PØDA: Prompt-driven Zero-shot Domain Adaptation

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
https://astra-vision.github.io/PODA

Abstract

Domain adaptation has been vastly investigated in computer vision but still requires access to target images at train time, which might be intractable in some uncommon conditions. In this paper, we propose the task of 'Prompt-driven Zero-shot Domain Adaptation', where we adapt a model trained on a source domain using only a general description in natural language of the target domain, i.e., a prompt. First, we leverage a pretrained contrastive vision-language model (CLIP) to optimize affine transformations of source features, steering them towards the target text embedding while preserving their content and semantics. To achieve this, we propose Prompt-driven Instance Normalization (PIN). Second, we show that these prompt-driven augmentations can be used to perform zero-shot domain adaptation for semantic segmentation. Experiments demonstrate that our method significantly outperforms CLIP-based style transfer baselines on several datasets for the downstream task at hand, even surpassing one-shot unsupervised domain adaptation. A similar boost is observed on object detection and image classification. The code is available at https://github.com/astra-vision/PODA .

1. Introduction

The last few years have witnessed tremendous success of supervised semantic segmentation methods towards better high-resolution predictions [5, 6, 9, 31, 50], multi-scale processing [30,57] or computational efficiency [56]. In controlled settings where segmentation models are trained using data from the targeted operational design domains, the accuracy can meet the high industry-level expectations on in-domain data; yet, when tested on out-of-distribution data, these models often undergo drastic performance drops [35]. This hinders their applicability in real-world scenarios for critical applications like in-the-wild autonomous driving.

To mitigate this domain-shift problem [2], unsupervised domain adaptation (UDA) [16, 18, 45, 46, 48, 59] has emerged as a promising solution. Training of UDA methods requires labeled data from *source* domain and unlabeled data from *target* domain. Though seemingly effortless, for some conditions even collecting unlabeled data is complex. For example, as driving through fire or sandstorm rarely occurs in real life, collecting raw data in such conditions is non-trivial. One may argue on using Internet images for UDA. However, in the industrial context, the practice of using public data is limited or forbidden. Recent works aim to reduce the burden of target data collection campaigns by devising one-shot [33,54] UDA methods, *i.e.*, using one target image for training. Pushing further this line of research, we frame the challenging new task of prompt-driven zero-shot domain adaptation where given a target domain description in natural language (*i.e.*, a *prompt*), our method accordingly adapts the segmentation model to this domain of interest. Figure 1 outlines the primary goal of our work with a few qualitative examples. Without seeing any fire or sandstorm images during training, the adapted models succeed in segmenting out critical scene objects, exhibiting fewer errors than the original source-only model.

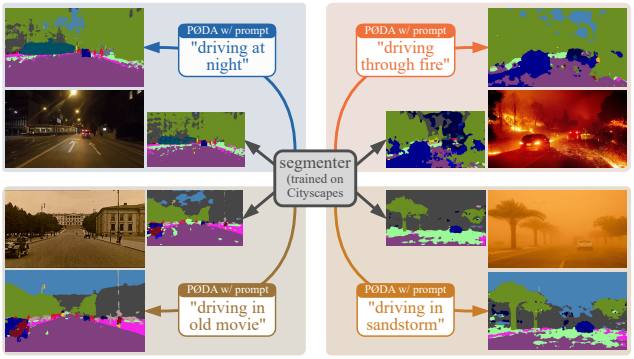

Figure 1. **Zero-shot adaptation with prompt.** PØDA enables the adaptation of a segmenter model (here, DeepLabv3+ *trained on the source dataset Cityscapes*) to unseen conditions with only a prompt. Source-only predictions are shown as smaller segmentation masks to the left or right of the test images.

Our method, illustrated in Fig. 2, is made possible by leveraging the vision-language connections from the seminal CLIP model [39]. Trained on 400M web-crawled image-text pairs, CLIP has revolutionized multi-modal representation learning, bringing outstanding capability to

zero-shot image synthesis [15, 24, 37], zero-shot multi-modal fusion [20], zero-shot semantic segmentation [26, 58], open-vocabulary object detection [34], few-shot learning [10], etc. In our work, we exploit the CLIP's latent space and propose a simple and effective feature stylization mechanism that converts source-domain *features* into target-domain ones (Fig. 2, left), which can be seen as a specific form of data *augmentation*. Fine-tuning the segmentation model on these zero-shot synthesized features (Fig. 2, middle) helps mitigating the distribution gap between the two domains thus improving performance on unseen domains (Fig. 2, right). Owing to the standard terminology of "prompt" that designates the input text in CLIP-based image generation, we coin our approach *Prompt-driven Zero-shot Domain Adaptation*, PØDA in short.

To summarize, our contributions are as follows:

- We introduce the novel task of prompt-driven zero-shot domain adaptation, which aims at adapting a source-trained model on a target domain provided *only* an arbitrary textual description of the latter.
- Unlike other CLIP-based methods that navigate CLIP latent space using direct image representations, we alter only the features, without relying on the appearance in pixel space. We argue that this is particularly useful for downstream tasks such as semantic segmentation where good features are decisive (and sufficient). We present a simple and effective *Prompt-driven Instance Normalization (PIN)* layer to augment source features, where affine transformations of low-level features are optimized such that the representation in CLIP latent space matches the one of target-domain prompt.
- We show the versatility of our method by adapting source-trained semantic segmentation models to different conditions: (i) from clear weather/daytime to adverse conditions (snow, rain, night), (ii) from synthetic to real, (iii) from real to synthetic. Interestingly, PØDA outperforms state-of-the-art one-shot unsupervised domain adaptation without using any target image.
- We show that PØDA can also be applied to object detection and image classfication.

2. Related works

Unsupervised Domain Adaptation. The UDA literature is vast and encompasses different yet connected approaches: adversarial learning [16, 46], self-training [29, 60], entropy minimization [36, 48], generative-based adaptation [18], etc. The domain gap is commonly reduced at the level of the input [18, 55], of the features [16, 32, 45, 52] or of the output [36, 46, 48].

Recently, the more challenging setting of One-Shot Unsupervised Domain Adaptation (OSUDA) has been proposed. To the best of our knowledge, two works on OSUDA for semantic segmentation exist [33, 54]. Luo et al. [33] show that traditional UDA methods fail when only a single unlabeled target image is available. To mitigate the risk of over-fitting on the style of the single available image, the authors propose a style mining algorithm, based on both a stylized image generator and a task-specific module. Wu et al. [54] introduce an approach based on style mixing and patch-wise prototypical matching (SM-PPM). During training, channel-wise mean and standard deviation of a randomly sampled source image's features are linearly mixed with the target ones. Patch-wise prototypical matching helps overcome negative adaptation [27].

In the more challenging zero-shot setting (where no target image is available), Lengyel et al. [25] tackle day-to-night domain adaptation using physics priors. They introduce a color invariant convolution layer (CIConv) that is added to make the network invariant to different lighting conditions. We note that this zero-shot adaption is orthogonal to ours and restricted to a specific type of domain gap.

Text-driven image synthesis. Recently, contrastive image-language pretraining has shown unprecedented success for multimodal learning in several downstream tasks such as zero-shot classification [39], multi-modal retrieval [21] and visual question answering [28]. This encouraged the community to modify images using text descriptions, a task that was previously challenging due to the gap between vision and language representations. For example, StyleCLIP [37] uses prompts to optimize StyleGAN [22] latent vectors and guide the generation process. However, the generation is limited to the training distribution of StyleGAN. To overcome this issue, StyleGAN-NADA [15] utilizes CLIP embeddings of text-prompts to perform domain adaptation of the generator, which is in this case trainable. Similarly, for text-guided semantic image editing, FlexIT [12] optimizes the latent code in VQGAN autoencoder's [13] space.

For text-guided style transfer, CLIPstyler [24] does not rely on a generative process. This setting is more realistic for not being restricted to a specific distribution, and challenging at the same time for the use of the encapsulated information in CLIP latent space. Indeed, there is no one-to-one mapping between image and text representations and regularization is needed to extract the useful information from a text embedding. Thus, in the same work [24], a U-net autoencoder that preserves the content is optimized while the output image embedding in CLIP latent space is varying during the optimization process.

We note that a common point in prior works is the mapping from pixel-space to CLIP latent space during the optimization process. In contrast with this, we directly manipulate deep features of the pre-trained CLIP visual encoder.

3. Prompt-driven Zero-shot Adaptation

Our framework, illustrated in Fig. 2, builds upon CLIP [39], a vision-language model pre-trained on 400M

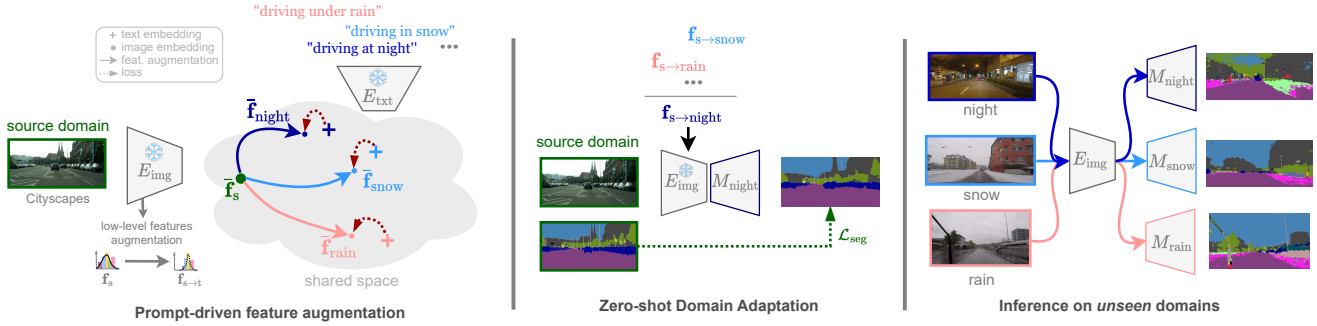

Figure 2. **Overview of PØDA, a Prompt-driven Zero-shot Domain Adaptation.** (*Left*) Using only a single textual description ("...") of an *unseen* target domain, we leverage a frozen ResNet encoder with CLIP weights to learn source→target low-level feature stylizations. Applied to a source image low-level feature map $\mathbf{f}_s$ with latent embedding $\bar{\mathbf{f}}_s$, these stylizations provide augmented features $\mathbf{f}_{s\to t}$ which embeddings (here, $\bar{\mathbf{f}}_{night}$, $\bar{\mathbf{f}}_{snow}$, $\bar{\mathbf{f}}_{rain}$) are closer to their respective target prompt embeddings (+). (*Middle*) Zero-shot domain adaptation is achieved by fine-tuning a segmenter model (M) on the feature-augmented source domain with the learned transformations. (*Right*) This enables inference on unseen domains.

image-text pairs crawled from the Internet. CLIP trains jointly an image encoder E_{img} and a text encoder E_{txt} over many epochs and learns an expressive representation space that effectively bridges the two modalities. In this work, we leverage this property to "steer" features from any source image towards a target domain in the CLIP latent space, with guidance from an arbitrary prompt describing the target domain, *e.g.*, "driving at night" or "navigating the roads in darkness" for the night domain. Our goal is to modify the style of source image features, bringing them closer to imaginary counterparts in the targeted domain (Fig. 2, left), while preserving their semantic content. The learned augmentations can then be applied on source images to generate features, in a zero-shot fashion, that correspond to the unseen target domain and can be further used to fine-tune the model towards handling target domain (Fig. 2, middle). This ultimately allows inference on unseen domains only described by a simple prompt at train time (Fig. 2, right).

Our approach faces several challenges: *(i)* How to generate informative features for the target domain without having access to any image from it; *(ii)* How to preserve pixel-wise semantics while augmenting features; *(iii)* Based on such features, how to adapt the source model to the *unseen* target domain. We address these questions in the following.

Problem formulation. Our main task is semantic segmentation, that is, pixel-wise classification of input image into semantic segments. We start from a K-class segmentation model M, pre-trained on a source domain dataset $\mathcal{D}_s = \{(\mathbf{x}_s, \mathbf{y}_s) \mid \mathbf{x}_s \in \mathbb{R}^{H\times W\times 3}, \mathbf{y}_s \in \{0,1\}^{H\times W\times K}\}$. By using a single predefined prompt TrgPrompt describing the targeted domain, we adapt the model M such that its performance on the unseen test target dataset $\mathcal{D}_t = \{\mathbf{x}_t \mid \mathbf{x}_t \in \mathbb{R}^{H\times W\times 3}\}$ is improved. The segmenter M is a DeepLabv3+ model [6] with CLIP image encoder E_{img} (*e.g.*, ResNet-50) as the frozen feature extractor backbone M_{feat} and a randomly initialized pixel classification head M_{cls}: $M = (M_{feat}, M_{cls})$. We train M in a supervised manner for the semantic segmentation task on the source domain. In order to preserve the compatibility of the encoder features with the CLIP latent space we keep M_{feat} frozen and train only the pixel classifier M_{cls}. Interestingly, we empirically show in Tab. 1 that keeping the feature extractor M_{feat} frozen also prevents overfitting to the source in favor of generalization. From the extractor we remove the attention pooling head of E_{img} to keep the spatial information for the pixel classifier. We denote $\mathbf{f}$ the intermediate features extracted by M_{feat} and $\bar{\mathbf{f}}$ their corresponding CLIP embedding computed with the attention pooling layer of E_{img}. In Fig. 3 we illustrate the difference between $\mathbf{f}$ and $\bar{\mathbf{f}}$.

Overview of the proposed method. Our solution is to mine styles using source-domain low level features set $\mathcal{F}_s = \{\mathbf{f}_s \mid \mathbf{f}_s = \texttt{feat-ext}(M_{feat}, \mathbf{x}_s)\}$ and TrgEmb, where $\text{TrgEmb} = E_{txt}(\text{TrgPrompt})$ is the CLIP text embedding of the target domain prompt. For generality, $\texttt{feat-ext}(\cdot)$ can pull features from any desired layer but we later show that using the lowest features works best.

The $\texttt{augment}(\cdot)$ operation, depicted in Fig. 3, augments the style-specific components of $\mathbf{f}_s$ with guidance from the target domain prompt, synthesizing $\mathbf{f}_{s\to t}$ with style information from the target domain. We emphasize that the features $\mathbf{f}_s$ and $\mathbf{f}_{s\to t}$ have the same size $h \times w \times c$ and identical seman-

M_{feat} ❄	CS	Night	Snow	Rain	GTA5
Yes	66.82	**18.31**	**39.28**	**38.20**	**39.59**
No	**69.17**	14.40	22.27	26.33	32.91

Table 1. **Segmentation with source-only trained models**. Performance (mIoU %) on "night", "snow" and "rain" parts of ACDC [44] *validation set* and on a subset of 1000 GTA5 images for models trained on Cityscapes (CS). 'M_{feat}❄': frozen backbone.

Algorithm 1: Style Mining (see Fig. 3)

Input : Set $\mathcal{F}_s$ of source image features
Target domain description embedding TrgEmb
Param : Number N of optimization steps
Learning rate lr and momentum m
of gradient descend (GD)
Output: Set $\mathcal{S}_{s\to t}$ of target styles

$\mathcal{S}_{s\to t} \leftarrow \emptyset$
foreach $\mathbf{f}_s \in \mathcal{F}_s$ **do**
  $\boldsymbol{\mu}^0 \leftarrow \texttt{mean}(\mathbf{f}_s)$
  $\boldsymbol{\sigma}^0 \leftarrow \texttt{std}(\mathbf{f}_s)$
  // Optimization
  for $i = 1, 2, \cdots, N$ **do**
    $\mathbf{f}^i_{s\to t} \leftarrow \texttt{PIN}(\mathbf{f}_s, \boldsymbol{\mu}^{i-1}, \boldsymbol{\sigma}^{i-1})$
    $\bar{\mathbf{f}}^i_{s\to t} \leftarrow \texttt{get-embedding}(\mathbf{f}^i_{s\to t})$
    $\boldsymbol{\mu}^i \leftarrow \texttt{GD}^{lr}_m(\boldsymbol{\mu}^{i-1}, \nabla_{\boldsymbol{\mu}}\mathcal{L}_{\boldsymbol{\mu},\boldsymbol{\sigma}}(\bar{\mathbf{f}}^i_{s\to t}, \text{TrgEmb}))$
    $\boldsymbol{\sigma}^i \leftarrow \texttt{GD}^{lr}_m(\boldsymbol{\sigma}^{i-1}, \nabla_{\boldsymbol{\sigma}}\mathcal{L}_{\boldsymbol{\mu},\boldsymbol{\sigma}}(\bar{\mathbf{f}}^i_{s\to t}, \text{TrgEmb}))$
  end
  $(\boldsymbol{\mu}_t, \boldsymbol{\sigma}_t) \leftarrow (\boldsymbol{\mu}^N, \boldsymbol{\sigma}^N)$
  $\mathcal{S}_{s\to t} \leftarrow \mathcal{S}_{s\to t} \cup \{(\boldsymbol{\mu}_t, \boldsymbol{\sigma}_t)\}$
end

tic content, though they encapsulate different visual styles. For adaptation, the source features $\mathbf{f}_s$ are augmented with the mined styles then used to fine-tune the classifier M_{cls}, resulting in the final adapted model. The overall pseudo-code is provided in Supplementary Material.

3.1. Zero-shot Feature Augmentation

We take inspiration from Adaptive Instance Normalization (AdaIN) [19], an elegant formulation for transferring style-specific components across deep features. In AdaIN, the styles are represented by the channel-wise mean $\boldsymbol{\mu} \in \mathbb{R}^c$ and standard deviation $\boldsymbol{\sigma} \in \mathbb{R}^c$ of features, with c the number of channels. Stylizing a source feature $\mathbf{f}_s$ with an arbitrary target style $(\mu(\mathbf{f}_t), \sigma(\mathbf{f}_t))$ reads:

$$\texttt{AdaIN}(\mathbf{f}_s, \mathbf{f}_t) = \sigma(\mathbf{f}_t)\left(\frac{\mathbf{f}_s - \mu(\mathbf{f}_s)}{\sigma(\mathbf{f}_s)}\right) + \mu(\mathbf{f}_t), \tag{1}$$

with $\mu(\cdot)$ and $\sigma(\cdot)$ as the two functions returning channel-wise mean and standard deviation of input feature; multiplications and additions are element-wise.

We design our augmentation strategy around AdaIN as it can effectively manipulate the style information with a small set of parameters. In the following, we present our augmentation strategy which mines target styles.

As we do not have access to any target (*i.e.* style) image, $\mu(\mathbf{f}_t)$ and $\sigma(\mathbf{f}_t)$ are unknown. Thus, we propose Prompt-driven Instance Normalization (PIN):

$$\texttt{PIN}(\mathbf{f}_s, \boldsymbol{\mu}, \boldsymbol{\sigma}) = \boldsymbol{\sigma}\left(\frac{\mathbf{f}_s - \mu(\mathbf{f}_s)}{\sigma(\mathbf{f}_s)}\right) + \boldsymbol{\mu}, \tag{2}$$

where $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ are optimizable variables driven by a prompt.

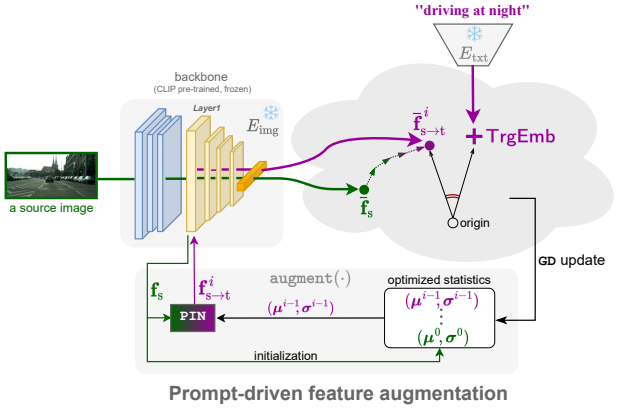

Figure 3. **Target style mining from a source image**. We illustrate here the optimization loops of Algorithm 1. The source image is forwarded through the CLIP image encoder E_{img} to extract low-level features $\mathbf{f}_s$ and subsequent CLIP embedding $\bar{\mathbf{f}}_s$. At each optimization step i, $\texttt{augment}(\cdot)$ takes the style of the previous iteration, $(\boldsymbol{\mu}^{i-1}, \boldsymbol{\sigma}^{i-1})$ and injects it within $\mathbf{f}_s$ via the PIN layer, to synthesize $\mathbf{f}^i_{s\to t}$ and the corresponding embedding $\bar{\mathbf{f}}^i_{s\to t}$. The loss $\mathcal{L}_{\boldsymbol{\mu},\boldsymbol{\sigma}}$ is the cosine distance between $\bar{\mathbf{f}}^i_{s\to t}$ and the target prompt embedding TrgEmb. Its optimization via gradient descent updates style to $(\boldsymbol{\mu}^i, \boldsymbol{\sigma}^i)$.

We aim to augment source image features $\mathcal{F}_s$ such that they capture the style of the target domain. Here, the prompt describing a target domain could be fairly generic. For instance, one can use prompts like "driving at night" or "driving under rain" to bring source features closer to the nighttime or rainy domains. The prompt is processed by the CLIP text encoder E_{txt} into the TrgEmb embedding.

We describe in Algorithm 1 the first step of our zero-shot feature augmentation procedure: mining the set $\mathcal{S}_{s\to t}$ of styles in targeted domain. For each source feature map $\mathbf{f}_s \in \mathcal{F}_s$, we want to mine style statistics corresponding to an imaginary target feature map $\mathbf{f}_t$. To this end, we formulate style mining as an optimization problem over the original source feature $\mathbf{f}_s$, *i.e.* optimizing $(\boldsymbol{\mu}, \boldsymbol{\sigma})$ in Eq. (2). The optimization objective is defined as the cosine distance in the CLIP latent space between the CLIP embedding $\bar{\mathbf{f}}_{s\to t}$ of the stylized feature $\mathbf{f}_{s\to t} = \texttt{PIN}(\mathbf{f}_s, \boldsymbol{\mu}, \boldsymbol{\sigma})$ and the description embedding TrgEmb of target domain:

$$\mathcal{L}_{\boldsymbol{\mu},\boldsymbol{\sigma}}(\bar{\mathbf{f}}_{s\to t}, \text{TrgEmb}) = 1 - \frac{\bar{\mathbf{f}}_{s\to t} \cdot \text{TrgEmb}}{\|\bar{\mathbf{f}}_{s\to t}\| \, \|\text{TrgEmb}\|}. \tag{3}$$

This CLIP-space cosine distance, already used in prior text-driven image editing works [37], aims to steer the stylized features in the direction of the target text embedding. One step of the optimization is illustrated in Fig. 3. In practice, we run several such steps leading to the mined target style denoted $(\boldsymbol{\mu}_t, \boldsymbol{\sigma}_t)$.

As there might be a variety of styles in a target domain, our mining populates the $\mathcal{S}_{s\to t}$ set with as many variations of

target style as there are source images, hence $|\mathcal{S}_{s\to t}| = |\mathcal{D}_s|$.

Intuitively, our simple augmentation strategy can be seen as a cost-efficient way to cover the distribution of the target domain by starting from different anchor points in the CLIP latent space coming from the source images and steering them in the direction of the target text embedding. This mitigates the diversity problem discussed in one-shot feature augmentation in [33, 54].

3.2. Fine-tuning for Adaptation

For adaptation, at each training iteration we stylize the source features using a mined target style $(\boldsymbol{\mu}_t, \boldsymbol{\sigma}_t)$ randomly selected from $\mathcal{S}_{s\to t}$. The augmented features are computed as $\mathbf{f}_{s\to t} = \texttt{PIN}(\mathbf{f}_s, \boldsymbol{\mu}_t, \boldsymbol{\sigma}_t)$ and are used for fine-tuning the classifier M_{cls} of the segmenter M (Fig. 2, middle). As we only adjust the feature style which keeps the semantic-content unchanged [19], we can still use the labels $\mathbf{y}_s$ to train the classifier with a standard segmentation loss. To this end, we simply forward augmented features through remaining layers in M_{feat} followed by M_{cls}. In the backward pass, only weights of M_{cls} are updated by the loss gradients. We denote the fine-tuned model as $M' = (M_{\text{feat}}, M'_{\text{cls}})$ and evaluate it on images with conditions and styles which were never seen during any of the training stages.

4. PØDA for semantic segmentation

4.1. Implementation details

We use the DeepLabv3+ architecture [6] with the backbone M_{feat} initialized from the image encoder E_{img} of the pre-trained CLIP-ResNet-50 model[1].

Source-only training. The network is trained for 200k iterations on random 768×768 crops with batch size 2. We use a polynomial learning rate schedule with initial $lr{=}10^{-1}$ for the classifier and $lr{=}10^{-4}$ for backbone when not frozen (see Tab. 1). We optimize with Stochastic Gradient Descent [4], momentum 0.9 and weight decay 10^{-4}. We apply standard color jittering and horizontal flip to crops.

Zero-shot feature augmentation. For the feature augmentation step, we use the source feature maps after the first layer (*Layer1*): $\mathbf{f}_s \in \mathbb{R}^{192\times192\times256}$. The style parameters $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ are 256D real vectors. The CLIP embeddings are 1024D vectors. We adopt the Imagenet templates from [39] to encode the target descriptions in TrgPrompt.

Classifier fine-tuning. Starting from the source-only pre-trained model, we fine-tune the classifier M_{cls} on batches of 8 augmented features $\mathbf{f}_{s\to t}$ for $2,000$ iterations. Polynomial schedule is used with the initial $lr = 10^{-2}$. We always use the last checkpoint for evaluation.

Datasets. As source, we use Cityscapes [11], composed of $2,975$ training and 500 validation images featuring 19 semantic classes. Though we adapt towards a prompt *not* a dataset, we need adhoc datasets to test on. We report main results using ACDC [44] because it has urban images captured in adverse conditions. We also study the applicability of PØDA to the two settings of real→synthetic (Cityscapes as source, and evaluating on GTA5 [42]) and synthetic→real (GTA5 as source, and evaluating on Cityscapes). We evaluate on the validation set when provided, and for GTA5 evaluation we use a random subset of $1,000$ images.

Evaluation protocol. Mean Intersection over Union (mIoU%) is used to measure adaptation performance. We test all models on target images at their original resolutions. For baselines and PØDA, we always report the mean and standard deviation over five models trained with different random seeds.

4.2. Main results

We consider the following adaptation scenarios: day→night, clear→snow, clear→rain, real→synthetic and synthetic→real. We report zero-shot adaption results of PØDA in the addressed set-ups, comparing against two state-of-the-art baselines: CLIPstyler [24] for zero-shot style transfer and SM-PPM [54] for one-shot UDA. Both PØDA and CLIPstyler models see no target images during training. In this study, we arbitrarily choose a simple prompt to describe each domain. We show later in Sec. 4.3 more results using other relevant prompts with similar meanings – showcasing that our adaptation gain is little sensitive to prompt selection. For SM-PPM, one random target image from the training set is used.

Comparison to CLIPstyler [24]. CLIPstyler is a style transfer method that also makes use of the pre-trained CLIP model but for zero-shot stylizing of source images. We consider CLIPstyler[2] as the most comparable zero-shot baseline for PØDA as both are built upon CLIP, though with different mechanisms and different objectives. Designed for style transfer, CLIPstyler produces images that exhibit characteristic styles of the input text prompt. However the stylized images can have multiple artifacts which hinder their usability in the downstream segmentation task. This is visible in Fig. 4 which shows stylized examples from CLIPstyler with PØDA target prompts. Zooming in, we note that stylization of snow or game added snowy roads or Atari game *on the buildings*, respectively.

Starting from source-only model, we fine-tune the classifier on stylized images, as similarly done in PØDA with the augmented features. Table 2 compares PØDA against the source-only model and CLIPstyler. PØDA consistently outperforms the two baselines. CLIPstyler brings some improvements over source-only in Cityscapes→Night and

[1] https://github.com/openai/CLIP

[2] We use official code https://github.com/cyclomon/CLIPstyler and follow the recommended configs.

Source	Target eval.	Method	mIoU[%]
CS	TrgPrompt = "driving at night"		
	ACDC Night	source-only	18.31
		CLIPstyler	21.38 ±0.36
		PØDA	**25.03** ±0.48
	TrgPrompt = "driving in snow"		
	ACDC Snow	source-only	39.28
		CLIPstyler	41.09 ±0.17
		PØDA	**43.90** ±0.53
	TrgPrompt = "driving under rain"		
	ACDC Rain	source-only	38.20
		CLIPstyler	37.17 ±0.10
		PØDA	**42.31** ±0.55
	TrgPrompt = "driving in a game"		
	GTA5	source-only	39.59
		CLIPstyler	38.73 ±0.16
		PØDA	**41.07** ±0.48
GTA5	TrgPrompt = "driving"		
	CS	source-only	36.38
		CLIPstyler	31.50 ±0.21
		PØDA	**40.08** ±0.52

Table 2. **Zero-shot domain adaptation in semantic segmentation.** Performance (mIoU%) of PØDA compared against CLIPstyler [24] and source-only baseline. Results are grouped by source domain and TrgPrompt. CS stands for Cityscapes [11]. The TrgPrompts are simply chosen, not engineered.

Figure 4. **CLIPstyler [24] stylization.** A sample Cityscapes image stylized using adhoc target prompts. Translated images exhibit visible artifacts, potentially harming adaptation, *e.g.* rain in Tab. 2

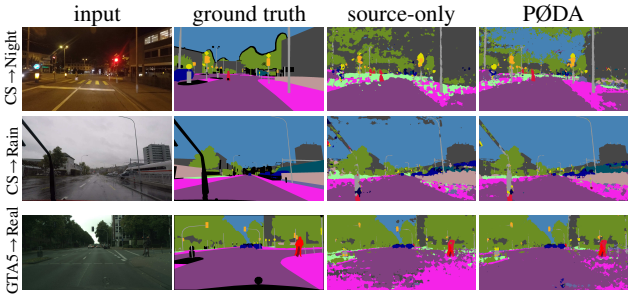

Figure 5. **Qualitative results of zero-shot adaptation.** (*Columns 1-2*) Input images and their ground truths; (*Columns 3-4*) Segmentation results of source-only and PØDA models.

Cityscapes→Snow. In other scenarios, *e.g.*, rain, CLIPstyler even performs worse than source-only.

Real→synthetic is an interesting though under-explored adaptation scenario. One potential application of real→synthetic is for model validation in the industry, where some hazardous validations like driving accidents must be done in the virtual space. Here we test if our zero-shot mechanism can be also applied to this particular setting. Similarly, PØDA outperforms both baselines. Also in the reverse synthetic→real setting, again our method performs the best. CLIPstyler undergoes almost 5% drops in mIoU compared to source-only.

We argue on the simplicity of our method that only introduces minimal changes to the feature statistics, yet such changes are crucial for target adaptation. CLIPstyler, designed for style transfer, involves training an additional StyleNet with ≈ 615k parameters for synthesizing the stylized images. We base on the simplicity merit of PØDA to explain why it is more favorable than CLIPstyler for downstream tasks like semantic segmentation: the minimal statistics changes help avoiding significant drifts on the feature manifold which may otherwise result in unwanted errors. For comparison, it takes us 0.3 seconds to augment one source feature, while stylizing an image with CLIPstyler takes 65 seconds (as measured on one RTX 2080TI GPU).

We show in Fig. 5 qualitative examples of predictions from source-only and PØDA models. We report class-wise performance in Supplementary Material.

Source	Target eval.	One-shot SM-PPM [54]	Zero-shot PØDA
CS	ACDC Night	13.07 / 14.60 (Δ=1.53)	18.31 / **25.03** (Δ=6.72)
	ACDC Snow	32.60 / 35.61 (Δ=3.01)	39.28 / **43.90** (Δ=4.62)
	ACDC Rain	29.78 / 32.23 (Δ=2.45)	38.20 / **42.31** (Δ=4.11)
GTA5	CS	30.48 / 39.32 (Δ=8.84)	36.38 / **40.08** (Δ=3.70)

Table 3. **Comparison with SM-PPM.** Semantic segmentation performance (mIoU%) for source / adapted models, and gain provided by adaptation (Δ in mIoU). For adaptation, SM-PPM (ResNet-101 DeepLabv2) has access to one target image, while PØDA (ResNet-50 DeepLabv3+) leverages a target prompt and a text encoder.

Comparison to one-shot UDA (OSUDA). We also compare PØDA against SM-PPM [54][3], a state-of-the-art OSUDA method, see Tab. 3. The OSUDA setting allows the access to a single unlabeled target domain image for DA. In SM-PPM, this image is considered as an anchor point for target style mining. Using 5 randomly selected target images, we trained, with each one, five models with different random seeds. The reported mIoUs are averaged over the 25 resulting models. We note that the absolute results of the two models are not directly comparable due to the differences in backbone (ResNet-101 in SM-PPM *vs*. ResNet-50 in PØDA) and in segmentation framework (DeepLabv2 in SM-PPM *vs*. DeepLabv3+ in PØDA). We thus analyze the improvement of each method over the corresponding naive source-only baseline while taking into account the source-only performance. We first no-

[3] We use official code https://github.com/W-zx-Y/SM-PPM

Method	ACDC Night	ACDC Snow	ACDC Rain	GTA5
Source only	18.31	39.28	38.20	39.59
Trg	"driving at night"	"driving in snow"	"driving under rain"	"driving in a game"
	25.03 ±0.48	43.90 ±0.53	42.31 ±0.55	41.07 ±0.48
ChatGPT-generated ← Relevant	"operating a vehicle after sunset"	"operating a vehicle in snowy conditions"	"operating a vehicle in wet conditions"	"piloting a vehicle in a virtual world"
	24.38 ±0.37	44.33 ±0.36	42.21 ±0.47	41.25 ±0.40
	"driving during the nighttime hours"	"driving on snow-covered roads"	"driving on rain-soaked roads"	"controlling a car in a digital simulation"
	25.22 ±0.64	43.56 ±0.62	**42.51** ±0.33	41.19 ±0.14
	"navigating the roads in darkness"	"piloting a vehicle in snowy terrain"	"navigating through rainfall while driving"	"maneuvering a vehicle in a computerized racing experience"'
	24.73 ±0.47	**44.67** ±0.18	41.11 ±0.69	40.34 ±0.49
	"driving in low-light conditions"	"driving in wintry precipitation"	"driving in inclement weather"	"operating a transport in a video game environment"
	24.68 ±0.34	43.11 ±0.56	40.68 ±0.37	41.34 ±0.42
	"travelling by car after dusk"	"travelling by car in a snowstorm"	"travelling by car during a downpour"	"navigating a machine through a digital driving simulation"
	24.89 ±0.24	43.83 ±0.17	42.05 ±0.35	**41.86** ±0.10
	24.82	*43.90*	*41.81*	*41.18*
← Irrelevant	"mesmerizing northern lights display"			
	20.05 ±0.77	40.07 ±0.66	38.43 ±0.82	37.98 ±0.31
	"playful dolphins in the ocean"			
	20.11 ±0.31	39.87 ±0.26	38.56 ±0.58	37.05 ±0.31
	"breathtaking view from mountaintop"			
	20.65 ±0.33	42.08 ±0.28	40.05 ±0.52	40.09 ±0.23
	"cheerful sunflower field in bloom"			
	21.10 ±0.50	39.85 ±0.68	40.09 ±0.41	37.93 ±0.55
	"dramatic cliff overlooking the ocean"			
	20.09 ±0.98	38.20 ±0.54	38.48 ±0.37	37.57 ±0.46
	"majestic eagle in flight over mountains"			
	20.70 ±0.38	39.60 ±0.27	40.38 ±0.86	38.52 ±0.21
	20.45	*39.95*	*39.33*	*38.19*

Table 4. **Effect of prompts on PØDA.** We show result for our TrgPrompt (top) as well as ChatGPT-generated relevant prompt (middle) and irrelevant prompt (bottom). Please refer to Sec. 4.3 for details. Best results (**bold**) are always obtained with relevant prompts for which mean mIoU (*italic*) also proves to be better.

tice that our source-only (CLIP ResNet) performs better than SM-PPM source-only (ImageNet pretrained ResNet), demonstrating the overall robustness of the frozen CLIP-based model. In Cityscapes→ACDC, both absolute and relative improvements of PØDA over source-only are greater than the ones of SM-PPM. Overall, PØDA exhibits on par or greater improvements over SM-PPM, despite the fact that our method is purely zero-shot.

Qualitative results on uncommon conditions. Figure 6 shows some qualitative results, training on Cityscapes, and adapting to uncommon conditions never found in datasets because they are either rare (*sandstorm*), dangerous (*fire*), or not labeled (*old movie*). For all, PØDA improves over source-only, which demonstrates its true benefit.

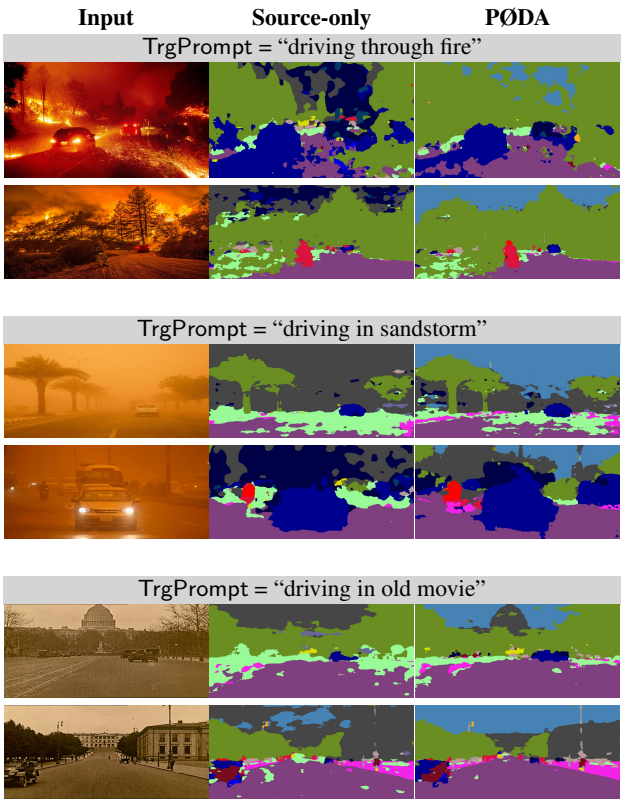

Figure 6. **PØDA on uncommon conditions.** Qualitative results here all use Cityscapes as source and PØDA uses TrgPrompt.

4.3. Ablation studies

TrgPrompt **selection.** Using any meaningful descriptions of the target domain, one should obtain similar adaptation gain with PØDA. To verify this, we generate other relevant prompts by querying ChatGPT[4] with `Give me 5 prompts that have the same exact meaning as [PROMPT]` using same prompts as in Tab. 2. Results in Tab. 4 show that adaptation gains are rather independent of the textual expression. Inversely, we query irrelevant prompts with `Give me 6 random prompts of length from 3 to 6 words describing a random photo`, which could result in negative transfer (See Tab. 4). By chance, small gains could occur; however we conjecture that such gains may originate from generalization by randomization rather than adaptation.

Choice of features to augment. DeepLabV3+ segmenter takes as inputs both low-level features from *Layer1* and high-level features from *Layer4*. In PØDA, we only augment the *Layer1* features and forward them through remaining layers 2-4 to obtain the *Layer4* features. The input to the classifier is the concatenation of both. We study in Tab. 5 if one should augment other features in addition to the ones in

[4]OpenAI's chatbot https://chat.openai.com/

Layer1	*Layer2*	*Layer3*	*Layer4*	ACDC Night
✓	✗	✗	✗	**25.03** ±0.48
✓	✓	✗	✗	23.43 ±0.51
✓	✗	✓	✗	22.93 ±0.53
✓	✗	✗	✓	21.05 ±0.55

Table 5. **Impact of selected layers for augmentation**. Performance (mIoU) of PØDA's day→night adaptation for different choices of ResNet layers for feature augmentation. In addition to augmenting features of *Layer1* (Row 1), one can augment *Layer2* or *Layer3* features (Rows 2-3), or of *Layer4* directly (Row 4).

Method	Night	Snow	Rain	GTA5
src-only*	18.31	39.28	38.20	39.59
PØDA*	**25.03** ±0.48	**43.90** ±0.53	**42.31** ±0.55	**41.07** ±0.48
src-only (*Layer1* ❄)	9.60	30.99	30.89	29.38
PØDA (*Layer1* ❄)	19.43 ±0.69	37.80 ±2.65	40.71 ±1.06	39.09 ±1.23

Table 6. **PØDA when freezing (❄) only *Layer1***. Both models with *, reported in Tab. 2, freeze the whole backbone *Layer1-4*.

Layer1: we observe the best performance with only *Layer1* augmentation. We conjecture that it is important to preserve the consistency between the two inputs to the classifier, *i.e.*, *Layer4* features should be derived from the augmented one from *Layer1*.

Number of mined styles. Our experiments use always $|\mathcal{S}_{s\to t}| = |\mathcal{D}_s|$ but we study the effect of changing the number $|\mathcal{S}_{s\to t}|$ of styles on the target domain performance. By performing ablation on CS→Night with $|\mathcal{S}_{s\to t}| = 1$, 10, 100, 1000, 2975 (i.e., $|\mathcal{D}_s|$), we obtain 16.00 ±5.01, 22.04 ±1.24, 23.90 ±0.96, 24.27 ±0.70, 25.03 ±0.48 respectively. For $|\mathcal{S}_{s\to t}| < |\mathcal{D}_s|$, the styles are sampled randomly from $\mathcal{D}_s$ and results are reported in average on 5 different samplings. Interestingly, we observe that the variance decreases with the increase of $|\mathcal{S}_{s\to t}|$. Results also suggest that only few styles (*e.g.* $|\mathcal{S}_{s\to t}| = 10$) could be sufficient for feature translation, similarly to few-shot image-to-image translation [38], though at the cost of higher variance.

Partial unfreezing of the backbone. While our experiments use a frozen backbone due to the observed good out-of-distribution performance (Tab. 1), we highlight that during training only *Layer1 must* be frozen to preserve its activation space where augmentations are done; the remaining three layers could be optionally fine-tuned. Results in Tab. 6 show that freezing the whole backbone (*i.e.*, *Layer1-4*) achieves the best results. In all cases, PØDA consistently improves the performance over source-only.

4.4. Further discussion

Generalization with PØDA. Inspired by the observation that some unrelated prompts improve performance on target domains (see Tab. 4), we study how PØDA can benefit from general style augmentation. First, we coin "Source-only-G" the generalized source-only model where we augment features by shifting the per-channel $(\boldsymbol{\mu}, \boldsymbol{\sigma})$ with Gaussian noises sampled for each batch of features, such that the signal to noise ratio is 20 dB. This source-only variant takes inspiration from [14] where simple perturbations of feature channel statistics could help achieve SOTA generalization performance in object detection. Tab. 7 shows that Source-only-G always improves over Source-only, demonstrating a generalization capability. When applying our zero-shot adaptation on Source-only-G (denoted "PØDA-G"), target performance again improves – always performing best on the desired target. Performance is further boosted by the style mixing strategy used in [54], *i.e.* source and augmented features statistics being linearly mixed.

Method	Night	Snow	Rain	GTA5
Source-only	18.31	39.28	38.20	39.59
Source-only-G	21.07	42.84	42.38	41.54
PØDA-G	**24.86** ±0.70	44.34 ±0.36	43.17 ±0.63	41.73 ±0.39
PØDA-G+style-mix	24.18 ±0.23	**44.46** ±0.34	**43.56** ±0.46	**42.98** ±0.12

Table 7. **Generalization with PØDA.** Source-only-G model is enhanced with a domain generalization technique. Training PØDA from Source-only-G ('PØDA-G') brings improvements. 'style-mix': style mixing as in [54].

Effect of priors. We now discuss existing techniques that approach zero-shot DA with different priors, revealing the potential combinations of different orthogonal methods. In Tab. 8, we report zero-shot results of CIConv [25] using physics priors, compared against CLIPstyler and PØDA, which use textual priors, *i.e.*, prompts. We also include the one-shot SM-PPM [54] model as the single target sample it requires can be considered as a prior. CIConv, a dedicated physics-inspired layer, is proven effective in enhancing backbone robustness on night scenes. The layer could be straightforwardly included in the CLIP image encoder to achieve the same effect. Albeit interesting, this combination would however require extremely high computational resources to re-train a CLIP variant equipped with CIConv. We leave open such a combination, as well as others like (i) combining image-level (CLIPstyler) and feature-level augmentation (PØDA) or (ii) additionally using style information from one target sample (like in SM-PPM) to help in guiding better the feature augmentation.

Other architectures. We show in Tab. 9 consistent gains brought by PØDA using new backbone (RN101 [17]) and segmenter (semantic FPN [23]).

5. PØDA for other tasks

PØDA operates at the features level, which makes it task-agnostic. We show in the following the effectiveness of our

Method	Prior	ACDC Night
CIConv* [25]	physics	30.60 / 34.50 (Δ=3.90)
SM-PPM [54]	1 target image	13.07 / 14.60 (Δ=1.53)
CLIPstyler [24]	1 prompt	18.31 / 21.38 (Δ=3.07)
PØDA	1 prompt	18.31 / 25.03 (Δ=6.72)

* Results of CIConv are on DarkZurich, a subset of ACDC Night [44].

Table 8. **Effect of different priors for zero-shot/one-shot adaptation.** We report mIoU% for source-only / adapted models, and gain brought by adaptation (Δ in mIoU). Note that [25, 54] use a deeper backbone making results not directly comparable.

Backbone	Method	Night	Snow	Rain	GTA5
Sem. FPN	src-only	18.10	35.75	36.07	40.67
	PØDA	**21.48** ±0.15	**39.55** ±0.13	**38.34** ±0.29	**41.59** ±0.24
DLv3+	src-only	22.17	44.53	42.53	40.49
	PØDA	**26.54** ±0.12	**46.71** ±0.43	**46.36** ±0.20	**43.17** ±0.13

Table 9. **PØDA with different architectures.** Backbones are RN50 for Semantic FPN ('Sem. FPN') and RN101 for DeepLabV3+ ('DLv3+').

method for object detection and image classification.

PØDA for Object Detection. We report in Tab. 10 some results when straightforwardly applying PØDA to object detection. Our Faster-RCNN [40] models, initialized with two backbones, are trained on two source datasets, either Cityscapes or the Day-Clear split in Diverse Weather Dataset (DWD) [53]. We report adaptation results on Cityscapes-Foggy [43] and four other conditions in DWD. For zero-shot feature augmentation in PØDA, we use simple prompts and take the default optimization parameters in previous experiments. PØDA obtains on par or better results than UDA methods [8, 41] (which use target images) and domain generalization methods [14, 47, 53]. We also experimented with YOLOF [7] for object detection in CS→Foggy; PØDA reaches 35.4%, improving 1.5% from the source-only model. These results open up potential combinations of PØDA with generalization techniques like [14] and [47] for object detection.

PØDA for Image Classification. We show that PØDA can be also applied for image classification. We use the same augmentation strategy to adapt a linear probe on top of CLIP-RN50 features. In a first experiment we train a linear classifier on the features of CUB-200 dataset [49] of 200 real bird species; we then perform zero-shot adaptation to classify bird paintings of CUB-200-Paintings dataset [51] using the single prompt "Painting of a bird". In our second experiment, we address the color bias in Colored MNIST [1]; while for training, even and odd digits are colored red and blue respectively, the test digits are randomly colored. We augment training digit features using the "Blue digit" and "Red digit" prompts for even and odd digits respectively, and create a separate set for each one to prevent styles from leaking, *i.e.* to avoid trivially using "red" styles coming from even digits to augment odd digits features and vice versa. Results in Tab. 11 show that PØDA significantly improves over the source-only models.

Method	Target	CS→ CS Foggy	DWD-Day Clear → Night Clear	Dusk Rainy	Night Rainy	Day Foggy
DA-Faster [8]	✓	32.0	-	-	-	-
ViSGA [41]	✓	43.3	-	-	-	-
NP+ [14]	✗	46.3	-	-	-	-
S-DGOD [53]	✗	-	36.6	28.2	16.6	33.5
CLIP The Gap [47]	✗	-	36.9	32.3	18.7	38.5
PØDA	✗	**47.3**	**43.4**	**40.2**	**20.5**	**44.4**

Table 10. **PØDA for object detection (mAP%).** For Cityscapes→Cityscapes-Foggy adaptation, the backbone is ResNet-50, while it is ResNet-101 for adaption from DWD-Day-Clear to other conditions in DWD.

Method	CUB-200 paintings	Colored MNIST
src-only	28.90	55.83
PØDA	**30.91** ±0.69	**64.16** ±0.41

Table 11. **PØDA for image classification (acc%).** The backbone is ResNet-50, and a linear classifier is fit on top of the features. The source domains are CUB-200 (*real* bird images) and colored MNIST with color bias, in second and third columns respectively.

6. Conclusion

In this work, we leverage the powerful zero-shot ability of the CLIP model to make possible a new challenging task of domain adaptation using prompts. We propose a cost-effective feature augmentation mechanism that adjusts the style-specific statistics of source features to synthesize augmented features in the target domain, guided by domain prompts in natural language. Extensive experiments have proven the effectiveness of our framework for semantic segmentation in particular. They also show its applicability to other tasks and various backbones. Our line of research aligns with the collective efforts of the community to leverage large-scale pre-trained models (so-called "foundation models" [3]) for data- and label-efficient training of perception models for real-world applications.

Acknowledgment. This work was partially funded by French project SIGHT (ANR-20-CE23-0016). The authors also thank Ivan Lopes and Fabio Pizzati for their kind proofreading.

References

[1] Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*, 2019. 9

[2] Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *ML*, 2010. 1

[3] Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021. 9

[4] Léon Bottou. Large-scale machine learning with stochastic gradient descent. In *COMPSTAT*, 2010. 5

[5] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE T-PAMI*, 2017. 1

[6] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *ECCV*, 2018. 1, 3, 5

[7] Qiang Chen, Yingming Wang, Tong Yang, Xiangyu Zhang, Jian Cheng, and Jian Sun. You only look one-level feature. In *CVPR*, 2021. 9

[8] Yuhua Chen, Haoran Wang, Wen Li, Christos Sakaridis, Dengxin Dai, and Luc Van Gool. Scale-aware domain adaptive faster r-cnn. *IJCV*, 2021. 9

[9] Bowen Cheng, Maxwell D. Collins, Yukun Zhu, Ting Liu, Thomas S. Huang, Hartwig Adam, and Liang-Chieh Chen. Panoptic-deeplab: A simple, strong, and fast baseline for bottom-up panoptic segmentation. In *CVPR*, 2020. 1

[10] Niv Cohen, Rinon Gal, Eli A Meirom, Gal Chechik, and Yuval Atzmon. "this is my unicorn, fluffy": Personalizing frozen vision-language representations. In *ECCV*, 2022. 2

[11] 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. 5, 6

[12] Guillaume Couairon, Asya Grechka, Jakob Verbeek, Holger Schwenk, and Matthieu Cord. Flexit: Towards flexible semantic image translation. In *CVPR*, 2022. 2

[13] Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *CVPR*, 2021. 2

[14] Qi Fan, Mattia Segu, Yu-Wing Tai, Fisher Yu, Chi-Keung Tang, Bernt Schiele, and Dengxin Dai. Towards robust object detection invariant to real-world domain shifts. In *ICLR*, 2023. 8, 9

[15] Rinon Gal, Or Patashnik, Haggai Maron, Gal Chechik, and Daniel Cohen-Or. StyleGAN-NADA: CLIP-guided domain adaptation of image generators. In *SIGGRAPH*, 2022. 2

[16] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *JMLR*, 2016. 1, 2

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 8

[18] Judy Hoffman, Eric Tzeng, Taesung Park, Jun-Yan Zhu, Phillip Isola, Kate Saenko, Alexei Efros, and Trevor Darrell. Cycada: Cycle-consistent adversarial domain adaptation. In *ICML*, 2018. 1, 2

[19] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *ICCV*, 2017. 4, 5

[20] Krishna Murthy Jatavallabhula, Alihusein Kuwajerwala, Qiao Gu, Mohd Omama, Tao Chen, Shuang Li, Ganesh Iyer, Soroush Saryazdi, Nikhil Keetha, Ayush Tewari, et al. Conceptfusion: Open-set multimodal 3d mapping. *arXiv preprint arXiv:2302.07241*, 2023. 2

[21] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *ICML*, 2021. 2

[22] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *CVPR*, 2019. 2

[23] Alexander Kirillov, Ross Girshick, Kaiming He, and Piotr Dollár. Panoptic feature pyramid networks. In *CVPR*, 2019. 8

[24] Gihyun Kwon and Jong Chul Ye. Clipstyler: Image style transfer with a single text condition. In *CVPR*, 2022. 2, 5, 6, 9

[25] Attila Lengyel, Sourav Garg, Michael Milford, and Jan C van Gemert. Zero-shot day-night domain adaptation with a physics prior. In *ICCV*, 2021. 2, 8, 9

[26] Boyi Li, Kilian Q Weinberger, Serge Belongie, Vladlen Koltun, and René Ranftl. Language-driven semantic segmentation. In *ICLR*, 2022. 2

[27] Guangrui Li, Guoliang Kang, Wu Liu, Yunchao Wei, and Yi Yang. Content-consistent matching for domain adaptive semantic segmentation. In *ECCV*, 2020. 2

[28] Junnan Li, Ramprasaath Selvaraju, Akhilesh Gotmare, Shafiq Joty, Caiming Xiong, and Steven Chu Hong Hoi. Align before fuse: Vision and language representation learning with momentum distillation. *NeurIPS*, 2021. 2

[29] Yunsheng Li, Lu Yuan, and Nuno Vasconcelos. Bidirectional learning for domain adaptation of semantic segmentation. In *CVPR*, 2019. 2

[30] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *CVPR*, 2017. 1

[31] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015. 1

[32] Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I Jordan. Conditional adversarial domain adaptation. In *NeurIPS*, 2018. 2

[33] Yawei Luo, Ping Liu, Tao Guan, Junqing Yu, and Yi Yang. Adversarial style mining for one-shot unsupervised domain adaptation. In *NeurIPS*, 2020. 1, 2, 5

[34] Matthias Minderer, Alexey Gritsenko, Austin Stone, Maxim Neumann, Dirk Weissenborn, Alexey Dosovitskiy, Aravindh Mahendran, Anurag Arnab, Mostafa Dehghani, Zhuoran Shen, et al. Simple open-vocabulary object detection with vision transformers. In *ECCV*, 2022. 2

[35] Yaniv Ovadia, Emily Fertig, Jie Ren, Zachary Nado, David Sculley, Sebastian Nowozin, Joshua Dillon, Balaji Lakshminarayanan, and Jasper Snoek. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. In *NeurIPS*, 2019. 1

[36] Fei Pan, Inkyu Shin, Francois Rameau, Seokju Lee, and In So Kweon. Unsupervised intra-domain adaptation for semantic segmentation through self-supervision. In *CVPR*, 2020. 2

[37] Or Patashnik, Zongze Wu, Eli Shechtman, Daniel Cohen-Or, and Dani Lischinski. Styleclip: Text-driven manipulation of stylegan imagery. In *ICCV*, 2021. 2, 4

[38] Fabio Pizzati, Jean-François Lalonde, and Raoul de Charette. Manifest: Manifold deformation for few-shot image translation. In *ECCV*, 2022. 8

[39] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, 2021. 1, 2, 5

[40] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NeurIPS*, 2015. 9

[41] Farzaneh Rezaeianaran, Rakshith Shetty, Rahaf Aljundi, Daniel Olmeda Reino, Shanshan Zhang, and Bernt Schiele. Seeking similarities over differences: Similarity-based domain alignment for adaptive object detection. In *ICCV*, 2021. 9

[42] Stephan R Richter, Vibhav Vineet, Stefan Roth, and Vladlen Koltun. Playing for data: Ground truth from computer games. In *ECCV*, 2016. 5

[43] Christos Sakaridis, Dengxin Dai, and Luc Van Gool. Semantic foggy scene understanding with synthetic data. *IJCV*, 2018. 9

[44] Christos Sakaridis, Dengxin Dai, and Luc Van Gool. Acdc: The adverse conditions dataset with correspondences for semantic driving scene understanding. In *ICCV*, 2021. 3, 5, 9

[45] Baochen Sun and Kate Saenko. Deep coral: Correlation alignment for deep domain adaptation. In *ECCV*, 2016. 1, 2

[46] Yi-Hsuan Tsai, Wei-Chih Hung, Samuel Schulter, Kihyuk Sohn, Ming-Hsuan Yang, and Manmohan Chandraker. Learning to adapt structured output space for semantic segmentation. In *CVPR*, 2018. 1, 2

[47] Vidit Vidit, Martin Engilberge, and Mathieu Salzmann. Clip the gap: A single domain generalization approach for object detection. *arXiv preprint arXiv:2301.05499*, 2023. 9

[48] Tuan-Hung Vu, Himalaya Jain, Maxime Bucher, Matthieu Cord, and Patrick Pérez. Advent: Adversarial entropy minimization for domain adaptation in semantic segmentation. In *CVPR*, 2019. 1, 2

[49] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. 2011. 9

[50] Jingdong Wang, Ke Sun, Tianheng Cheng, Borui Jiang, Chaorui Deng, Yang Zhao, Dong Liu, Yadong Mu, Mingkui Tan, Xinggang Wang, et al. Deep high-resolution representation learning for visual recognition. *TPAMI*, 2020. 1

[51] Sinan Wang, Xinyang Chen, Yunbo Wang, Mingsheng Long, and Jianmin Wang. Progressive adversarial networks for fine-grained domain adaptation. In *CVPR*, 2020. 9

[52] Yifei Wang, Wen Li, Dengxin Dai, and Luc Van Gool. Deep domain adaptation by geodesic distance minimization. In *ICCV Workshops*, 2017. 2

[53] Aming Wu and Cheng Deng. Single-domain generalized object detection in urban scene via cyclic-disentangled self-distillation. In *ECCV*, 2022. 9

[54] 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, 2, 5, 6, 8, 9

[55] Yanchao Yang and Stefano Soatto. Fda: Fourier domain adaptation for semantic segmentation. In *CVPR*, 2020. 2

[56] Hengshuang Zhao, Xiaojuan Qi, Xiaoyong Shen, Jianping Shi, and Jiaya Jia. Icnet for real-time semantic segmentation on high-resolution images. In *ECCV*, 2018. 1

[57] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *CVPR*, 2017. 1

[58] Chong Zhou, Chen Change Loy, and Bo Dai. Extract free dense labels from clip. In *ECCV*, 2022. 2

[59] Yang Zou, Zhiding Yu, BVK Kumar, and Jinsong Wang. Unsupervised domain adaptation for semantic segmentation via class-balanced self-training. In *ECCV*, 2018. 1

[60] Yang Zou, Zhiding Yu, Xiaofeng Liu, BVK Kumar, and Jinsong Wang. Confidence regularized self-training. In *ICCV*, 2019. 2