Augmentation Consistency-guided Self-training for Source-free Domain Adaptive Semantic Segmentation

Anonymous Author(s)
Affiliation
Address
email

Abstract

We focus on source-free domain adaptation for semantic segmentation, wherein a source model must adapt itself to a new target domain given only unlabeled target data. We propose Augmentation Consistency-guided Self-training (AUGCO), an adaptation algorithm that uses the model's pixel-level predictive consistency across diverse, automatically generated views of each target image along with model confidence to identify reliable pixel predictions, and selectively self-trains on those, leading to state-of-the-art performance within a simple to implement and fast to converge approach.

1 Introduction

Consider a deep model trained to perform semantic segmentation deployed atop an autonomous vehicle. While unsupervised domain adaptation (DA) has been extensively studied [1–5], most prior DA methods assume continued access to labeled source data during adaptation. However, this may be impractical due to the limitations of on-board compute and memory, particularly so for a compute-heavy task such as segmentation.

We consider the problem of adapting such a trained semantic segmentation model to a new target domain given only its trained parameters and unlabeled target data. The absence of source data for regularization makes this setting very challenging and highly susceptible to divergence from original task training. We build upon recent work in parameter constrained self-training called TENT [6], which constrains optimization to only update the model's batch-norm parameters (both affine and normalization), and self-trains on unlabeled target data by minimizing a conditional entropy [7] loss. While TENT leads to modest performance improvements on standard domain shifts, it performs self-training on *all* model predictions. Under a domain shift, many of the model's predictions may initially be incorrect, and entropy minimization encourages the model to increase its confidence even on such incorrect predictions! As a result, unconstrained self-training leads to error accumulation [8–10], particularly on categories on which the source model does poorly to begin with.

To address this, prior work has proposed *selective* self-training on instances deemed *reliable* via model confidence [11] or consistency under random image augmentations [10]. However, model confidence from deep networks is known to be miscalibrated under a domain shift [12], and the suitability of augmentation consistency for semantic segmentation has not been previously studied. We propose AUGCO, a selective self-training algorithm that makes use of a novel selection strategy based on combines pixel-level predictive consistency across diverse, automatically generated target image views with per-class confidence.

2 AUGCO: Augmentation Consistency-guided Self-Training

Setup and Notation. In semantic segmentation we are given an input image, $\mathbf{x} \in \mathbb{R}^{H \times W \times 3}$, and the goal is to label every pixel, $\mathbf{x}_{ij}$, with one of C semantic labels, $y_{ij} \in \{1, 2, \ldots, C\}$, producing

Submitted to the Workshop on Distribution Shifts, 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

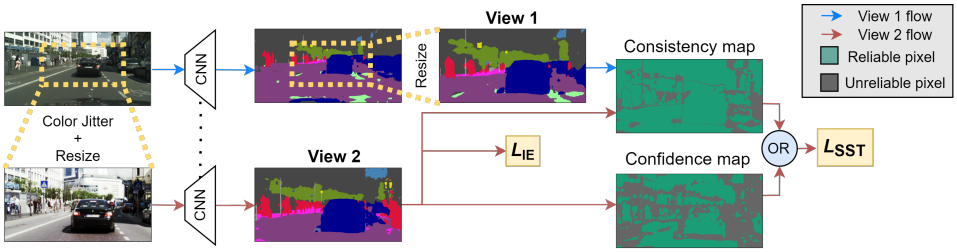

Figure 1: Overview of Augmentation Consistency-guided Self-Training (AUGCO). **Left:** First, the model makes predictions on two views of each target image that differ in scale, spatial context and color statistics, that are generated via a random crop, resize, and jitter strategy (Sec. 2.1). **Right:** Next, reliable pixel predictions for self-training are identified based on pixel-level consistency across aligned predictions and class-conditioned confidence thresholding, followed by selective self-training (Sec. 2.2).

an output label image, $y \in \mathbb{R}^{H \times W}$. To do this, we will learn a function h (CNN in our case) which takes images as input and produces a probabilistic output over C classes for each output pixel: $h : \mathbf{x} \rightarrow \mathbf{p} \in \mathbb{R}^{H \times W \times C}$. We produce a *pseudolabel* by taking the argmax of the output probabilities: $\hat{y} = \mathbf{p}$. In source-free domain adaptation, we assume access to a model trained on labeled source ($\mathcal{S}$) data, $h_{\mathcal{S}}$, as well as N *unlabeled* instances $\mathbf{x}_{\mathcal{T}} \sim \mathcal{P}_{\mathcal{T}}(\mathcal{X})$ from a target domain $\mathcal{T}$.

Overview. Our method first uses a random crop, resize, and jitter strategy to generate two aligned predictive views of each target image that capture model predictions across varying object scale, spatial context, and color statistics (Sec. 2.1). Next, AUGCO identifies *reliable* model predictions on which to self-train using self-supervised signals in the form of pixel-level predictive consistency across the two aligned views, as well as model confidence. Finally, the model is self-trained using pseudolabels for reliable predictions. See Fig. 1.

2.1 Aligned predictive view generation

A key facet of our approach will be to identify pixels for which model predictions are deemed reliable. To do this we ensemble model predictions over random image regions that differ in scale and spatial context. We begin by randomly selecting a bounding box with coordinates, (r_1, c_1, r_2, c_2), for each target image that satisfies two constraints: i) it spans an area that is 25-50% of the area of the original image and ii) it matches the aspect ratio of the original image (i.e. $(r_2 - r_1)/(c_2 - c_1) = H/W$).

View 1 (resized crop of prediction): To create the first output prediction, we pass the original image, $\mathbf{x}_{\mathcal{T}}$, through the current model, h, to produce an output probabilistic prediction, $\mathbf{p} = h(\mathbf{x}_{\mathcal{T}})$. This original output prediction will be cropped using the random bounding box coordinates and resized to the original output image size: $V = \texttt{resize}(\mathbf{p}[r_1 : r_2, c_1 : c_2], H, W)$

View 2 (prediction on resized image crop): For our second output prediction we first modify image appearance by applying a pixel-level color jitter $\mathbf{x}'_{\mathcal{T}} = \texttt{jitter}(\mathbf{x}_{\mathcal{T}})$. We then use the same bounding box coordinates to extract a cropped image region and resize that region to the original image size to produce a rescaled image view $\tilde{\mathbf{x}}_{\mathcal{T}} = \texttt{resize}(\mathbf{x}'_{\mathcal{T}}[r_1 : r_2, c_1 : c_2], H, W)$. This jittered, cropped, and resized image is then passed through the model to produce a probabilistic output, $\tilde{\mathbf{p}} = h(\tilde{\mathbf{x}}_{\mathcal{T}})$ and associated predicted view, $\tilde{V} = \tilde{\mathbf{p}}$.

We thus obtain aligned predictive views V and $\tilde{V}$, which capture model predictions made at varying object scale (*e.g.* in Fig. 1, the size of the car in the secondary view is larger than in the original), spatial context (*e.g.* additional cars are absent in the secondary view), and color statistics.

2.2 Selective Self-Training

2.2.1 Measuring Reliability

Pixel-level predictive consistency. First, we measure pixel-level consistency between the model's aligned predictions V and $\tilde{V}$, and mark pixels with identical predictions ($V_{ij} == \tilde{V}_{ij}$) across the two views as "consistent" and those with different predicted labels as "inconsistent".

Class-conditioned confidence thresholding. In addition to predictive consistency, we also aim to capture a notion of the intrinsic model confidence. We compute a per-category empirical range

Method	G→C	S→C	C→DZN
source	34.4	29.4/34.1	28.8
TENT [6]	38.9	35.5/41.6	26.6
Test-time BN [18]	37.7	35.0/40.8	28.0
SFDA [19]	43.2	39.2/**45.9**	-
AUGCO (**ours**)	**47.1**	**39.5/45.9**	**32.4**

Table 1: **Results**: We report mIoU over the target test set. On S→C we follow prior work and report mIoU over 16 and 13 categories.

to choose an adaptive per-category confidence threshold. Given a batch, we gather all output probabilities and select a confidence threshold per category, $t_c \in \mathbb{R}$, corresponding to the top K-th percentile (K=50 in our experiments) of observed confidence values for category c. We consider an output prediction to be high confidence if its top score is greater than the corresonding category threshold: $\max \mathbf{p}_{ij} > t_{\mathbf{p}_{ij}}$.

Overall, for a pixel, $\mathbf{x}_{ij}$, with per-view probabilistic and categorical predictions, $\mathbf{p}, V$ and $\tilde{\mathbf{p}}, \tilde{V}$, we define a binary reliability value, r_{ij}, in the following way:

$$r_{ij} = \begin{cases} 1 & \text{if } \overbrace{V_{ij} = \tilde{V}_{ij}}^{\text{consistent}} \text{ or } \overbrace{\max \tilde{\mathbf{p}}_{ij} > t_{\tilde{V}_{ij}}}^{\text{confident}} \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

Selective self-training. Having obtained pseudolabels and reliability assignments, we update model parameters via self-training. To prevent task divergence in the source-free setting, we update only the model's batch-norm parameters (affine and normalization), as proposed in Wang *et al.* [6].

We then minimize a cross-entropy loss L_{CE} over reliable predictions. The self-training objective we minimize is:

$$L_{SST}(\mathbf{x}_{ij}) = r_{ij} L_{CE}(\tilde{\mathbf{p}}_{ij}, \tilde{V}_{ij}) \tag{2}$$

Finally, to encourage the model to make diverse predictions over the target domain, we add a target "information entropy" loss L_{IE} proposed in Li *et al.* [13]: we update the model to maximize entropy over the running average of its predictions q. L_{IE} is given by: $L_{IE}(\mathbf{x}_{ij}) = \sum_{c=1}^{C} \tilde{\mathbf{p}}_{ijc} \log q_c$ For L_{IE} loss weight α, the complete AUGCO loss objective that is backpropagated is given by:

$$L_{\texttt{AUGCO}} = \mathbb{E}_{\mathbf{x} \sim \mathcal{P}_{\mathcal{T}}} \left[\frac{1}{HW} \sum_{i=1,j=1}^{H,W} L_{SST}(\mathbf{x}_{ij}) + \alpha L_{IE}(\mathbf{x}_{ij}) \right] \tag{3}$$

3 Experiments

Setup. We evaluate AUGCO on 3 shifts: GTA5[14]→Cityscapes [15] (G→C) SYNTHIA [16]→Cityscapes (S→C) , and Cityscapes→Dark Zurich Night (C→DZN [17]). We report mean Intersection-over-Union (mIoU) across classes on the target test set. Across settings, we evaluate our method (AUGCO) after a *single pass* over the unlabeled target data (*i.e.* one epoch)

Baselines. We use DeepLabV3 [20] with a ResNet50 [21] backbone and compare against state-of-the art methods for test-time and source free adaptation: **TENT** [6], **Test-time BN** [18], and **SFDA** [19].

Results. Table 1 presents results. Across shifts, AUGCO outperforms prior work, often by significant margins (eg. 3.9 points and 13/19 categories over SFDA [19] on G→C), despite being considerably simpler (SFDA uses 120 epochs of adversarial learning whereas AUGCO uses 1 epoch of selective self-training). AUGCO also significantly outperforms TENT [6] (+8.2 on G→C).

Ablations. In Table 2 we present ablations of AUGCO for both architectures. We observe:

▷ **Unconstrained self-training leads to suboptimal performance (Row 1).** We first try self-training on all pixels by minimizing cross-entropy with respect to predictions. As seen, this achieves 34.04, underperforming even the source model (mIoU=34.4).

Reg.	Selection strategy		Class bal.	Loss		mIoU ↑
H(Y) loss	Confidence	Consistency	Loss wts.	Reliable	Unreliable	
(Unconstrained self-training on all predictions)				CE	CE	34.04
✓				CE	CE	39.11
✓	✓			CE	None	16.93
✓		✓		CE	None	47.05
✓	✓	✓		CE	None	47.12
✓	✓	✓	✓	CE	None	47.12

Table 2: Ablating AUGCO on GTA→Cityscapes. We report mIoU over all categories. CE = cross-entropy against predicted pseudolabel.

▷ **Pixel-level predictive consistency is an effective selection strategy (Row 3-6).** To regularize self-training, we first add a target information entropy regularizer (Sec 2) and find that mIoU increases to 39.11(Row 2). Next, to validate our selection criterion, we first use only confidence for selection (we select predictions in the top-50 %ile by confidence per-category) – despite careful tuning this leads to a low mIoU of 16.93 (Row 3), indicating that confidence alone is a poor indicator of reliability. Next, we try using predictive consistency for selection, and find this improves mIoU to 47.05 (Row 4), validating the hypothesis that predictive consistency is an effective proxy for reliability. Finally, combining consistency and confidence obtains the same best performance of 47.12.

▷ **Varying optimization parameters.** We now try alternatively training all model parameters instead of just batch norm. We observe rapid task with larger learning rates and carefully tune optimizers to obtain an mIoU of 46.74 with a learning rate of 5×10^{-6} and weight decay of 5×10^{-4}, worse than when training batch-norm parameters alone.

Analysis. To measure AUGCO's reliability measure, we evaluate if it is indeed a good indicator of reliability. We first measure the accuracy of pseudolabels marked as reliable and unreliable – reliable pseudolabels have an accuracy of **86.2**%, whereas unreliable ones have a low accuracy of **19.1**%; further these statistics are stable over the course of training.

Next, we evaluate the reliability measure by category. For each category, we report i) *precision* of reliability with respect to correctness (when a pixel prediction is reliable, how often is it actually correctly classified?), and ii) precision of unreliability with respect to incorrectness. We find that unreliable predictions are highly correlated with being incorrect across categories, which explains the effectiveness of excluding them from training. However, the precision of the reliability measure is significantly higher for head categories (*e.g.* road, building, car) than for the tail (*e.g.* bicycle, bus).

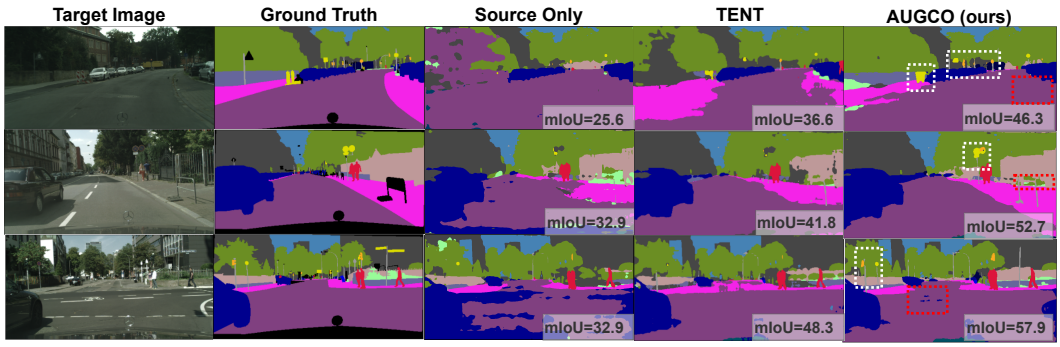

Figure 2: Qualitative segmentation results of the source model, TENT [6], and AUGCO. White boxes highlight categories recovered by AUGCO, whereas red boxes show failure cases.

References

[1] K. Saenko, B. Kulis, M. Fritz, and T. Darrell, "Adapting visual category models to new domains," in *European conference on computer vision*, pp. 213–226, Springer, 2010.

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

[3] M. Long, Y. Cao, J. Wang, and M. Jordan, "Learning transferable features with deep adaptation networks," in *International Conference on Machine Learning*, pp. 97–105, 2015.

[4] 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*, pp. 1989–1998, 2018.

[5] T.-H. Vu, H. Jain, M. Bucher, M. Cord, and P. Pérez, "Advent: Adversarial entropy minimization for domain adaptation in semantic segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2517–2526, 2019.

[6] D. Wang, E. Shelhamer, S. Liu, B. Olshausen, T. Darrell, U. Berkeley, and A. Research, "Tent: Fully test-time adaptation by entropy minimization," in *International Conference on Learning Representations*, vol. 4, p. 6, 2021.

[7] Y. Grandvalet, Y. Bengio, *et al.*, "Semi-supervised learning by entropy minimization.," in *CAP*, pp. 281–296, 2005.

[8] C. Chen, W. Xie, W. Huang, Y. Rong, X. Ding, Y. Huang, T. Xu, and J. Huang, "Progressive feature alignment for unsupervised domain adaptation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 627–636, 2019.

[9] X. Jiang, Q. Lao, S. Matwin, and M. Havaei, "Implicit class-conditioned domain alignment for unsupervised domain adaptation," in *International Conference on Machine Learning*, pp. 4816–4827, PMLR, 2020.

[10] V. Prabhu, S. Khare, D. Kartik, and J. Hoffman, "Sentry: Selective entropy optimization via committee consistency for unsupervised domain adaptation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 8558–8567, 2021.

[11] S. Tan, X. Peng, and K. Saenko, "Class-imbalanced domain adaptation: An empirical odyssey," in *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, September 2020.

[12] J. Snoek, Y. Ovadia, E. Fertig, B. Lakshminarayanan, S. Nowozin, D. Sculley, J. Dillon, J. Ren, and Z. Nado, "Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift," in *Advances in Neural Information Processing Systems*, pp. 13969–13980, 2019.

[13] B. Li, Y. Wang, T. Che, S. Zhang, S. Zhao, P. Xu, W. Zhou, Y. Bengio, and K. Keutzer, "Rethinking distributional matching based domain adaptation," *arXiv preprint arXiv:2006.13352*, 2020.

[14] S. R. Richter, V. Vineet, S. Roth, and V. Koltun, "Playing for data: Ground truth from computer games," in *European conference on computer vision*, pp. 102–118, Springer, 2016.

[15] 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 conference on computer vision and pattern recognition*, pp. 3213–3223, 2016.

[16] 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 conference on computer vision and pattern recognition*, pp. 3234–3243, 2016.

[17] C. Sakaridis, D. Dai, and L. V. Gool, "Guided curriculum model adaptation and uncertainty-aware evaluation for semantic nighttime image segmentation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 7374–7383, 2019.

[18] Z. Nado, S. Padhy, D. Sculley, A. D'Amour, B. Lakshminarayanan, and J. Snoek, "Evaluating prediction-time batch normalization for robustness under covariate shift," *arXiv preprint arXiv:2006.10963*, 2020.

[19] Y. Liu, W. Zhang, and J. Wang, "Source-free domain adaptation for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2021.

[20] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation," *arXiv preprint arXiv:1706.05587*, 2017.

[21] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.