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

Mohammad Fahes[1] Tuan-Hung Vu[1,2] Andrei Bursuc[1,2] Patrick Pérez[1,2] Raoul de Charette[1]
[1]Inria [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.

Algorithm 2: Prompt-driven Zero-shot DA

Input: Source dataset $\mathcal{D}_\mathrm{s} = \{(\mathbf{x}_\mathrm{s}, \mathbf{y}_\mathrm{s})\}$
CLIP encoders E_img and E_txt
Target domain description $\mathsf{TrgPrompt}$
Feature backbone $M_\mathrm{feat} \leftarrow E_\mathrm{img}$
Source model: $M = (M_\mathrm{feat}, M_\mathrm{cls})$
Result: Target-adapted model $M' = (M_\mathrm{feat}, M'_\mathrm{cls})$
// Initialization
$\mathsf{TrgEmb} = E_\mathrm{txt}(\mathsf{TrgPrompt})$
$M_\mathrm{cls} \leftarrow \mathtt{train}(M_\mathrm{cls}, \mathcal{D}_\mathrm{s})$ ▷ source-only training
// Feature Augmentation
$\mathcal{F}_\mathrm{s} \leftarrow \mathtt{feat\text{-}ext}(M_\mathrm{feat}, \{\mathbf{x}_\mathrm{s}\})$
$\mathcal{S}_{\mathrm{s}\rightarrow\mathrm{t}} \leftarrow \mathtt{augment}(\mathcal{F}_\mathrm{s}, \mathsf{TrgEmb})$
// Adaptation
$M'_\mathrm{cls} \leftarrow \mathtt{fine\text{-}tune}(M_\mathrm{cls}, \mathcal{F}_\mathrm{s}, \mathcal{S}_{\mathrm{s}\rightarrow\mathrm{t}}, \{\mathbf{y}_\mathrm{s}\})$ ▷ fine-tuning

B. Experimental details

Feature augmentation. PIN operates on image features. For augmentation, we optimize $(\boldsymbol{\mu}, \boldsymbol{\sigma})$ of source feature map $\mathbf{f}_\mathrm{s}$; 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 $\boldsymbol{\mu}_\mathrm{mix}, \boldsymbol{\sigma}_\mathrm{mix}$ are given by:

$$\boldsymbol{\mu}_\mathrm{mix} = \boldsymbol{\alpha}\boldsymbol{\mu}_\mathrm{t} + (1-\boldsymbol{\alpha})\boldsymbol{\mu}_\mathrm{s}\,, \quad (4)$$

$$\boldsymbol{\sigma}_\mathrm{mix} = \boldsymbol{\alpha}\boldsymbol{\sigma}_\mathrm{t} + (1-\boldsymbol{\alpha})\boldsymbol{\sigma}_\mathrm{s}\,, \quad (5)$$

where $\boldsymbol{\alpha} \in \mathbb{R}^c$ 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:

$$\mathbf{f}_{\mathrm{s}\rightarrow\mathrm{t}} = \mathtt{PIN}(\mathbf{f}_\mathrm{s}, \boldsymbol{\mu}_\mathrm{mix}, \boldsymbol{\sigma}_\mathrm{mix}), \quad (6)$$

with prompt-driven instance normalization $\mathtt{PIN}$ defined in Eq. 2.

C. Additional experiments

Effect of style mining initialization. In our feature optimization step, we initialize $(\boldsymbol{\mu}, \boldsymbol{\sigma})$ with $(\mu(\mathbf{f}_\mathrm{s}), \sigma(\mathbf{f}_\mathrm{s}))$. 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 $\bar{\mathbf{f}}_\mathrm{s}$ and better preserving the semantics.

$\boldsymbol{\mu}^0$	$\boldsymbol{\sigma}^0$	mIoU
$\mu(\mathbf{f}_\mathrm{s})$	$\sigma(\mathbf{f}_\mathrm{s})$	**25.03**±0.48
$\mathbf{0}$	$\mathbf{1}$	8.59±0.82
$\sim\mathcal{N}(\mathbf{0},\mathbf{I})$	$\sim\mathcal{N}(\mathbf{0},\mathbf{I})$	6.80±0.92

Table 12: **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 works substantially better.

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

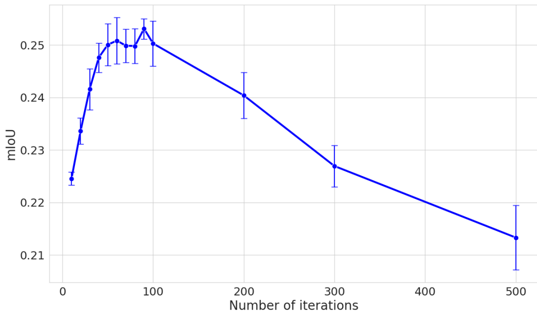

Figure 7: **Effect of the number of optimization iterations.** Performance (mIoU %) of PØDA adaptation from Cityscapes to ACDC-Night as a function of the number of statistics optimization iterations. The values are averages over 5 runs and the bars represent the standard deviation.

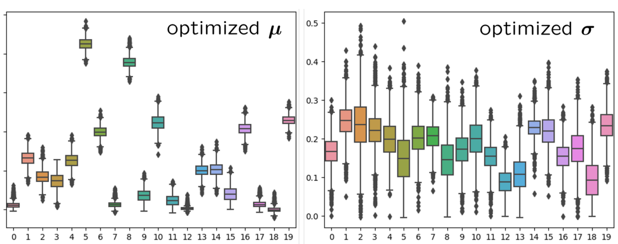

Figure 8: **Per-channel optimized statistics.** Distributions of the first 20 channels of the optimized statistics of $\boldsymbol{\mu}$ (*Left*) and $\boldsymbol{\sigma}$ (*Right*). Each boxplot shows the interquartile range (IQR) that contains 50% of the data: Its bottom and top edges delimit the first 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).

for the "over-stylization" problem in this case.

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 $\bar{\mathbf{f}}_\mathrm{s}$ — are diverse, we show in Fig. 8 the boxplots of optimized parameters on the first 20 channels of $\mathbf{f}_{\mathrm{s}\rightarrow\mathrm{t}}$ (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

Method	Night	Snow	Rain	GTA5
PØDA no src pretrain	22.46	36.73	39.70	39.57
PØDA	**25.03**	**43.90**	**42.31**	**41.07**

Table 13: **Importance of source-only pre-training.** Semantic segmentation performance (mIoU %) of PØDA *vs.* its variant without source-only training, when adapting from Cityscapes to ACDC Nigt/Snow/Rain and to GTA5.

	ACDC Night	Nighttime Driving [2]	NightCity [5]
Source-only	18.31	29.61	25.63
PØDA	**25.03**$\pm$0.48	**33.98**$\pm$0.61	**28.90**$\pm$0.61

Table 14: **PØDA at night.** Segmentation performance (mIoU %) of PØDA adapted from Cityscapes to nighttime with TrgPrompt = "driving at night", on three different night-time driving datasets.

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 1920×1080.
- **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 1024×512.

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.[1] With Cityscapes as source

[1] https://github.com/open-mmlab/mmdetection

Source	Target eval.	Method	road	sidewalk	building	wall	fence	pole	traffic light	traffic sign	vegetation	terrain	sky	person	rider	car	truck	bus	train	motorcycle	bicycle	mIoU%
		TrgPrompt = "driving at night"																				
CS	ACDC Night	source-only	70.42	18.32	**43.83**	6.11	17.08	23.52	**24.51**	19.76	39.74	6.11	0.78	21.62	8.96	23.08	2.53	0.00	3.27	8.42	9.87	18.31
		CLIPstyler	73.96	23.26	42.16	3.31	7.21	**35.49**	23.34	19.01	**45.41**	8.81	**27.87**	21.06	8.48	38.17	1.84	0.00	11.54	**10.38**	4.89	21.38$\pm$0.36
		PØDA	**77.54**	**26.90**	42.71	**13.51**	**21.36**	33.52	23.70	**21.73**	39.91	**9.51**	19.40	**28.80**	**11.85**	**50.89**	**10.14**	0.00	**20.76**	8.76	**14.50**	**25.03**$\pm$0.48
		TrgPrompt = "driving in snow"																				
	ACDC snow	source-only	70.47	23.50	63.80	17.96	**27.36**	**38.52**	**56.26**	**45.00**	**83.00**	10.75	83.65	47.73	0.72	61.42	21.87	5.90	21.58	35.83	31.01	39.28
		CLIPstyler	74.29	31.25	69.17	15.21	25.21	36.83	44.79	42.56	76.87	11.07	91.48	**53.23**	0.13	67.66	23.88	**9.14**	36.48	**42.67**	28.76	41.09$\pm$0.17
		PØDA	**75.40**	**34.61**	**75.22**	**26.77**	27.34	35.20	52.68	44.37	82.01	**14.16**	**93.72**	50.51	**0.99**	**69.11**	**26.64**	2.72	**46.98**	42.64	**33.09**	**43.90**$\pm$0.53
		TrgPrompt = "driving under rain"																				
	ACDC rain	source-only	74.10	31.98	63.07	**15.08**	**23.92**	**41.31**	**50.12**	**44.43**	79.93	22.07	87.45	47.99	4.39	68.92	10.35	18.52	13.64	7.03	21.58	38.20
		CLIPstyler	73.71	36.09	68.91	3.77	16.99	36.94	39.75	36.44	78.21	20.64	91.79	40.34	**9.65**	**74.54**	13.16	**20.33**	12.73	14.06	18.26	37.17$\pm$0.10
		PØDA	**76.60**	**38.52**	**78.01**	15.02	22.53	40.33	45.39	41.40	**86.85**	**37.97**	**96.46**	**50.39**	6.35	74.19	**19.19**	7.98	**22.06**	**21.04**	**23.65**	**42.31**$\pm$0.55
		TrgPrompt = "driving in a game"																				
	GTA5	source-only	68.72	22.65	78.79	36.81	**17.31**	39.66	**39.33**	14.84	**72.61**	22.53	87.31	57.50	26.14	74.29	**44.57**	**20.45**	0.00	18.30	10.35	39.59
		CLIPstyler	73.06	**29.89**	77.86	25.50	11.69	**39.72**	35.88	**24.04**	67.38	12.75	88.77	46.58	33.38	72.03	42.79	11.12	0.00	28.84	**14.61**	38.73$\pm$0.16
		PØDA	**73.93**	22.69	**78.82**	**37.52**	14.17	36.97	33.14	17.34	72.44	**26.22**	**88.85**	**62.69**	**37.04**	**74.33**	43.03	11.91	0.00	**35.33**	13.91	**41.07**$\pm$0.48
		TrgPrompt = "driving"																				
GTA5	CS	source-only	58.97	20.92	72.84	16.53	**24.58**	31.37	34.77	23.62	82.12	17.04	66.28	**63.46**	14.72	**81.27**	**20.83**	17.19	4.68	**20.57**	19.56	36.38
		CLIPstyler	66.70	23.63	64.12	5.08	3.66	20.67	19.31	18.10	81.68	12.36	**81.04**	54.64	0.52	73.47	20.65	**22.30**	4.03	15.79	10.73	31.50$\pm$0.21
		PØDA	**84.34**	**36.73**	**79.43**	**18.33**	16.54	**36.93**	**38.45**	**33.81**	**82.44**	**19.14**	75.90	62.65	**16.47**	75.48	15.68	19.57	**11.28**	16.53	**21.76**	**40.08**$\pm$0.52

Table 15: **Zero-shot domain adaptation in semantic segmentation.** Performance (mIoU%) of PØDA compared against CLIPstyler [3] and source-only baseline. Results are grouped by source domain and target domain with associated TrgPrompt. CS stands for Cityscapes [1]. This table provides details of the main results in Tab. 2.

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

[1] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *CVPR*, 2016. 3

[2] Dengxin Dai and Luc Van Gool. Dark model adaptation: Semantic image segmentation from daytime to nighttime. In *ITSC*, 2018. 2

[3] Gihyun Kwon and Jong Chul Ye. Clipstyler: Image style transfer with a single text condition. In *CVPR*, 2022. 1, 3

[4] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NeurIPS*, 2015. 2

[5] Xin Tan, Ke Xu, Ying Cao, Yiheng Zhang, Lizhuang Ma, and Rynson WH Lau. Night-time scene parsing with a large real dataset. *IEEE T-IP*, 2021. 2

[6] Xinyi Wu, Zhenyao Wu, Yuhang Lu, Lili Ju, and Song Wang. Style mixing and patchwise prototypical matching for one-shot unsupervised domain adaptive semantic segmentation. In *AAAI*, 2022. 1