

Domain Adaptation with a Single Vision-Language Embedding

Mohammad Fahes^{1*}, Tuan-Hung Vu^{1,2}, Andrei Bursuc^{1,2}, Patrick Pérez³,
Raoul de Charette¹

¹Inria, Paris, France.

²Valeo.ai, Paris, France.

³Kyutai, Paris, France.

*Corresponding author(s). E-mail(s): mohammad.fahes@inria.fr;

Contributing authors: tuan-hung.vu@valeo.com; andrei.bursuc@valeo.com;
patrick@kyutai.org; raoul.de-charette@inria.fr;

Abstract

Domain adaptation has been extensively investigated in computer vision but still requires access to target data at the training time, which might be difficult to obtain in real-world autonomous driving scenarios, especially under rare or adverse conditions. In this paper, we present a new framework for domain adaptation relying on a single Vision-Language (VL) latent embedding instead of full target data. First, leveraging a contrastive language-image pre-training model (CLIP), we propose prompt/photo-driven instance normalization (PIN). PIN is a feature augmentation method that mines multiple visual styles using a single target VL latent embedding, by optimizing affine transformations of low-level source features. The VL embedding can come from a language prompt describing the target domain, a partially optimized language prompt, or a single unlabeled target image. Second, we show that these mined styles (*i.e.*, augmentations) can be used for zero-shot (*i.e.*, target-free) and one-shot unsupervised domain adaptation. Experiments on semantic segmentation in real-world driving datasets, including Cityscapes and ACDC (adverse conditions), demonstrate the effectiveness of the proposed method, which outperforms relevant baselines in the practical zero-shot and one-shot settings.

Keywords: Zero-shot domain adaptation, vision-language models, feature augmentation, style mining, prompt/photo-driven instance normalization

1 Introduction

The success of deep learning in image classification (Krizhevsky et al, 2012) has led to a swift paradigm shift over the past decade. This shift has propelled the rapid evolution of algorithms, with deep neural networks (DNNs) forming the backbone of nearly every modern computer

vision method. For instance, supervised semantic segmentation methods have achieved remarkable success in improving high-resolution predictions (Long et al, 2015; Chen et al, 2017, 2018; Cheng et al, 2020; Wang et al, 2020a), incorporating multi-scale processing (Zhao et al, 2017; Lin et al, 2017) and enhancing computational efficiency (Zhao et al, 2018). Yet, DNN-based methods are still far from being reliably applied in

many scenarios for critical applications like real-world autonomous driving. The reason is that learning-based systems suppose the training and testing data to be independent and identically distributed (i.i.d.), a hypothesis that is violated most of the times as the training data tend to under-represent the true generative distribution. Consequently, 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 (Ovadia et al, 2019).

To mitigate the so-called “domain-shift” problem (Ben-David et al, 2010), unsupervised domain adaptation (UDA) (Ganin et al, 2016; Sun and Saenko, 2016; Hoffman et al, 2018; Zou et al, 2018; Tsai et al, 2018; Vu et al, 2019) has been proposed; and aims at training on labeled data from a *source* domain and unlabeled data from the *target* domain. This alleviates the need for annotating data, which is often laborious, taking 2 to 3 hours per image on average (Cordts et al, 2016; Sakaridis et al, 2021). Moreover, modern architectures can require massive amounts of annotated images (Kirillov et al, 2023). UDA is thus seen as a label-efficient framework for dealing with domain shifts.

Despite its apparent simplicity, even gathering unlabeled data can be challenging under certain conditions. For example, as driving through fire or sandstorm rarely occurs in real life and can be dangerous, capturing these conditions is not 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 unsupervised domain adaptation (OSUDA) (Luo et al, 2020; Wu et al, 2022) methods, *i.e.*, using one target image for training, which substitutes label-efficiency by a more challenging data-efficiency setting.

In this paper, we frame the challenging new task of domain adaptation using a single Vision-Language (VL) latent embedding. At training time, our method adapts the segmentation model to the domain defined by either a single description in natural language (*i.e.*, a *prompt*), a partially optimized prompt, or a single unlabeled

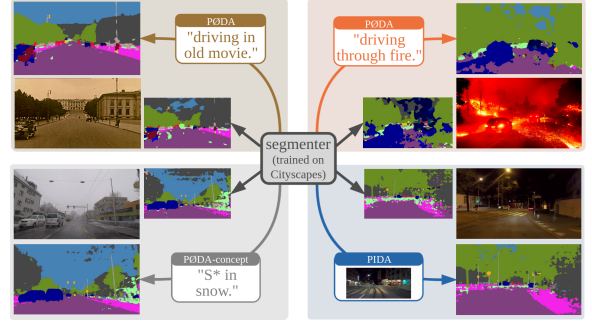


Fig. 1: Domain adaptation using a single vision-language (VL) embedding. The proposed framework adapts a segmenter model (here DeepLabv3+), initially trained on the source domain (Cityscapes), to unseen target conditions using only one embedding in a shared VL space. (*Top*) PØDA performs adaptation via a single text prompt. (*Bottom-left*) PØDA-concept employs a prompt where the concept S^* , optimized from source images, is combined with textual descriptions of target conditions. (*Bottom-right*) PIDA adapts the model using a single unlabeled target image. Smaller segmentation masks adjacent to each test image show predictions from the source-only model for comparison.

target image. This makes the method suitable for zero-shot domain adaptation when the embedding comes from a prompt, and for one-shot unsupervised domain adaptation (OSUDA) when it comes from an unlabeled target image. Since our approach leverages a single VL target embedding which may originate from a prompt or an image, we coin it *Prompt-driven Zero-shot Domain Adaptation* (PØDA) when using a fully verbalized target prompt, *PØDA-concept* when a part of the target prompt is optimized using source images, and *Photo-driven one-shot Domain Adaptation* (PIDA¹) when using a target image.

Fig. 1 outlines the primary goal of our work with a few qualitative examples. Without seeing any fire or old movie images during training, and using a simple description of these conditions with a language prompt (*i.e.*, PØDA), the adapted models succeed in segmenting out critical scene objects, exhibiting fewer errors than the original source-only model. Fig. 1 additionally shows the improvements brought by our framework when part of the prompt is optimized from source data

¹In PIDA the “T” stands for the roman number 1.

(i.e., PØDA-concept) or when the VL embedding comes from an unlabeled target image (i.e., PIDA).

Our method, illustrated in Fig. 2, is made possible by leveraging the VL connections from the seminal CLIP model (Radford et al, 2021). Trained on 400M web-crawled image-text pairs, CLIP has revolutionized multi-modal representation learning, bringing outstanding transfer capability to tasks such as image synthesis (Kwon and Ye, 2022; Gal et al, 2022; Patashnik et al, 2021), multi-modal fusion (Jatavallabhula et al, 2023), semantic segmentation (Li et al, 2022; Zhou et al, 2022a), few-shot learning (Zhou et al, 2022c; Gao et al, 2024; Fahes et al, 2026), as well as open-vocabulary object detection (Minderer et al, 2022).

Our work exploits CLIP latent space and proposes a simple and effective method that converts source-domain embeddings into target-domain ones (Fig. 2, left), by optimizing style-specific components of low-level features. This procedure can be seen as a specific form of feature *augmentation* utilizing a single VL embedding guidance coming from either a prompt or an image. Fine-tuning the segmentation model using the optimized styles (Fig. 2, middle) helps mitigating the distribution gap between the two domains, thus improving the performance on target domains (Fig. 2, right).

This paper extends our previous work (Fahes et al, 2023), PØDA, and proposes a more general formulation. The extension is threefold. (i) We further analyze the effectiveness of relevant prompts in PØDA showing that it originates from the combined description of content (e.g., “driving”) and style (e.g., “at night”). (ii) Inspired by prompt learning (Zhou et al, 2022c,b), we show that rather than encompassing content and style in the prompt, the content *concept* S^* can be optimized from source data and combined with a textual style (e.g., “ S^* at night”). (iii) Since our framework requires only a single VL embedding, we propose PIDA, which utilizes an unlabeled image instead of a prompt, thus allowing adaptation to conditions that are hard to describe with a textual prompt. These novel contributions not only improve the original PØDA but also extend its scope of application.

Overall, our contributions are the following:

- We define a novel task of domain adaptation, which aims at adapting a source-trained model to a target domain provided *only* a single VL embedding, and demonstrate its effectiveness with either a prompt or an image.
- 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 the pixel space. We argue that this is particularly useful for downstream tasks such as semantic segmentation, where good features are decisive for the performance of the segmentation head. Specifically, we introduce prompt/photo-driven instance normalization (PIN), which optimizes affine transformations of the low-level source features such that their embeddings match that of a single image/text condition describing the unseen target domain.
- We propose Prompt-driven Zero-shot Domain Adaptation (Fahes et al, 2023) (PØDA), which utilizes a *single prompt* to describe the content and style of the target condition, demonstrating its performance on a wide variety of scenarios: (i) from clear weather/daytime to adverse conditions (snow, rain, night), (ii) from synthetic to real, (iii) from real to synthetic.
- We introduce the novel PØDA-concept, building on a *concept optimization* strategy to relax the necessity of describing the content of the source images using language. Experiments show that when the optimized concept is coupled with a visual appearance (i.e., style) description, the downstream performance is significantly improved.
- We introduce PIDA which instead utilizes the VL embedding of a *single image* to adapt a trained model. For peculiar visual conditions which are difficult to describe with text prompts, we demonstrate that it further improves the performance of our method.

2 Related works

Unsupervised Domain Adaptation (UDA). The UDA literature is vast and encompasses different yet connected approaches: adversarial learning (Ganin et al, 2016; Tsai et al, 2018), self-training (Zou et al, 2019; Li et al, 2019), entropy minimization (Vu et al,

2019; Pan et al, 2020), generative-based adaptation (Hoffman et al, 2018), etc. The domain gap is commonly reduced at the level of the input (Hoffman et al, 2018; Yang and Soatto, 2020), of the features (Ganin et al, 2016; Sun and Saenko, 2016; Wang et al, 2017; Long et al, 2018) or of the output (Tsai et al, 2018; Vu et al, 2019; Pan et al, 2020).

Recently, the more challenging setting of One-Shot Unsupervised Domain Adaptation (OSUDA) has been proposed. Luo et al (2020) show that traditional UDA methods fail when only a single unlabeled target image is available. To mitigate the risk of overfitting 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 (2022) 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 addressing negative adaptation (Li et al, 2020).

In the more challenging zero-shot setting (where no target image is available), Lengyel et al (2021) tackle day-to-night domain adaptation using physics priors. They introduce a color invariant convolution layer (CICov) that is added to make the network invariant to different lighting conditions. We note that this zero-shot adaptation method is orthogonal to ours and is restricted to a specific type of domain gap.

Text-driven image synthesis. Recently, contrastive vision-language pretraining has shown unprecedented success for multimodal learning in several downstream tasks such as zero-shot classification (Radford et al, 2021), multi-modal retrieval (Jia et al, 2021) and visual question answering (Li et al, 2021a). This encouraged the community to edit images using text descriptions, a task that was previously challenging due to the gap between vision and language representations. For example, StyleCLIP (Patashnik et al, 2021) uses prompts to optimize StyleGAN (Karras et al, 2019) latent vectors and guide the generation process. However, the generation is limited to the training distribution of StyleGAN. To overcome this issue, StyleGAN-NADA (Gal et al, 2022) utilizes CLIP embeddings of text prompts to perform

domain adaptation of the generator, which is in this case trainable.

For text-guided style transfer, CLIP-styler (Kwon and Ye, 2022) 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 (Kwon and Ye, 2022), a U-net (Ronneberger et al, 2015) 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 shallow features of the pre-trained CLIP visual encoder.

Prompt Learning. Inspired by prompt embedding optimization and automatic prompt generation in natural language processing (Shin et al, 2020; Jiang et al, 2020; Zhong et al, 2021), Zhou et al (2022c) apply prompt learning for context optimization in VL models, and show that it outperforms prompt engineering in the few-shot setting for different image recognition downstream tasks. CoOp (for Context Optimization) (Zhou et al, 2022c) consists of replacing the prompt context by learnable vectors, which are optimized by minimizing a classification loss using few labeled support images. The weights of both the image and text encoders are frozen during this process. Subsequently, CoCoOp (Zhou et al, 2022b) adds a lightweight neural network that predicts input-conditional tokens, and improves the performance on unseen classes compared to CoOp. Instead of learning a single prompt, ProDA (Lu et al, 2022) estimates the distribution of prompts in the output embedding space, *i.e.*, modeling the classifier weights distribution by a multivariate Gaussian. Such a strategy handles the variability of visual representations.

In this work, the embeddings of the source training images are used to optimize their common concept in the language space. This differs from strategies like (Zhou et al, 2022c) which fix

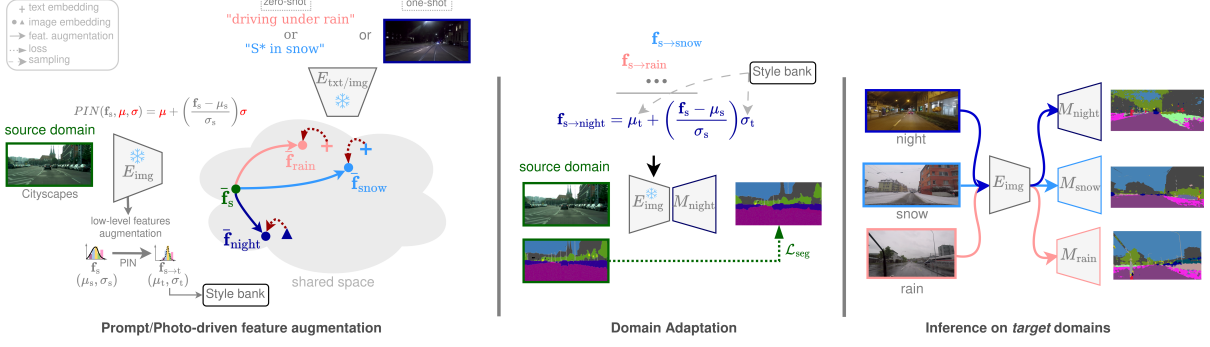


Fig. 2: Overview of our framework for domain adaptation using a single vision-language (VL) embedding. (Left) Given a single VL embedding representing the target domain — either from a text prompt $+$ or a target image $\blacktriangle$ — we use a frozen ResNet encoder initialized with CLIP weights to optimize low-level feature transformations from source to target. The transformation is based on our proposed prompt/photo-driven instance normalization (PIN), which optimizes the two vectors μ and σ shown in red. These are initialized from the source domain (μ_s and σ_s) and optimized such that the latent features better align with the target VL embedding ($+$ or $\blacktriangle$). The resulting parameters (μ_t, σ_t) are saved in a style bank. A detailed description of PIN is provided in Fig. 4. (Middle) Zero-shot or one-shot unsupervised domain adaptation is then achieved by fine-tuning the segmentation model (M) on features that are affinely transformed using styles sampled from the corresponding style bank (e.g., $f_{s \rightarrow \text{night}}$). (Right) This enables inference on previously unseen target domains.

the class name and learn the context in a supervised way using a cross-entropy loss. Our goal is to “search” in the text space for a representation that expresses the meta-domain common to the integrity of our source images, *i.e.*, supposing that, globally, all of them share the same semantic class (*i.e.*, driving scenes).

Normalization in Deep Learning of CNNs. To accelerate and stabilize the training of CNNs, Batch Normalization (BN) was proposed in (Ioffe and Szegedy, 2015). During mini-batch training, BN operation consists of normalizing the batch of features channel-wise and applying a learnable affine transformation. Specifically, BN reads as:

$$\text{BN}(\mathbf{f}) = \gamma \left(\frac{\mathbf{f} - \mu_c(\mathbf{f})}{\sigma_c(\mathbf{f})} \right) + \beta, \quad (1)$$

where $\mathbf{f} \in \mathbb{R}^{B \times C \times H \times W}$ and B, C, H, W are respectively the batch size, the number of channels, the height and the width of the feature activation; $\gamma, \beta \in \mathbb{R}^C$ are learnable parameters; $\mu_c(\cdot)$ and $\sigma_c(\cdot)$ are two functions that return the mean and standard deviation for each channel, computed across the remaining dimensions of $\mathbf{f}$. Specifically, for a channel c :

$$\mu_c(\mathbf{f}) = \frac{1}{BHW} \sum_{b=1}^B \sum_{h=1}^H \sum_{w=1}^W \mathbf{f}_{bchw}, \quad (2)$$

$$\sigma_c(\mathbf{f}) = \sqrt{\frac{1}{BHW} \sum_{b=1}^B \sum_{h=1}^H \sum_{w=1}^W (\mathbf{f}_{bchw} - \mu_c(\mathbf{f}))^2 + \epsilon}. \quad (3)$$

ϵ is a small positive constant used for numerical stability (*e.g.*, $\epsilon = 10^{-5}$). The effectiveness of BN was initially associated to its effect on reducing internal covariate shift, while following research rather argued for its role in smoothing the optimization landscape (Santurkar et al, 2018). Ulyanov et al (2016, 2017) show that replacing BN by Instance Normalization (IN) improves image stylization. IN reads as:

$$\text{IN}(\mathbf{f}) = \gamma \left(\frac{\mathbf{f} - \mu(\mathbf{f})}{\sigma(\mathbf{f})} \right) + \beta, \quad (4)$$

with $\mu(\cdot)$ and $\sigma(\cdot)$ computed across the spatial dimension for each channel c of an instance b in a mini-batch:

$$\mu_{bc}(\mathbf{f}) = \frac{1}{HW} \sum_{h=1}^H \sum_{w=1}^W \mathbf{f}_{bchw}, \quad (5)$$

$$\sigma_{bc}(\mathbf{f}) = \sqrt{\frac{1}{HW} \sum_{h=1}^H \sum_{w=1}^W (\mathbf{f}_{bchw} - \mu_{bc}(\mathbf{f}))^2 + \epsilon}. \quad (6)$$

Leveraging the relation between low-level feature statistics and image style, Adaptive Instance

Normalization (AdaIN) (Huang and Belongie, 2017) transfers style-specific components across features. In AdaIN, the styles are represented by the channel-wise mean $\mu(\mathbf{f}) \in \mathbb{R}^C$ and standard deviation $\sigma(\mathbf{f}) \in \mathbb{R}^C$ of features. Styling a source feature $\mathbf{f}_s$ with an arbitrary target style $(\mu(\mathbf{f}_t), \sigma(\mathbf{f}_t))$ reads:

$$\text{AdaIN}(\mathbf{f}_s, \mu(\mathbf{f}_t), \sigma(\mathbf{f}_t)) = \sigma(\mathbf{f}_t) \left(\frac{\mathbf{f}_s - \mu(\mathbf{f}_s)}{\sigma(\mathbf{f}_s)} \right) + \mu(\mathbf{f}_t), \quad (7)$$

where $\mu(\cdot)$ and $\sigma(\cdot)$ are defined as in Eq. (5) and Eq. (6). Note that there are no learnable parameters in Eq. (7).

These normalization methods, sharing similar mathematical formulations, have all different goals. In this paper, we propose prompt/photo-driven instance normalization (PIN), which is inspired by AdaIN, yet supposes having no access to $\mu(\mathbf{f}_t)$ and $\sigma(\mathbf{f}_t)$.

3 Domain Adaptation with a single VL Embedding

Our framework, illustrated in Fig. 2, builds upon CLIP (Radford et al, 2021), a Vision-Language model pre-trained on 400M image-text pairs crawled from the Internet. CLIP trains jointly an image encoder E_{img} and a text encoder E_{txt} to learn an expressive representation space that effectively bridges the two modalities. We use such bimodal space in this work to bring image embeddings from a source domain closer to a target domain defined by a single VL embedding, coming either from a textual description (*i.e.*, zero-shot) or an unlabeled target image (*i.e.*, one-shot). This is done by optimizing the style-specific components of the source low-level features, with the final values of these components corresponding to augmented data.

For zero-shot, we explore two types of guidance without access to target data: using either a full textual prompt or a textual prompt combined with an optimized concept. For the former, we explore the use of simple general prompts describing the target domain, *e.g.*, “driving at night” for the night domain. Because simple prompts may not be sufficient to represent complex semantics of scenes, we also suggest optimizing a concept S^* from the source images, inspired by context

optimization (Zhou et al, 2022c). To form the target prompt, this concept is complemented with our target description, *e.g.*, “ S^* at night”. Our zero-shot domain adaptation framework can interchangeably leverage any of these guidance signals.

For one-shot, we simply replace the single language embedding by the embedding of a single unlabeled target image. The optimization thus operates in the image embedding space, *i.e.*, the space defined by the image encoder. We later show that this setting might be useful when the target domain conditions are nuanced and not trivial to verbalize.

Overall, our goal is to optimize affine transformations of low-level source features so that the embeddings of these features become closer to their imaginary counterparts in the targeted domain (Fig. 2, left), while crucially preserving their semantic content. The learned augmentations can then be applied to synthesize, in a zero-shot/one-shot fashion, features that correspond to the target domain and can subsequently be used to fine-tune the model (Fig. 2, middle). This ultimately allows inference on domains only described by either a single prompt or an image at training time (Fig. 2, right).

Our approach faces several challenges: (i) How to “mine” style information relative to a target domain using only one VL embedding? (ii) How to preserve pixel-wise semantics (*i.e.*, content) during style mining? (iii) Based on such mined styles, how to adapt the source model to the target domain? We address these questions in the following.

Problem formulation. Our main task is semantic segmentation, that is, pixel-wise classification of input images into semantic segments. We start from a K -class segmentation model M , trained on a labeled source domain dataset $\mathcal{D}_s = \{(\mathbf{I}_s, \mathbf{y}_s) \mid \mathbf{I}_s \in \mathbb{R}^{H \times W \times 3}, \mathbf{y}_s \in \{0, 1\}^{H \times W \times K}\}$, where $\mathbf{I}_s$ represents an image and $\mathbf{y}_s$ its ground-truth annotation. The segmenter M is composed of a 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} : $M = (M_{\text{feat}}, M_{\text{cls}})$. Our goal is to adapt the model M such that its performance on a test target dataset $\mathcal{D}_t = \{\mathbf{I}_t \mid \mathbf{I}_t \in \mathbb{R}^{H \times W \times 3}\}$ is improved, provided solely a single VL latent embedding as information about the target

$M_{\text{feat}} *$	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 performance of source-only models. Mean Intersection-over-Union (mIoU %) is reported on the “night”, “snow”, and “rain” subsets of the ACDC (Sakaridis et al, 2021) *validation set*, as well as on a subset of 1000 images from GTA5. Models are trained on Cityscapes (‘CS’). ‘ $M_{\text{feat}} *$ ’ indicates a frozen backbone.

domain. This is enabled by access to a target embedding, $\bar{\mathbf{f}}_t$, which is derived either from a prompt using the frozen text encoder, or from a single unlabeled target image using the frozen image encoder.

We train M in a supervised manner for the semantic segmentation task on the source domain. Interestingly, we empirically show in Table 1 that keeping the feature extractor M_{feat} frozen helps alleviate the overfitting risk to the source domain in favor of generalization; not a universal finding, this observation is limited to our experimental setting specified in Section 4.1. To minimize the interference of generalization effects caused by different training strategies in the adaptation results, we systematically freeze M_{feat} when training on source and later fine-tuning for adaptation.

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 feature activations extracted by M_{feat} and $\bar{\mathbf{f}}$ their corresponding CLIP embeddings. In Fig. 4 we illustrate the difference between $\mathbf{f}$ and $\bar{\mathbf{f}}$.

Overview of the proposed method. Our solution is to mine styles (*i.e.* channel-wise statistics of the features) using source-domain low-level features set $\mathcal{F}_s = \{\mathbf{f}_s | \mathbf{f}_s = \text{feat-ext}(M_{\text{feat}}, \mathbf{I}_s)\}$ and $\bar{\mathbf{f}}_t$. For generality, **feat-ext**($\cdot$) can pull features from any desired layer. However, we later show that using the features from the earliest layers yields the best results.

The **augment**($\cdot$) operation, depicted in Fig. 4, augments the style-specific components of $\mathbf{f}_s$ with guidance from $\bar{\mathbf{f}}_t$, synthesizing $\mathbf{f}_{s \rightarrow t}$ with style information from the target domain. We emphasize that the features $\mathbf{f}_s$ and $\mathbf{f}_{s \rightarrow t}$ have the same size and identical semantic 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} , leading to the final adapted model.

3.1 Instance Normalization guided by single VL embedding

We draw inspiration from Adaptive Instance Normalization (AdaIN) (Huang and Belongie, 2017), an elegant formulation for transferring style-specific components across features. AdaIN is defined in Eq. (7).

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.

3.1.1 Prompt-driven Instance Normalization

In the zero-shot setting, we suppose having no access to any image from the target domain. Consequently, $\mu(\mathbf{f}_t)$ and $\sigma(\mathbf{f}_t)$ in Eq. (7) are unknown. Yet, we suppose having a single text condition in natural language (*i.e.*, a prompt P) describing the target domain. Thus, we propose Prompt-driven Instance Normalization (PIN) as

$$\text{PIN}_{\bar{\mathbf{f}}_t}(\mathbf{f}_s, \mu, \sigma) := \sigma \left(\frac{\mathbf{f}_s - \mu(\mathbf{f}_s)}{\sigma(\mathbf{f}_s)} \right) + \mu, \quad (8)$$

where $\mu \in \mathbb{R}^C$ and $\sigma \in \mathbb{R}^C$ are optimizable variables driven by $\bar{\mathbf{f}}_t = E_{\text{txt}}(P)$.

We aim to augment source image features $\mathcal{F}_s$ such that they capture the style of the unseen 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.

3.1.2 Concept optimization

In domain adaptation, despite the discrepancies between source and target data, they still share common characteristics (*e.g.*, “driving” data). Interestingly, our experiments show that prompt-based adaptation performs better when the prompt includes a word describing the common characteristics (*e.g.*, the word “driving” in “driving at night”, “driving under rain”, *etc.*), though this improvement comes at the cost of

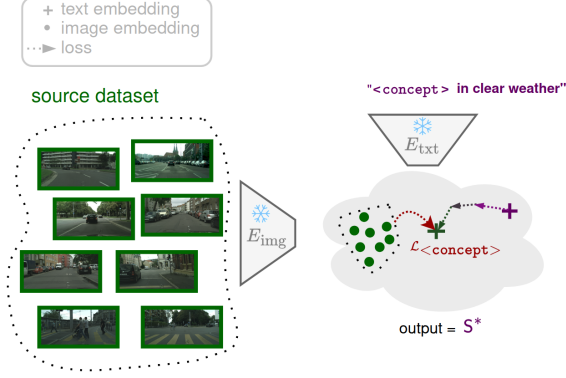


Fig. 3: Concept optimization. A learnable $\langle \text{concept} \rangle$ token is optimized in the word embedding space using a cosine distance objective $\mathcal{L}_{\langle \text{concept} \rangle}$, encouraging the text embedding to align closely with embeddings of source images. The final optimized word embedding is denoted by S^* .

prompt engineering. To prevent this, we draw inspiration from CoOp (Zhou et al, 2022c) and seek to optimize a word embedding describing the common characteristics. While similar in spirit, our approach differs from CoOp in methodology and implementation as the latter requires access to target data and their labels. Instead, we optimize a concept using only *source* images and a textual style description as guidance (e.g., “in clear weather”).

In practice, we apply concept optimization in two steps. In the first step, we construct a prompt P in the form of $\langle \text{concept} \rangle + \text{“in clear weather”}$, where $\langle \text{concept} \rangle$ represents optimizable parameters in the word embedding space. The optimization is done using SGD with the following objective:

$$\mathcal{L}_{\langle \text{concept} \rangle} = 1 - \frac{E_{\text{txt}}(P) \cdot E_{\text{img}}(\mathbf{I}_s)}{\|E_{\text{txt}}(P)\| \|E_{\text{img}}(\mathbf{I}_s)\|}. \quad (9)$$

The word embedding is optimized by minimizing the cosine similarity between the representations of the source images and the prompt representation. The final value of $\langle \text{concept} \rangle$ is denoted S^* . In the second step, the target prompt P^* is constructed by concatenating S^* with target-specific texts, such as “ S^* at night” or “ S^* under rain”. Our proposed *concept optimization* is illustrated in Fig. 3. We then apply PIN (Eq. (8)), though using as $\bar{\mathbf{f}}_t$ the VL embedding of P^* , i.e., $\bar{\mathbf{f}}_t = E_{\text{txt}}(P^*)$.

Algorithm 1: Style Mining (see Fig. 4)

Input : Set $\mathcal{F}_s$ of source image features.
 VL embedding $\bar{\mathbf{f}}_t$.
Param : Number N of optimization steps.
 Learning rate lr and momentum m of gradient descend (GD).
Output: Set $\mathcal{S}_{s \rightarrow t}$ of target styles.

```

1  $\mathcal{S}_{s \rightarrow t} \leftarrow \emptyset$ 
  foreach  $\mathbf{f}_s \in \mathcal{F}_s$  do
2    $\mu^0 \leftarrow \text{mean}(\mathbf{f}_s)$ 
3    $\sigma^0 \leftarrow \text{std}(\mathbf{f}_s)$ 
  // Optimization
4   for  $i = 1, 2, \dots, N$  do
5      $\mathbf{f}_{s \rightarrow t}^i \leftarrow \text{PIN}_{\bar{\mathbf{f}}_t}(\mathbf{f}_s, \mu^{i-1}, \sigma^{i-1})$ 
6      $\bar{\mathbf{f}}_{s \rightarrow t}^i \leftarrow \text{get-embedding}(\mathbf{f}_{s \rightarrow t}^i)$ 
7      $\mu^i \leftarrow \text{GD}_m^{lr}(\mu^{i-1}, \nabla_{\mu} \mathcal{L}_{\mu, \sigma}(\bar{\mathbf{f}}_{s \rightarrow t}^i, \bar{\mathbf{f}}_t))$ 
8      $\sigma^i \leftarrow \text{GD}_m^{lr}(\sigma^{i-1}, \nabla_{\sigma} \mathcal{L}_{\mu, \sigma}(\bar{\mathbf{f}}_{s \rightarrow t}^i, \bar{\mathbf{f}}_t))$ 
9   end
10   $(\mu_t, \sigma_t) \leftarrow (\mu^N, \sigma^N)$ 
11   $\mathcal{S}_{s \rightarrow t} \leftarrow \mathcal{S}_{s \rightarrow t} \cup \{(\mu_t, \sigma_t)\}$ 
  end

```

3.1.3 Photo-driven Instance Normalization

Within our framework, $\bar{\mathbf{f}}_t$ is simply an embedding guiding the optimization of μ and σ . While we have explored the use of text to obtain $\bar{\mathbf{f}}_t$, certain peculiar conditions may be challenging to describe this way. Alternatively, our framework also allows getting the VL embedding from a single unlabeled target image $\mathbf{I}_t$.

In such case, PIN (Eq. (8)) is applied using the embedding of the target image (i.e., $\bar{\mathbf{f}}_t = E_{\text{img}}(\mathbf{I}_t)$) as guidance. In this setting, PIN refers to *Photo-driven Instance Normalization*.

3.2 Style mining with PIN

In the sections above, we described how PIN can leverage either a target prompt (Section 3.1.1), the combination of a source concept and a target prompt (Section 3.1.2), or a target image (Section 3.1.3). We now detail how to mine a collection of styles.

We describe in Algorithm 1 the first step of our feature augmentation procedure: mining the set $\mathcal{S}_{s \rightarrow t}$ of styles in targeted domain. For each source feature activation $\mathbf{f}_s \in \mathcal{F}_s$, we want to mine style statistics corresponding to an imaginary target feature activation $\mathbf{f}_t$. To this end, we formulate style mining as an optimization problem over the original source feature $\mathbf{f}_s$, i.e., optimizing (μ, σ)

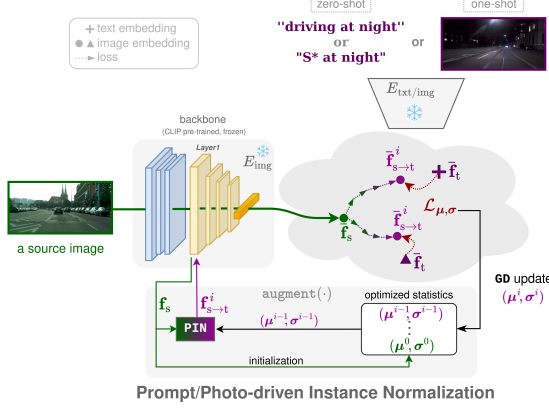


Fig. 4: Target style mining from a source image. This figure illustrates the optimization loop described in Algorithm 1. A source image is passed through the CLIP image encoder E_{img} to extract low-level features $\mathbf{f}_s$ and the corresponding CLIP embedding $\mathbf{f}_s$. At each optimization step i , $\text{augment}(\cdot)$ injects the previous style parameters $(\mu^{i-1}, \sigma^{i-1})$ into the features $\mathbf{f}_s$ using the PIN layer, producing the stylized features $\mathbf{f}_{s \rightarrow t}^i$ and their embedding $\tilde{\mathbf{f}}_{s \rightarrow t}^i$. The loss $\mathcal{L}_{\mu, \sigma}$ is computed as the cosine distance between $\tilde{\mathbf{f}}_{s \rightarrow t}^i$ and the target embedding $\tilde{\mathbf{f}}_t$, which can come from a text prompt (e.g., “driving at night”), a partially optimized prompt (e.g., “S* at night”), or an unlabeled target image. Gradients from this loss update the style parameters to (μ^i, σ^i) via gradient descent.

in Eq. (8). The optimization objective is defined as the cosine distance in the CLIP latent space between the CLIP embedding $\tilde{\mathbf{f}}_{s \rightarrow t}$ of the stylized feature $\mathbf{f}_{s \rightarrow t} = \text{PIN}(\mathbf{f}_s, \mu, \sigma)$ and the VL embedding $\tilde{\mathbf{f}}_t$ of target domain:

$$\mathcal{L}_{\mu, \sigma}(\tilde{\mathbf{f}}_{s \rightarrow t}, \tilde{\mathbf{f}}_t) = 1 - \frac{\tilde{\mathbf{f}}_{s \rightarrow t} \cdot \tilde{\mathbf{f}}_t}{\|\tilde{\mathbf{f}}_{s \rightarrow t}\| \|\tilde{\mathbf{f}}_t\|}. \quad (10)$$

This CLIP-space cosine distance, already used in prior text-driven image editing works (Patashnik et al, 2021), aims to steer the stylized features in the direction of the target VL embedding. One step of the optimization is illustrated in Fig. 4. In practice, we run several such steps per image (i.e., instance) leading to the mined target style denoted (μ_t, σ_t) .

As there might be a variety of styles in a target domain, our mining populates the $\mathcal{S}_{s \rightarrow t}$ set with as many variations of target style as there are source images, hence $|\mathcal{S}_{s \rightarrow t}| = |\mathcal{D}_s|$.

Intuitively, our simple augmentation strategy can be seen as a cost-efficient way to cover the distribution of the target domain by starting from

Algorithm 2: VL-guided UDA

Input: Source dataset $\mathcal{D}_s = \{(\mathbf{I}_s, \mathbf{y}_s)\}$.
CLIP encoders E_{img} and E_{txt} .
Target VL embedding $\tilde{\mathbf{f}}_t$.
Feature backbone $M_{\text{feat}} \leftarrow E_{\text{img}}$.
Source model: $M = (M_{\text{feat}}, M_{\text{cls}})$.
Result: Target-adapted model $M' = (M_{\text{feat}}, M'_{\text{cls}})$.

// Initialization
1 $M \leftarrow \text{train}(M, \mathcal{D}_s)$
// Feature Augmentation
2 $\mathcal{F}_s \leftarrow \text{feat-ext}(M_{\text{feat}}, \{\mathbf{I}_s\})$
3 $\mathcal{S}_{s \rightarrow t} \leftarrow \text{augment}(\mathcal{F}_s, \tilde{\mathbf{f}}_t)$
// Adaptation
4 $M' \leftarrow \text{fine-tune}(M, \mathcal{F}_s, \mathcal{S}_{s \rightarrow t}, \{\mathbf{y}_s\})$

different anchor points in CLIP latent space coming from the source images and steering them in the direction of the target VL embedding. This mitigates the diversity problem discussed in one-shot feature augmentation (Luo et al, 2020; Wu et al, 2022).

3.3 Fine-tuning for Adaptation

For adaptation, at each training iteration, we stylize the source features using a mined target style (μ_t, σ_t) randomly selected from $\mathcal{S}_{s \rightarrow t}$. The augmented features are computed as $\mathbf{f}_{s \rightarrow t} = \text{AdaIN}(\mathbf{f}_s, \mu_t, \sigma_t)$ 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 (Huang and Belongie, 2017), we can still use the labels $\mathbf{y}_s$ to train the classifier in a supervised manner with a standard segmentation loss (i.e., cross-entropy). 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 $M' = (M_{\text{feat}}, M'_{\text{cls}})$.

Algorithm 2 presents the high-level pseudocode for VL-guided adaptation: from source-only training as model initialization, to VL embedding-driven feature augmentation, to zero-shot/one-shot model adaptation.

In PØDA and PØDA-concept, M' is evaluated on images with conditions and styles that were never seen during any of the training stages yet described by a single prompt, while in PIDA it is

evaluated on the validation set of the same dataset from which the target image was sampled.

4 Experiments

4.1 Implementation details

We use the DeepLabv3+ architecture (Chen et al, 2018) with the backbone M_{feat} initialized from the image encoder E_{img} of the pre-trained CLIP-ResNet-50 model.²

Source-only training. The network is trained for 200k iterations on random 768×768 crops with a batch size of 2. We use a polynomial learning rate schedule with initial $lr=10^{-1}$ for the classifier and $lr=10^{-4}$ for backbone when it is not frozen (see Table 1). Optimization is done with Stochastic Gradient Descent (SGD) (Bottou, 2010), momentum 0.9 and weight decay 10^{-4} . We apply standard color jittering and horizontal flip to crops.

Concept optimization. S^* is optimized from the source images using SGD for 10 epochs, with a batch size of 16. The learning rate is set to 10^{-4} .

PIN optimization. For the style mining step, we use the source feature activations after the first layer (*Layer1*): $\mathbf{f}_s \in \mathbb{R}^{192 \times 192 \times 256}$. The style parameters μ and σ are 256-dim real vectors. The CLIP embeddings are 1024-dim vectors for CLIP-ResNet-50 backbone, and 512-dim vectors for CLIP-ResNet-101. We adopt the Imagenet templates from (Radford et al, 2021) to encode the target descriptions P . For PØDA, PØDA-concept and PIDA, PIN is optimized for 100 iterations using Gradient Descent (GD) on batches of 16 feature instances with a learning rate $lr=1.0$.

Classifier fine-tuning. Starting from the source-only trained model, we fine-tune the classifier M_{cls} on batches of 8 augmented features $\mathbf{f}_{s \rightarrow t}$ 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 (Cordts et al, 2016), composed of 2,975 training and 500 validation images featuring 19 semantic classes. Though we adapt towards a prompt or an image *not* a dataset, we need ad-hoc datasets to test on. We report main results using ACDC (Sakaridis

et al, 2021) because it has urban images captured in adverse conditions. We also study the applicability of our method to the two settings of real→synthetic (Cityscapes as source, and evaluating on GTA5 (Richter et al, 2016)) 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. We always report the mean and standard deviation over five models trained with different random seeds.

4.2 PØDA

We consider the following adaptation scenarios: day→night, clear→snow, clear→rain, real→synthetic and synthetic→real. We report zero-shot adaptation results of PØDA in the addressed set-ups, comparing against two main baselines: CLIPstyler (Kwon and Ye, 2022) for zero-shot style transfer and SM-PPM (Wu et al, 2022) for one-shot UDA. Both PØDA and CLIPstyler models do not see any target images during training. In this study, we arbitrarily choose a simple prompt to describe each domain. We show later in Section 4.2.1 more results using other relevant prompts with similar meanings – showing that our adaptation gain exhibits only a slight sensitivity to prompt selection. For SM-PPM, one random target image from the training set is used.

Comparison to CLIPstyler. CLIPstyler (Kwon and Ye, 2022) 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 CLIPstyler³ 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. 5 which shows stylized examples from CLIPstyler

²<https://github.com/openai/CLIP>

³We use official code <https://github.com/cyclomon/CLIPstyler> and follow the recommended configs.

with PØDA target prompts. Zooming in, we note that stylization of snow or game added snowy roads or Atari game *on the buildings*, respectively.



Fig. 5: CLIPstyler (Kwon and Ye, 2022) stylization. An example Cityscapes image stylized using ad-hoc target prompts. The resulting translated images exhibit visible artifacts, which may negatively affect domain adaptation performance, *e.g.*, in the rain scenario shown in Table 2.

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 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 note that PØDA does not yield equal improvements over the source-only model across all domains, nor does CLIPstyler. This phenomenon is common in domain adaptation, where the effectiveness might depend on multiple factors, including the architecture, the pretraining strategy, the source domain, the target domain, and the adaptation method itself. This observation holds in unsupervised domain adaptation (Zou et al, 2018; Hoyer et al, 2022; Ganin and Lempitsky, 2015; Kang et al, 2019; Xiao and Zhang, 2021) and domain generalization (Zhou et al, 2021; Wang

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

Table 2: Zero-shot domain adaptation for semantic segmentation. We report mIoU (%) performance of PØDA compared to CLIPstyler (Kwon and Ye, 2022) and the source-only baseline. Results are grouped by source domain and target prompts (P). ‘CS’ denotes Cityscapes dataset (Cordts et al, 2016).

et al, 2021b; Li et al, 2021b; Choi et al, 2021; Lee et al, 2022).

Domains like night exhibit a larger gap from the source domain (*i.e.*, day and clear weather) due to significant illumination variations and increased noise. This gap is reflected in the low performance of source-only model on night images (*cf.* Table 2). Our method is particularly effective at addressing this domain, leading to a performance gain of $\approx +7$ mIoU over the source-only model. This improvement is enabled by the use of a simple prompt such as “driving at night”, which helps bridge the gap by aligning low-level feature statistics. In contrast, domains like snow and rain, which share more visual cues with the source domain, exhibit a smaller domain gap, resulting in $\approx +4\%$ mIoU gain.

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 $\approx 615k$ parameters for synthesizing the stylized images. We draw on the simplicity of PØDA to explain why it is more advantageous than CLIPstyler for downstream tasks like semantic segmentation: the minimal statistics changes help

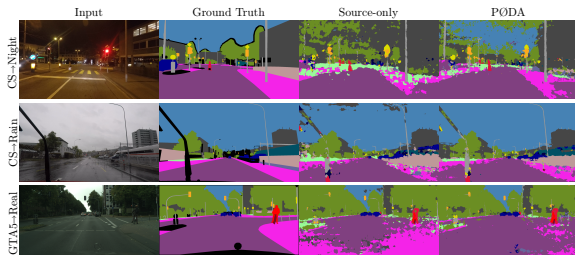


Fig. 6: Qualitative results. (Columns 1-2) Input images and their corresponding ground-truth segmentation; (Columns 3-4) Segmentation predictions from the source-only model and the proposed PØDA model, respectively.

avoiding significant drifts on the feature manifold which may otherwise result in unwanted errors.

Computation time. We measure the average computation time of PØDA using a ResNet-50 backbone and a DeepLabV3+ decoder on an RTX 2080TI GPU, showing that our method incurs a minimal overhead. In details, style mining with PIN (Section 3.2) takes 0.3 second per instance, being $200\times$ faster than CLIPStyler, and resulting in ≈ 15 minutes for the whole Cityscapes dataset. Fine-tuning for adaptation (Section 3.3) with our stylization takes 14 minutes for the 2k iterations – being 30 seconds longer than Empirical Risk Minimization (ERM). Finally, our method adds no computational overhead at inference time, as it does not introduce any additional parameters. Inference on a full-resolution ACDC image takes ≈ 0.11 seconds on average.

Qualitative results. We show in Fig. 6 qualitative examples of predictions from source-only and PØDA models.

Comparison to one-shot UDA (OSUDA). We also compare PØDA against SM-PPM (Wu et al, 2022),⁴ a state-of-the-art OSUDA method, see Table 3. The OSUDA setting allows the access to a single unlabeled target domain image for domain adaptation. 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

Source	Target eval.	1-shot SM-PPM	0-shot PØDA
CS	ACDC Night	13.07 / 14.60 ($\Delta=1.53$)	18.31 / 25.03 ($\Delta=6.72$)
	ACDC Snow	32.60 / 35.61 ($\Delta=3.01$)	39.28 / 43.90 ($\Delta=4.62$)
	ACDC Rain	29.78 / 32.23 ($\Delta=2.45$)	38.20 / 42.31 ($\Delta=4.11$)
GTA5	CS	30.48 / 39.32 ($\Delta=8.84$)	36.38 / 40.08 ($\Delta=3.70$)

Table 3: Comparison with one-shot UDA. We compare PØDA to SM-PPM (Wu et al, 2022), a one-shot unsupervised domain adaptation method. The table reports semantic segmentation performance in terms of mIoU (%) for both source-only / adapted models, along with the improvement from adaptation (Δ mIoU). For adaptation, SM-PPM (using DeepLabv2 with a ResNet-101 backbone) leverages a single target image, while PØDA (using DeepLabv3+ with a ResNet-50 backbone) relies on a target prompt and a text encoder.

not directly comparable due to the differences in backbone (ResNet-101 in SM-PPM *vs.* ResNet-50 in PØDA) and in segmentation head (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 notice 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 $\rightarrow$ 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 7 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.

4.2.1 Ablation studies

Effect of prompts on PØDA. Using any meaningful descriptions of the target domain as prompt P, one should obtain similar adaptation gain with PØDA. To verify this, we generate

⁴We use official code <https://github.com/W-zx-Y/SM-PPM>

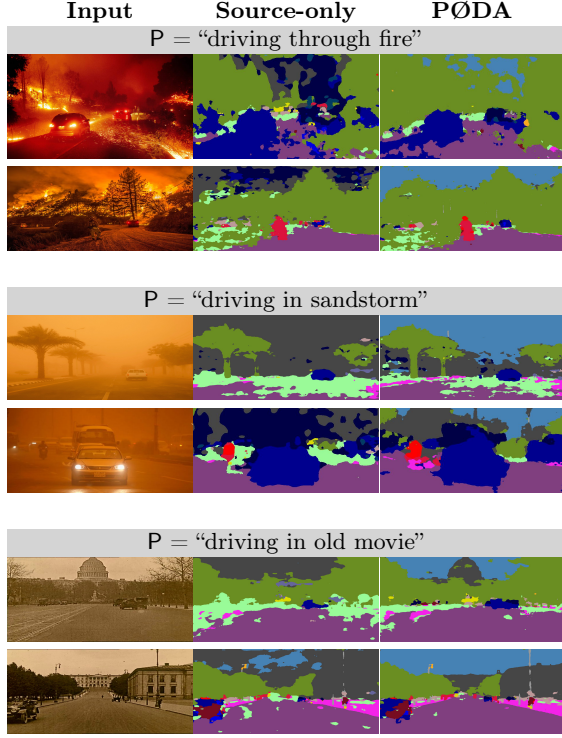


Fig. 7: Qualitative results on uncommon conditions. The source-only model is trained on Cityscapes and PØDA uses the single prompt P on top of the source-only model.

other **relevant prompts** by querying ChatGPT⁵ with Give me 5 prompts that have the same exact meaning as [PROMPT] using same prompts as in Table 2. Results in Table 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 Table 4). In some cases, small gains could occur; however we conjecture that such gains may originate from generalization by randomization rather than adaptation. The generated **relevant prompts** cover a broad spectrum of conceptually synonymous expressions. For instance, the term “driving” may be replaced with phrases such as “operating a vehicle”, “navigating the roads”, “traveling by car”. Similarly, stylistic variations like “at night” are reformulated as “after

Method	ACDC Night	ACDC Snow	ACDC Rain	GTA5
Source only	18.31	39.28	38.20	39.59
P	“driving at night” 25.03 ± 0.48	“driving in snow” 43.90 ± 0.53	“driving under rain” 42.31 ± 0.55	“driving in a game” 41.07 ± 0.48
Relevant ChatGPT-generated	“operating a vehicle after sunset” 24.38 ± 0.37	“operating a vehicle in snowy conditions” 44.33 ± 0.36	“operating a vehicle in wet conditions” 42.21 ± 0.47	“piloting a vehicle in a virtual world” 41.25 ± 0.40
	“driving during the nighttime hours” 25.22 ± 0.64	“driving on snow-covered roads” 43.56 ± 0.62	“driving on rain-soaked roads” 42.51 ± 0.33	“controlling a car in a digital simulation” 41.19 ± 0.14
	“navigating the roads in darkness” 24.73 ± 0.47	“piloting a vehicle in snowy terrain” 44.67 ± 0.18	“navigating through rainfall while driving” 41.11 ± 0.69	“maneuvering a vehicle in a computerized racing experience” 40.34 ± 0.49
	“driving in low-light conditions” 24.68 ± 0.34	“driving in wintry precipitation” 43.11 ± 0.56	“driving in inclement weather” 40.68 ± 0.37	“operating a transport in a video game environment” 41.34 ± 0.42
	“travelling by car after dusk” 24.89 ± 0.24	“travelling by car in a snowstorm” 43.83 ± 0.17	“travelling by car during a downpour” 42.05 ± 0.35	“navigating a machine through a digital driving simulation” 41.86 ± 0.10
	<i>24.82</i>	<i>43.90</i>	<i>41.81</i>	<i>41.18</i>
	“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
Irrelevant	“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
	<i>20.45</i>	<i>39.95</i>	<i>39.33</i>	<i>38.19</i>

Table 4: Effect of prompts on PØDA. This table presents results for three types of prompts: our hand-crafted prompt P (*top*), **relevant prompt** generated by ChatGPT (*middle*), and **irrelevant prompt** also generated by ChatGPT (*bottom*). For more details, refer to Section 4.2.1. The best results (in **bold**) are consistently achieved with **relevant prompts**, which also yield higher average mIoU scores (*italic*).

“sunset”, “in darkness”, or “in low-light conditions”. As shown in Table 4, these alternative phrasings maintain the effectiveness of PØDA, provided that they convey the same underlying semantic meaning. In contrast, **irrelevant prompts**—those with semantically unrelated content—demonstrate noticeably reduced performance.

Effect of prompt length on PØDA. In addition to semantic consistency, we investigate the effect of prompt length by generating a new set of **relevant prompts** querying ChatGPT with: Give me 5 prompts that have the same exact meaning as [PROMPT] with increasing length. Results presented in Table 5 indicate consistent improvements over the source-only

⁵OpenAI’s chatbot <https://chatgpt.com/>

Method	ACDC Night	ACDC Snow	ACDC Rain	GTA5
Source only	18.31	39.28	38.20	39.59
Longer ← ChatGPT-generated P	"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
	"night drive"	"snowy drive"	"rainy drive"	"game driving"
	24.97 ± 0.52	42.61 ± 0.53	41.26 ± 0.52	41.03 ± 0.40
	"driving through the night"	"driving through snow"	"driving through rain"	"driving in a video game"
	24.71 ± 0.61	42.92 ± 0.28	42.43 ± 0.37	40.85 ± 0.45
	"operating a vehicle after dark"	"navigating a vehicle in snowy conditions"	"navigating a vehicle in rainy weather"	"controlling a car in a virtual game"
	25.63 ± 0.75	43.52 ± 0.26	42.19 ± 0.45	40.67 ± 0.42
	"cruising down the road in the nighttime"	"carefully driving a car while snow is falling"	"driving a car while raindrops fall on the windshield"	"simulating vehicle driving within a video game"
	24.87 ± 0.38	43.33 ± 0.52	41.56 ± 0.50	41.07 ± 0.51
	"traveling by car under the stars during the late hours of the night"	"Operating a vehicle on snow-covered roads during a winter storm"	"traveling by car on wet roads as steady rain pours from the sky"	"engaging in interactive car driving as part of a gaming experience"
	25.82 ± 0.43	44.38 ± 0.43	42.32 ± 0.36	41.72 ± 0.40

Table 5: Effect of prompt length on PØDA. This table presents results for prompts with increasing length (from top to bottom), generated by ChatGPT. For more details, refer to Section 4.2.1.

model across all prompt lengths, with no clear correlation between length and performance. Notably, some of the longest prompts (last row in Table 5) yield slightly better results, suggesting that additional descriptive detail can occasionally enhance the adaptation effect, though not uniformly.

Impact of selected layers for feature augmentation. 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 (by optimizing PIN) and forward them through remaining layers 2-4 to obtain the *Layer4* features. The input to the segmentation head is the concatenation of both. We study in Table 6 if one should augment other features in addition to the ones in *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 ones of *Layer1*.

We highlight that augmenting low-level features is detrimental for style manipulation. A decade of research in neural style transfer with CNNs has solidified the principle that CNN low-level features capture localized structures and visual style, whereas deeper features encodes global semantics. Methodologically, style manipulation is typically

<i>Layer1</i>	<i>Layer2</i>	<i>Layer3</i>	<i>Layer4</i>	ACDC Night
✓	✗	✗	✗	25.03 ± 0.48
✓	✓	✗	✗	23.43 ± 0.51
✓	✗	✓	✗	22.93 ± 0.53
✓	✗	✗	✓	21.05 ± 0.55

Table 6: Impact of selected layers for feature augmentation. This table shows the semantic segmentation performance (mIoU) of PØDA for day→night adaptation, using different ResNet layers for feature augmentation. In addition to augmenting features of *Layer1* (Row 1), one can augment *Layer2*, *Layer3*, or *Layer4* features (Rows 2-4).

achieved through two avenues: calculating feature correlations via Gram matrices (Gatys et al, 2015, 2016), or manipulating feature channel statistics (Huang and Belongie, 2017; Pan et al, 2018; Fan et al, 2023). Following the latter, we extract stylistic information from textual prompts by optimizing low-level channel statistics. This preserves the content features, avoiding the degradation CLIP’s semantic representations suggested by the lower performance in Table 6 when PIN is applied in deeper blocks.

Number of mined styles. In our experiments, $|\mathcal{S}_{s \rightarrow t}| = |\mathcal{D}_s|$. Here we study the effect of changing the number $|\mathcal{S}_{s \rightarrow t}|$ of styles on the target domain performance. By performing ablation on CS→Night with $|\mathcal{S}_{s \rightarrow t}| = 1, 10, 100, 1000, 2975$ (*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 (mIoU %) respectively. For $|\mathcal{S}_{s \rightarrow t}| < |\mathcal{D}_s|$, 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 $|\mathcal{S}_{s \rightarrow t}|$. Results also suggest that only few styles (*e.g.*, $|\mathcal{S}_{s \rightarrow t}| = 10$) could be sufficient for feature translation, similarly to few-shot image-to-image translation (Pizzati et al, 2022), 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 (Table 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 Table 7 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.

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 (<i>Layer1</i> *)	9.60	30.99	30.89	29.38
PØDA (<i>Layer1</i> *)	19.43 ± 0.69	37.80 ± 2.65	40.71 ± 1.06	39.09 ± 1.23

Table 7: Partial unfreezing of the backbone. PØDA results when only *Layer1* is frozen (*). In contrast, the models marked with * (reported in Table 2) freeze the entire backbone, *i.e.*, *Layers 1 through 4*.

Note that for fine-tuning the backbone, different hyperparameters, optimizer, and/or fine-tuning strategies might bring performance improvement *w.r.t.* to the current results. This is beyond the scope of this work, which aims to show that PIN is a plug-and-play module that can be used for any CNN-based vision-language pretrained model. Other works have for example shown that PIN can be used for state-of-the-art language-driven domain generalization (Fahes et al, 2024).

We show in Appendix A that PØDA can be applied on object detection and image classification. Additional ablations on PIN initialization and on training for adaptation are included in Appendix B.

4.3 PØDA-concept

Our initial prompts are intuitive descriptions verbalizing the target domains, *e.g.*, “driving at night”, “driving under rain”. Each prompt is the natural combination of a global *content*, (*i.e.*, “driving”) and a target style condition (*e.g.*, “at night”, “under rain”). Equivalently, the *relevant prompts* queried using ChatGPT (See Table 4) follow the same construction. For instance, in “navigating the roads in darkness”, the concept is expressed by “navigating the roads” and the style condition is described by “in darkness”. The presence of both elements in the target prompt is crucial for mining relevant styles. Indeed, in the rows of PØDA of Table 8, we show that while using either element alone improves over source-only, yet better results are obtained when combining both. Interestingly, optimizing towards “driving” improves all the target performances

Method	ACDC Night	ACDC Snow	ACDC Rain	GTA5
Source only	18.31	39.28	38.20	39.59
PØDA	“driving”	“driving”	“driving”	“driving”
	22.81 ± 0.34	43.34 ± 0.42	40.65 ± 0.99	40.33 ± 0.46
	“night”	“snow”	“rain”	“game”
	23.74 ± 0.55	40.45 ± 0.37	39.74 ± 0.35	40.69 ± 0.38
	“driving at night”	“driving in snow”	“driving under rain”	“driving in a game”
PØDA-concept	25.03 ± 0.48	43.90 ± 0.53	42.31 ± 0.55	41.07 ± 0.48
	“S*”	“S*”	“S*”	“S*”
	21.90 ± 0.19	44.04 ± 0.34	43.25 ± 0.45	42.54 ± 0.29
	“S* at night”	“S* in snow”	“S* under rain”	“S* in a game”
	25.16 ± 0.31	44.72 ± 0.24	42.78 ± 0.35	42.90 ± 0.17

Table 8: Effect of prompt construction. mIoU (%) performance on the target datasets under different prompt configurations. For PØDA, we evaluate prompts based on the text concept “driving”, the target style descriptions (*e.g.*, “night”), and their combination (prompt P, *e.g.*, “driving at night”). For PØDA-concept, we report results using either the optimized concept (“S*”) alone or combined with a target style description (*e.g.*, “S* at night”).

without any use of a textual style condition. Such prompt representation could describe driving scene images with all the styles included, bringing some generalization effects.

Concept optimization from source data. As introduced in Section 3.1.2, since the images of the source and unseen target domains share similar general content, we propose to optimize the latter using the source images instead of relying on text. This is done by optimizing a word embedding such that the text latent embedding is close to the source images’ embeddings in the bimodal space (See Eq. (9)). Results of PØDA-concept in Table 8 show superior performance compared to all variants of PØDA. Using a concept optimized from images better encodes the semantic content common to the data and results in better mined styles. Interestingly, using S* alone improves the performance *w.r.t.* source-only. This case is similar to the use of PØDA with “driving”, yet S* is more data-specialized. Note that for severe style shift, *e.g.*, night, the mIoU still significantly lags behind S* + “at night”.

Fig. 8 shows that including an optimized concept in the prompt not only improves performance but also makes style mining less sensitive to the number of iterations. Indeed, optimizing towards a prompt including S* restricts the search space to a region encoding the semantics of the images, which reduces the risk of drifting from

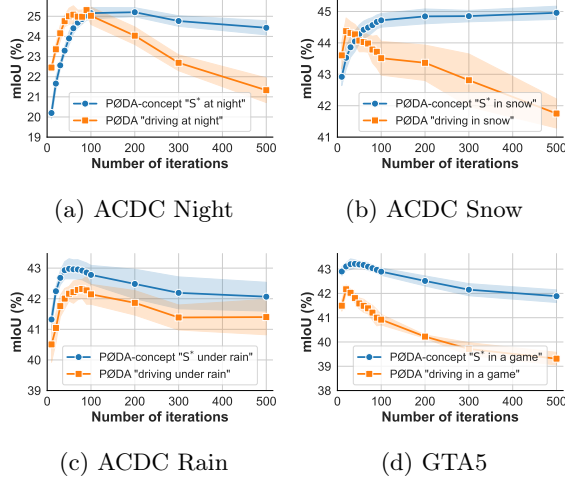


Fig. 8: Effect of the number of optimization iterations. mIoU (%) performance of PØDA and PØDA-concept as a function of the number of PIN optimization iterations. For example, 200 iterations means that PIN is optimized for 200 steps for both PØDA and PØDA-concept; the resulting styles are saved and then used to fine-tune the segmentation model in each case. Results are shown for zero-shot adaptation from Cityscapes to ACDC subsets — (a) night, (b) snow, (c) rain — as well as to (d) GTA5. Reported values are averaged over 5 runs.

the content. For a high number of iterations, the performance is significantly harmed when using a concept expressed through language (*i.e.*, “driving”). Interestingly, using an optimized S^* , the same observations hold, however the decrease is significantly smaller. We conjecture that in both cases, this might be due to over-stylization (Kwon and Ye, 2022).

In short, using an optimized concept along with the style condition has two advantages: better performance and lower sensitivity to the optimization iterations.

4.4 PIDA

We now evaluate our one-shot adaptation, PIDA, which instead of a target prompt leverages a target image. PIDA relies on the same mining strategy (Section 3.2) but using a target embedding derived from a target image $\mathbf{I}_t$ (cf. Section 3.1.3). As in prior experiments, we always report the average performance over 5 runs though here, using a different $\mathbf{I}_t$ randomly sampled from the

Method	ACDC Night	ACDC Snow	ACDC Rain	GTA5
source-only	18.31	39.28	38.20	39.59
0-shot				
PØDA	“driving at night” 25.03 ± 0.48	“driving in snow” 43.90 ± 0.53	“driving under rain” 42.31 ± 0.55	“driving in a game” 41.07 ± 0.48
AdaIN	18.15 ± 3.65	40.25 ± 1.72	42.61 ± 0.80	40.56 ± 0.97
PIDA	24.29 ± 1.07	43.66 ± 0.99	42.66 ± 1.21	41.02 ± 1.86

Table 9: PIDA on discrete domains. mIoU (%) performance for 0-shot adaptation using a prompt (PØDA) and two 1-shot variants —AdaIN and PIDA —that leverage a single target image.

target training set. Because one-shot adaptation enables nuanced settings that are hard to verbalize, we explore varying levels of granularity in the following experiments.

Discrete domains. Table 9 reports performance on the same discrete domains (*i.e.*, rain, night, etc.) as done previously. Since PIDA has access to a target image, we also report a vanilla AdaIN-adaptation baseline which seeks direct alignment with the style of the target image. As expected, PIDA significantly outperforms AdaIN, a result we attribute to our mining strategy providing greater diversity. Conversely, PIDA is outperformed almost consistently by its 0-shot counterpart, PØDA. We conjecture that this is due to the prompts of PØDA better capturing the overall appearance of the target domain, whereas randomly selected target images may not adequately represent the entire distribution of appearances in the target domain. This interpretation is supported by the high variance of PIDA, suggesting that mined styles are highly dependent on the choice of the target image.

Continuous domains. To better show the interest of PIDA for nuanced target conditions, harder to verbalize, we report performance on timelapse data. As such data are lacking in the driving domain, we use CoMoGAN (Pizzati et al, 2021), a continuous image-to-image translation method trained on Waymo Open dataset (Sun et al, 2020), where the translation is parametrized by $\phi \in [0, \pi]$, mapping the sun elevation in $[+30^\circ, -40^\circ]$ where negative values indicates that the sun is below the horizon (*i.e.*, nighttime). Using CoMoGAN, we translate Cityscapes training and validation sets for different ϕ values. For each ϕ , we leverage PIDA using for adaptation a random image from the ϕ -translated training set.

In Fig. 9, we plot the Cityscapes validation performance for each ϕ value, which shows

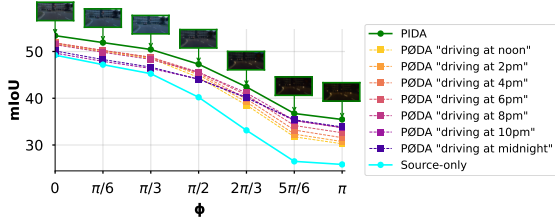


Fig. 9: PIDA on continuous time-of-day domains. Average performance (mIoU %) over 5 runs on ϕ -translated Cityscapes for source-only, PIDA, and PØDA with different prompts. Example one-shot target images used for PIDA are shown as insets.

that PIDA consistently outperforms the source-only model. In the latter plot, we also report performance of the 0-shot PØDA using various prompts constructed with “driving at <time>”. Interestingly, prompts referring to daytime (e.g., “driving at noon” or “at 2pm”) perform better than other prompts when the sun is higher (i.e., $\phi < \pi/2$) while night prompts (e.g., “driving at 10pm” or “at midnight”) are better after sunset (i.e., $\phi > \pi/2$). Still, for all ϕ values, PIDA consistently outperforms PØDA regardless of the prompt used. Such results hint that using an image as adaptation guidance can lead to better performance for peculiar conditions which are harder to describe with text.

5 Discussion

Generalization with PØDA. Inspired by the observation that some unrelated prompts improve performance on target domains (see Table 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 (μ, σ) 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 (Fan et al, 2023) where simple perturbations of feature channel statistics could help achieve SOTA generalization performance in object detection. Table 10 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

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 10: Generalization with PØDA. The “Source-only-G” model applies a domain generalization technique by perturbing low-level feature statistics with Gaussian noise. Building on this, PØDA-G (PØDA applied to Source-only-G) yields further performance gains. ‘style-mix’ refers to a style mixing strategy inspired by (Wu et al, 2022).

improves – always performing best on the desired target.

To further boost the performance, we perform style mixing. Wu et al (2022) showed that for OSUDA, mixing the source feature statistics and those of the single target image brings generalization effects. Later, Fahes et al (2024) showed that there is no need for the target image: mixing original and PIN-augmented statistics helps generalization. We follow the same strategy here. The mixed statistics $\mu_{\text{mix}}, \sigma_{\text{mix}}$ are given by:

$$\mu_{\text{mix}} = \alpha \odot \mu_t + (1 - \alpha) \odot \mu_s, \quad (11)$$

$$\sigma_{\text{mix}} = \alpha \odot \sigma_t + (1 - \alpha) \odot \sigma_s, \quad (12)$$

where $\alpha \in \mathbb{R}^c$ are per-channel mixing weights uniformly sampled in $[0, 1]$; $\odot$ denotes element-wise multiplication. Addition and subtraction are also element-wise. Finally, the augmented features are computed as follows:

$$\mathbf{f}_{s \rightarrow t} = \text{AdaIN}(\mathbf{f}_s, \mu_{\text{mix}}, \sigma_{\text{mix}}), \quad (13)$$

with AdaIN defined in Eq. (7).

PØDA with different architectures. We show in Table 11 consistent gains brought by PØDA using other backbone (RN101 (He et al, 2016)) and segmentation head (semantic FPN (Kirillov et al, 2019)).

Topic importance. In the era of large multimodal foundation models achieving unprecedented performance on similar tasks (Bai et al, 2025; Kirillov et al, 2023), we believe our work tackles a complementary and still critical problem space. Specifically, we argue that making domain adaptation faster and more accessible for smaller models—such as ResNet-50 with DeepLabV3+,

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 11: PØDA with different architectures. We evaluate PØDA using two segmentation architectures: Semantic FPN (‘Sem. FPN’) with a ResNet-50 backbone and DeepLabV3+ (‘DLv3+’) with a ResNet-101 backbone. This comparison highlights the effectiveness of PØDA across different model designs.

which are widely used in practice due to their efficiency and deployability—is of significant importance. Moreover, lightweight adaptation is highly practical in cases where large-scale retraining is not viable and fast inference is required, such as in resource-constrained environments or real-time applications. Lastly, while large-scale models are less prone to distribution shifts, we argue that understanding domain gaps in the context of smaller models could reveal a deeper theoretical understanding of domain adaptation mechanisms more broadly.

Limitations. PIN assumes that domain shift is entirely encoded in the visual style, therefore overlooking any camera and content variations. Another underlying assumption is that the style is entirely modeled by the channel-wise mean and standard deviations. While both assumptions are widely adopted for image translation and domain adaptation, they further limit the applicability of PIN. Investigating this is essential for deeper understanding of PIN’s role and its synergy with other adaptation methods.

Future directions. Future research could explore integrating our framework with online techniques like test-time adaptation (TTA) (Wang et al, 2021a), which enables on-the-fly learning during inference. We envision a dual-stage adaptation pipeline where our “offline” approach would provide a robust foundation by adapting to broad target characteristics through generic prompts (e.g., *driving at night*), while TTA would handle instance-specific shifts. Combining of the latter would allow to benefit from both general domain alignment and fine-grained, real-time adjustments. Investigating this synergy is a promising direction for achieving more resilient generalization in dynamic environments.

6 Conclusion

In this work, we leverage the CLIP model to make possible a new challenging task of domain adaptation using a single VL embedding. 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 a single VL embedding coming either from a prompt in natural language, a partially optimized prompt, or a single target image. Extensive experiments, particularly for semantic segmentation, have proven the effectiveness of our framework for semantic segmentation in particular. They also show its applicability to other tasks and various backbones. Our extensive experiments on the Cityscapes and ACDC datasets—which reflect real-world driving conditions, including challenging scenarios such as night, snow, and rain—demonstrate the practical relevance of our method in safety-critical applications. By relying only on a language prompt or a single image, our approach significantly reduces the dependency on large-scale target data collection, representing a step toward scalable and efficient deployment in real-world autonomous driving systems. Our line of research aligns with the collective efforts of the community to leverage large-scale pre-trained models (so-called “foundation models” (Bommasani et al, 2021)) for data- and label-efficient training of perception models for real-world applications. Future research may explore hybrid solutions for robust segmentation based on both large systems and our lightweight adaptation strategy.

Data availability statement. All the datasets used in this paper are publicly available.

Acknowledgment. This work was partially funded by French project SIGHT (ANR-20-CE23-0016) and the European Union under grant agreement No. 101070617.

A PØDA on other tasks

PØDA operates at the features level, which makes it task-agnostic. We show in the following the effectiveness of our method for object detection and image classification.

Method	Target	CS→ CS Foggy	DWD-Day Clear →			
			Night Clear	Dusk Rainy	Night Rainy	Day Foggy
DA-Faster (Chen et al, 2021b)	✓	32.0	-	-	-	-
ViSGA (Rezaeianaran et al, 2021)	✓	43.3	-	-	-	-
NP+ (Fan et al, 2023)	✗	46.3	-	-	-	-
S-DGOD (Wu and Deng, 2022)	✗	-	36.6	28.2	16.6	33.5
CLIP The Gap (Vidit et al, 2023)	✗	-	36.9	32.3	18.7	38.5
PØDA	✗	47.3	43.3	40.2	20.5	43.1

Table 12: 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.

PØDA for Object Detection. We report results in Table 12 from straightforwardly applying PØDA to object detection. Our Faster-RCNN (Ren et al, 2015) models, initialized with two CLIP-pretrained backbones, are trained on two source datasets, either Cityscapes or the Day-Clear split in Diverse Weather Dataset (DWD) (Wu and Deng, 2022). We report adaptation results on Cityscapes-Foggy (Sakaridis et al, 2018) 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 (Chen et al, 2021b; Rezaeianaran et al, 2021) (which use target images) and domain generalization methods (Fan et al, 2023; Wu and Deng, 2022; Vidit et al, 2023). We also experimented with YOLOF (Chen et al, 2021a) 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 (Fan et al, 2023) and (Vidit et al, 2023) for object detection.

In the Cityscapes-Foggy experiment, we freeze *Layer1* of the backbone, while in the DWD experiments we additionally freeze *Layer2*. The remaining layers are fine-tuned as commonly done. The code for PØDA in object detection was built upon the MMDetection library.⁶

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

μ^0	σ^0	mIoU
$\mu(\mathbf{f}_s)$	$\sigma(\mathbf{f}_s)$	25.03 ± 0.48
0	1	8.59 ± 0.82
$\sim \mathcal{N}(\mathbf{0}, \mathbf{I})$	$\sim \mathcal{N}(\mathbf{0}, \mathbf{I})$	6.80 ± 0.92

Table 13: Effect of style initialization. Performance (in mIoU) of PØDA on ACDC-Night val set (Cityscapes as source), with different initializations of style statistics. Starting from source images’ statistics works substantially better.

dataset (Wah et al, 2011) of 200 real bird species. We then perform zero-shot adaptation to classify bird paintings of CUB-200-Paintings dataset (Wang et al, 2020b) using the single prompt “Painting of a bird”. PØDA improves the source-only performance from 28.90 to 30.91 ± 0.69 . In our second experiment, we address the color bias in Colored MNIST (Arjovsky et al, 2019). 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. Again, PØDA brings significant improvement over the source-only model (64.16 ± 0.41 *vs.* 55.83).

B Additional experiments

Effect of style mining initialization. In our feature optimization step, we initialize (μ, σ) with $(\mu(\mathbf{f}_s), \sigma(\mathbf{f}_s))$. In Table 13, 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 $\mathbf{f}_s$ and better preserving the semantics.

Diversity of optimized statistics. To verify that the statistics — optimized for the same number of iterations with the same P but from different starting anchor points $\mathbf{f}_s$ — are diverse, we

⁶<https://github.com/open-mmlab/mmdetection>

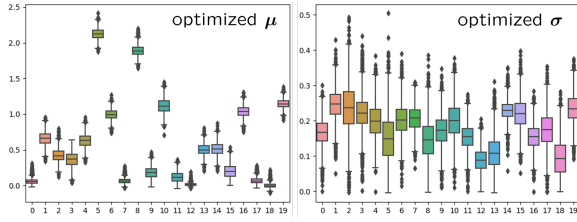


Fig. 10: Per-channel optimized statistics. Distributions of the first 20 channels of the optimized statistics of μ (Left) and σ (Right). Each boxplot shows the interquartile range (IQR) that contains 50% of the data: Its bottom and top edges delimit the first and third quartiles respectively. The horizontal line inside the box denotes the data’s 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).

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 14: 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.

show in Fig. 10 the boxplots of optimized parameters on the first 20 channels of $\mathbf{f}_{s \rightarrow t}$ (for prompt “driving at night”).

Training from scratch on augmented features. In PØDA, we start with a source-only trained model (Algorithm 2, line 1) then we fine-tune it on augmented features (Algorithm 2, line 4). 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 (Table 14) show the importance of the first, source-only training step.

References

Arjovsky M, Bottou L, Gulrajani I, et al (2019) Invariant risk minimization. arXiv preprint arXiv:190702893

Bai S, Chen K, Liu X, et al (2025) Qwen2.5-vl technical report. arXiv preprint arXiv:250213923

Ben-David S, Blitzer J, Crammer K, et al (2010) A theory of learning from different domains. ML

Bommasani R, Hudson DA, Adeli E, et al (2021) On the opportunities and risks of foundation models. arXiv preprint arXiv:210807258

Bottou L (2010) Large-scale machine learning with stochastic gradient descent. In: COMP-STAT

Chen LC, Papandreou G, Kokkinos I, et al (2017) Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. IEEE T-PAMI

Chen LC, Zhu Y, Papandreou G, et al (2018) Encoder-decoder with atrous separable convolution for semantic image segmentation. In: ECCV

Chen Q, Wang Y, Yang T, et al (2021a) You only look one-level feature. In: CVPR

Chen Y, Wang H, Li W, et al (2021b) Scale-aware domain adaptive faster r-cnn. IJCV

Cheng B, Collins MD, Zhu Y, et al (2020) Panoptic-deeplab: A simple, strong, and fast baseline for bottom-up panoptic segmentation. In: CVPR

Choi S, Jung S, Yun H, et al (2021) Robust-net: Improving domain generalization in urban-scene segmentation via instance selective whitening. In: CVPR

Cordts M, Omran M, Ramos S, et al (2016) The cityscapes dataset for semantic urban scene understanding. In: CVPR

Fahes M, Vu TH, Bursuc A, et al (2023) Poda: Prompt-driven zero-shot domain adaptation. In: ICCV

Fahes M, Vu TH, Bursuc A, et al (2024) A simple recipe for language-guided domain generalized segmentation. In: CVPR

- Fahes M, Vu TH, Bursuc A, et al (2026) Clip’s visual embedding projector is a few-shot cornucopia. In: WACV
- Fan Q, Segu M, Tai YW, et al (2023) Towards robust object detection invariant to real-world domain shifts. In: ICLR
- Gal R, Patashnik O, Maron H, et al (2022) StyleGAN-NADA: CLIP-guided domain adaptation of image generators. In: SIGGRAPH
- Ganin Y, Lempitsky V (2015) Unsupervised domain adaptation by backpropagation. In: ICML
- Ganin Y, Ustinova E, Ajakan H, et al (2016) Domain-adversarial training of neural networks. JMLR
- Gao P, Geng S, Zhang R, et al (2024) Clip-adapter: Better vision-language models with feature adapters. IJCV
- Gatys LA, Ecker AS, Bethge M (2015) A neural algorithm of artistic style. arXiv preprint arXiv:150806576
- Gatys LA, Ecker AS, Bethge M (2016) Image style transfer using convolutional neural networks. In: CVPR
- He K, Zhang X, Ren S, et al (2016) Deep residual learning for image recognition. In: CVPR
- Hoffman J, Tzeng E, Park T, et al (2018) Cycada: Cycle-consistent adversarial domain adaptation. In: ICML
- Hoyer L, Dai D, Van Gool L (2022) Daformer: Improving network architectures and training strategies for domain-adaptive semantic segmentation. In: CVPR
- Huang X, Belongie S (2017) Arbitrary style transfer in real-time with adaptive instance normalization. In: ICCV
- Ioffe S, Szegedy C (2015) Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: ICML
- Jatavallabhula KM, Kuwajerwala A, Gu Q, et al (2023) Conceptfusion: Open-set multimodal 3d mapping. arXiv preprint arXiv:230207241
- Jia C, Yang Y, Xia Y, et al (2021) Scaling up visual and vision-language representation learning with noisy text supervision. In: ICML
- Jiang Z, Xu FF, Araki J, et al (2020) How can we know what language models know? ACL
- Kang G, Jiang L, Yang Y, et al (2019) Contrastive adaptation network for unsupervised domain adaptation. In: CVPR
- Karras T, Laine S, Aila T (2019) A style-based generator architecture for generative adversarial networks. In: CVPR
- Kirillov A, Girshick R, He K, et al (2019) Panoptic feature pyramid networks. In: CVPR
- Kirillov A, Mintun E, Ravi N, et al (2023) Segment anything. In: ICCV
- Krizhevsky A, Sutskever I, Hinton GE (2012) Imagenet classification with deep convolutional neural networks. In: Neurips
- Kwon G, Ye JC (2022) Clipstyler: Image style transfer with a single text condition. In: CVPR
- Lee S, Seong H, Lee S, et al (2022) Wildnet: Learning domain generalized semantic segmentation from the wild. In: CVPR
- Lengyel A, Garg S, Milford M, et al (2021) Zero-shot day-night domain adaptation with a physics prior. In: ICCV
- Li B, Weinberger KQ, Belongie S, et al (2022) Language-driven semantic segmentation. In: ICLR
- Li G, Kang G, Liu W, et al (2020) Content-consistent matching for domain adaptive semantic segmentation. In: ECCV
- Li J, Selvaraju R, Gotmare A, et al (2021a) Align before fuse: Vision and language representation learning with momentum distillation. In: NeurIPS

- Li P, Li D, Li W, et al (2021b) A simple feature augmentation for domain generalization. In: ICCV
- Li Y, Yuan L, Vasconcelos N (2019) Bidirectional learning for domain adaptation of semantic segmentation. In: CVPR
- Lin TY, Dollár P, Girshick R, et al (2017) Feature pyramid networks for object detection. In: CVPR
- Long J, Shelhamer E, Darrell T (2015) Fully convolutional networks for semantic segmentation. In: CVPR
- Long M, Cao Z, Wang J, et al (2018) Conditional adversarial domain adaptation. In: NeurIPS
- Lu Y, Liu J, Zhang Y, et al (2022) Prompt distribution learning. In: CVPR
- Luo Y, Liu P, Guan T, et al (2020) Adversarial style mining for one-shot unsupervised domain adaptation. In: NeurIPS
- Minderer M, Gritsenko A, Stone A, et al (2022) Simple open-vocabulary object detection with vision transformers. In: ECCV
- Ovadia Y, Fertig E, Ren J, et al (2019) Can you trust your model’s uncertainty? evaluating predictive uncertainty under dataset shift. In: NeurIPS
- Pan F, Shin I, Rameau F, et al (2020) Unsupervised intra-domain adaptation for semantic segmentation through self-supervision. In: CVPR
- Pan X, Luo P, Shi J, et al (2018) Two at once: Enhancing learning and generalization capacities via ibn-net. In: ECCV
- Patashnik O, Wu Z, Shechtman E, et al (2021) Styleclip: Text-driven manipulation of stylegan imagery. In: ICCV
- Pizzati F, Cerri P, de Charette R (2021) Comogan: continuous model-guided image-to-image translation. In: CVPR
- Pizzati F, Lalonde JF, de Charette R (2022) Manifold: Manifold deformation for few-shot image translation. In: ECCV
- Radford A, Kim JW, Hallacy C, et al (2021) Learning transferable visual models from natural language supervision. In: ICML
- Ren S, He K, Girshick R, et al (2015) Faster r-cnn: Towards real-time object detection with region proposal networks. In: NeurIPS
- Rezaeianaran F, Shetty R, Aljundi R, et al (2021) Seeking similarities over differences: Similarity-based domain alignment for adaptive object detection. In: ICCV
- Richter SR, Vineet V, Roth S, et al (2016) Playing for data: Ground truth from computer games. In: ECCV
- Ronneberger O, Fischer P, Brox T (2015) U-net: Convolutional networks for biomedical image segmentation. In: MICCAI
- Sakaridis C, Dai D, Van Gool L (2018) Semantic foggy scene understanding with synthetic data. IJCV
- Sakaridis C, Dai D, Van Gool L (2021) Acdc: The adverse conditions dataset with correspondences for semantic driving scene understanding. In: ICCV
- Santurkar S, Tsipras D, Ilyas A, et al (2018) How does batch normalization help optimization? In: NeurIPS
- Shin T, Razeghi Y, Logan IV RL, et al (2020) Autoprompt: Eliciting knowledge from language models with automatically generated prompts. In: EMNLP
- Sun B, Saenko K (2016) Deep coral: Correlation alignment for deep domain adaptation. In: ECCV
- Sun P, Kretzschmar H, Dotiwalla X, et al (2020) Scalability in perception for autonomous driving: Waymo open dataset. In: CVPR
- Tsai YH, Hung WC, Schuster S, et al (2018) Learning to adapt structured output space for

- semantic segmentation. In: CVPR
- Ulyanov D, Vedaldi A, Lempitsky V (2016) Instance normalization: The missing ingredient for fast stylization. arXiv preprint arXiv:160708022
- Ulyanov D, Vedaldi A, Lempitsky V (2017) Improved texture networks: Maximizing quality and diversity in feed-forward stylization and texture synthesis. In: CVPR
- Vidit V, Engilberge M, Salzmann M (2023) Clip the gap: A single domain generalization approach for object detection. arXiv preprint arXiv:230105499
- Vu TH, Jain H, Bucher M, et al (2019) Advent: Adversarial entropy minimization for domain adaptation in semantic segmentation. In: CVPR
- Wah C, Branson S, Welinder P, et al (2011) The caltech-ucsd birds-200-2011 dataset. California Institute of Technology
- Wang D, Shelhamer E, Liu S, et al (2021a) Tent: Fully test-time adaptation by entropy minimization. In: ICLR
- Wang J, Sun K, Cheng T, et al (2020a) Deep high-resolution representation learning for visual recognition. TPAMI
- Wang S, Chen X, Wang Y, et al (2020b) Progressive adversarial networks for fine-grained domain adaptation. In: CVPR
- Wang Y, Li W, Dai D, et al (2017) Deep domain adaptation by geodesic distance minimization. In: ICCV Workshops
- Wang Z, Luo Y, Qiu R, et al (2021b) Learning to diversify for single domain generalization. In: ICCV
- Wu A, Deng C (2022) Single-domain generalized object detection in urban scene via cyclic-disentangled self-distillation. In: ECCV
- Wu X, Wu Z, Lu Y, et al (2022) Style mixing and patchwise prototypical matching for one-shot unsupervised domain adaptive semantic segmentation. In: AAAI
- Xiao N, Zhang L (2021) Dynamic weighted learning for unsupervised domain adaptation. In: CVPR
- Yang Y, Soatto S (2020) Fda: Fourier domain adaptation for semantic segmentation. In: CVPR
- Zhao H, Shi J, Qi X, et al (2017) Pyramid scene parsing network. In: CVPR
- Zhao H, Qi X, Shen X, et al (2018) Icnet for real-time semantic segmentation on high-resolution images. In: ECCV
- Zhong Z, Friedman D, Chen D (2021) Factual probing is [mask]: Learning vs. learning to recall. In: NAACL
- Zhou C, Loy CC, Dai B (2022a) Extract free dense labels from clip. In: ECCV
- Zhou K, Yang Y, Qiao Y, et al (2021) Domain generalization with mixstyle. In: ICLR
- Zhou K, Yang J, Loy CC, et al (2022b) Conditional prompt learning for vision-language models. In: CVPR
- Zhou K, Yang J, Loy CC, et al (2022c) Learning to prompt for vision-language models. IJCV
- Zou Y, Yu Z, Kumar B, et al (2018) Unsupervised domain adaptation for semantic segmentation via class-balanced self-training. In: ECCV
- Zou Y, Yu Z, Liu X, et al (2019) Confidence regularized self-training. In: ICCV