C-SFDA: A Curriculum Learning Aided Self-Training Framework for Efficient Source Free Domain Adaptation

(Supplementary Material)

1. Overview

In Section 2, we provide more details about the notations used in the main paper. We describe the detailed training algorithms for image classification and semantic segmentation in Section 3. We evaluate the impact of misleading/noisy neighbors on a clustering-based source-free domain adaptation (SFDA) method in Section 4. Section 5 discusses more on memory overhead. In Section 6, we present more training details. We conduct a qualitative evaluation of semantic segmentation results in Section 7. We provide comparisons with more baselines in Section 8.

2. *N*otation

Notation Table: As shown in Table. 1, we provide symbols and brief descriptions of the notations used in our work. We have categorized the notations into 4 parts: *a) Data, b) Networks, c) Outputs/Thresholds, d) General*.

3. Proposed Algorithm

We describe the training details for Image Classification in Algorithm 1. For Semantic Segmentation, we summarize the training details in Algorithm 2. Both of these algorithms follow a similar training pipeline with minor changes. In selective pseudo-labeling, we have used *Difference of Confidence (DoC)* which was not clearly discussed in the main paper. We have also used class balancing which was not clearly discussed. We discuss them below.

Difference of Confidence (DoC): After separating the input batch into $\mathbb{D}_R$ and $\mathbb{D}_U$, we take difference of top-2 confidence scores of model prediction after sorting,

$$\hat{h}_t = \frac{1}{L}\sum_{l=1}^{l=L} \hat{h}_t^l = \frac{1}{L}\sum_{l=1}^{l=L} f_{\hat{\theta}_t}(\hat{x}_t^l)) \quad (1)$$

$$\hat{h}_t = Sort\{\hat{h}_t\} \quad (2)$$

$$q = DoC(\hat{h}_t) = \hat{h}_t[0] - \hat{h}_t[1] \quad (3)$$

Class Balancing, λ_k: We apply class balancing only for the cross-entropy loss. The process of estimating λ_k is given in Algorithm. 1.

Table 1. **Notation Table**

	Symbol	Description
Data	$\mathcal{D}_s$	Source dataset
	$\mathcal{D}_t$	Unlabeled target dataset
	$\mathbb{D}_R$	Reliable Sample Set
	$\mathbb{D}_U$	Unreliable Sample Set
Networks	f	DNN Model
	f_{θ_t}	Student Model
	$f_{\hat{\theta}_t}$	Teacher Model
	H	Contrastive output head
	C	Classifier
	G	CNN Backbone
Outputs/ Thresholds	(x_s, y_s)	Labeled source sample
	$\mathcal{T}_l(\cdot)$	l^{th} augmentation
	x_t	Unlabeled target sample
	$(x_t, \hat{y}_t)$	Pseudo-labeled target sample
	$\hat{h}_t$	Model Output Probabilities
	g_u	Uncertainty Measure
	d_j	Difficulty Score of j^{th} Batch
	r^i	Reliability of i^{th} sample
	τ_c	Confidence Threshold
	τ_u	Uncertainty Threshold
	τ_d	DoC Threshold
General	K	Number of Available Class
	L	Number of Augmentations
	B	Batch Size
	T	Total Number of Iterations
	γ	EMA Update Coefficient
	μ_r	Labelled Loss Coefficient
	μ_c	Contrastive Loss Coefficient
	μ_e	Entropy Loss Coefficient
	α	Labelled Loss Constant
	β	Contrastive Loss Constant
	P	Percentile Threshold for Conf. and Unc.
	H	Height of an image
	W	Width of an image
	λ_k	Loss Re-Weighting Coefficients

Algorithm 1 TRAINING DETAILS FOR IMAGE CLASSIFICATION

Input: Trained Source Model f_{θ_s}, and unlabeled target dataset $\mathcal{D}_t$

$\triangleright$ *Let* $[\cdot]$ *denote the indexing operation,* $\cdot||\cdot$ *denote the append operation,* $|\cdot|$ *denote the cardinality,* $std(.)$ *denote standard deviation, and* K *be the number of classes.*

Initialize: Student Model, $f_{\theta_t} = f_{\theta_s}$ and Teacher Model, $f_{\hat{\theta}_t} = f_{\theta_s}$
for $iter < MaxIter$ **do**:
    $\mathcal{X}_t \leftarrow$ batch sampled from $\mathbb{D}_t$
    Step 1: ***Confidence and Uncertainty***
    $X_b, Y_b, W, V \leftarrow \{\}, \{\}, \{\}, \{\}$ $\triangleright$ Empty ordered lists
    for x in $\mathcal{X}_t$ **do**:
        $\hat{w}_L, \hat{h}_L \leftarrow \{\}, \{\}$ $\triangleright$ Empty ordered list
        for l in L **do**:
            $x_l = \mathcal{T}_l(x)$ $\triangleright$ $\mathcal{T}_l$ is the l^{th} augmentation
            $h = f_{\hat{\theta}_t}(x_l)$
            $w \leftarrow \max_{k \in K} h[k]$
            $\hat{w}_L \leftarrow \hat{w}_L || w$
            $\hat{h}_L \leftarrow \hat{h}_L || h$ $\triangleright$ Class Predictions
        end for
        $h = \frac{1}{L} \sum_{l=1}^{l=L} \hat{h}_L$ $\triangleright$ Augmented Average Prediction
        $y_b = \arg\max_{k \in K} h[k]$ $\triangleright$ Predicted class
        $w = \frac{1}{L} \sum_{l=1}^{l=L} \hat{w}_L$ $\triangleright$ Predicted class probabilities
        $v = std(\hat{w}_L)$ $\triangleright$ Prediction Uncertainty
        $q = DoC(\hat{w}_L)$ $\triangleright$ Difference of Top-2 Confidence Scores
        $W, V, Q, Y_b, X_b \leftarrow W \,||\, w, U \,||\, v, Q \,||\, q, Y_b \,||\, y_b, X_b \,||\, x$
    end for
    Step 2: ***Calculate Thresholds***
    $\tau_c \leftarrow \frac{1}{B} \sum_{i=1}^{i=B} W$ $\triangleright$ Confidence threshold
    $\tau_u \leftarrow \frac{1}{B} \sum_{i=1}^{i=B} V$ $\triangleright$ Uncertainty threshold
    Step 3: ***Selective Pseudo-labelling***
    $\mathbb{D}_R, \mathbb{D}_U, Q_U \leftarrow \{\}, \{\}, \{\}$ $\triangleright$ Empty ordered lists
    for y_b, w, v, q, x_b in Y_b, W, V, Q, X_b **do**:
        Following eqn.(4) of main paper, calculate r^b
        $\mathbb{D}_R \leftarrow \mathbb{D}_R \,||\, (x_b, y_b)$; if $r^b = 1$
        $\mathbb{D}_U \leftarrow \mathbb{D}_U \,||\, (x_b, y_b)$; if $r^b = 0$
        $Q_U \leftarrow Q_U \,||\, q$; if $r^b = 0$
    end for
    $\tau_d \leftarrow \frac{1}{|Q_U|} \sum_{i=1}^{i=|Q_U|} Q_U$ $\triangleright$ DoC threshold
    for $(x_b, y_b), q$ in $\mathbb{D}_U, Q_U$ **do**:
        $\mathbb{D}_R \leftarrow \mathbb{D}_R \,||\, (x_b, y_b)$; if $q > \tau_d$
    end for
    Step 4: ***Calculate Losses and Update the Model***
    Compute loss weights for $\mathcal{L}_{ce}^R$, $\lambda_k = \frac{1}{n_k}$; $\forall$ k $\in \{1, \ldots, K\}$ $\triangleright$ n_k = Number of samples in $\mathbb{D}_R$ with label k
    Compute $\mathcal{L}_{ce}^R, \mathcal{L}_{\mathcal{P}}, \mathcal{L}_{\mathcal{C}}$ using $\mathbb{D}_R, \mathbb{D}_U$.
    Update θ_t by minimizing $\mathcal{L}_{tot}$ (in eqn.(9) of main paper) using SGD optimizer
    Update $\hat{\theta}_t$ using eqn.(3) of main paper
    Update loss coefficients μ_r, μ_c
end for

Output: Updated θ_t

Algorithm 2 TRAINING DETAILS FOR SEMANTIC SEGMENTATION

Input: Trained Source Model f_{θ_s}, and unlabeled target dataset $\mathcal{D}_t$

$\triangleright$ *Let* $[\cdot]$ *denote the indexing operation,* $\cdot||\cdot$ *denote the append operation,* $|\cdot|$ *denote the cardinality,* $std(.)$ *denote standard deviation, and* K *be the number of classes.*

Initialize: Student Model, $f_{\theta_t} = f_{\theta_s}$ and Teacher Model, $f_{\hat{\theta}_t} = f_{\theta_s}$
for $iter < MaxIter$ **do**:
  $\mathcal{X}_t \leftarrow$ batch sampled from $\mathbb{D}_t$
  Step 1: ***Confidence and Uncertainty Calculations***
  $X_b, Y_b, W, V \leftarrow \{\}, \{\}, \{\}, \{\}$ $\triangleright$ Empty ordered lists
  for x in $\mathcal{X}_t$ **do**:
    $\hat{h} = f(x)$
    $y_b \leftarrow \arg\max_{k \in K} \hat{h}[k]$ $\triangleright$ Class predictions
    $w \leftarrow \max_{k \in K} \hat{h}[k]$ $\triangleright$ Predicted class probabilities
    $\hat{w}_L \leftarrow \{\}$ $\triangleright$ Empty ordered list
    for l in L **do**:
      $w_l \leftarrow \max_{k \in K} f(x_l)[k]$
      $\hat{w}_L \leftarrow \hat{w}_L || w_l$
    end for
    $v = std(\hat{w}_L)$ $\triangleright$ Prediction Uncertainty
    $W, V, Y_b, X_b \leftarrow W \,||\, w, V \,||\, v, Y_b \,||\, y_b, X_b \,||\, x$
  end for
  Step 2: ***Calculate Thresholds***
  $\tau_c, \tau_u \leftarrow \{\}, \{\}$ $\triangleright$ List of class-wise confidence thresholds
  for k in range(K) **do**:
    $p_c \leftarrow W[Y_b == k]$ $\triangleright$ Store all prediction probabilities of class k in p_c
    $p_v \leftarrow V[Y_b == k]$ $\triangleright$ Store all prediction uncertainties of class k in p_u
    $p_c, p_v \leftarrow$ sort(p_c), sort(p_v)
    $\tau_c \leftarrow \tau_c \,||\, p_c[0.55|p_c|]$ $\triangleright$ Set threshold at top most 45% confident predictions
    $\tau_u \leftarrow \tau_u \,||\, p_v[0.55|p_v|]$ $\triangleright$ Set threshold at top most 45% uncertain predictions
  end for
  Step 3: ***Selective Pseudo-Labelling***
  $\hat{\mathcal{B}}_t \leftarrow \{\}$ $\triangleright$ Empty ordered list
  for y_b, w, v, x_b in Y_b, W, V, X_b **do**:
    for k in range(K) **do**:
      $y_b[(w < \tau_c[k]) \& (v > \tau_u[k]) \& (y_b == k)] \leftarrow K + 1$ $\triangleright$ Assign class-id, $K + 1$ representing '*unknown*'
    end for
    $\hat{\mathcal{B}}_t \leftarrow \hat{\mathcal{B}}_t \,||\, (x_b, y_b)$
  end for
  Step 4: ***Conditional Update***
  if $mean(W) > \tau_{thr}$ **then:**
    $x_t, y_t \leftarrow \hat{\mathcal{B}}_t$
    $\hat{h} \leftarrow f(x_t)$
    Compute $\mathcal{L}_{ce}^R \leftarrow$ CE$(\hat{h}, y_t)$
    Compute $\mathcal{L}_E$ using eq.(12) of main paper
    Update trainable parameters of θ_t by minimizing $\mathcal{L}_{tot}$ (in eq.(13)) using SGD optimizer
    Update $\hat{\theta}_t$ using eqn.(3) of main paper
    Update loss coefficient, μ_e
  end if
end for

Output: Updated θ_t

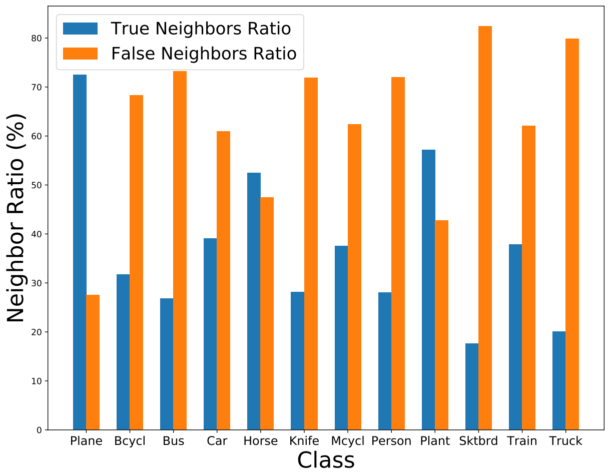

Figure 1. Quality evaluation of clustering-based neighbor pseudo-labels on VisDA dataset. We use k (=3) similar neighbors of central data instances and evaluate their quality. Here, true neighbors have the same pseudo-label as the central data instance (x_c) ground truth label. In such a scenario, using cluster knowledge for pseudo-label refinement may lead to severely noisy labels. Broadly speaking, for label refinement of x_c, if we take the neighbor pseudo-labels it inevitably leads to the wrong label for x_c.

Both DoC and λ_k are only used for image classification.

4. Noisy Neighbors in Cluster-based Methods

For analyzing the noisy neighbors in VisDA dataset, we show the noisy neighbor scenario for cluster-based SFDA method in Figure 1. We take SHOT [15] as the clustering-based method and consider the pseudo-labels generated on the first iteration. As there is a huge number of false neighbors, the labels collected from these neighbors lead to noisy labels. To identify if a neighbor is true or false, we compare pseudo-labels of neighbors with the central instance's ground truth. If they are the same, we call them true neighbors and vice versa. The true neighbor ratio is the percentage of true neighbors among all pseudo-labels.

5. Memory Overhead

The performance of SOTA SFDA methods (e.g., [42, 4]) relies heavily on using a large-size memory bank for label refinement. [42] requires global memory banks or processing the entire dataset before the adaptation. [4] requires more than 4% of the dataset size as a memory queue to perform reasonably, with the best performance reported with the entire dataset size (e.g. 55K for VisDA-C) as the queue. These methods are likely to face significant scale-up issues in online adaptation or in scenarios requiring adaptation to *millions of target samples*. Our method eliminates the requirement of memory banks completely making it scalable while producing better performance.

Table 2. Effect of Batch Size on the Performance

Batch Size	64	128	256
VisDA-C	87.8	87.8	87.7
DomainNet	69.0	69.0	69.0

Table 3. **List of Augmentations** used for both Image Classification and Semantic Segmentation (Sem. Seg.).

Task	Augmentation
Image Classification	RandomResizedCrop(224, scale=(0.2, 1.0)) ColorJitter(0.8, 0.8, 0.5, 0.2) RandomGrayscale(p=0.2) RandomRotation(degrees = [-2,2]) RandomPosterize(8, p=0.2) RandomEqualize(p=0.2) GaussianBlur([0.1, 2]), RandomHorizontalFlip(), ToTensor() Normalize(mean=[0.485, 0.456, 0.406], std=[0.229, 0.224, 0.225])
Sem. Seg.	ColorJitter(0.6, 0.6, 0.6, 0.15) RandomGrayscale(p=0.5) GaussianBlur([0.1, 2]) Normalize(mean = [104.00699, 116.66877, 122.67892])

We compare the performance of C-SFDA under different batch sizes on DomainNet and VisDA-C (Table 2). We find C-SFDA performs consistently/comparably with different batch sizes. It is evident from the experiment that large batch sizes are not required for the success of C-SFDA.

6. Training Details

6.1. Source Model Training

For the image classification task, we create source models for 4 different datasets. For Office-31 and Office-Home, we follow [15] to train the model for 50 and 100 epochs with a learning rate of $1e-3$ and weight decay of $1e-3$. We use a learning rate of $1e-2$ for the bottleneck layer (=256 dim) and task-specific FC layers. We consider the bottleneck layer as the contrastive head (H). For VisDA and DomainNet, we train the model for 10 and 60 epochs.

For GTA5 and SYNTHIA, we train on the source domain for 35 epochs (GTA5) and 15 epochs (SYNTHIA), making use of Gaussian blur and random flip augmentations. We use a batch size of 16, and a learning rate of 1×10^{-4} for GTA5 and 2×10^{-5} for SYNTHIA, with weight decay of 5×10^{-4}. For source augmentations, we use snow and frost augmentations with uniformly sampled severity between 1 and 3 (maximum severity possible in [9] is 5). We follow [28] for Cityscapes training and apply random cropping of size 512×512 on the scale between 0.5 and 1.

This increases the training data size and produces a validation mIoU of 66.37 with DeepLabV2 architecture. We use SGD optimizer with a momentum of 0.9, a learning rate of $2.5e-4$, and a weight decay of $5e-4$ and consider poly learning rate policy with a power of 0.9. We use 2 NVIDIA A40 GPUs for all the training.

6.2. Target Domain Training

In table 3, we list the augmentations used during the target domain training. For all augmentations, we use PyTorch default implementations. For the baseline models, we follow their GitHub implementations [1,2]. However, we directly report most of the baseline results for Office-31, Office-Home, and VisDA-C from SFDA-DE [2]. For DomainNet, we follow AdaCon [1]. We only consider AdaCon for online adaptation since it is the previous state-of-the-art benchmark.

For Semantic Segmentations, we follow HCL [6] [3] implementations. We also report the baseline method results from HCL [6]. For online adaptations, we follow AUGCO [22] to report the baseline results. Note that, we find ourselves in a bit of a conundrum in comparing against the state-of-the-art works in SFDA semantic segmentation. Since different works consider different training environments and a number of add-ons, it is hard to find suitable techniques that match the adaptation scenarios we consider here. Moreover, SFDA gained wide interest very recently from researchers and continues to be a very challenging task. Therefore, the number of baseline methods for semantic segmentation is limited. *In Cityscapes→ Dark Zurich adaptation, we only report online adaptation results, as offline methods are very rarely reported in prior works. To the best of our ability, we could not find any SFDA technique for this task where training has been done in an offline fashion. When we run C-SFDA in an offline manner, we obtain a mIoU of 35.1 after training for 10K iterations.*

7. Qualitative Evaluation

In Figure 2, we show some qualitative results for GTA5→Cityscapes adaptation. The first two columns show several validation images with their ground truth segmentation maps. For the baseline comparison, we choose state-of-the-art for semantic segmentation, HCL [6]. As evident from Figure 2, Our proposed method performs significantly better in detecting the edges and reducing noisy predictions compared to HCL. Here, we choose a few crowded scenes for comparison to show the effectiveness of C-SFDA in challenging scenarios.

[1] https://github.com/DianCh/AdaContrast
[2] https://github.com/tim-learn/SHOT
[3] https://github.com/jxhuang0508/HCL/tree/225b791e08cfa976885f6b7386b0e53674a28035

8. More Comparison

For comparison, we consider a number of baselines that work with or without source data. SFAN [30], STAR [17] RWOT [31], SE [3] are among the source-dependent UDA techniques. For source-free settings, we consider SFDA [10], 3C-GAN [12], SHOT [15], A^{2}Net [29], G-SFDA [32], SFDA-DE [2], AdaCon [1]. For Segmentation, we consider SOTA SFDA techniques such as UR [23], SFDA [16], HCL [6]. For online semantic segmentation benchmarks, we consider Test Time BN [20], TENT [27], AUGCO [22]. However, due to the page limit, we put more comparisons with other relevant baselines here in Table 4-7.

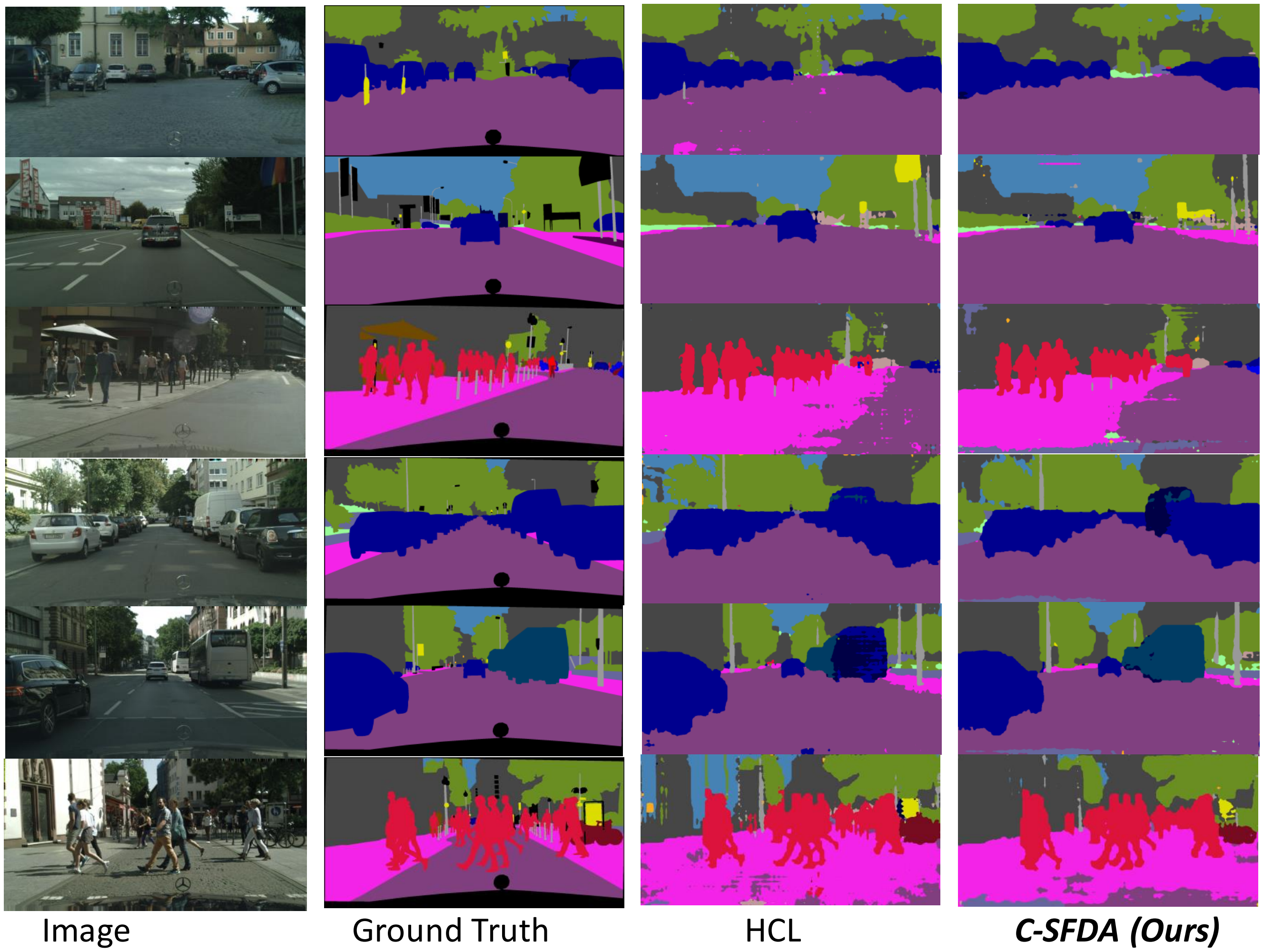

Figure 2. Qualitative Evaluation of GTA5→Cityscapes source-free domain adaptation for semantic segmentation. Compared to the state-of-the-art method HCL [6], we observe that the proposed C-SFDA performs better at edge classification. We also encouragingly find our method performing significantly better at distinguishing between building and sky (whereas the baseline HCL struggles due to the similar colors and positions of the sky and building class pixels). Comparing our results with the ground truth, we find the proposed C-SFDA to perform satisfactorily in most cases.

Table 4. Classification performance (%) under UDA and SFDA settings on **Office-Home** dataset (ResNet50 backbone). We report Top-1 accuracy on 12 domain shifts (→) and take the average (Avg.) over them. Our method achieves SOTA performance on 8 of these shifts.

Method	SF	Ar→Cl	Ar→Pr	Ar→Rw	Cl→Ar	Cl→Pr	Cl→Rw	Pr→Ar	Pr→Cl	Pr→Rw	Rw→Ar	Rw→Cl	Rw→Pr	Avg.
GSDA [5]	×	61.3	76.1	79.4	65.4	73.3	74.3	65.0	53.2	80.0	72.2	60.6	83.1	70.3
RSDA [4]	×	53.2	77.7	81.3	66.4	74.0	76.5	67.9	53.0	82.0	75.8	57.8	85.4	70.9
TSA [14]	×	57.6	75.8	80.7	64.3	76.3	75.1	66.7	55.7	81.2	75.7	61.9	83.8	71.2
SRDC [24]	×	52.3	76.3	81.0	69.5	76.2	78.0	68.7	53.8	81.7	76.3	57.1	85.0	71.3
FixBi [19]	×	58.1	77.3	80.4	67.7	79.5	78.1	65.8	57.9	81.7	76.4	62.9	86.7	72.7
SFDA [10]	✓	48.4	73.4	76.9	64.3	69.8	71.7	62.7	45.3	76.6	69.8	50.5	79.0	65.7
G-SFDA [32]	✓	57.9	78.6	81.0	66.7	77.2	77.2	65.6	56.0	82.2	72.0	57.8	83.4	71.3
SHOT [15]	✓	57.1	78.1	81.5	68.0	78.2	78.1	67.4	54.9	82.2	73.3	58.8	84.3	71.8
A^{2}Net [29]	✓	58.4	79.0	82.4	67.5	79.3	78.9	**68.0**	56.2	82.9	**74.1**	60.5	85.0	72.8
SFDA-DE [2]	✓	59.7	79.5	82.4	**69.7**	78.6	**79.2**	66.1	57.2	82.6	73.9	60.8	85.5	72.9
C-SFDA (Ours)	✓	**60.3**	**80.2**	**82.9**	69.3	**80.1**	78.8	67.3	**58.1**	**83.4**	73.6	**61.3**	**86.3**	**73.5**

References

[1] Dian Chen, Dequan Wang, Trevor Darrell, and Sayna Ebrahimi. Contrastive test-time adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 295–305, 2022. 5, 7

[2] Ning Ding, Yixing Xu, Yehui Tang, Chao Xu, Yunhe Wang, and Dacheng Tao. Source-free domain adaptation via dis-

Table 5. Source-free (SF) domain adaptation performance on **VisDA** dataset (ResNet-101 backbone) shown by per-class accuracy (%) and their average (Avg.). Our method improves the average accuracy by 1% compared to the previous SOTA, Adacon [1]. C-SFDA also achieves a significant performance gain (3.5% in Avg.) for online test-time domain adaptation settings.

Method	SF	plane	bike	bus	car	horse	knife	mcycle	person	plant	sktbrd	train	truck	Avg.
SFAN [30]	×	93.6	61.3	84.1	70.6	94.1	79.0	91.8	79.6	89.9	55.6	89.0	24.4	76.1
SWD [11]	×	90.8	82.5	81.7	70.5	91.7	69.5	86.3	77.5	87.4	63.6	85.6	29.2	76.4
MCC [8]	×	88.7	80.3	80.5	71.5	90.1	93.2	85.0	71.6	89.4	73.8	85.0	36.9	78.8
STAR [17]	×	95.0	84.0	84.6	73.0	91.6	91.8	85.9	78.4	94.4	84.7	87.0	42.2	82.7
RWOT [31]	×	95.1	80.3	83.7	90.0	92.4	68.0	92.5	82.2	87.9	78.4	90.4	68.2	84.0
SE [3]	×	95.9	87.4	85.2	58.6	96.2	95.7	90.6	80.0	94.8	90.8	88.4	47.9	84.3
Source only	-	57.2	11.1	42.4	66.9	55.0	4.4	81.1	27.3	57.9	29.4	86.7	5.8	43.8
3C-GAN [12]	✓	94.8	73.4	68.8	74.8	93.1	95.4	88.6	84.7	89.1	84.7	83.5	48.1	81.6
SHOT [15]	✓	94.3	88.5	80.1	57.3	93.1	94.9	80.7	80.3	91.5	89.1	86.3	58.2	82.9
A^2Net [29]	✓	94.0	87.8	85.6	66.8	93.7	95.1	85.8	81.2	91.6	88.2	86.5	56.0	84.3
G-SFDA [32]	✓	96.1	88.3	85.5	**74.1**	97.1	95.4	89.5	79.4	95.4	92.9	89.1	42.6	85.4
SFDA-DE [2]	✓	95.3	**91.2**	77.5	72.1	95.7	**97.8**	85.5	**86.1**	**95.5**	**93.0**	86.3	61.6	86.5
AdaCon [1]	✓	97.0	84.7	84.0	77.3	96.7	93.8	91.9	84.8	94.3	93.1	**94.1**	49.7	86.8
C-SFDA (Ours)	✓	**97.6**	88.8	**86.1**	72.2	**97.2**	94.4	**92.1**	84.7	93.0	90.7	93.1	**63.5**	**87.8**
AdaCon [1] (Online)	✓	95.0	68.0	82.7	**69.6**	94.3	80.8	90.3	79.6	**90.6**	69.7	**87.6**	36.0	78.7
C-SFDA (Online)	✓	**95.9**	**75.6**	**88.4**	68.1	**95.4**	**86.1**	**94.5**	**82.0**	89.2	**81.4**	87.3	**43.8**	**82.2**

Table 6. Performance evaluation on **GTA5→Cityscapes** (DeepLabV2 with ResNet101) where we report mean IoU (mIoU) over 19 categories on Cityscapes validations set. Our method achieves the best mIoU in SFDA and online test-time adaptation.

Method	SF	Road	SW	Build	Wall	Fence	Pole	TL	TS	Veg.	Terrain	Sky	PR	Rider	Car	Truck	Bus	Train	Motor	Bike	mIoU
CBST [34]	×	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
AdvEnt [26]	×	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
IDA [21]	×	90.6	37.1	82.6	30.1	19.1	29.5	32.4	20.6	85.7	40.5	79.7	58.7	31.1	86.3	31.5	48.3	0.0	30.2	35.8	46.3
CRST [35]	×	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
CrCDA [7]	×	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
IAST [18]	×	93.8	57.8	85.1	39.5	26.7	26.2	43.1	34.7	84.9	32.9	88.0	62.6	29.0	87.3	39.2	49.6	23.2	34.7	39.6	51.5
ProDA [33]	×	91.5	52.4	82.9	42.0	35.7	40.0	44.4	43.3	87.0	43.8	79.5	66.5	31.4	86.7	41.1	52.5	0.0	45.4	53.8	53.7
CPSL [13]	×	91.7	52.9	83.6	43.0	32.3	43.7	51.3	42.8	85.4	37.6	81.1	69.5	30.0	88.1	44.1	59.9	24.9	47.2	48.4	55.7
Source Only	-	69.7	20.5	73.3	22.1	12.3	23.5	31.8	17.9	78.7	18.7	68.2	53.9	26.5	70.6	32.2	4.5	8.1	26.8	31.5	36.4
UR [23]	✓	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
SFDA [16]	✓	91.7	52.7	82.2	28.7	20.3	36.5	30.6	23.6	81.7	35.6	84.8	59.5	22.6	83.4	29.6	32.4	11.8	23.8	39.6	45.8
HCL [6]	✓	92.0	55.0	80.4	33.5	24.6	37.1	35.1	28.8	83.0	37.6	82.3	59.4	27.6	83.6	32.3	36.6	14.1	28.7	43.0	48.1
C-SFDA (ours)	✓	90.4	42.2	83.2	34.0	29.3	34.5	36.1	38.4	84.0	43.0	75.6	60.2	28.4	85.2	33.1	46.4	3.5	28.2	44.8	**48.3**
TENT [27] (Online)	✓	87.3	39.0	79.8	24.3	19.6	21.2	25.1	16.6	83.8	34.7	77.7	57.9	17.8	85.0	24.9	20.8	2.0	16.6	4.5	38.9
AUGCO [22] (Online)	✓	90.3	41.2	81.8	26.5	21.4	34.5	404.	33.3	83.6	34.6	79.7	61.4	19.3	84.7	30.3	39.5	7.3	27.6	34.6	45.9
C-SFDA (Online)	✓	84.7	37.8	82.4	29.7	28.0	31.8	34.8	29.3	83.7	43.8	76.9	58.8	28.4	84.9	33.5	44.1	0.5	24.5	39.1	**46.3**

tribution estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7212–7222, 2022. 5, 6, 7

[3] Geoffrey French, Michal Mackiewicz, and Mark Fisher. Self-ensembling for visual domain adaptation. In *International Conference on Learning Representations*, number 6, 2018. 5, 7

[4] Xiang Gu, Jian Sun, and Zongben Xu. Spherical space domain adaptation with robust pseudo-label loss. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 6

[5] Lanqing Hu, Meina Kan, Shiguang Shan, and Xilin Chen. Unsupervised domain adaptation with hierarchical gradient synchronization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4043–4052, 2020. 6

[6] Jiaxing Huang, Dayan Guan, Aoran Xiao, and Shijian Lu. Model adaptation: Historical contrastive learning for unsupervised domain adaptation without source data. *Advances in Neural Information Processing Systems*, 34:3635–3649, 2021. 5, 6, 7, 8

[7] Jiaxing Huang, Shijian Lu, Dayan Guan, and Xiaobing Zhang. Contextual-relation consistent domain adaptation for semantic segmentation. In *European conference on computer vision*, pages 705–722. Springer, 2020. 7, 8

[8] Ying Jin, Ximei Wang, Mingsheng Long, and Jianmin Wang. Minimum class confusion for versatile domain adaptation. In *European Conference on Computer Vision*, pages 464–480. Springer, 2020. 7

[9] Alexander B. Jung, Kentaro Wada, Jon Crall, Satoshi Tanaka, Jake Graving, Christoph Reinders, Sarthak Yadav, Joy Banerjee, Gábor Vecsei, Adam Kraft, Zheng Rui, Jirka Borovec, Christian Vallentin, Semen Zhydenko, Kilian Pfeiffer, Ben Cook, Ismael Fernández, François-Michel De Rainville, Chi-Hung Weng, Abner Ayala-Acevedo, Raphael Meudec, Matias Laporte, et al. imgaug. https://github.com/aleju/imgaug, 2020. Online; accessed 01-Feb-2020. 4

[10] Youngeun Kim, Donghyeon Cho, Kyeongtak Han, Priyadarshini Panda, and Sungeun Hong. Domain adaptation without source data. *IEEE Transactions on Artificial Intelligence*, 2021. 5, 6

[11] Chen-Yu Lee, Tanmay Batra, Mohammad Haris Baig, and Daniel Ulbricht. Sliced wasserstein discrepancy for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10285–10295, 2019. 7

Table 7. Performance evaluation on **SYNTHIA→Cityscapes**. We report mean IoU (mIoU) over 16 common categories between SYNTHIA and Cityscapes. mIoU* are calculated over 13 categories. Our method achieves SOTA performance in both mIoU and mIoU*.

Method	SF	Road	SW	Build	Wall*	Fence*	Pole*	TL	TS	Veg.	Sky	PR	Rider	Car	Bus	Motor	Bike	mIoU	mIoU*
AdaptSeg [25]	×	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
AdvEnt [26]	×	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	41.2	48.0
IDA [21]	×	84.3	37.7	79.5	5.3	0.4	24.9	9.2	8.4	80.0	84.1	57.2	23.0	78.0	38.1	20.3	36.5	41.7	48.9
CRST [35]	×	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	43.8	50.1
CrCDA [7]	×	86.2	44.9	79.5	8.3	0.7	27.8	9.4	11.8	78.6	86.5	57.2	26.1	76.8	39.9	21.5	32.1	42.9	50.0
ProDA [33]	×	87.1	44.0	83.2	26.9	0.7	42.0	45.8	34.2	86.7	81.3	68.4	22.1	87.7	50.0	31.4	38.6	51.9	58.5
CPSL [13]	×	87.3	44.4	83.8	25.0	0.4	42.9	47.5	32.4	86.5	83.3	69.6	29.1	89.4	52.1	42.6	54.1	54.4	61.7
Source Only	-	45.2	19.6	72.0	6.7	0.1	24.3	5.5	7.8	74.4	81.9	57.3	17.3	39.0	19.5	7.0	6.2	31.3	36.2
UR [23]	✓	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
SFDA [16]	✓	67.8	31.9	77.1	8.3	1.1	35.9	21.2	26.7	79.8	79.4	58.8	27.3	80.4	25.3	19.5	37.4	42.4	48.7
HCL [6]	✓	80.9	34.9	76.7	6.6	0.2	36.1	20.1	28.2	79.1	83.1	55.6	25.6	78.8	32.7	24.1	32.7	43.5	50.2
C-SFDA (Ours)	✓	87.0	39.0	79.5	12.2	1.8	32.2	20.4	24.3	79.5	82.2	51.5	24.5	78.7	31.5	21.3	47.9	**44.6**	**51.3**
TENT [27] (Online)	✓	88.1	44.9	74.4	4.3	0.1	21.8	2.0	7.8	77.3	82.8	52.9	9.7	77.6	7.5	0.2	15.8	35.5	41.6
AUGCO [22] (Online)	✓	74.8	32.1	79.2	5.0	0.1	29.4	3.0	11.1	78.7	83.1	57.5	26.4	74.3	20.5	12.1	39.3	39.2	45.5
C-SFDA (Online)	✓	85.9	38.1	79.2	11.9	1.1	32.0	17.1	22.9	79.7	89.4	46.6	22.0	78.4	29.6	17.4	46.0	**43.0**	**49.5**

[12] Rui Li, Qianfen Jiao, Wenming Cao, Hau-San Wong, and Si Wu. Model adaptation: Unsupervised domain adaptation without source data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9641–9650, 2020. 5, 7

[13] Ruihuang Li, Shuai Li, Chenhang He, Yabin Zhang, Xu Jia, and Lei Zhang. Class-balanced pixel-level self-labeling for domain adaptive semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11593–11603, 2022. 7, 8

[14] Shuang Li, Mixue Xie, Kaixiong Gong, Chi Harold Liu, Yulin Wang, and Wei Li. Transferable semantic augmentation for domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11516–11525, 2021. 6

[15] Jian Liang, Dapeng Hu, and Jiashi Feng. Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation. In *International Conference on Machine Learning*, pages 6028–6039. PMLR, 2020. 4, 5, 6, 7

[16] Yuang Liu, Wei Zhang, and Jun Wang. Source-free domain adaptation for semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1215–1224, 2021. 5, 7, 8

[17] Zhihe Lu, Yongxin Yang, Xiatian Zhu, Cong Liu, Yi-Zhe Song, and Tao Xiang. Stochastic classifiers for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9111–9120, 2020. 5, 7

[18] Ke Mei, Chuang Zhu, Jiaqi Zou, and Shanghang Zhang. Instance adaptive self-training for unsupervised domain adaptation. In *European conference on computer vision*, pages 415–430. Springer, 2020. 7

[19] Jaemin Na, Heechul Jung, Hyung Jin Chang, and Wonjun Hwang. Fixbi: Bridging domain spaces for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1094–1103, June 2021. 6

[20] Zachary Nado, Shreyas Padhy, D Sculley, Alexander D'Amour, Balaji Lakshminarayanan, and Jasper Snoek. Evaluating prediction-time batch normalization for robustness under covariate shift. *arXiv preprint arXiv:2006.10963*, 2020. 5

[21] Fei Pan, Inkyu Shin, Francois Rameau, Seokju Lee, and In So Kweon. Unsupervised intra-domain adaptation for semantic segmentation through self-supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3764–3773, 2020. 7, 8

[22] Viraj Uday Prabhu, Shivam Khare, Deeksha Kartik, and Judy Hoffman. Augmentation consistency-guided self-training for source-free domain adaptive semantic segmentation. In *NeurIPS 2022 Workshop on Distribution Shifts: Connecting Methods and Applications*. 5, 7, 8

[23] Prabhu Teja Sivaprasad and Francois Fleuret. Uncertainty reduction for model adaptation in semantic segmentation. In *2021 Ieee/Cvf Conference On Computer Vision And Pattern Recognition, Cvpr 2021*, number CONF, pages 9608–9618. IEEE, 2021. 5, 7, 8

[24] Hui Tang, Ke Chen, and Kui Jia. Unsupervised domain adaptation via structurally regularized deep clustering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8725–8735, 2020. 6

[25] Yi-Hsuan Tsai, Wei-Chih Hung, Samuel Schulter, Kihyuk Sohn, Ming-Hsuan Yang, and Manmohan Chandraker. Learning to adapt structured output space for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7472–7481, 2018. 8

[26] Tuan-Hung Vu, Himalaya Jain, Maxime Bucher, Matthieu Cord, and Patrick Pérez. Advent: Adversarial entropy minimization for domain adaptation in semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2517–2526, 2019. 7, 8

[27] Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. Tent: Fully test-time adaptation by entropy minimization. In *International Conference on Learning Representations*, 2021. 5, 7, 8

[28] Xinyi Wu, Zhenyao Wu, Hao Guo, Lili Ju, and Song Wang. Dannet: A one-stage domain adaptation network for unsupervised nighttime semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15769–15778, 2021. 4

[29] Haifeng Xia, Handong Zhao, and Zhengming Ding. Adaptive adversarial network for source-free domain adaptation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9010–9019, 2021. 5, 6, 7

[30] Ruijia Xu, Guanbin Li, Jihan Yang, and Liang Lin. Larger norm more transferable: An adaptive feature norm approach for unsupervised domain adaptation. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1426–1435, 2019. 5, 7

[31] Renjun Xu, Pelen Liu, Liyan Wang, Chao Chen, and Jindong Wang. Reliable weighted optimal transport for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4394–4403, 2020. 5, 7

[32] Shiqi Yang, Yaxing Wang, Joost van de Weijer, Luis Herranz, and Shangling Jui. Generalized source-free domain adaptation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8978–8987, 2021. 5, 6, 7

[33] Pan Zhang, Bo Zhang, Ting Zhang, Dong Chen, Yong Wang, and Fang 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*, pages 12414–12424, 2021. 7, 8

[34] Yang Zou, Zhiding Yu, BVK Kumar, and Jinsong Wang. Unsupervised domain adaptation for semantic segmentation via class-balanced self-training. In *Proceedings of the European conference on computer vision (ECCV)*, pages 289–305, 2018. 7

[35] Yang Zou, Zhiding Yu, Xiaofeng Liu, BVK Kumar, and Jinsong Wang. Confidence regularized self-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5982–5991, 2019. 7, 8