UP/Threshold	MC	mIoU
√	√		$\tau = 0.90$	1	52.8
√	√	√	UP	3	53.1
√	√	√	UP	5	**53.7**

TABLE IX: blue Comparison between our BNN, Dropout based [35] uncertainty estimation (dropout rate 0.5) and Deep Ensembles [37]. Setting: GTA5 $\rightarrow$ Cityscapes with DeepLab-V2 and ResNet-101.

Method	# MC Samples / # Ensemble Models	mIoU
MC Dropout [35]	3	48.5
Deep Ensembles [37]	3	51.2
Our BNN	3	**53.1**
MC Dropout [35]	5	49.3
Deep Ensembles [37]	5	52.0
Our BNN	5	**53.7**

results are comparable in Table X, demonstrating our method is insensitive to the type of BNN posterior distributions. While a spherical Gaussian is preferred when there is a constraint about the total trainable parameters.

blue

g) The Effect of KL Regularization: Table XI shows the effect of KL regularization. We found that performance increases when the KL regularization is lowered. This is reasonable in SFDA as the target domain at deployment is

TABLE X: Diagonal vs. Spherical Gaussian Posteriors. Task: GTA5 $\rightarrow$ Cityscapes. Archtecture: DeepLab-V2 with ResNet-101.

Diagonal/Spherical	mIoU
Spherical	53.4
Diagonal	**53.7**

quite different from the source domain, forcing the target posterior close to the source prior model may harm the performance.

TABLE XI: blue The effect of KL regularization.

λ	1	1e-1	1e-2	1e-3	0
mIoU	50.1	51.0	52.2	52.9	**53.7**

blue

h) Evaluation on Other Architectures: In Table XII, we further evaluate our method on another architecture, *i.e.*, FCN8s [67] with the VGG-16 [68] backbone. To save space, we cluster 19 classes into four groups. That is, Background (BG): building, wall, fence, vegetation, terrain, sky; Minority Class (MC): rider, train, motorcycle, bicycle; Road Infrastructure Vertical (RIV): pole, traffic light, traffic sign; Road Infrastructure Ground (RIG): road, sidewalk; and Dynamic Stuff (DS): person, car, truck, bus. The results indicate that our method keeps the superiority on various architectures.

TABLE XII: blue Evaluation on the architecture – FCN8s with the VGG-16 backbone.

Method	BG	MC	RIV	RIG	DS	mIoU
SFDA [18]	51.6	7.8	15.9	58.6	43.7	35.8
GTA [17]	49.9	30.3	32.9	74.9	50.8	45.9
Ours	**52.1**	**32.2**	**39.8**	**78.2**	**53.7**	**49.1**

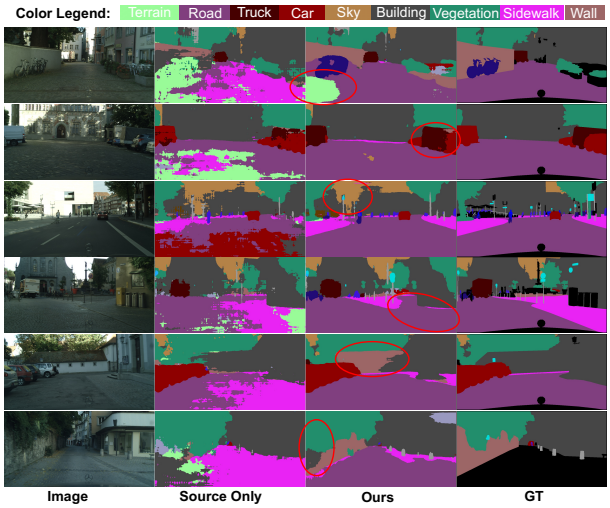


Fig. 7: blue Failure cases.

blue

i) Failure Case: Figure 7 shows some failure cases produced by our method. We found the failure often happens when two categories share a similar appearance. For example, part of the "road" is mis-segmented as "terrain" in the 1^{st} row, some cars are regarded as trucks in the 2^{nd} row, the white building similar to "sky" is mis-classified (row 3), "sidewalk" mis-classified to "road" (row 4), "building" mis-classified to "wall" (row 5) and "vegetation" on the wall mis-classified to "wall" (row 6). This observation can be used to further advance our method.

V. Conclusion

We proposed a Bayesian Neural Network (BNN) based approach to addressing pseudo-label noise in SFDA for semantic-segmentation. Our BNN's improved uncertainty estimation underpins two novel self-training components including Uncertainty-aware Online Teacher-Student Learning (UOTSL) and our simple but effective feature augmentation, Uncertainty-aware FeatureMix (UFM). These self-training objectives are effective on their own and together set a new state-of-the-art performance on two standard SFDA segmentation benchmarks. Further analysis shows the efficacy of each component.

References

[1] P. Arbeláez, B. Hariharan, C. Gu, S. Gupta, L. Bourdev, and J. Malik, "Semantic segmentation using regions and parts," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2012, pp. 3378–3385.

[2] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 12, pp. 2481–2495, 2017.

[3] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *Proceedings of the European Conference on Computer Vision*, 2014, pp. 740–755.

[4] V. M. Patel, R. Gopalan, R. Li, and R. Chellappa, "Visual domain adaptation: A survey of recent advances," *IEEE signal processing magazine*, vol. 32, no. 3, pp. 53–69, 2015.

[5] G. Csurka, T. M. Hospedales, M. Salzmann, and T. Tommasi, "Visual domain adaptation in the deep learning era," *Synthesis Lectures on Computer Vision*, vol. 11, no. 1, pp. 1–190, 2022.

[6] G. Csurka, R. Volpi, and B. Chidlovskii, "Unsupervised domain adaptation for semantic image segmentation: a comprehensive survey," *arXiv preprint arXiv:2112.03241*, 2021.

[7] G. Kang, Y. Wei, Y. Yang, Y. Zhuang, and A. Hauptmann, "Pixel-level cycle association: A new perspective for domain adaptive semantic segmentation," in *Advances in Neural Information Processing Systems*, vol. 33, 2020, pp. 3569–3580.

[8] J. Hoffman, E. Tzeng, T. Park, J.-Y. Zhu, P. Isola, K. Saenko, A. Efros, and T. Darrell, "Cycada: Cycle-consistent adversarial domain adaptation," in *International Conference on Machine Learning*, 2018, pp. 1989–1998.

[9] Z. Wu, X. Han, Y.-L. Lin, M. G. Uzunbas, T. Goldstein, S. N. Lim, and L. S. Davis, "Dcan: Dual channel-wise alignment networks for unsupervised scene adaptation," in *Proceedings of the European Conference on Computer Vision*, 2018, pp. 518–534.

[10] E. Tzeng, J. Hoffman, T. Darrell, and K. Saenko, "Simultaneous deep transfer across domains and tasks," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2015, pp. 4068–4076.

[11] Z. Lu, Y. Yang, X. Zhu, C. Liu, Y.-Z. Song, and T. Xiang, "Stochastic classifiers for unsupervised domain adaptation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 9111–9120.

[12] Y. Ganin and V. Lempitsky, "Unsupervised domain adaptation by backpropagation," in *International Conference on Machine Learning*, 2015, pp. 1180–1189.

[13] Z. Wu, X. Wang, J. E. Gonzalez, T. Goldstein, and L. S. Davis, "Ace: Adapting to changing environments for semantic segmentation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 2121–2130.

[14] 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 *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 2100–2110.

[15] Y. Luo, P. Liu, T. Guan, J. Yu, and Y. Yang, "Significance-aware information bottleneck for domain adaptive semantic segmentation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 6778–6787.

[16] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 2507–2516.

[17] J. N. Kundu, A. Kulkarni, A. Singh, V. Jampani, and R. V. Babu, "Generalize then adapt: Source-free domain adaptive semantic segmentation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 7046–7056.

[18] Y. Liu, W. Zhang, and J. Wang, "Source-free domain adaptation for semantic segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 1215–1224.

[19] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On calibration of modern neural networks," in *International Conference on Machine Learning*, 2017, pp. 1321–1330.

[20] A. Kendall and Y. Gal, "What uncertainties do we need in bayesian deep learning for computer vision?" *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[21] M. Abdar, F. Pourpanah, S. Hussain, D. Rezazadegan, L. Liu, M. Ghavamzadeh, P. Fieguth, X. Cao, A. Khosravi, U. R. Acharya *et al.*, "A review of uncertainty quantification in deep learning: Techniques, applications and challenges," *Information Fusion*, vol. 76, pp. 243–297, 2021.

[22] V. Olsson, W. Tranheden, J. Pinto, and L. Svensson, "Classmix: Segmentation-based data augmentation for semi-supervised learning," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2021, pp. 1369–1378.

[23] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7472–7481.

[24] Y.-H. Tsai, K. Sohn, S. Schulter, and M. Chandraker, "Domain adaptation for structured output via discriminative patch representations," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 1456–1465.

[25] Y. Zou, Z. Yu, X. Liu, B. Kumar, and J. Wang, "Confidence regularized self-training," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 5982–5991.

[26] Y. Zou, Z. Yu, B. Kumar, and J. Wang, "Unsupervised domain adaptation for semantic segmentation via class-balanced self-training," in *Proceedings of the European Conference on Computer Vision*, 2018, pp. 289–305.

[27] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," *Advances in Neural Information Processing Systems*, 2014.

[28] A. Tarvainen and H. Valpola, "Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results," in *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[29] J. Liang, D. Hu, and J. Feng, "Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation," in *International Conference on Machine Learning*, 2020, pp. 6028–6039.

[30] R. Li, Q. Jiao, W. Cao, H.-S. Wong, and S. Wu, "Model adaptation: Unsupervised domain adaptation without source data," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 9641–9650.

[31] M. Bateson, H. Kervadec, J. Dolz, H. Lombaert, and I. Ben Ayed, "Source-relaxed domain adaptation for image segmentation," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 2020, pp. 490–499.

[32] F. Fleuret *et al.*, "Uncertainty reduction for model adaptation in semantic segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 9613–9623.

[33] B. Lakshminarayanan, A. Pritzel, and C. Blundell, "Simple and scalable predictive uncertainty estimation using deep ensembles," in *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[34] Z. Zheng and Y. Yang, "Rectifying pseudo label learning via uncertainty estimation for domain adaptive semantic segmentation," *International Journal of Computer Vision*, vol. 129, no. 4, pp. 1106–1120, 2021.

[35] Y. Gal and Z. Ghahramani, "Dropout as a bayesian approximation: Representing model uncertainty in deep learning," in *International Conference on Machine Learning*, 2016, pp. 1050–1059.

[36] K. Ciosek, V. Fortuin, R. Tomioka, K. Hofmann, and R. Turner, "Conservative uncertainty estimation by fitting prior networks," in *International Conference on Learning Representations*, 2020.

[37] B. Lakshminarayanan, A. Pritzel, and C. Blundell, "Simple and scalable predictive uncertainty estimation using deep ensembles," *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[38] W. Ji, S. Yu, J. Wu, K. Ma, C. Bian, Q. Bi, J. Li, H. Liu, L. Cheng, and Y. Zheng, "Learning calibrated medical image segmentation via multi-rater agreement modeling," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 12 341–12 351.

[39] S. Yu, H.-Y. Zhou, K. Ma, C. Bian, C. Chu, H. Liu, and Y. Zheng, "Difficulty-aware glaucoma classification with multi-rater consensus modeling," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 2020, pp. 741–750.

[40] A. Foong, D. Burt, Y. Li, and R. Turner, "On the expressiveness of approximate inference in bayesian neural networks," *Advances in Neural Information Processing Systems*, vol. 33, pp. 15 897–15 908, 2020.

[41] A. Graves, "Practical variational inference for neural networks," in *Advances in Neural Information Processing Systems*, vol. 24, 2011.

[42] C. Zhang, J. Bütepage, H. Kjellström, and S. Mandt, "Advances in variational inference," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 8, pp. 2008–2026, 2019.

[43] J. Huang, S. Lu, D. Guan, and X. Zhang, "Contextual-relation consistent domain adaptation for semantic segmentation," in *Proceedings of the European Conference on Computer Vision*, 2020, pp. 705–722.

[44] F. Lv, T. Liang, X. Chen, and G. Lin, "Cross-domain semantic segmentation via domain-invariant interactive relation transfer," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 4334–4343.

[45] I. Shin, S. Woo, F. Pan, and I. S. Kweon, "Two-phase pseudo label densification for self-training based domain adaptation," in *Proceedings of the European Conference on Computer Vision*, 2020, pp. 532–548.

[46] Y. Zhang, Z. Qiu, T. Yao, C.-W. Ngo, D. Liu, and T. Mei, "Transferring and regularizing prediction for semantic segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 9621–9630.

[47] H. Wang, T. Shen, W. Zhang, L.-Y. Duan, and T. Mei, "Classes matter: A fine-grained adversarial approach to cross-domain semantic segmentation," in *Proceedings of the European Conference on Computer Vision*, 2020, pp. 642–659.

[48] K. Mei, C. Zhu, J. Zou, and S. Zhang, "Instance adaptive self-training for unsupervised domain adaptation," in *Proceedings of the European Conference on Computer Vision*, 2020, pp. 415–430.

[49] W. Tranheden, V. Olsson, J. Pinto, and L. Svensson, "Dacs: Domain adaptation via cross-domain mixed sampling," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2021, pp. 1379–1389.

[50] Q. Wang, D. Dai, L. Hoyer, L. Van Gool, and O. Fink, "Domain adaptive semantic segmentation with self-supervised depth estimation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 8515–8525.

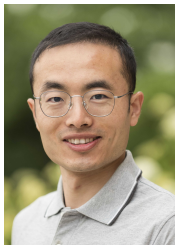
[51] P. Zhang, B. Zhang, T. Zhang, D. Chen, Y. Wang, and F. Wen, "Prototypical pseudo label denoising and target structure learning for domain adaptive semantic segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 12 414–12 424.

[52] S. R. Richter, V. Vineet, S. Roth, and V. Koltun, "Playing for data: Ground truth from computer games," in *Proceedings of the European Conference on Computer Vision*, 2016, pp. 102–118.

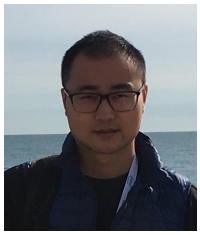
[53] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2016, pp. 3213–3223.

[54] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2016, pp. 3234–3243.

[55] Y. Yang and S. Soatto, "Fda: Fourier domain adaptation for semantic segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 4085–4095.
[56] 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.
[57] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.
[58] Y. Li, L. Yuan, and N. Vasconcelos, "Bidirectional learning for domain adaptation of semantic segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 6936–6945.
[59] P. T. Jackson, A. A. Abarghouei, S. Bonner, T. P. Breckon, and B. Obara, "Style augmentation: data augmentation via style randomization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, vol. 6, 2019, pp. 10–11.
[60] X. Huang and S. Belongie, "Arbitrary style transfer in real-time with adaptive instance normalization," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2017, pp. 1501–1510.
[61] C. Michaelis, B. Mitzkus, R. Geirhos, E. Rusak, O. Bringmann, A. S. Ecker, M. Bethge, and W. Brendel, "Benchmarking robustness in object detection: Autonomous driving when winter is coming," *arXiv preprint arXiv:1907.07484*, 2019.
[62] A. B. Jung, K. Wada, J. Crall, S. Tanaka, J. Graving, S. Yadav, J. Banerjee, G. Vecsei, A. Kraft, J. Borovec *et al.*, "imgaug," *GitHub project*, 2020.
[63] Q. Zhang, J. Zhang, W. Liu, and D. Tao, "Category anchor-guided unsupervised domain adaptation for semantic segmentation," in *Advances in Neural Information Processing Systems*, vol. 32, 2019.
[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 *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 07, 2020, pp. 12 613–12 620.
[65] Q. Lian, F. Lv, L. Duan, and B. Gong, "Constructing self-motivated pyramid curriculums for cross-domain semantic segmentation: A non-adversarial approach," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 6758–6767.
[66] Z. Zheng and Y. Yang, "Unsupervised scene adaptation with memory regularization in vivo," in *International Joint Conferences on Artificial Intelligence*, 2019.
[67] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3431–3440.
[68] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *International Conference on Learning Representations*, 2015.

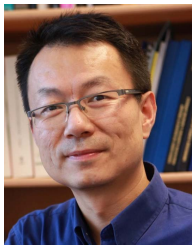


Zhihe Lu is currently pursuing the Ph.D. degree at Centre for Vision, Speech and Signal Processing (CVSSP), University of Surrey. He received his master degree at Chinese Academy of Sciences, Institute of Automation (CASIA) in 2019. His research interest centers around deep learning with limited annotated data. Recently, he focuses on two applications including domain adaptation and few-shot learning. He serves as regular reviewers for flagship conferences and journals, *e.g.*, CVPR, AAAI, IJCV, TIP and TNNLS.



Da Li is a Senior Research Scientist within the Machine Learning and Data Intelligence group in Samsung AI Centre Cambridge and a Visiting Researcher of the Machine Intelligence Research group at the University of Edinburgh. His research interests span transfer learning, meta-learning and semi-supervised learning. He serves as regular Reviewers for flagship venues, *e.g.*, CVPR, ICML, NeurIPS and journals, *e.g.*, TPAMI, JMLR, ML, and has served as a Senior Program Committee (Area Chair) of AAAI 2022.

Yi-Zhe Song (Senior Member, IEEE) received the bachelor's degree (Hons.) from the University of Bath in 2003, the M.Sc. degree from the University of Cambridge in 2004, and the Ph.D. degree in computer vision and machine learning from the University of Bath in 2008. He is currently a Professor of computer vision and machine learning and the Director of the SketchX Laboratory, Centre for Vision Speech and Signal Processing (CVSSP), University of Surrey. Previously, he was a Senior Lecturer at the Queen Mary University of London and a Research and Teaching Fellow at the University of Bath. He is a fellow of the Higher Education Academy, as well as a full member of the EPSRC Review College, the UK's main agency for funding research in engineering and the physical sciences. He received the Best Dissertation Award for his M.Sc. degree. He is the Program Chair of the British Machine Vision Conference (BMVC) in 2021 and regularly serves as the Area Chair (AC) for flagship computer vision and machine learning conferences, most recently at ICCV'21 and BMVC'20.



Tao Xiang received the Ph.D. degree in electrical and computer engineering from the National University of Singapore in 2001. He is currently a Professor with the Department of Electrical and Electronic Engineering, University of Surrey, and a Research Scientist at Facebook AI. He has published over 200 papers in international journals and conferences. His research interests include computer vision, machine learning, and data mining.

Timothy M. Hospedales (Senior Member, IEEE) is a Professor with the School of Informatics, The University of Edinburgh. He is also a Principal Scientist and Program Director for Machine Learning and Data Intelligence at Samsung AI Centre Cambridge. His research interests include machine learning and computer vision. He is an Associate Editor of the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE (TPAMI). He serves as an Area Chair of several major events including ICCV, CVPR, ECCV, AAAI, and ACL and a Program Chair of BMVC 2018.