PØDA: Prompt-driven Zero-shot Domain Adaptation
– Supplementary Material –

Mohammad Fahes\textsuperscript{1} Tuan-Hung Vu\textsuperscript{1,2} Andrei Bursuc\textsuperscript{1,2} Patrick Pérez\textsuperscript{1,2} Raoul de Charette\textsuperscript{1}
\textsuperscript{1}Inria \textsuperscript{2}Valeo.ai

A. Overall pseudo-code of PØDA

Algorithm 2 presents the high-level pseudo-code of PØDA: from source-only training as model initialization, to prompt-driven feature augmentation, to zero-shot model adaptation.

B. Experimental details

Feature augmentation. PIN operates on image features. For augmentation, we optimize \((\mu, \sigma)\) of source feature map \(f_s\); it is done in batches for the sake of speed. We fix the batch size \(b = 16\) and the learning rate \(lr = 1.0\).

Style mixing. In the discussion of PØDA (Sec. 4.4 and Tab. 7), we presented the performance gains that style-mixing [6] brings to our method in three settings. By randomly mixing original and augmented statistics, we introduce certain perturbations to the final augmented features. The mixed statistics \(\mu_{\text{mix}}, \sigma_{\text{mix}}\) are given by:

\[
\mu_{\text{mix}} = \alpha \mu + (1 - \alpha) \mu_s, \quad \sigma_{\text{mix}} = \alpha \sigma + (1 - \alpha) \sigma_s,
\]

where \(\alpha \in \mathbb{R}^c\) are per-channel mixing weights uniformly sampled in \([0, 1]\), similarly to [6]: multiplications are element-wise. Finally, the augmented features are computed as follows:

\[
f_{s\rightarrow t} = \text{PIN}(f_s, \mu_{\text{mix}}, \sigma_{\text{mix}}),
\]

with prompt-driven instance normalization \(\text{PIN}\) defined in Eq. 2.

C. Additional experiments

Effect of style mining initialization. In our feature optimization step, we initialize \((\mu, \sigma)\) with \((\mu(f_s), \sigma(f_s))\). In Tab. 12, we report results using different initialization strategies. Starting from pre-defined or random initialization, instead of from original statistics, degrades badly the performance. As we do not use any regularization term in the CLIP cosine distance loss, we argue that initializing the optimized statistics with those of the source images is a form of regularization, favoring augmented features in a neighborhood of \(f_s\) and better preserving the semantics.

Optimization steps. In all our experiments, 100 iterations of optimization are performed for each batch of source features. We show in Fig. 7 the effect of the total number of iterations. We see an inflection point at around 80-100 iterations. Using few iterations is not sufficient for style alignment. Above 100, we also observe a performance drop. We refer to [3] and argue

\begin{table}[h]
\centering
\begin{tabular}{ccc}
\hline
\(\mu^0\) & \(\sigma^0\) & mIoU \\
\hline
\(\mu(f_s)\) & \(\sigma(f_s)\) & \textbf{25.03}\pm0.48 \\
0 & 1 & 8.59\pm0.82 \\
\sim \mathcal{N}(0, I) & \sim \mathcal{N}(0, I) & 6.80\pm0.92 \\
\hline
\end{tabular}
\caption{Effect of style initialization. Performance (in mIoU) of PØDA on ACDC-Night val set (Cityscapes as source), with different style statistics initializations. Starting from source images’ statistics \(\mu^0\) and \(\sigma^0\) works substantially better.}
\end{table}
Figure 7: Effect of the number of optimization iterations. Performance (mIoU %) of PØDA adaptation from Cityscapes to ACDC-Night as a function of the number of statistics optimization iterations. The values are averages over 5 runs and the bars represent the standard deviation.

Figure 8: Per-channel optimized statistics. Distributions of the first 20 channels of the optimized statistics of $\mu$ (Left) and $\sigma$ (Right). Each boxplot shows the interquartile range (IQR) that contains 50% of the data: Its bottom and top edges delimit the first and third quartiles respectively. The horizontal line inside the box denotes the data median. The whiskers extend from the edges of the box to the furthest point within 1.5 times the IQR, in each direction. Outlier points beyond these limits are individually plotted (diamonds).

Diversity of optimized statistics. To verify that the global statistics — optimized for the same number of iterations with the same TrgPrompt but from different starting anchor points $\bar{f}_s$ — are diverse, we show in Fig. 8 the boxplots of optimized parameters on the first 20 channels of $f_{s\rightarrow t}$ (for prompt “driving at night”).

Training from scratch on augmented features. In PØDA, we start with a source-only trained model (Algorithm 2, line 2) then we fine-tune it on augmented features (Algorithm 2, line 5). This is the general setting for domain adaptation. However, since our method performs domain adaptation under the assumption of label preservation, we also experimented training the model from scratch on augmented features. The results (Tab. 13) show the importance of the first, source-only training step.

Testing PØDA on other datasets. PØDA does not use target datasets at any point in training. Although there is no reason for the improvements observed to be specific for the datasets we test on, we show in Tab. 14 the performance of the model adapted using “driving at night” on two additional night-time driving scenes datasets:

- **Nighttime Driving** [2], test set, which consists of 50 annotated images of night driving scenes, with resolution of $1920 \times 1080$.
- **NightCity** [5], which is a large dataset of 4297 night-time driving scenes collected from many cities around the world; We tested on the validation/testing set, which consists of 1299 images of resolution $1024 \times 512$.

D. Class-wise performance

We report class-wise IoUs in Tab. 15.

E. PØDA for Object Detection

Here, we share the implementation details for our object detection experiments (Sec. 5 and Tab. 10). We used the implementation of Faster R-CNN [4] from the MMDetection library.\footnote{https://github.com/open-mmlab/mmdetection} With Cityscapes as source
dataset, we trained all models for 8 epochs using the SGD optimizer with 0.9 momentum and 1e−4 weight decay. The initial learning rate $lr$ is set as 1e−2 and is dropped by a factor of 10 after the 7th epoch; the same $lr$ scheme is used in source only and PØDA trainings. With Day-Sunny split of the DWD dataset as source, models are trained for 20 epochs using a similar SGD optimizer. When training on source, the learning rate starts at 1e−3 and drops at the 9th epoch to 1e−4; in PØDA training, the learning rate is ten times less.

### References


