The supplementary material provides further details of the proposed approach, additional quantitative results, ablations, and implementation details. We have released our code on our project page: https://val.cds.iisc.ac.in/C-SFTrans/. The remainder of the supplementary material is organized as follows:

- Section 1: Proposed Approach (Table 1, Algorithm 1)
- Section 2: Implementation Details
  - Datasets (Section 2.1)
  - Style augmentations (Section 2.2)
  - Experimental Settings (Section 2.3)
- Section 3: Additional Comparisons (Tables 2)
- Section 4: Ablations on target-side goal task training (Tables 3, 4, and 5)

1. Proposed Approach

We summarize all the notations used in the paper in Table 1. The notations are grouped into the following 6 categories - models, transformers, datasets, spaces, losses, and criterion. Our proposed method has been outlined in Algorithm 1

### Target adaptation losses. We use the Information Maximization loss [8] that consists of entropy loss $L_{ent}$ and diversity loss $L_{div}$.

\[
L_{ent} = -\mathbb{E}_{x \in \mathcal{X}} \sum_{k=1}^{K} \delta_k(f_g(z_c)) \log \delta_k(f_g(z_c)) \tag{1}
\]

\[
L_{div} = \sum_{k=1}^{K} \hat{p}_k \log \hat{p}_k = KL(\hat{p}, \frac{1}{K}) \tag{2}
\]

where $\delta_k(b) = \frac{\exp(b_k)}{\sum_i \exp(b_i)}$ is the $k$th element of softmax output of $b \in \mathbb{R}^K$. The entropy loss $L_{ent}$ ensures that the model predicts more confidently for a particular label and the diversity loss $L_{div}$ ensures that the predictions are well-balanced across different classes. We optimize all parameters of the transformer backbone $h$, except the non-causal heads $h_{ln}$.

### Table 1. Notation Table

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$f$</td>
<td>Backbone feature extractor</td>
</tr>
<tr>
<td>$f_g$</td>
<td>Goal task classifier</td>
</tr>
<tr>
<td>$f_n$</td>
<td>Style classifier</td>
</tr>
<tr>
<td>$z_c$</td>
<td>Class token of last layer</td>
</tr>
<tr>
<td>$z_n$</td>
<td>Style token of last layer</td>
</tr>
<tr>
<td>$N_p$</td>
<td>Number of patch tokens</td>
</tr>
<tr>
<td>$h_l^n$</td>
<td>Non-causal heads of layer $l$</td>
</tr>
<tr>
<td>$h_l'$</td>
<td>All attention-heads of layer $l$</td>
</tr>
<tr>
<td>$h_l \backslash h_l^n$</td>
<td>Causal heads of layer $l$</td>
</tr>
<tr>
<td>$W_K$</td>
<td>Key weights</td>
</tr>
<tr>
<td>$W_Q$</td>
<td>Query weights</td>
</tr>
<tr>
<td>$W_V$</td>
<td>Value weights</td>
</tr>
<tr>
<td>$D_s$</td>
<td>Labeled source dataset</td>
</tr>
<tr>
<td>$D_t$</td>
<td>Unlabeled target dataset</td>
</tr>
<tr>
<td>$a_i$</td>
<td>Augmentation function $i$</td>
</tr>
<tr>
<td>$D_i[s]$</td>
<td>$i$th augmented source dataset</td>
</tr>
<tr>
<td>$D_i[t]$</td>
<td>$i$th augmented target dataset</td>
</tr>
<tr>
<td>$(x_i, y_i)$</td>
<td>Labeled source sample</td>
</tr>
<tr>
<td>$(x_i[s], y_i)$</td>
<td>Augmented source sample</td>
</tr>
<tr>
<td>$x_i$</td>
<td>Unlabeled target sample</td>
</tr>
<tr>
<td>$(x_i[t], y_i)$</td>
<td>Target augmented sample</td>
</tr>
<tr>
<td>$x$</td>
<td>Clean input sample</td>
</tr>
<tr>
<td>$x_{SCI}$</td>
<td>Style Characterizing Input</td>
</tr>
<tr>
<td>$\mathcal{X}$</td>
<td>Input space</td>
</tr>
<tr>
<td>$\mathcal{C}$</td>
<td>Label set for goal task</td>
</tr>
<tr>
<td>$\mathcal{Z}_c$</td>
<td>Class token feature space</td>
</tr>
<tr>
<td>$\mathcal{Z}_n$</td>
<td>Style token feature space</td>
</tr>
<tr>
<td>$\mathcal{Z}<em>1, \ldots, \mathcal{Z}</em>{N_p}$</td>
<td>Patch tokens</td>
</tr>
<tr>
<td>$\mathcal{L}_{style}$</td>
<td>Style Classification loss</td>
</tr>
<tr>
<td>$\mathcal{L}_{cls}$</td>
<td>Task Classification loss</td>
</tr>
<tr>
<td>$\mathcal{L}_{ent}$</td>
<td>Entropy loss</td>
</tr>
<tr>
<td>$\mathcal{L}_{div}$</td>
<td>Diversity loss</td>
</tr>
<tr>
<td>$\beta_1$</td>
<td>Importance weight for style feature</td>
</tr>
<tr>
<td>$\beta_2$</td>
<td>Importance weight for task feature</td>
</tr>
<tr>
<td>$CIS_i$</td>
<td>Causal Influence Score for head $i$</td>
</tr>
<tr>
<td>$\tau$</td>
<td>Threshold</td>
</tr>
</tbody>
</table>

*Equal Contribution*
\[
\min_{h^i \setminus h^t, f_g} \mathbb{E}_{D_t} [\mathcal{L}_{ent} + \mathcal{L}_{div}] 
\] (3)

**Pseudo-labeling.** We use the clustering method of SHOT [8] to obtain pseudo-labels. At first, the centroid of each class is calculated using the following,

\[
c_k = \frac{\sum_{z_c \in \mathcal{X}} \delta_k(f_g(z_c))z_c}{\sum_{z_c \in \mathcal{X}} \delta_k(f_g(z_c))} 
\] (4)

The closest centroid is chosen as the pseudo-label for each sample using the following cosine distance formulation,

\[
g_c = \arg \min_k D_c(z_c, c_k) 
\] (5)

where \(D_c\) denotes the cosine-distance in the class-token feature space \(Z_c\) between a centroid \(c_k\) and the input sample features \(z_c\). In successive iterations, the centroids keep updating and the pseudo-labels get updates with respect to the new centroids.

**Attention heads in vision transformers.** A ViT takes an image \(x\) as input of size \(H \times W \times C\) and divides it into \(N_p\) patches of size \((P, P)\) each. The total number of patches is \(N_p = \frac{H \times W}{P \times P}\). In every layer, \(i\), a head \(h^i_l\) takes the patches as input and transforms a patch into \(K, Q, V\) using the weights \(W_K, W_Q, W_V\), respectively. The self-attention [16] is computed as follows,

\[
h^i_l = \text{Softmax} \left( \frac{QK^T}{\sqrt{d_k}} \right) V 
\] (6)

where \(d_k\) represents the dimension of the keys/queries.

### 2. Implementation details

#### 2.1. Datasets

We use four standard object classification benchmarks for DA to evaluate our approach. The **Office-Home** dataset [17] consists of images from 65 categories of everyday objects from four domains - Art (Ar), Clipart (Cl), Product (Pr), and Real World (Rw). Office-31 [12] is a simpler benchmark containing images from 31 categories belonging to three domains of objects in office settings - Amazon (A), Webcam (W), and DSLR (D). **VisDA** [11] is a large-scale benchmark containing images from two domains - 152,397 synthetic source images and 55,388 real-world target images. Lastly, **DomainNet** [10] is the largest and the most challenging dataset due to severe class imbalance and diversity of domains. It contains 345 categories of objects from six domains - Clipart (clp), Infograph (inf), Painting (pmt), Quickdraw (qdr), Real (rel), Sketch (skt).

---

**Algorithm 1** C-SFTrans Training Algorithm

**Vendor-side training**

1. **Input:** Let \(D_s\) be the source data, \(D_{sty}\) be the style dataset, ImageNet pre-trained DeiT-B backbone \(h\) from [19], randomly initialized goal classifier \(f_g\) and randomly initialized style classifier \(f_s\).

**Non-causal attention heads selection**

\([\therefore]\) Fig. 3A (main paper)

2. for \(\text{iter} < \text{MaxTaskIters}\) do:
   3. Sample batch from \(D_s\)
   4. Construct \(x_{SCI}\) from \(x_i\)
   5. Compute \(A^i\) using Eq. 3 (main paper)
   6. Compute \(L_{cls}\) using Eq. 4 (main paper)
   7. update \(\beta_1, \beta_2\) for head \(j\) by minimizing \(L_{cls}\)
   8. end for

**Goal task training**

\([\therefore]\) Fig. 3B (main paper)

9. for \(\text{iter} < \text{MaxIter}\) do:
   10. for \(\text{iter} < \text{MaxTaskIters}\) do:
       11. Sample batch from \(D_s\)
       12. Compute \(L_{cls}\) using Eq. 6 (main paper)
       13. update \(\theta_{h^t}, \theta_{h^t}, \theta_f\) by minimizing \(L_{cls}\)
   14. end for

**Style classifier training**

\([\therefore]\) Fig. 3B (main paper)

15. for \(\text{iter} < \text{MaxDomainIters}\) do:
   16. Sample batch from \(D_{sty}\)
   17. Compute \(L_{dom}\) using Eq. 1 (main paper)
   18. update \(\theta_{h^t}, \theta_f\) by minimizing \(L_{dom}\)
   19. end for

\([\therefore]\) The two steps are carried out alternatively

**Client-side training**

20. end for

**Goal Task Training**

\([\therefore]\) Fig. 3B (main paper)

21. **Input:** Target data \(D_t\), Target augmented DRI data \(D_t^{qdr}\), source-side pretrained backbone \(h\), goal classifier \(f_g\) and domain classifier \(f_d\).

22. for \(\text{iter} < \text{MaxIter}\) do:

**Style classifier training**

\([\therefore]\) Fig. 3B (main paper)

23. for \(\text{iter} < \text{MaxTaskIters}\) do:
   24. Sample batch from \(D_t\)
   25. Compute \(L_{im} + L_{div}\) using Eq. 1, 2 (suppl.)
   26. update \(\theta_{h^t}, \theta_f\) by minimizing \(L_{im} + L_{div}\)
   27. end for

28. for \(\text{iter} < \text{MaxDomainIters}\) do:
   29. Sample batch from \(D_{sty}\)
   30. Compute \(L_{dom}\) using Eq. 1 (main paper)
   31. update \(\theta_{h^t}, \theta_f\) by minimizing \(L_{dom}\)
   32. end for

\([\therefore]\) The two steps are carried out alternatively

33. end for
2.2. Style augmentations

We construct novel stylised images using 5 label-preserving augmentations on the original clean images to enable non-causal factor alignment during the training process. The augmentations are as follows:

1. FDA augmentation: We use the FDA augmentation [20] to generate stylized images based on a fixed set of style images [3]. In this augmentation, a given input image is stylized by interchanging the low-level frequencies between the FFTs of the input image and the reference style image.

2. Weather augmentations: We employ the frost and snow augmentations from [5] to simulate the weather augmentation. Specifically, we use the lowest severity of frost and snow (severity = 1) to augment the input images.

3. AdaIN augmentation: AdaIN [3] uses a reference style image to stylize a given input image by altering the feature statistics in an instance normalization (IN) layer [15]. We use the same reference style image set as in FDA, and set the augmentation strength to 0.5.

4. Cartoon augmentation: We employ the cartoonization augmentation from [5] to produce cartoon-style images with reduced texture from the input.

5. Style augmentation: We use the style augmentation from [4] that augments an input image through random style transfer. This augmentation alters the texture, contrast and color of the input while preserving its geometric features.

2.3. Experimental settings

In all our experiments, we use the Stochastic Gradient Descent (SGD) optimizer [6] with a momentum of 0.9 and batch size of 64. We follow [8] and use label smoothing in the training process. For the source-side, we train the goal task classifier for 20 epochs, and the style classifier until it achieves 80% accuracy. On the target-side, we train the goal task classifier for 2 epochs, and use the same criteria for the style classifier as the source-side. The first 5 epochs of the source-side training are used for warm-up with a warm-up factor of 0.01. On the source-side, we use a learning rate of $8 \times 10^{-4}$ for the VisDA dataset, and $8 \times 10^{-3}$ for the remaining benchmarks. For the target-side goal task training, we use a learning rate of $5 \times 10^{-5}$ for VisDA, $2 \times 10^{-3}$ for DomainNet, and $8 \times 10^{-3}$ for the rest. Our proposed method comprises an alternate training mechanism where the goal task training and style classifier training are done alternatively for a total of 25 rounds, which is equivalent to 50 epochs of target adaptation in [8]. For comparisons, we implement the source-free methods DIPE [18] and Feature Mixup [7] by replacing the backbone with DeiT-B. While CDTrans [19] uses the entire domain for training and evaluation with the DomainNet dataset, we follow the setup of [13] to ensure fair comparisons. We train on the train split and evaluate on the test split of each domain.

3. Additional comparisons

We present additional comparisons with the DomainNet benchmark in Table 2. Our method achieves the best results.
We select a set of non-causal heads. The goal task epochs are varied from 1 to 5.

Table 4. Ablation study for the three components of the target-side goal task training. SSPL denotes self-supervised pseudo-labeling.

<table>
<thead>
<tr>
<th>Method</th>
<th>(L_{\text{ent}})</th>
<th>(L_{\text{div}})</th>
<th>textitSSPL</th>
<th>Avg.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source-Only</td>
<td>\textit{x}</td>
<td>\textit{x}</td>
<td>76.4</td>
<td></td>
</tr>
<tr>
<td>C-SFTrans</td>
<td>\checkmark</td>
<td>\checkmark</td>
<td>\checkmark</td>
<td>79.7 (+5.7)</td>
</tr>
</tbody>
</table>

among the existing source-free prior arts and outperforms the source-free SHOT-B by 3.6%. We also observe that C-SFTrans surpasses the non-source-free method CDTrans by an impressive 5.5%.

4. Ablations on target-side goal task training

(a) Target-side goal task training loss. Table 4 shows the influence of the three loss terms in the target-side goal task training - entropy loss \(L_{\text{ent}}\), diversity loss \(L_{\text{div}}\) and self-supervised pseudo-labeling SSPL. We observe that using \(L_{\text{ent}}\) alone produces lower results even compared to the source baseline. On the other hand, using both \(L_{\text{ent}}\) and \(L_{\text{div}}\) significantly improves the performance, which highlights the importance of the diversity term \(L_{\text{div}}\). Finally, we obtain the best results when all three components are used together for target-side adaptation, further showing the significance of the pseudo-labeling step.

(b) Sensitivity analysis of alternate training. In our proposed method, we perform style classifier training and goal task training in an alternate fashion, i.e. the task classifier \(f_g\) is trained for a few epochs, followed by the training of the style classifier \(f_s\) until it reaches a certain accuracy threshold (empirically set to 80%). In Table 3, we show the effect of varying the number of epochs of the goal task training from 1 to 5, and observe the impact on the goal task accuracy during non-causal factor alignment. We observe that 2 epochs of goal task training achieves the optimal balance between target accuracy and training effort. We observe that just a single epoch of task classifier training negatively impacts the goal task performance. While 3 epochs achieves the best performance, it involves significant training effort for merely 0.5% improvement in the task accuracy. Therefore, 2 epochs of goal task training achieves the optimal balance between target accuracy and training effort.

(c) Selection of non-causal heads. We select a set of non-causal attention heads based on their Causal Influence Score (CIS). We sort the CIS in descending order and select the top \(\lambda\%\) of heads satisfying the condition \(CIS > \tau\). In Table 5, we present the effect of altering this hyperparameter \(\lambda\) on the overall performance. We observed that with a lower value of \(\lambda\), the pathways formed by non-causal heads do not adequately extract and learn the non-causal factors, which consequently hinders the domain-invariant alignment and leads to non-optimal task performance. Similarly, increasing \(\lambda\) too much reduces the ability of the network to learn causal factors and leads to lower performance. Overall, our approach is not very sensitive towards this hyperparameter.

Table 6 demonstrates that fewer augmentations for the style classifier significantly deteriorate the adaptation performance in comparison to the full set of augmentations. This indicates that a more complex style classification task better facilitates the non-causal factor alignment. However, due to the scarcity of more complex augmentations, we use the six outlined in Sec. 2.2.

Table 5. Sensitivity analysis on non-causal heads (%) for Single-Source DA on 4 settings of Office-Home

<table>
<thead>
<tr>
<th>(\lambda)</th>
<th>(Ar \rightarrow Cl)</th>
<th>(Cl \rightarrow Pr)</th>
<th>(Pr \rightarrow Rw)</th>
<th>(Rw \rightarrow Ar)</th>
<th>Avg.</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>70.2</td>
<td>86.7</td>
<td>87.5</td>
<td>82.4</td>
<td>81.7</td>
</tr>
<tr>
<td>0.2</td>
<td>70.0</td>
<td>86.8</td>
<td>87.6</td>
<td>82.5</td>
<td>81.7</td>
</tr>
<tr>
<td>0.3</td>
<td>70.3</td>
<td>86.9</td>
<td>87.7</td>
<td>82.6</td>
<td>81.9</td>
</tr>
<tr>
<td>0.4</td>
<td>70.2</td>
<td>86.5</td>
<td>87.2</td>
<td>82.1</td>
<td>81.5</td>
</tr>
</tbody>
</table>

(d) Effect of augmentations. Table 6 demonstrates that fewer augmentations for the style classifier significantly deteriorate the adaptation performance in comparison to the full set of augmentations. This indicates that a more complex style classification task better facilitates the non-causal factor alignment. However, due to the scarcity of more complex augmentations, we use the six outlined in Sec. 2.2.

References


