A PIPELINE OF THE WHOLE FRAMEWORK

Finally, we list the algorithm implementation pipeline of the whole framework both on the source and target domain in the following Algorithm 1 and Algorithm 2.

Algorithm 1 Pipeline of AudoFormer on the source domain.

Input: Source data $x_i^s \in \mathcal{X}_i^s$ with $\mathcal{F}_s$, $\mathcal{C}_s$ and $\mathcal{G}_s$ and iteration T_s.
Output: Trained model of source domain $\mathcal{F}_s$, $\mathcal{C}_s$ and $\mathcal{G}_s$.
Initializtion: Initialize $\mathcal{F}_t$ with pre-trained parameters.

for *epoch = 1* to T_s **do**
  Step 1. Aggregate the features of each layer to produce global feature maps $\hat{x}_a$ of by EMA.
  Step 2. Extract the feature representations $\hat{f}_{a,s}$ and logits $\hat{z}_{a,s}$ from multilevel global attention fusion by ADM block $\mathcal{G}_s$.
  Step 3. Extract feature representations f_s and logits z_s $\mathcal{F}_s$ by source classifier $\mathcal{C}_s$.
  Step 4. Train the source domain by supervised labels.
  Step 5. Distill knowledge to optimize the ADM by Eq. 4
end for

Algorithm 2 Pipeline of AudoFormer on the target domain.

Input: A pre-trained model $\mathcal{F}_s$, $\mathcal{C}_s$ and $\mathcal{G}_s$ on the source domain, Target domain $\mathcal{X}_t$, Hyper-parameters α, β, iteration T_t.
Output: Trained model for target domain $\mathcal{F}_t$, $\mathcal{C}_t$ and $\mathcal{G}_t$.
Initializtion: Initialize $\mathcal{F}_t$ with parameters pre-trained on the source domain, and freeze the classifier layer $\mathcal{C}_t$ and $\mathcal{G}_t$.

for *epoch = 1* to T_t **do**
  for *i = 1* to n_t **do**
    Step 1. Initiate the memory bank with full representations.
    Step 2. Extract feature representations $\hat{f}_{a,t}$ and logits $\hat{z}_{a,t}$ by $\mathcal{G}_t$ to construct an auxiliary domain.
    Step 3. Extract feature representations f_t from output features and logits z_t by $\mathcal{F}_t$ and $\mathcal{C}_t$.
    Step 4. Dynamic centroid evaluation to determine pseudo-labels of both domains.
    Step 5. Distinguish easy and hard samples by consistent strategies and store in the memory bank $\mathcal{M}_e$, $\mathcal{M}_h$, respectively.
    Step 6. Re-evaluate hard samples by consistent neighbors.
  end for
  Step 7. Optimize the whole model by information maximization with loss function $\mathcal{L}_{im}$ (i.e., Eq. 10).
  Step 8. Train the target domain with pseudo-labels by consistent self-supervised training with $\mathcal{L}_{cst}$ (i.e., E.q. 6).
  Step 9. Align the target-specific domain to source-like domain by $\mathcal{L}_{cmk}$ (i.e., Eq. 9).
end for

B ABLATION STUDY

B.1 EFFECT OF ADM ON SOURCE DOMAIN

Since the ADM block is introduced for our AudoFormer. Thus, we first validate the effect of ADM block for supervised training and 'Source-only' adaptation in the source domain, as shown in Tab. 4. We verify the effectiveness of the ADM block from two aspects. First, we verify the model by pure ViT-base (i.e., w/o ADM) and our AudoFormer with the distilled training (i.e., w ADM). When exploiting the pure ViT-base model to train both three datasets by supervised learning. In the Office-31 dataset, we can obtain an average score of 95.9%, while exploiting AudoFormer i.e., adding the ADM block to the ViT-base model, we can achieve 96.5% on average, roughly a 0.6% improvement. On both three datasets, AudoFormer can improve the average result of the model by 1.5%. Based on the above two forms of architecture, we apply them to the target domain without exploiting any

adaptation strategies, i.e., 'Soure-only' adaptation. The score of 'Source-only' by AudoFormer on three datasets can significantly be improved by 1.2% on average. We believe AudoFormer enhances the ability of ViT to learn the spatial context from the global features, thus enhancing the inductive bias to some extent.

Table 4: Effect of MAD block for Supervised training and 'Source-only' adaptation in the source domain.

DataSets	Supervised		Soure-only	
	ViT-base	AudoFormer	ViT-base	AudoFormer
Office-31	95.9	96.5	84.6	86.2
Ofiice-Home	86.5	89.3	72.7	73.6
VISDA-C	98.5	99.5	64.9	66.1
Avg.	93.6	95.1	74.1	75.3

B.2 Effect of Strategies

To validate the effectiveness of the different components of our method, we exploit the ViT-base backbone to perform ablation experiments on the three datasets for each component. The corresponding results are reported in Tab. 5. In the first row, training the original model on the source domain without any adaptation, we obtain 86.2%, 73.6%, and 66.1%. If we directly utilize the information maximization loss $\mathcal{L}_{im}$ to optimize the model (i.e., Baseline), we can achieve 91.3%, 79.4%, and 85.6%, respectively. Moreover, by utilizing the $\mathcal{L}_{cst}$ to supervise the alignment of the source and target domain, the average result can be improved by 1.4%, which proves that our dynamic consistency strategy can effectively distinguish inconvenient features, so as to distinguish between easy samples and hard samples and improve the effect of the model through multiple consistency.

Finally, both $\mathcal{L}_{cst}$ and $\mathcal{L}_{cmk}$ are employed to verify our approach, which improves the baseline by about 2.1%. This demonstrates that both the proposed three components are critical for AudoFormer to perform well on SFDA.

Table 5: The ablation study of our approaches exploited by ViT-backbone with different components. '+' denotes the add operation.

Method	Office-31	Office-Home	VISDA-C	Avg.
Source-only	86.2	73.6	66.1	75.3
Baseline	91.3	79.4	85.6	85.4
+ $\mathcal{L}_{cst}$	92.5	81.1	86.7	86.8
+ $\mathcal{L}_{cst}$ + $\mathcal{L}_{cmk}$	93.0	81.7	87.8	87.5

B.3 Visualization

To explore the alignment effect of the final source domain and target domain, we conduct visualization experiments from two aspects: the attention maps by grad-cam Gildenblat & contributors (2021) and feature alignment by t-SNE.

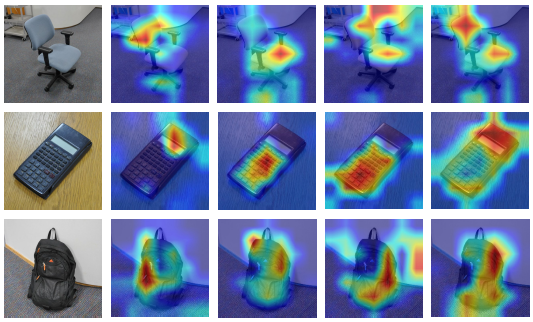
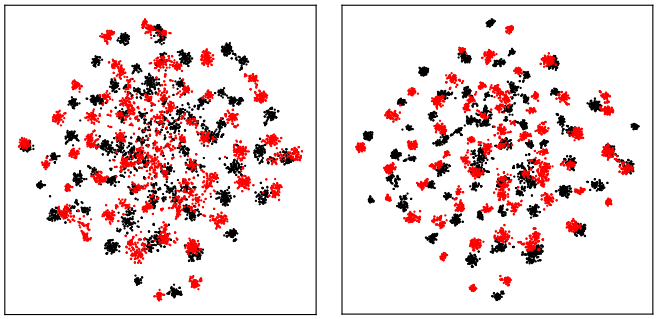


Figure 3: Attention maps of images about desk chair, calculator, and black package in the Office-31 dataset.



(a) Before Adaptation (b) After Adaptation

Figure 4: t-SNE visualization for domain adaptation on Office-Home (A $\rightarrow$ P), Red denotes the source domains, while black denotes target domains.

Attention Map: In the ViT model, the attention maps represent the degree of attention to the target region. Normally, the hotter the color of the area, the higher the attention to the area. According to attention maps shown in Figure 3, the original 'Source-only' can hardly pay attention to the main regions. Although ViT-base G-SFDA and SHOT++ can focus on the main objects, it is not comprehensive enough. Compared with the former methods, our proposed method AudoFormer can accurately capture discriminative region features. For example, the 'Source only' method only pays attention to the background of the images w.r.t. calculator, while our proposed method focuses on most areas of the target, which proves that the attention effect was effectively improved after alignment.

t-SNE: To demonstrate the effect of different methods on domain alignment, we utilize t-SNE to visualize the distributions of feature representations, which are obtained from the penultimate layer in both source and target domains. As can be seen from Figure 4(a), the sample features of the same category are more dispersed before adaptation. This might be due to the severe domain shift problem with source data. Benefiting from $\mathcal{L}_{cst}$ and $\mathcal{L}_{cmk}$ loss, after adaptation, we can observe that the category distance is significantly reduced and the sample categories are distributed more clearly.

C More Details about ADM and Attention

C.1 The Architecture of ADM block

To mitigate the effect of lacking inductive bias and obtain invariant feature representations, we make full use of the advantages of CNN to compensate for this deficiency. Therefore, we construct an ADM black to extract the feature representations $\hat{f}_a$ and logits $\hat{z}_a$ from the aggregated multilevel global features of intermediate layers. We first reshape the self-attention to feature map $\hat{x}_a \in \mathcal{R}^{\mathrm{D}\times\mathrm{H}\times\mathrm{W}}$. In our reshaped map, the hidden vector can be seen as a channel-wise feature, therefore, we exploit the convolution to downsample the feature map. As can be seen from Tab. 6, our ADM consists of three convolution layers. By downsampling operation, we can obtain the related context of attention features, which covers the major object. For example, the ViT-base model takes the 768 dimensional features as a hidden vector. To make the dimension consistent with the classifier, we extract 256 dimensional features and map the features to logits by the FC layer which consists of two linear layers. This module can be applied dynamically in ViT-base and DeiT-base models to assist the model generate the auxiliary domain.

Table 6: The architechure of ADM block.

Channel	Kernel	Component
$768 \rightarrow 512$	3×3, stride=2	Conv, BN, ReLU
$512 \rightarrow 256$	3×3, stride=1	Conv, BN, ReLU
$256 \rightarrow 256$	3×3, stride=1	Conv, BN, AvP
$256 \rightarrow C$	3×3, stride=1	Linear, ReLU, Linear

* C is the output dimension of the linear layer.

C.2 VISUALIZATION OF MULTILEVEL ATTENTION

To study whether the target is noticed or not by attention, we visualize heatmap attention feature maps of each layer by the grad-cam and calculate the final global attention visualized features by EMA. From Fig. 5, we can observe that the target objects are not completely covered by the attention maps in each layer. For example, in layers of 1st-4th, the target samples are barely focused on. And the others such as 5th-10th, etc., these layers can focus on different local features of targets. In the last 12th layer, the target object can be noticed to a large extent. Besides, we exploit the EMA to calculate the global attention maps, which can focus on the vast majority of the target regions and fully cover the front attention regions. This proves that the EMA method we adopted can effectively capture the global features of the object. Based on such feature maps, the ADM block can obtain more accurate feature representations.

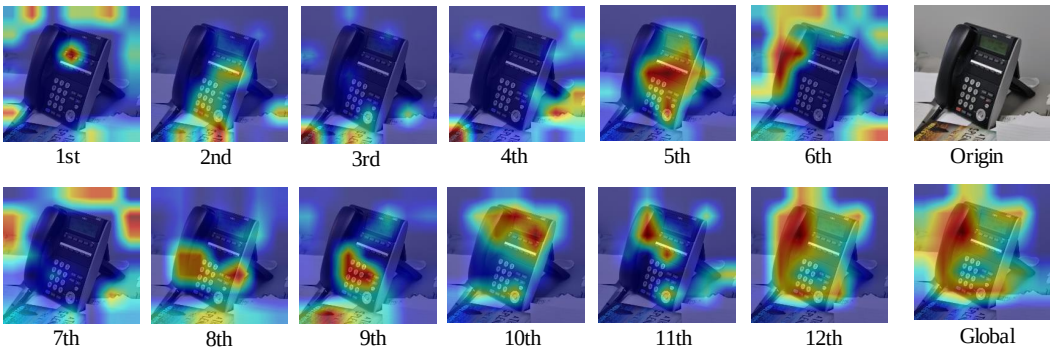
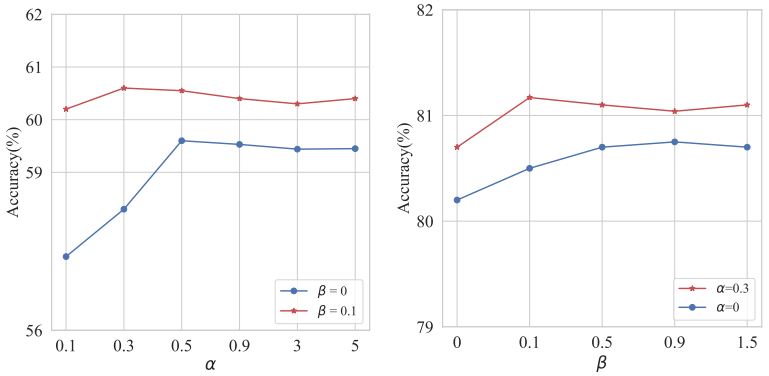


Figure 5: Attention maps of the intermediate layers in ViT-base model.

C.3 EFFECT OF HYPER-PARAMETERS

In our total loss function (Eq. 11), α and β are the major hyper-parameters for balancing the loss terms in our framework. To test their effect on the final performance, we conduct an experiment on the following two tasks, Ar→Cl and Pr→Ar. As depicted in Figure 6(a), our model is less sensitive to the change of α and β, and the results are significantly improved when α and β are larger than 0. For the Ar→Cl task, when β is set to 0.1, the discrepancy of α over the interval is only about 0.5%. While for the task Pr→Ar, the best performance is achieved when β is set to 0.1 and α is set to 0.3, as shown in Figure 6(b).



(a) α (b) β

Figure 6: Sensitivity of hyper-parameter α and β in specific tasks (Ar→Cl and Pr→Ar) on Office-Home Dataset.