Supplementary Material for Contrast, Stylize and Adapt: Unsupervised Contrastive Learning Framework for Domain Adaptive Semantic Segmentation

The supplementary material is organized as follows: in Sec. A we detail all the losses used for training our method. In Sec. B we provide additional discussion on the qualitative results.

A. Additional Training Details

Due to lack of space we provide additional training losses that are used to train CONFETI in the supplementary material. The weighted prototype estimation (described in Sec. 3.2.1 of the main paper) is summarized in Algo. 1.

Class Activation Maps. To recall, in order to obtain the class activation maps (CAM) we learn the linear layer $\mathbf{w} \in \mathbb{R}^{K \times D}$, where K and D are the number of classes and channel dimension of the latent features, respectively. The predicted probability scores are obtained by first applying global average pooling on the features obtained from the backbone f_b, followed by projecting them with the linear layer as:

$$\hat{p}_c = \sigma\Big(\frac{1}{H'W'}\sum_{d=1}^{D}\mathbf{w}_{c,d}\sum_{j=1}^{H'\times W'}\mathbf{f}_{d,j}\Big) \tag{A1}$$

where $\sigma(\cdot)$ is the sigmoid function.

Given, objects from multiple classes can be present in an image, we use a multi-label classification loss to train the network. In details, we construct the binary labels $y \in \{0,1\}^K$, from the ground truth labels of the source and the pseudo-labels of the target domain, which indicate the existence of a class in an image. The network is then trained with a standard binary cross-entropy (BCE) loss as:

$$\mathcal{L}_{\text{CAM}} = -\frac{1}{K}\sum_{c=1}^{K} y_c \log \hat{p}_c + (1-y_c)\log(1-\hat{p}_c) \tag{A2}$$

Diversity Regularization Loss. The prototypical contrastive loss $\mathcal{L}_{\text{PCL}}$ (described in Sec. 3.2.1) can be prone to trivial solution where all the features are assigned to a single prototype. This can happen because of the absence of labels in the target domain and the possibly noisy and erroneous pseudo-label, especially in the presence of a learnable projection. We thus follow [3], and apply a regularization term to guarantee feature diversity:

$$\mathcal{L}_{\text{Reg}} = \frac{1}{K\log K}\sum_{c=1}^{K}\frac{\exp(\overline{\mathbf{v}}\cdot\mathbf{p}_c/T)}{\sum_{k=1}^{K}\exp(\overline{\mathbf{v}}\cdot\mathbf{p}_k/T)} \tag{A3}$$

where $\overline{\mathbf{v}} = \frac{1}{|B||H'W'|}\sum_{i=1}^{B}\sum_{j=1}^{H'\times W'}\mathbf{v}_{i,j}$ is the mean feature computed from a mini-batch of size $|B|$ and $\mathbf{v}_{i,j}$ is the projected features from the j^{th} pixel location in the i^{th} image.

Algorithm 1 Weighted prototype estimation

Input: CAM $\mathbf{M}$, projected features $\mathbf{v}_j$, set of pixel locations $\mathcal{N}$, semantic categories K, momentum update parameter γ
Output: Prototypes $\{\mathbf{p}_c\}_{c=1}^{K}$
for c=1:K **do**
  Select features $\mathbf{v}_j$ belongs to class c
  Select N features with highest CAM score
  $\mathbf{p}'_c = \frac{\sum_{j\in\mathcal{N}_c}\mathbf{M}_{c,j}\mathbf{v}_j}{\sum_{j'\in\mathcal{N}_c}\mathbf{M}_{c,j'}}$
  $\mathbf{p}_c \leftarrow \gamma\mathbf{p}_c + (1-\gamma)\mathbf{p}'_c$
  $\mathbf{p}_c \leftarrow \frac{\mathbf{p}_c}{\|\mathbf{p}_c\|_2}$
end for
return $\{\mathbf{p}_c\}_{c=1}^{K}$

Style Transfer Losses. Besides the patch-based NCE loss for style transfer (described in Sec. 3.2.2), we also use the adversarial loss to train the generator $\mathcal{G}$ and the discriminator $\mathcal{E}$. The adversarial loss encourages the stylized source images to be visually similar to the target distribution, and is given as:

$$\begin{aligned}\mathcal{L}_{\text{GAN}} = {} & \mathbb{E}_{X^{\text{T}}\sim\mathcal{D}_{\text{T}}}\log\mathcal{E}(X^{\text{T}}) \\ & + \mathbb{E}_{X^{\text{S}}\sim\mathcal{D}_{\text{S}}}(1-\log\mathcal{E}(\mathcal{G}(X^{\text{S}})))\end{aligned} \tag{A4}$$

CUT Architecture. The CUT [2] adopts an encoder-decoder architecture for its generator $\mathcal{G}$. The first half of $\mathcal{G}$ is regarded as the encoder $\mathcal{G}_{enc}$ and the second half is the decoder $\mathcal{G}_{dec}$. Thus, the generator is the composition $\mathcal{G}_{enc}\circ\mathcal{G}_{dec}$.

Given an image X^{S}, the L layers features are extracted from $\mathcal{G}_{enc}$ for computing $\mathcal{L}_{PatchNCE}$. Naturally, the features from the deeper layers in the encoder corresponds to a

larger patch due to the increasing receptive field. At layer l, the features $\{\mathbf{z}_{i,l}\}_{i=1}^{N}$ is produced as:

$$\mathbf{z}_{i,l}^{S} = H_l(\mathcal{G}_{enc,l}(X^{\mathrm{S}}))_i \tag{A5}$$

where H_l is a 2-layer MLP followed by a ℓ_2 normalization. Similarly, the stylized features $\{\hat{\mathbf{z}}_{i,l}^{S\to\mathrm{T}}\}_{i=1}^{N}$ is obtained in the same manner with the stylized image $\hat{X}^{S\to\mathrm{T}} = \mathcal{G}(X^{\mathrm{S}})$.

The consistency in the translation is guarantee by the $\mathcal{L}_{PatchNCE}$ computed with properly chosen positive and negative pairs. Since the patches in the same location should have similarity before and after the stylization, for $\hat{\mathbf{z}}_{i,l}^{S\to\mathrm{T}}$, the positive patch $\mathbf{z}_{i,l,+}^{\mathrm{S}} = \mathbf{z}_{i}^{\mathrm{S}}$and the negative patches $\{\mathbf{z}_{i,l,-}^{\mathrm{S}}\} = \{\mathbf{z}_{k,l}^{\mathrm{S}}\}_{k\in\{1\ldots N\}/i}$. Finally, the PatchNCE loss is computed as:

$$\mathcal{L}_{PatchNCE} = \sum_{l=1}^{L}\sum_{i=1}^{N} \ell_{PatchNCE}(\hat{\mathbf{z}}_{i,l}^{\mathrm{S}\to\mathrm{T}}, \mathbf{z}_{i,l,+}^{\mathrm{S}}, \{\mathbf{z}_{i,l,-}^{\mathrm{S}}\}) \tag{A6}$$

Overall Training Objective. The final training objective is given as:

$$\begin{aligned}\mathcal{L}_{\text{Joint}} =& \mathcal{L}_{\text{CE}} + \lambda_{\text{PCL}}\mathcal{L}_{\text{PCL}} + \lambda_{\text{CAM}}\mathcal{L}_{\text{CAM}} + \lambda_{\text{Reg}}\mathcal{L}_{\text{Reg}} + \\ & \lambda_{\text{style}}(\mathcal{L}_{\text{GAN}} + \mathcal{L}_{\text{PatchNCE}} + \mathcal{L}_{\text{SC}})\end{aligned} \tag{A7}$$

where λ_{PCL}, λ_{CAM}, λ_{Reg}, λ_{style} are the weighing factors.

B. Qualitative Analysis

As shown in Fig. 5 our method performs better on several difficult situations than [1] and [3]. Our method better classify the same objects in the same category correctly, such as the truck in the first row and the fence and wall in the third row. CONFETI also outperforms other methods on small and fine objects, for instance the poles and traffic lights in the second row. In particular, CONFETI is able to distinguish better similar objects, for instance, in the example of the first row, our methods classified correctly the truct while other methods confused it with the sky. Another example is shown in the second row, the bikes and motorbikes are easy to be confused, yet CONFETI gives a better segmentation at the junction of two similar objects.

References

[1] Zhengkai Jiang, Yuxi Li, Ceyuan Yang, Peng Gao, Yabiao Wang, Ying Tai, and Chengjie Wang. Prototypical contrast adaptation for domain adaptive semantic segmentation. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXIV*, pages 36–54. Springer, 2022. 2

[2] Taesung Park, Alexei A Efros, Richard Zhang, and Jun-Yan Zhu. Contrastive learning for unpaired image-to-image translation. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IX 16*, pages 319–345. Springer, 2020. 1

[3] Binhui Xie, Shuang Li, Mingjia Li, Chi Harold Liu, Gao Huang, and Guoren Wang. Sepico: Semantic-guided pixel contrast for domain adaptive semantic segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023. 1, 2