Supplementary Materials for Learning Domain-Aware Detection Head with Prompt Tuning

Anonymous Author(s)
Affiliation
Address
`email`

A Additional Experimental Results

A.1 Effect of Prompt Design and Loss Function

In order to further verify the effectiveness of our proposed prompt and the validation of the objective function, we verified the performance of combinations of different prompt designs and loss functions. For convenience, we rewrite the loss function Eq.9, Eq.11 and Eq.12 as follows.

Given an image $I \in \mathcal{X}^s$ from the source domain, the generated regions and their class label are denoted as $\{\mathbf{r}_j^s\}_{j=1}^{N_r}$ and $\{y_j^s\}_{j=1}^{N_r}$. Following the prediction probability calculated by Eq.6, we denote the single-term loss computed with the source data as:

$$\mathcal{L}_{s,D} = \mathbb{E}_{\mathcal{X}^s}[-\frac{1}{N_r}\sum_{j=1}^{N_r}\log p(\hat{y} = y_j^s|\mathbf{r}_j^s, s, D)] \tag{14}$$

where $D \in \{\{s\}, \{t\}, \{s,t\}\}$ indicates probability is calculated with the class embedding generated by the source prompt $\{\mathbf{t}_i^s\}_{i=1}^K$, the target prompt$\{\mathbf{t}_i^t\}_{i=1}^K$, or both prompts $\{\mathbf{t}_i^s\}_{i=1}^K \cup \{\mathbf{t}_i^t\}_{i=1}^K$. With Eq.14, we can equivalently express the loss function described in Eq.9 as:

$$\mathcal{L}_s = \mathcal{L}_{s,\{s\}} + \mathcal{L}_{s,\{t\}} + \mathcal{L}_{s,\{s,t\}} \tag{15}$$

Given an image $I \in \mathcal{X}^t$ from the target domain, the generated regions and their pseudo label are denoted as $\{\mathbf{r}_j^t\}_{j=1}^{N_r}$ and $\{y_j^t\}_{j=1}^{N_r}$. Similar to Eq.14, the cross-entropy and information entropy are computed by:

$$\mathcal{L}_{t,\mathcal{D}} = \mathbb{E}_{\mathcal{X}^t}[-\frac{1}{N_r}\sum_{j=1}^{N_r}\mathbf{I}(p(\hat{y} = y_j^t|\mathbf{r}_j^t, t, \{t\}) \geq \tau)\log p(\hat{y} = y_j^t|\mathbf{r}_j^t, t, \mathcal{D})] \tag{16}$$

$$\mathcal{L}_{ent} = \mathbb{E}_{\mathcal{X}^t}[-\frac{1}{N_r}\sum_{j=1}^{N_r}\mathbf{I}(p(\hat{y} = y_j^t|\mathbf{r}_j^t, t, \{t\}) \geq \tau)p(\hat{y} = y_j^t|\mathbf{r}_j^t, t, \{t\})\log p(\hat{y} = y_j^t|\mathbf{r}_j^t, t, \{t\})] \tag{17}$$

With Eq.16 and Eq.17, the loss function in Eq.11 can also be rewritten as:

$$\mathcal{L}_t = \mathcal{L}_{t,\{t\}} + \mathcal{L}_{t,\{s,t\}} + \mathcal{L}_{ent} \tag{18}$$

Therefore, the total objective function is also denoted as:

$$\mathcal{L} = \mathcal{L}_s + \lambda\mathcal{L}_t = \mathcal{L}_{s,\{s\}} + \mathcal{L}_{s,\{t\}} + \mathcal{L}_{s,\{s,t\}} + \lambda[\mathcal{L}_{t,\{t\}} + \mathcal{L}_{t,\{s,t\}} + \mathcal{L}_{ent}] \tag{19}$$

To be more specific, we took the Cross-Weather adaptation scenario for validation. All the training configurations are the same as in Section 4.2 of the main text. We set the total number of tokens

Table 3: The influence (%) of loss design on Cross-Weather adaptation scenario Cityscapes→Foggy Cityscapes. Total number of tokens is set to 16. - stands for the prompt is not compatible with the loss. Without the historical prompt ensemble strategy.

Prompt Design	$\mathcal{L}_{s,\{s\}} + \mathcal{L}_{t,\{t\}}$	$\mathcal{L}_{s,\{t\}}$	$\mathcal{L}_{s,\{s,t\}} + \mathcal{L}_{t,\{s,t\}}$	$\mathcal{L}_{ent}$	mAP
A photo of a [class][domain]	-	-	-	-	52.6
$[\mathbf{v}_1^c][\mathbf{v}_2^c]...[\mathbf{v}_M^c][\mathbf{c}_i]$	✓	-	-	-	53.0
$[\mathbf{v}_1^c][\mathbf{v}_2^c]...[\mathbf{v}_M^c][\mathbf{c}_i][\mathbf{dl}_d]$	✓				53.5
$[\mathbf{v}_1^c][\mathbf{v}_2^c]...[\mathbf{v}_M^c][\mathbf{c}_i][\mathbf{dl}_d]$	✓	✓			53.8
$[\mathbf{v}_1^c][\mathbf{v}_2^c]...[\mathbf{v}_M^c][\mathbf{c}_i][\mathbf{dl}_d]$	✓	✓	✓		53.8
$[\mathbf{v}_1^c][\mathbf{v}_2^c]...[\mathbf{v}_M^c][\mathbf{c}_i][\mathbf{dl}_d]$	✓	✓	✓	✓	53.7
$[\mathbf{v}_1^d][\mathbf{v}_2^d]...[\mathbf{v}_N^d][\mathbf{c}_i][\mathbf{dl}_d]$	✓				52.9
$[\mathbf{v}_1^d][\mathbf{v}_2^d]...[\mathbf{v}_N^d][\mathbf{c}_i][\mathbf{dl}_d]$	✓	✓			53.1
$[\mathbf{v}_1^d][\mathbf{v}_2^d]...[\mathbf{v}_N^d][\mathbf{c}_i][\mathbf{dl}_d]$	✓	✓	✓		52.7
$[\mathbf{v}_1^d][\mathbf{v}_2^d]...[\mathbf{v}_N^d][\mathbf{c}_i][\mathbf{dl}_d]$	✓	✓	✓	✓	52.9
$[\mathbf{v}_1^c][\mathbf{v}_2^c]...[\mathbf{v}_M^c][\mathbf{v}_1^d][\mathbf{v}_2^d]...[\mathbf{v}_N^d][\mathbf{c}_i][\mathbf{dl}_d]$	✓				54.4
$[\mathbf{v}_1^c][\mathbf{v}_2^c]...[\mathbf{v}_M^c][\mathbf{v}_1^d][\mathbf{v}_2^d]...[\mathbf{v}_N^d][\mathbf{c}_i][\mathbf{dl}_d]$	✓	✓			54.8
$[\mathbf{v}_1^c][\mathbf{v}_2^c]...[\mathbf{v}_M^c][\mathbf{v}_1^d][\mathbf{v}_2^d]...[\mathbf{v}_N^d][\mathbf{c}_i][\mathbf{dl}_d]$	✓	✓	✓		55.2
$[\mathbf{v}_1^c][\mathbf{v}_2^c]...[\mathbf{v}_M^c][\mathbf{v}_1^d][\mathbf{v}_2^d]...[\mathbf{v}_N^d][\mathbf{c}_i][\mathbf{dl}_d]$	✓	✓	✓	✓	55.5

Table 4: The influence (%) of λ for the loss with target data $\mathcal{L}_t$.

Cityscapes→ FoggyCityscapes					
λ	0.25	0.5	1.0	5.0	10.0
mAP	55.6	55.8	55.9	55.7	55.4
Sim10K→ Cityscapes					
λ	0.25	0.5	1.0	5.0	10.0
mAP	62.5	62.9	62.9	62.4	62.8

to 16, *i.e.* $M/N = 16$ for the prompt with only domain-invariant/domain-specific tokens, and $M = 8, N = 8$ for the domain-adaptive prompt. Table 3 shows that each term of loss function contributes to improvement in the adaption performance. To reduce the influence of other factors, we do not use the historical prompt ensemble strategy. With $\mathcal{L}_{s,\{s\}} + \mathcal{L}_{t,\{t\}}$, four types of learnable prompts have improved $0.3 \sim 1.8\%$ on the hand-crafted prompt. Combined with $\mathcal{L}_{s,\{t\}}$, which commands the target prompt to classify the source image accurately, further gains $0.2 \sim 0.4\%$ on three prompts distinguished across domains. This reveals that by constraining the detection heads in the source and target domains to classify the input image as accurately as possible, the prompt can learn more domain-invariant knowledge. Further, though introduce $\mathcal{L}_{s,\{s,t\}} + \mathcal{L}_{t,\{s,t\}}$ has few effects on prompt with only domain-invariant/domain-specific tokens, it improves 0.4% on the domain-adaptive prompt. We conclude that in two respect. On one hand, just using domain-invariant/domain-specific tokens cannot model shared knowledge between domains and domain-specific knowledge at the same time. On the other hand, constraining the detection head generated by one domain to output higher confidence in its own domain than the other domain contributes to learning domain-specific knowledge. Moreover, adding $\mathcal{L}_{ent}$ brings extra 0.3% improvement.

A.2 Hyperparameter Search

To select hyperparameters for our loss functions, we perform experiments of different choices of the weight values $\lambda(\mathcal{L}_t)$. We conduct the experiment on DA-Pro on both Cityscapes→FoggyCityscapes and Sim10K→Cityscapes adaptation scenarios. Table 4 shows that our DA-Pro is robust with different settings of λ. Considering that the best result is achieved under 1.0, we take $\lambda = 1.0$ as the default.

Table 5: The influence (%) of the historical prompt ensemble on Cross-Weather adaptation scenario Cityscapes→Foggy Cityscapes. Total number of tokens is set to 16.

Historical Prompt Ensemble	$\mathcal{L}_{s,\{s\}} + \mathcal{L}_{t,\{t\}}$	$\mathcal{L}_{s,\{t\}}$	$\mathcal{L}_{s,\{s,t\}} + \mathcal{L}_{t,\{s,t\}}$	$\mathcal{L}_{ent}$
	54.4	54.8	55.2	55.5
✓	54.7	55.0	55.6	55.9
Gain	0.3	0.2	0.4	0.4

A.3 Effect of the Historical Prompt Ensemble Strategy

To verify the effect of the proposed historical prompt ensemble strategy, we calculate the mAP of whether apply prompt ensemble under different loss function. We conduct the experiment on DA-Pro on Cityscapes→FoggyCityscapes scenarios. Table 5 shows the effect of our historical prompt ensemble strategy. It improves the performance of all forms of learnable prompts without any inference overhead. In particular, the best case of improvement is 0.4% for the domain-adaptive prompt.