Black-box Unsupervised Domain Adaptation with Bi-directional Atkinson-Shiffrin Memory
(Supplemental Materials)

We provide more experiment details in Section A (including datasets, implementation details and discussions), qualitative results (including visualization of the denoised pseudo labels and qualitative comparisons) in Section B, theoretical insights in Section C and social impacts and limitations in Section D.

A. Experiment Details and Discussions

A.1. Datasets

We evaluate the proposed BiMem over multiple datasets as listed below:

Black-box UDA for Semantic Segmentation: We study two domain adaptive semantic segmentation tasks GTA5 [18] → Cityscapes [7] and SYNTHIA [19] → Cityscapes. Cityscapes has 19 categories with pixel-wise annotations. GTA5 has $24,966$ synthetic images and shares 19 categories with Cityscapes. For SYNTHIA, we use 'SYNTHIA-RAND-CITYSCAPES' which contains $9,400$ synthetic images and shares 16 categories with Cityscapes. For the two tasks, we adopt the 2975 training images in Cityscapes as target domain and perform evaluation on the 500 validation images in Cityscapes.

Black-box UDA for Object Detection: We study two domain adaptive detection tasks Cityscapes [7] → Foggy Cityscapes [22] and SYNTHIA [19] → Cityscapes [7]. Cityscapes has 2975 training images and 500 validation images, where the bounding boxes are generated from pixel-wise instance annotations as in [6, 21]. Foggy Cityscapes is translated from Cityscapes by adding simulated fog, which shares 8 instance categories with Cityscapes. For task Cityscapes → Foggy Cityscapes, we adopt 2975 training images in Foggy Cityscapes as target domain and perform evaluation on the 500 validation images in Foggy Cityscapes. For SYNTHIA, we use 'SYNTHIA-RAND-CITYSCAPES' which contains $9,400$ synthetic images and shares 6 instance categories with Cityscapes. We adopt 2975 training images in Cityscapes as target domain and evaluate on the 500 validation images in Cityscapes under task SYNTHIA → Cityscapes.

Black-box UDA for Image Classification: We study two UDA-based image classification tasks Office-Home [25] and Office-31 [20]. Office-home consists of 12 adaptation tasks with 4 domains: Art, Clipart, Product and Real-World. Office-31 includes 6 adaptation tasks with 3 domains: Amazon, DSLR and Webcam. Office-Home has images of 65 classes from Art (A), Clipart (C), Product (P) and Real-World (R) which consist of 2496, 4464, 4503 and 4450 images, respectively. Following [36, 20], we study 12 adaptation tasks: A→C, A→P, A→R, C→A, C→P, C→R, P→A, P→C, P→R, R→A, R→C and R→P. Office-31 has images of 31 classes from Amazon (A), Webcam (W) and DSLR (D) which have 2817, 795 and 498 images, respectively. Following [36, 20], we study 6 adaptation tasks: A→W, D→W, W→D, A→D, D→A, and W→A.

A.2. Implementation Details

Semantic Segmentation: We adopt DeepLab-V2 [4] with ResNet-101 [11] (pretrained on ImageNet [8]) as the segmentation network as in [24, 35]. We adopt SGD optimizer [1] with a momentum 0.9 and a weight decay $1e-4$. The initial learning rate is $1e-4$ and decayed by a polynomial policy of power 0.9 [4].

Object Detection: We adopt deformable-DETR [34] with ResNet-50 [11] (pretrained on ImageNet [8]) as detection network as in [2, 34]. We adopt SGD optimizer [1] with a momentum 0.9 and a weight decay $1e-4$. The initial learning rate is $2e-4$.

Image Classification: Following [15], we adopt ResNet-50 [11] (pretrained on ImageNet [8]) for the tasks Office-Home and Office-31. We adopt SGD optimizer [1] with a momentum 0.9 and a weight decay $1e-3$. The initial learning rates are $1e-3$ and $1e-2$ for ResNet-50 feature extractor and classifier, respectively.

For all experiments, we set momentum coefficient γ at 0.999 and update coefficient γ' in Eq.6 at 0.999. We set the size of short-term memory M at 65536 as in [10] and the number of features N in every active selection at 256. For all experiments, we warm up the target model with soft pseudo-labels predicted by the source model, and the category-wise centroids in long-term memory are initialized by all predictions from the warm-up target model.

A.3. Discussions

A.3.1 Parameter Analysis

We study the update coefficient γ' used in Eq.4 for long-term memory and the parameter N used in active selection for short-term memory.

Update Coefficient γ'. The update coefficient γ' in Eq. 4 (in the main text) controls the update speed of long-term memory, $e.g.$, the larger it is, the slower the long-term memory updates. We study how it affects the adaptation over task GTA5 $\rightarrow$ Cityscapes. As shown in Table 1, BiMem yields robust performance when γ' is large enough (from 0.99 to 0.9999) while its performance starts to drop slightly when γ' becomes too small. This shows that a large update coefficient with smooth and slow update helps maintain stable long-term memorization with effective memorization calibration and superior adaptation performance, whereas a too small update coefficient leads to fast update of the long-term memory and results in unstable long-term memorization and less effective backward calibration.

	Update Coefficient γ'				
Method	0.5	0.9	0.99	0.999	0.9999
BiMem	46.2	47.4	47.9	**48.2**	48.1

Table 1: The update coefficient γ' defined in Eq.4 affects domain adaptation. The experiments are conducted over semantic segmentation task GTA5 $\rightarrow$ Cityscapes.

Number of Features N for Active Selection. The parameter N controls the number of features that are actively selected by short-term memory from sensory memory in every training iteration. We study how parameter N affects the adaptation over task GTA5 $\rightarrow$ Cityscapes. As shown in Table 2, the proposed BiMem is tolerant to the variation of parameter N where the best performance is achieved when N is set at 256.

	Number of Features N in Active Selection				
Method	128	256	384	512	640
BiMem	47.9	**48.2**	48.1	47.8	48.0

Table 2: The number of features N used in the active selection design affects domain adaptation. The experiments are conducted over semantic segmentation task GTA5 $\rightarrow$ Cityscapes.

A.3.2 Difference to Memory-based UDA Methods

Several recent studies [12, 26, 13] introduce memorization into network training by memorizing historical models [12] for source-free UDA, and memorizing source and target features [26, 13] for conventional UDA. Different from [12, 26, 13] that memorize source features/models for conventional or source-free UDA, BiMem focuses on black-box UDA and relies on neither source models nor source data/features during adaptation. Specifically, BiMem constructs three types of memory that interact with each other in a bi-directional manner, which memorizes and calibrates useful target information learnt during black-box adaptation to make up for the absence of source data and models. Table 3 shows that BiMem clearly outperforms [12, 26, 13], largely because they were not designed for black-box UDA and their designed memorization mechanisms cannot well handle the absence of source data and models.

A.3.3 Difference to Noisy Label Learning

In this work, we leverage our constructed BiMem to denoise the source-predicted pseudo labels for black-box UDA, which share similar ideas as in noisy label learning. Here we study the difference between black-box UDA and noisy label learning by providing insights and experiments.

Methods	MeGA [26]	MemSAC [13]	HCL [12]	**BiMem**
mIoU	43.1	44.7	45.7	**48.2**

Table 3: Comparison with existing memory-based UDA methods [26, 13, 12] over GTA5 → Cityscapes semantic segmentation.

Learning from noisy labels is a critical task in deep learning due to the lack of high-quality labels in many real-world data [23, 17, 30, 9, 29, 27, 3, 31, 14, 33]. In prior studies on learning with noisy labels, the correct labels are randomly corrupted by a noise transition matrix that defines the probability of flipping correct labels to false labels [23, 17, 30, 9, 29]. Different from the label noise simulated by random corruptions in [29, 27, 3, 31, 14, 33] that is normally independent to data distributions, the pseudo label noises in black-box UDA are caused by the 'distribution gaps' between source and target domains, which is explicitly correlated to source-target distribution ($e.g.$, data with different categories and styles generally experience different noise degrees across domains).

To further investigate the difference between the label noises simulated by random corruptions and the pseudo label noises caused by domain gaps, we experimentally compare BiMem with several noisy label learning methods [28, 5, 32] over task GTA5 → Cityscapes. Experimental results in Table 4 show that BiMem outperforms [28, 5, 32] by large margins, demonstrating that noisy label learning methods [28, 5, 32] cannot well handle the pseudo label noises in black-box UDA that is caused by the 'distribution gaps' between source and target domains and quite different to the label noise simulated by random corruptions.

Method	SCE [28]	INCV [5]	AdaCorr [32]	**BiMem**
mIoU	39.2	43.6	42.7	**48.2**

Table 4: Comparison with existing noisy label learning techniques [28, 5, 32] over semantic segmentation task GTA5 → Cityscapes.

A.3.4 Effectiveness of Memory Consolidation in Long-term Memory

As described in Section Method in the main text, long-term memory consolidates the calibrated sensory and short-term memories iteratively, the features stored in which tends to gradually move closer to the true feature centroid of each category while the adaptation moves on. In this subsection, we provide quantitative results to support this claim. Specifically, we measure the quality of long-term features by calculating the $l1$ distance between the true feature centroids (acquired by using the category ground-truth) and the feature centroids stored in long-term memory (obtained by consolidating the calibrated sensory and short-term memories iteratively). As shown in Fig. 1, by consolidating sensory memory, the long-term features ($i.e.$, the category-wise feature centroids) gradually move close to the true feature centroids as the adaptation moves on (orange line), showing that our memory consolidation design enables the long-term memory to capture less-noisy representative information. In addition, consolidating the hard features (from short-term memory) into long-term memory further pushes the long-term features closer to the true feature centroids (blue line), showing that the hard features captured by short-term memory help build more comprehensive long-term memorization as the hard features are generally rare.

A.3.5 Adaptation across Different Models

Black-box UDA is flexible and allows different target networks regardless of source networks as described in Section Introduction in the main text. We examine this property by evaluating BiMem over two flexible adaptation scenarios across different model architectures or sizes (on GTA5 → Cityscapes). The *first column* of Table 5 shows the results of adaptation across different model architectures, $i.e.$, from SegFormer (MiT-B5) to DeepLab-v2 (ResNet-101). It can be seen that all black-box UDA methods perform much better with the stronger source model SegFormer while BiMem achieves the best performance of 52.6% in mIoU. The *second column* of Table 5 presents the results of adaptation from large model to light-weight model $i.e.$, DeepLab-v2 (ResNet-101) → DeepLab-v2 (ResNet-50). It shows that the light-weight target models still achieve competitive performance as compared with the large target models in the *last column*, and BiMem achieves the best performance of 46.5 % in mIoU.

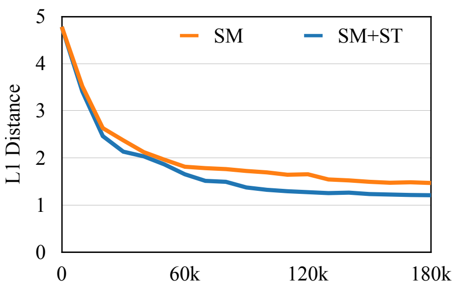


Figure 1: Effectiveness of memory consolidation in long-term Memory. 'SM' denotes consolidation of sensory memory and 'SM + ST' denotes consolidation of both sensory memory and short-term memory.

These results indicate that BiMem is scalable as it could bring further improvements by employing stronger source models and allows to adapt towards light-weight models in specific scenarios such as edge computing.

Source Model ↓ Target Model	SegFormer (MiT-B5) ↓ DeepLab-v2 (ResNet-101)	DeepLab-v2 (ResNet-101) ↓ DeepLab-v2 (ResNet-50)	DeepLab-v2 (ResNet-101) ↓ DeepLab-v2 (ResNet-101)
Source only	44.0	36.6	36.6
CBST [35]	46.9	38.9	40.3
SFDA [16]	48.2	41.2	43.3
DINE [15]	50.4	44.3	46.7
BiMem	**52.6**	**46.5**	**48.2**

Table 5: Black-box adaptation across different models.

A.3.6 Adaptation with Predictions from Multiple Source Domains

Black-box UDA is flexible and allows effective and efficient adaption from multiple source domains while raising little concern in data privacy and introducing much less computation overhead as compared with traditional UDA and source-free UDA. We examine this property by evaluating BiMem over multi-source adaptation scenario, $i.e.$, GTA5 & SYNTHIA → Cityscapes. For each unlabeled target sample, we simply average the two category-wise probability vector predicted by GTA5-trained model and SYNTHIA-trained model respectively to obtain its pseudo label. As shown in Table 6, all black-box UDA methods perform much better when adapting with predictions from two sources, while BiMem achieves the best performance clearly. This shows that the information from multiple source domains are complementary for domain adaptation while black-box UDA enables effective usage of this property as black-box UDA introduces little computation overhead when increasing the number of source domains. In another word, these results indicate that our proposed BiMem is scalable and can be easily improved by fusing the information from additional source domains with a simple average operation.

Source ↓ Target	GTA5 & SYNTHIA ↓ Cityscapes	GTA5 ↓ Cityscapes	SYNTHIA ↓ Cityscapes
Source only	42.3	39.8	33.5
CBST [35]	46.8	44.2	36.8
SFDA [16]	49.2	47.4	38.8
DINE [15]	52.8	50.4	40.9
BiMem	**54.7**	**52.3**	**42.2**

Table 6: Black-box adaptation with the pseudo labels predicted from multiple source models. The results are evaluated over 16 categories shared by GTA5, SYNTHIA and Cityscapes.

A.3.7 Analysis of the 'Forgetting' Issue in Black-box UDA

As discussed in Sec.4.7 and Figure 3 in the main text, we examine the source of the 'forgetting' by splitting target training data into two portions according to their initial pseudo labels and evaluate the model (trained using full training data) on these two portions respectively, as shown in the Columns 1 and 2 (copied from the main manuscript) of Fig. 2. Here we additionally provide the controlled experiment on target validation data. Similarly, we split the target validation data into two subsets according to their initial pseudo labels (predicted by the black-box predictor). This produces target validation data with correct initial pseudo labels ($i.e.$, $X_{t\_val}^{\text{correct}}$) and target validation data with incorrect initial pseudo labels ($i.e.$, $X_{t\_val}^{\text{incorrect}}$), where the splitting allows training models with full data but evaluating them over decomposed validation data. As shown in Columns 3 and 4 of Fig 2, we can observe similar phenomenon as in Fig.3 in the main text. For vanilla self-training, the mIoU of $X_{t\_val}^{\text{correct}}$ increases stably in the left graph while the mIoU of $X_{t\_val}^{\text{incorrect}}$ increases at the early stage but decreases gradually as shown in the right graph. This indicates that vanilla self-training learns useful information to generate correct predictions for $X_{t\_val}^{\text{incorrect}}$ at the early training stage but tends to forget these information at a later training stage. Differently, BiMem builds comprehensive and robust memorization that memorizes and calibrates useful and representative information on the fly, leading to stabler black-box UDA without performance degradation and training collapse.

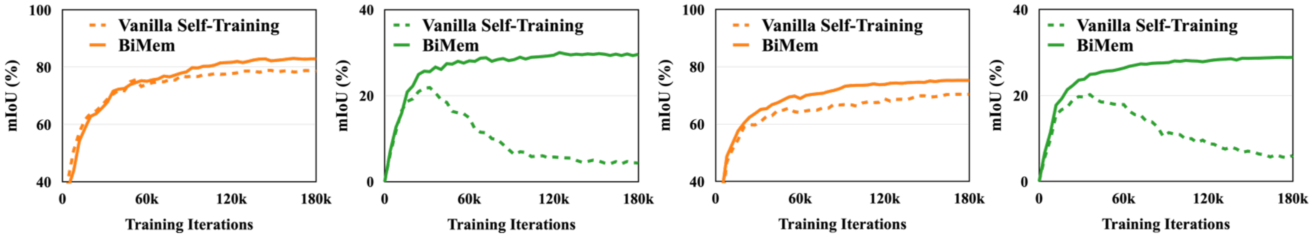


Figure 2: Performance of vanilla self-training and our BiMem on the decomposed target validation data (Columns 3 and 4), $i.e.$, target validation data with correct initial pseudo labels (Column 3) and target validation data with incorrect initial pseudo labels (Column 3). Columns 1 and 2 (copied from the main manuscript) show the results on target training data.

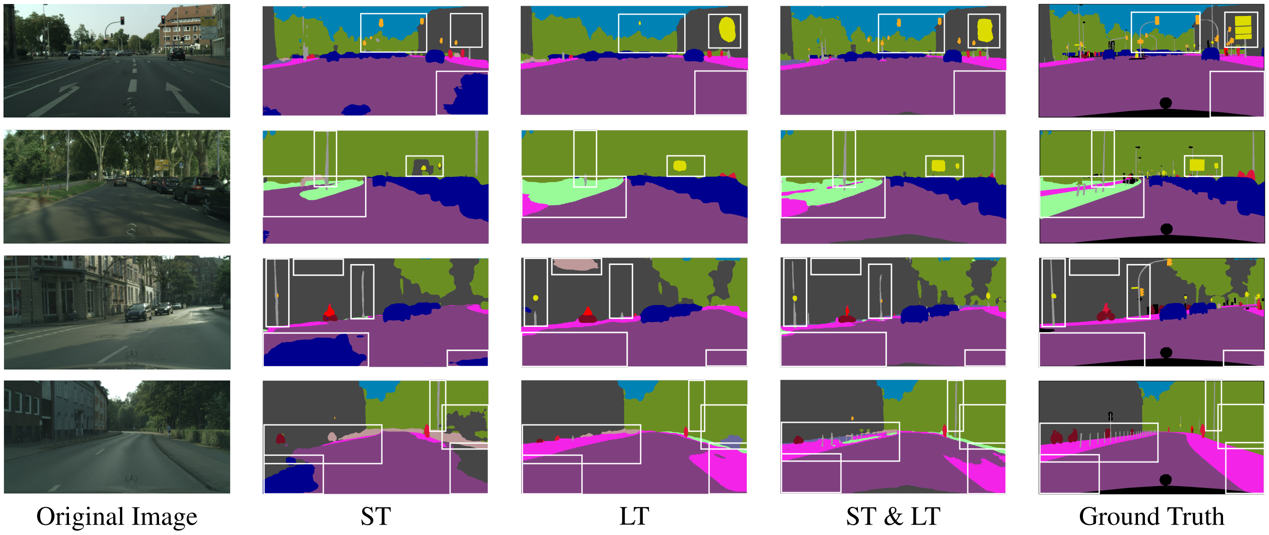


Figure 3: Visualization of the denoised pseudo labels over task GTA → Cityscapes. 'ST' denotes short-term memory, 'LT' denotes long-term memory and 'ST&LT' denotes including both short-term memory and long-term memory.

B. Qualitative Results

B.1. Visual Illustrations of Pseudo Label Denoising with BiMem

As discussed in Section 4.6 in the main text, the experimental results in Rows 2-4 of Table 8 (in the main text) quantitatively demonstrate that short-term memory and long-term memory are complementary for denoising source-predicted pseudo labels. In this subsection, we provide the respective denoised pseudo labels to qualitatively illustrate this property. It can be observed that the short-term memory largely improves the detection of small-scale objects ($e.g.$, traffic light and pole) as shown in the 2nd column of Fig. 3, while the long-term memory mainly helps denoise large-scale objects ($e.g.$, road and vegetation) as shown in the 3rd column of Fig. 3. This shows that the short-term memory and the long-term memory capture different types of information learnt during adaptation. In addition, we can observe that the combination of short-term memory and long-term memory performs the best clearly as shown in the 4th column of Fig. 3, showing that the different types of information captured by the short-term memory and the long-term memory respectively can work synergically and are complementary for label denoising.

B.2. Qualitative Comparisons

We present qualitative illustrations and comparisons over tasks GTA5 → Cityscapes and SYNTHIA → Cityscapes. As shown in Fig. 4, BiMem yields the best segmentation consistently which is well aligned with the quantitative results.

C. Theoretical Insights

BiMem can be modelled as a memory-calibrated classification maximum likelihood (CML) problem optimized via classification expectation maximization.

Proof: The objective of BiMem self-training is to find the parameters θ_G that maximizes the classification log-likelihood function [37] of the observed target samples x_t:

$$\theta_G^* = \operatorname*{argmax}_{\theta_G} \sum_{x_t} \sum_{c=1}^{C} y_t^{(c)} \log p(c|x_t;\theta_G), \tag{1}$$

where C stands for the number of categories, $y_t \in \{0,1\}^C$ for all t, and $p(c|x_t;\theta_G)$ is the posterior probability.

Eq. 1 can be maximized by CEM. Compared with traditional expectation maximization that has an "expectation" step and a "maximization" step, CEM has an additional "classification" step that assigns a label to each target sample with a maximal posterior probability.

The CEM steps of our BiMem are illustrated as the following:

Expectation Step: Given the model parameters θ_G, estimate the posterior probability: $p(c|x_t;\theta_G)$, for all x_t.

Classification Step: Fix model parameters θ_G and find the pseudo label with memory-based calibration weight $\hat{p}$ as following:

$$\hat{y}_t = \operatorname*{argmax}_{y_t} \sum_{c=1}^{C} y_t^{(c)} \log p(c|x_t;\theta_G)\ \hat{p} \tag{2}$$

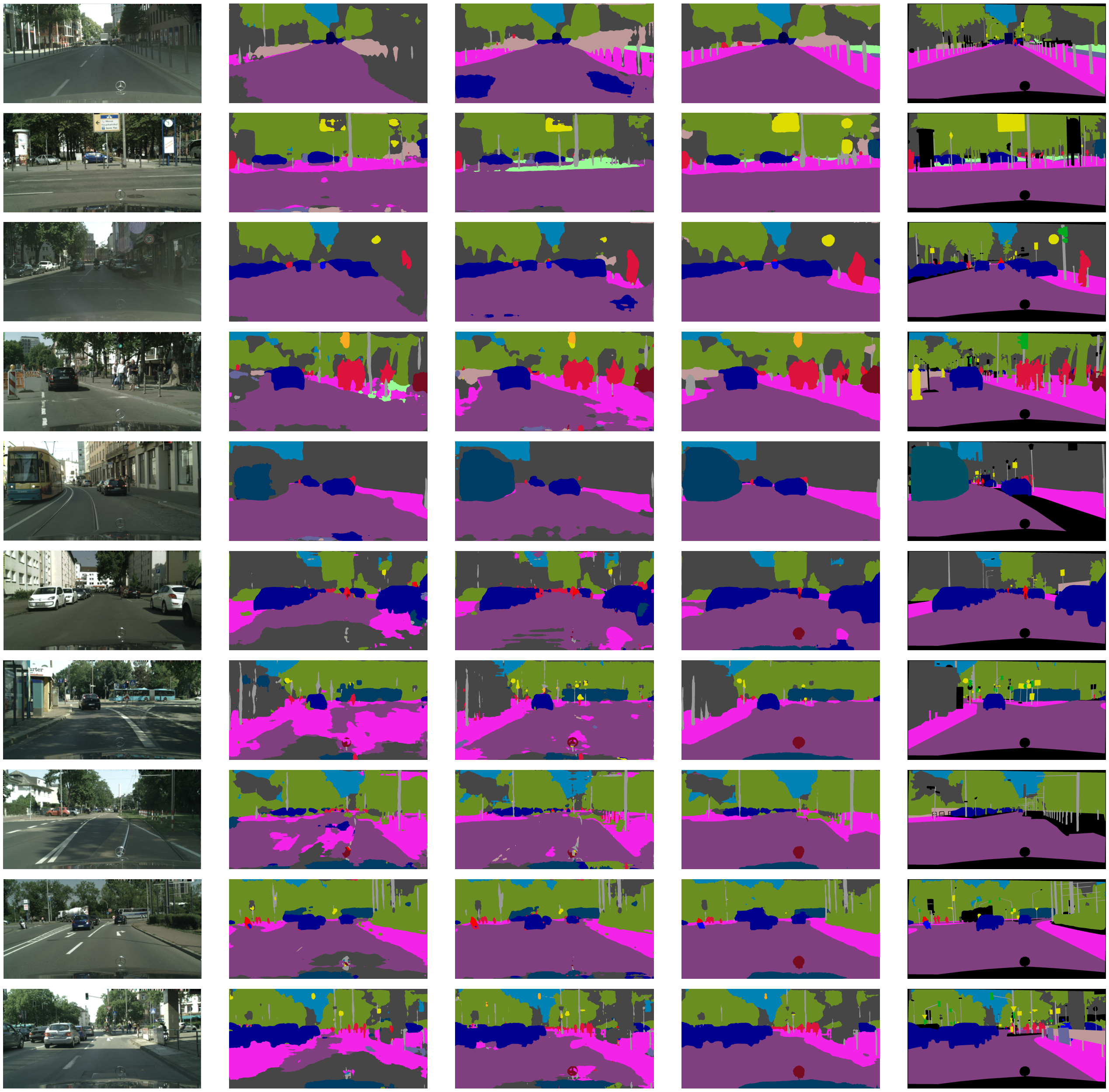
Maximization Step: Now, we are ready to maximize the classification log-likelihood as follows:

$$\max_{\theta_G} \sum_{x_t} \sum_{c=1}^{C} \hat{y}_t^{(c)} \log p(c|x_t;\theta_G). \tag{3}$$

Thus, the classification log-likelihood defined in Eq. 1 can be maximized by iterating the above three steps until convergence.

D. Social Impacts and Limitations

This work explores a new transfer learning pipeline named black-box UDA, which has clear advantages in little data privacy concerns and the flexibility of allowing different target networks regardless of the source-trained black-box models. In another word, black-box UDA benefits the computer vision community by providing a new transfer learning solution that raises little data privacy issue. On the other hand, the explored techniques in this work are still at an early stage and thus our



Original Image Baseline [4] DINE [15] BiMem(Ours) Ground Truth

Figure 4: Qualitative comparison of BiMem with the baseline model [4] and DINE [15] over two tasks including GTA5 → Cityscapes as shown in rows 1-5 and SYNTHIA → Cityscapes as shown in rows 6-10. The proposed BiMem yields the best segmentation over two adaptation tasks consistently.

proposed method can serve as an auxiliary tool in applications instead of the hard control system that could lead to harmful consequences.

References

[1] Léon Bottou. Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010*, pages 177–186. Springer, 2010. 1

[2] Qi Cai, Yingwei Pan, Chong-Wah Ngo, Xinmei Tian, Lingyu Duan, and Ting Yao. Exploring object relation in mean teacher for cross-

domain detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11457–11466, 2019. 1

[3] Haw-Shiuan Chang, Erik Learned-Miller, and Andrew McCallum. Active bias: Training more accurate neural networks by emphasizing high variance samples. *Advances in Neural Information Processing Systems*, 30, 2017. 3

[4] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2017. 1, 7

[5] Pengfei Chen, Ben Ben Liao, Guangyong Chen, and Shengyu Zhang. Understanding and utilizing deep neural networks trained with noisy labels. In *International Conference on Machine Learning*, pages 1062–1070. PMLR, 2019. 3

[6] Yuhua Chen, Wen Li, Christos Sakaridis, Dengxin Dai, and Luc Van Gool. Domain adaptive faster r-cnn for object detection in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3339–3348, 2018. 1

[7] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016. 1

[8] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 1

[9] Jacob Goldberger and Ehud Ben-Reuven. Training deep neural-networks using a noise adaptation layer. 2016. 3

[10] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9729–9738, 2020. 2

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 1

[12] Jiaxing Huang, Dayan Guan, Aoran Xiao, and Shijian Lu. Model adaptation: Historical contrastive learning for unsupervised domain adaptation without source data. *arXiv preprint arXiv:2110.03374*, 2021. 2, 3

[13] Tarun Kalluri, Astuti Sharma, and Manmohan Chandraker. Memsac: Memory augmented sample consistency for large scale domain adaptation. *arXiv preprint arXiv:2207.12389*, 2022. 2, 3

[14] Xuefeng Li, Tongliang Liu, Bo Han, Gang Niu, and Masashi Sugiyama. Provably end-to-end label-noise learning without anchor points. In *International Conference on Machine Learning*, pages 6403–6413. PMLR, 2021. 3

[15] Jian Liang, Dapeng Hu, Jiashi Feng, and Ran He. Dine: Domain adaptation from single and multiple black-box predictors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8003–8013, 2022. 1, 4, 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. 4

[17] Nagarajan Natarajan, Inderjit S Dhillon, Pradeep K Ravikumar, and Ambuj Tewari. Learning with noisy labels. *Advances in neural information processing systems*, 26, 2013. 3

[18] Stephan R Richter, Vibhav Vineet, Stefan Roth, and Vladlen Koltun. Playing for data: Ground truth from computer games. In *European conference on computer vision*, pages 102–118. Springer, 2016. 1

[19] German Ros, Laura Sellart, Joanna Materzynska, David Vazquez, and Antonio M Lopez. The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3234–3243, 2016. 1

[20] Kate Saenko, Brian Kulis, Mario Fritz, and Trevor Darrell. Adapting visual category models to new domains. In *European conference on computer vision*, pages 213–226. Springer, 2010. 1

[21] Kuniaki Saito, Yoshitaka Ushiku, Tatsuya Harada, and Kate Saenko. Strong-weak distribution alignment for adaptive object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6956–6965, 2019. 1

[22] Christos Sakaridis, Dengxin Dai, and Luc Van Gool. Semantic foggy scene understanding with synthetic data. *International Journal of Computer Vision*, 126(9):973–992, 2018. 1

[23] Hwanjun Song, Minseok Kim, Dongmin Park, Yooju Shin, and Jae-Gil Lee. Learning from noisy labels with deep neural networks: A survey. *IEEE Transactions on Neural Networks and Learning Systems*, 2022. 3

[24] 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. 1

[25] Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. Deep hashing network for unsupervised domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5018–5027, 2017. 1

[26] Vibashan VS, Vikram Gupta, Poojan Oza, Vishwanath A Sindagi, and Vishal M Patel. Mega-cda: Memory guided attention for category-aware unsupervised domain adaptive object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4516–4526, 2021. 2, 3

[27] Ruxin Wang, Tongliang Liu, and Dacheng Tao. Multiclass learning with partially corrupted labels. *IEEE transactions on neural networks and learning systems*, 29(6):2568–2580, 2017. 3

[28] Yisen Wang, Xingjun Ma, Zaiyi Chen, Yuan Luo, Jinfeng Yi, and James Bailey. Symmetric cross entropy for robust learning with noisy labels. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 322–330, 2019. 3

[29] Tong Xiao, Tian Xia, Yi Yang, Chang Huang, and Xiaogang Wang. Learning from massive noisy labeled data for image classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2691–2699, 2015. 3

[30] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization (2016). *arXiv preprint arXiv:1611.03530*, 2017. 3

[31] HaiYang Zhang, XiMing Xing, and Liang Liu. Dualgraph: A graph-based method for reasoning about label noise. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9654–9663, 2021. 3

[32] Songzhu Zheng, Pengxiang Wu, Aman Goswami, Mayank Goswami, Dimitris Metaxas, and Chao Chen. Error-bounded correction of noisy labels. In *International Conference on Machine Learning*, pages 11447–11457. PMLR, 2020. 3

[33] Tianyi Zhou, Shengjie Wang, and Jeff Bilmes. Robust curriculum learning: from clean label detection to noisy label self-correction. In *International Conference on Learning Representations*, 2020. 3

[34] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. *arXiv preprint arXiv:2010.04159*, 2020. 1

[35] 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. 1, 4

[36] Yang Zou, Zhiding Yu, Xiaofeng Liu, BVK Kumar, and Jinsong Wang. Confidence regularized self-training. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5982–5991, 2019. 1

[37] Yang Zou, Zhiding Yu, Xiaofeng Liu, B.V.K. Vijaya Kumar, and Jinsong Wang. Confidence regularized self-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019. 6