

DiffPrompter: Differentiable Implicit Visual Prompts for Semantic-Segmentation in Adverse Conditions

Sanket Kalwar^{*1}, Mihir Ungarala^{*1}, Shruti Jain^{*1}, Aaron Monis¹, Krishna Reddy Konda³
 Sourav Garg², K Madhava Krishna¹

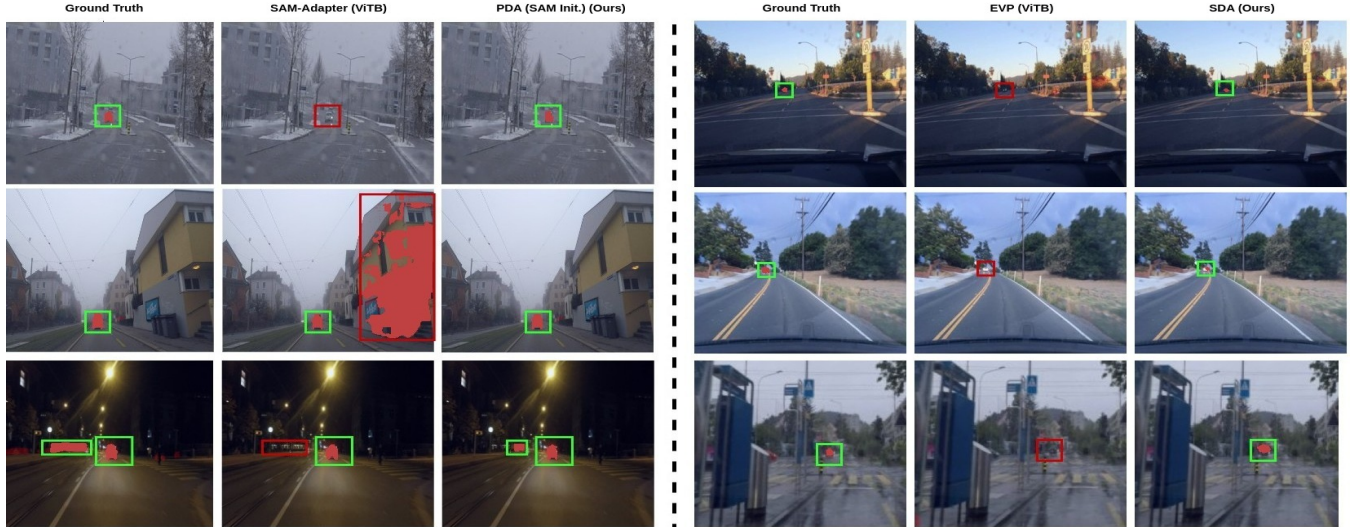


Fig. 1: **Overview:** To tackle Semantic Segmentation in adverse weather conditions, we introduce the DiffPrompter framework (explained later in Fig. 2), through which we create a *Serial* Differentiable Adaptor (SDA) and *Parallel* Differentiable Adaptor (PDA). Both these adaptors achieve superior results as compared to the current state-of-the-art (SOTA) methods: SAM-Adapter [1] (left) and EVP [2] (right), where columns in each case respectively represent the ground-truth segmentation mask, the baseline method (SAM-Adapter/EVP), and our proposed method (PDA/SDA). In this representation, the model’s masked output is shown in red, with green bounding boxes highlighting correct predictions and red bounding boxes indicating missed segmentation outputs (false negatives) or incorrect segmentation outputs (false positives). It is evident that our proposed methods, PDA and SDA, demonstrate superior qualitative performance compared to the SOTA methods, EVP and SAM-Adapter.

Abstract—Semantic segmentation in adverse weather scenarios is a critical task for autonomous driving systems. While foundation models have shown promise, the need for specialized adaptors becomes evident for handling more challenging scenarios. We introduce *DiffPrompter*, a novel differentiable visual and latent prompting mechanism aimed at expanding the learning capabilities of existing adaptors in foundation models. Our proposed ∇HFC (High Frequency Components) based image processing block excels particularly in adverse weather conditions, where conventional methods often fall short. Furthermore, we investigate the advantages of *jointly training visual and latent prompts*, demonstrating that this combined approach significantly enhances performance in out-of-distribution scenarios. Our differentiable visual prompts leverage *parallel* and *series* architectures to generate prompts, effectively improving object segmentation tasks in adverse conditions. Through a comprehensive series of experiments and evaluations, we provide empirical evidence to support the efficacy of our approach. Project page: diffprompter.github.io

I. INTRODUCTION

Scene understanding or semantic segmentation for Autonomous Vehicles (AV) is an important part of their perception stack through which they can recognize the map elements, open space and other vehicles and obstacles. Most deep-learning approaches for semantic segmentation work well in ideal environmental conditions [3] but fail under adverse conditions such as low light, fog, snow, rain and glare [4], [5]. However, this strategy requires one to store and deploy a separate copy of the backbone parameters for every single task. This is computationally expensive and infeasible in most real-time scenarios. Other strategies [6] include fine-tuning only the last few layers or the neck of the model but these methods underperform as compared to complete fine-tuning of the model.

To boost the performance, [1], [2] introduced visual prompts as inputs to the Adaptors. However, these methods fail to generalize well in out-of-distribution settings because the visual prompts fail to adapt to new domains and new data distributions.

To address this issue, we use *differentiable visual prompts* which are also learned during training. The prompts focus on enhancing the image using differentiable parameters as

* denotes equal contribution

¹are with RRC, IIIT Hyderabad, India {sankethkalwar, mihir.ungarala, shrutipraveenjain, aaronmonis22}@gmail.com, mkrishna@iiit.ac.in

²is with the Australian Institute for Machine Learning, University of Adelaide, Australia sourav.garg@adelaide.edu.au

³is with ZF TCI, Hyderabad, India krishna.konda@zf.com

†Code: <https://github.com/DiffPrompter/diff-prompter>

◊Paper full version: <https://arxiv.org/pdf/2310.04181>

demonstrated by [7]. A similar idea has also been explored in [8] where they introduced a differentiable FFT block as a visual prompt to the Adaptor. With our differentiable image processing block, we provide an array of image pre-processing techniques that can be combined with each other to produce a more meaningful image prompt for any domain.

The contributions of this paper are listed below:

- 1) a *differentiable* visual prompting mechanism to extend the generalizing abilities in out-of-distribution settings of existing adaptors for foundation models (see Table II);
- 2) a ∇HFC (High Frequency Components) retaining differentiable image processing block, which is particularly robust to adverse conditions (see Sec. V-D and Table IV);
- 3) *joint learning* of visual prompts and latent prompts, which outperforms their individually-learned versions (see Sec. V-D and Table III); and
- 4) *parallel* and *serial* versions of our architecture to generate visual and latent prompts that improve the object-segmentation task in adverse conditions (see Fig. 1, Table I and II).

II. RELATED WORK

Foundation Segmentation Models: Semantic segmentation task is often tackled using methods like Fully Convolutional Networks (FCNs) [9], encoder-decoder structures [10], atrous convolutions [11], pyramid modules [12] and attention modules [13]. Segformer [14] uses a novel hierarchically structured transformer encoder and a decoder that aggregates this information allowing for high performance even on zero-shot distributions. The Segment Anything Model (SAM) [15] introduces a foundational model in the form of a ViT-based model trained upon a large visual corpus. It has been designed and trained to be transferred zero-shot to new image distributions and tasks.

Visual Prompt tuning: Prompting was initially introduced in NLP [16]. Recent works have introduced prompting to vision tasks [1], [2], [8], [17], [18]. [18] introduces a set of learnable embedding vectors at each layer of a transformer. Visual Prompt Tuning (VPT) [17] modifies the input to the Vision Transformer by introducing a small amount of task-specific learnable parameters into the input space by simply prepending into the input sequence and are learned together using a linear head during fine-tuning. Explicit Visual Prompting [2] introduces tunable parameters that extract explicit visual information from frozen patch embeddings and the high-frequency components of input images to create explicit prompts. The explicit prompt created from these features is used as an Adaptor for each layer of the ViT.

Differentiable Image enhancement: IA-Yolo [19] uses a CNN to predict differentiable image processing parameters trained in conjunction with Yolov3 [20] for object detection in an end-to-end fashion. GDIP [7] proposes a domain-agnostic novel gating mechanism for learning parameters of multiple image pre-processing techniques that operate concurrently. This mechanism can be plugged into existing

object detection networks like YOLOv3 [20] and trained end-to-end on adverse condition images. [2], [8] extracts the high-frequency components of an image to create the explicit visual prompts.

III. PROPOSED METHOD

We propose a Differentiable Visual Prompt block, referred to as *DiffVP*, as shown in Fig. 2. This block learns visual prompts using *DiffIP* and latent prompts through a shallow vision encoder. The Differentiable Adaptor, denoted as *DiffAdaptor*, employs *VPTune* to learn local information of the visual prompt. Local information from the transformer layer of the encoder is provided to the *Embedding Tune* layer. The output of the *Embedding Tune* and *VPTune* layers is combined and fed into the adaptor layer, which outputs local features that when added to the transformer layers, learn local invariant features. To introduce global invariance in the features, the latent embedding generated by the shallow vision encoder is added to the transformer output features. This combination of local and global invariance aids the Parallel Differentiable Adaptor (PDA) and Sequential Differentiable Adaptor (SDA) in improving semantic segmentation tasks.

A. Differentiable Visual Prompt (*DiffVP*) Block:

We utilize a shallow vision encoder, denoted as F_{enc} , in combination with the differential image processing block *DiffIP*, which incorporates our proposed ∇HFC image processing operation. This combination is used to generate the visual prompt output I_n along with the output latent embedding f_{emb}^n as follows:

$$f_{emb}^n = F_{enc}(I_{n-1}) \quad (1)$$

$$I_n = DiffIP(f_{emb}^n) \quad (2)$$

The n indicates the total number of transformer layers in the ViT-B encoder discussed in Section III-C. In the section below, we will delve into the building blocks of *DiffVP*, which include the shallow vision encoder, the ∇HFC image processing block, and *DiffIP*.

1) **Shallow Vision Encoder:** This lightweight vision encoder, denoted as F_{enc} , takes an image $I \in R^{H \times W \times 3}$ as input and learns a latent embedding f_{emb} that encapsulates the global context of the image. This embedding is valuable for predicting the parameters of the image processing blocks in *DiffIP*. The encoder comprises five consecutive 2D convolution layers with a kernel size of 3 and a stride of 1, along with the *ReLU* non-linearity. Finally, adaptive average pooling is applied to reduce the spatial dimensions, followed by a fully connected layer that produces the latent embedding f_{emb} . This process can be expressed using Eq.1.

2) **∇HFC Image Processing Block:** We propose a method to learn high-frequency components directly by parameterizing the HFC component of the image as mentioned in the work [2], where low-frequency components of the image are removed and high-frequency components are kept

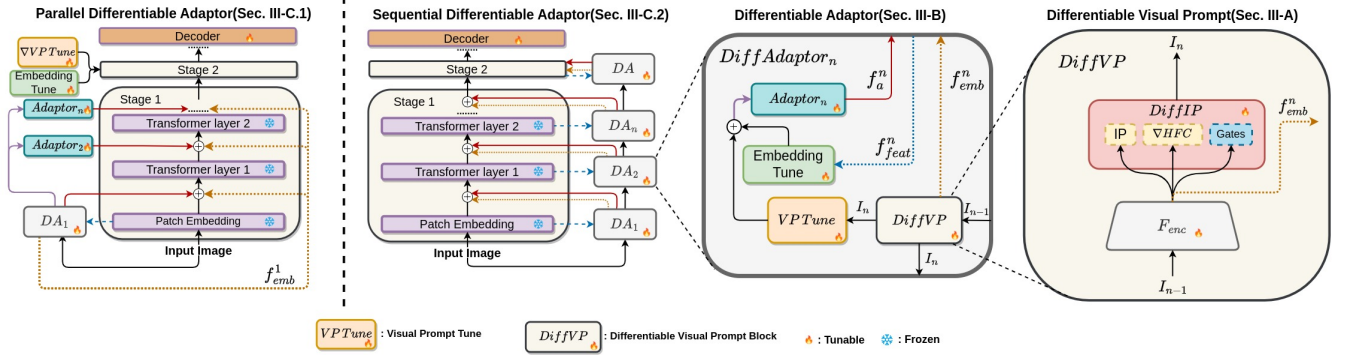


Fig. 2: Framework overview, First, the input image is sent to a Parallel Differentiable Adaptor (*PDA*, Sec.III-C.1) and Sequential Differentiable Adaptor (*SDA*, Sec.III-C.2) segmentation model through various stages of transformer encoder and decoder layers which in turn generates semantic segmentation mask as output, details in Sec.III-C, *DiffAdaptor* is used as a building block in both *PDA* and *SDA* which contains *DiffVFP* block that generates differentiable visual prompt details of both are discussed in Sec.III-B and Sec.III-A.

by masking the $z = fft(I)$ by the mask $\mathbf{M}_{hfc} \in \{0, 1\}^{H \times W}$ which retains high-frequency components of z and that this retained component is inverse Fourier transformed to get the high-frequency component image as $I_{hfc} = ifft(z\mathbf{M}_{hfc})$. In [2] τ rate parameter is tuned manually for the given dataset. However, this parameter can be made learnable by predicting it directly using shallow vision encoder output f_{emb} along with F_{hfc} fully-connected layer and sigmoid non-linearity σ as shown in the Eq.(3) denoted by $\tilde{\tau}$.

$$\tilde{\tau} = \max(0.1, \min(\sigma(F_{hfc}(f_{emb})), 0.8)) \quad (3)$$

$$\tilde{\mathbf{M}}_{\nabla hfc}^{i,j} = \begin{cases} 0, & \text{if } \frac{HW-4|(i-\frac{H}{2})(j-\frac{W}{2})|}{HW} < \tilde{\tau} \\ 1, & \text{otherwise} \end{cases} \quad (4)$$

$$\tilde{I}_{\nabla hfc} = ifft(z\tilde{\mathbf{M}}_{\nabla hfc}) \quad (5)$$

Mask $\tilde{\mathbf{M}}_{\nabla hfc}^{i,j}$ can be further generated using $\tilde{\tau}$ learned during training as shown in the Eq. (4), which enables learning high-frequency component image $\tilde{I}_{\nabla hfc}$ directly in end-to-end fashion by the process mentioned in Eq. (5).

3) **Differentiable Image Processing Block (*DiffIP*):** We propose a new image processing operation, denoted as ∇HFC , for extracting high-frequency components from an image. This operation is introduced in addition to the image processing techniques utilized in GDIP model [7], which includes tone adjustment, contrast enhancement, sharpening, defogging, gamma correction, white balance, and identity operation. When these image processing operations are combined, they form a new block known as *DiffIP*. The *DiffIP* block consists of image-processing operations represented as ip and their corresponding gates denoted as w_{ip} . The parameters for these operations are predicted using f_{emb} , which is obtained from a shallow vision encoder. The *DiffIP* block can be succinctly summarized by combining Eq.1 and Eq.6, where f_{ip} represents the output image resulting from the image processing operation F_{ip} , and N denotes a min-max normalization operation.

$$I_n = N\left(\sum_{ip} w_{ip} N(F_{ip}(f_{emb}^n))\right) \quad (6)$$

B. Differentiable Adaptor (*DiffAdaptor*):

The Differentiable Adaptor takes image I_{n-1} as input, which is initially passed through the *DiffVFP* block. This block outputs a differentiable visual prompt I_n and f_{emb}^n , and the I_n is utilized by the *VPTune* layer. The *VPTune* layer performs patch-wise convolution to generate patch embeddings f_{VPTune} . The embedding layer takes the features f_{feat}^n from the corresponding n^{th} layer of the ViT encoder and generates the output embedding $f_{embedding_tune}$. This output embedding is then added to the patch embeddings generated by the *VPTune* layer and is finally fed into the adaptor layer.

The adaptor block is composed of two components: a shared MLP denoted as MLP_{shared} and unshared MLPs represented as MLP_n for each of the transformer layers at all stages of the encoder. This composition can be succinctly expressed using Equations 7 and 8.

$$x = f_{VPTune} + f_{embedding_tune} \quad (7)$$

$$f_a^n = MLP_{shared}(GELU(MLP_n(x))) \quad (8)$$

C. Parallel-Sequential Differentiable Adaptor Models:

PDA and *SDA* comprise a ViT-B encoder and a progressive upsampling ConvNet layer as its decoder, as illustrated in Fig. 2. In the ViT encoder, all transformer layers remain frozen at every stage, while the decoder, Adaptor layers, and the *DiffAdaptor* units are fine-tuned during training. In the section below, we will delve into the details of *PDA* and *SDA*.

1) **Parallel Differentiable Adaptor Model (*PDA*):** The *PDA* encoder follows a standard forward pass procedure. The input image is processed through various transformer layers and ultimately reaches the decoder layer. Additionally, at each skip connection, features computed from the Adaptor, along with the f_{emb} features, are incorporated. In the initial stage of the ViT-B encoder, only one *DiffAdaptor* is employed. Its output is computed by *DiffVFP*, which takes the input image and generates the visual prompts, along with the latent prompt f_{emb} . Subsequently, this output is fed into *VPTune*, which utilizes the

visual prompts, and *Embedding Tune*, which takes patch embeddings as input. The sum of the outputs from *VPTune* and *Embedding Tune* is then provided to all the Adaptor layers, the details of which are outlined in Sec. III-B.

In PDA, the computations for *DiffVP* are performed only once and are concurrently utilized by all the Adaptor layers associated with that particular stage.

2) Sequential Differentiable Adaptor Model (SDA):

In contrast to the PDA model, the SDA incorporates *DiffAdaptor* blocks at all stages of the encoder. The outputs of the *DiffAdaptor*, namely the latent prompt f_{emb} and the Adaptor output f_a^n , are added to the skip-connection (plus sign between the transformer layer in the Fig.2) of the ViT-B encoder. Additionally, the I_n image is generated as an output by the *DiffAdaptor* and subsequently serves as an input to the next *DiffAdaptor* block in a sequential manner.

IV. EXPERIMENTAL SETUP

A. Datasets

For the vehicle foreground segmentation task, we use **BDD100K** dataset [4] (approximately 7,000 train and 1,000 test images), **ACDC** test dataset [21] (406 images), **Wild-Dash** test dataset [22], [23] (70 images) and **Dark-Zurich** test dataset [5] (50 images). These datasets specialize in semantic segmentation in adverse weather conditions. BDD100K specializes in scenarios with rain, snow, clear, overcast, foggy and cloudy conditions. ACDC dataset has conditions with fog, nighttime, rain and snow. Wild-Dash comprises real-world driving scenarios in challenging conditions and Dark-Zurich consists of images in nighttime, twilight and daytime. Focusing mainly on vehicle segmentation, we combine the vehicle classes in the above-mentioned datasets, as a single foreground class, while the rest are considered as a background class. In the BDD100K dataset, out of the 19 classes, we use 6 classes for foreground - car, truck, bus, train, motorcycle, and bicycle. Similarly, in ACDC, Wild-Dash and Dark-Zurich datasets, from the 34 labels in the annotated data, we use 8 classes for foreground i.e. car, truck, bus, caravan, trailer, train, motorcycle and bicycle.

TABLE I: Comparison between EVP-Net and SAM-Adapter with our proposed PDA/SDA trained on BDD100k trainset and tested on BDD100k testset. The colors denoted in green and red correspond to a relative increase or decrease in the performance of PDA and SDA, respectively, in comparison to the baseline methods.

Method	BDD100K			
	$S_\alpha \uparrow$	$E_\phi \uparrow$	$F_{\beta}^w \uparrow$	$MAE \downarrow$
EVP-Net	.8466	.9225	.7521	.03033
PDA(Ours)	.8513 (+0.56)	.9274 (+0.53)	.7590 (+0.92)	.0294 (-3.1)
SDA(Ours)	.8523 (+0.67)	.9294 (+0.75)	.7624 (+1.37)	.0296 (-2.4)
SAM-Adapter	.9222	.9561	.8571	.0155
PDA(Ours)	.9251 (+0.31)	.9592 (+0.32)	.8656 (+0.99)	.0144 (-7.63)
SDA(Ours)	.9279 (+0.62)	.9653 (+0.96)	.8495 (-0.89)	.0100 (-55.10)

B. Training Setup

PDA and SDA models are trained on input images with a size of $352 \times 352 \times 3$ and a batch size of 4. Additionally, we

employ the *AdamW* optimizer with a *lr* of $2e-4$, a multi-step LR scheduler with a gamma parameter of 0.1, and a milestone parameter set to 1. We train PDA and SDA for 50 epochs on the BDD100K dataset and subsequently evaluate their performance on BDD100K, ACDC, Wild-Dash, and Dark-Zurich. For both setups, we use Binary Cross Entropy Loss during backpropagation.

PDA(SAM Init) undergoes training in a similar fashion as mentioned above, with the difference that the image dimensions are set to $1024 \times 1024 \times 3$, with a batch size of 3. Similarly, for SDA(SAM Init), the training happens in a similar fashion except that the batch size is 1. Our proposed method PDA and SDA, are constructed on the foundation of the ViT-B architecture, selected for its suitability in real-time autonomous driving applications. In pursuit of fairness and transparency, we have chosen to utilize ViT-B as the backbone for both the SAM-Adapter and EVP models, ensuring an impartial comparison with our own PDA and SDA innovations.

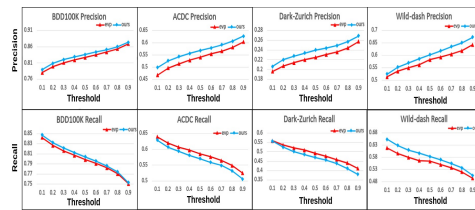


Fig. 3: Precision & Recall graph, where row 1 corresponds to Precision and row 2 corresponds to Recall, also columns 1-4 belong to BDD100K, ACDC, Dark-Zurich and Wild-Dash datasets respectively. The red colour belongs to the EVP method while the blue colour belongs to SDA(ours). Better viewed at 400x zoom.

V. RESULTS AND ANALYSES

A. Qualitative Analysis

We compare the results of PDA and SDA with the foundation model SAM-Adapter and EVP as shown in Fig.1 for semantic segmentation on real-world data. On the left side of the vertical dashed line, SAM-Adapter has missed many instances (row 1 and row 3) or classified incorrectly (row 2) as compared to our proposed method which does better on objects far away as well as on adverse conditions images like snow, fog and night scenarios. Likewise, we see similar trends to the right side of the vertical dashed line where SDA does better qualitatively as compared to EVP on the far away objects in adverse conditions like dusk, cloudy and rainy scenarios. We quantify the improvements extended by PDA and SDA in the next section.

B. Quantitative Analysis

We compare our proposed method with the SOTA method EVP and SAM-Adapter on the BDD100K, ACDC, Wild-Dash and Dark-Zurich test dataset as shown in Table I and Table II. These results are discussed in the subsequent Sec. V-B.0.a and Sec. V-B.0.b. We use the following evaluation metrics: S-measure S_α [24] provides similarity score for objects, background and regions structure with respect to the

TABLE II: Comparison of EVP-Net and SAM-Adapter with our proposed PDA/SDA on ACDC, Wild-Dash and Dark-Zurich dataset. Note that the training set only comprises BDD100K and is evaluated on the ACDC, Wild-Dash and Dark-Zurich for testing the generalization ability of the models.

Method	ACDC				Wild-Dash				Dark-Zurich			
	$S_\alpha \uparrow$	$E_\phi \uparrow$	$F_\beta^w \uparrow$	$MAE \downarrow$	$S_\alpha \uparrow$	$E_\phi \uparrow$	$F_\beta^w \uparrow$	$MAE \downarrow$	$S_\alpha \uparrow$	$E_\phi \uparrow$	$F_\beta^w \uparrow$	$MAE \downarrow$
EVP-Net	.6859	.7710	.4298	.0249	.7409	.8218	.3741	.0297	.6202	.6037	.1766	.0194
PDA(Ours)	.6941 (+0.1)	.7755 (+0.58)	.4406 (+2.51)	.0245 (-1.6)	.7289 (-1.64)	.7996 (-2.78)	.3627 (-3.14)	.0317 (+6.7)	.6245 (+0.69)	.6216 (+2.97)	.1903 (+7.76)	.0169 (-14.7)
SDA(Ours)	.6948 (+1.3)	.7886 (+2.28)	.4542 (+5.68)	.0233 (-6.8)	.7480 (+0.95)	.8273 (+0.67)	.3976 (+6.28)	.0282 (-5.3)	.6294 (+1.48)	.6573 (+8.88)	.1938 (+9.74)	.0148 (-31.1)
SAM-Adapter	.8366	.8871	.6756	.0107	.8873	.9207	.6147	.0093	.7033	.7349	.3125	.0132
PDA(Ours)	.8448 (+0.98)	.8939 (+0.77)	.6751 (-0.07)	.0106 (-0.94)	.8944 (+0.80)	.9220 (+0.14)	.6346 (+3.24)	.0094 (+0.1)	.7100 (+0.95)	.7385 (+0.49)	.3036 (-2.93)	.0143 (+8.34)
SDA(Ours)	.8460 (+1.12)	.9001 (+1.47)	.6690 (-0.98)	.0100 (-7)	.8820 (-0.60)	.9310 (+1.11)	.6658 (+8.31)	.0097 (+4.3)	.6718 (-4.69)	.7412 (+0.85)	.2627 (-18.95)	.0087 (-51)

ground truth mask; mean E-measure E_ϕ [25] combines local pixel values with the image-level mean value in one term, thus jointly capturing image-level statistics and local pixel matching information; and as standard, weighted F-measure F_β^w and MAE (Mean Absolute Error).

The colors denoted in green and red correspond to a relative increase or decrease in the performance of PDA and SDA, respectively, in comparison to the baseline methods, and this convention is followed in Table I and II. In the Quantitative section, we use the absolute difference between baseline and PDA/SDA method by default unless specified explicitly.

a) Performance of EVP vs PDA and SDA - BDD100K:

On the BDD100K dataset, SDA performs well as compared to EVP on S_α , E_ϕ and F_β^w by an average increase of 0.76% with the better MAE score as shown in the Table I. PDA performs well on S_α , E_ϕ and F_β^w by an average increase of 0.55%, and with the best MAE score as compared to both EVP and SDA. Similarly, In the case of ACDC, Wild-Dash and Dark-Zurich datasets, SDA demonstrates significant performance gains as compared to EVP by 1.7%, 1.2% and 2.67%, with the best MAE scores, which demonstrates the generalization capability of SDA on the challenging adverse weather datasets. Also in terms of pure quantitative numbers shown in the Table II, the SDA model outperforms EVP and PDA on almost all the datasets.

b) Performance of SAM-Adapter vs PDA (SAM Init.) and SDA (SAM Init.) - BDD100K:

PDA is initialized by SAM ViT-B weights at the start of the training, and compared with SAM-Adapter with ViT-B architecture, trained on the BDD100K dataset. The Table I, PDA performs better on average at BDD100K and ACDC by 0.48% with the best MAE as compared to SAM-Adapter. In the Wild-Dash dataset, PDA has a performance gain of 0.94% with comparable MAE. In Table II Dark-Zurich there is an increase of 0.67% and 0.36% on S_α and E_ϕ with a decrease in F_β^w score by 0.89%, MAE score of SAM-Adapter is better than PDA in Dark-Zurich dataset. In most diverse settings PDA performance is better as compared to the SAM-Adapter. In the case of SDA with SAM-initialized weights, it performs 0.24% on the BDD100K dataset, and 0.54% on the ACDC dataset which is better than SAM-Adapter along with better MAE scores on both the datasets. On the Wild-Dash dataset SDA has a performance boost of 1.87 % with slightly worse MAE scores, while on the Dark-Zurich dataset, there is a performance decrease of almost 2.5 % along with the best MAE score as compared to SAM-Adapter. In terms of pure

quantitative numbers in the Table I and II, SDA is ahead of PDA and SAM-Adapter on most of the metrics on all the datasets.

C. Segmentation Statistics

In Fig.3 we show through precision & recall graph, how well our proposed SDA method performs against the EVP model on the contrary to evaluation metrics S_α , E_ϕ , F_β^w and MAE.

In terms of precision, our method performs the best as compared to EVP which indicates that our method has fewer false positives in all the above test datasets. Recall of our system is better on BDD100K and Wild-Dash dataset while there is a slight drop in recall as the threshold increases in ACDC and Dark-Zurich.

TABLE III: Ablation for PDA design choice.

Method	BDD100K			
	$S_\alpha \uparrow$	$E_\phi \uparrow$	$F_\beta^w \uparrow$	$MAE \downarrow$
Vanilla ViT-B	.8454	.9217	.7542	.0316
EVP	.8466	.9225	.7521	.03033
PDA (w/o f_{emb})	.8479	.9229	.7534	.03065
PDA (w/o $DiffIP$)	.846	.9222	.7515	.03082
PDA (w/o $DiffIP$ & w/o f_{emb})	.8449	.9213	.7492	.03109
PDA (ours)	.8513	.9270	.759	.02940

TABLE IV: Comparison between EVP with only HFC, ∇HFC with f_{emb} .

Method	BDD100K			
	$S_\alpha \uparrow$	$E_\phi \uparrow$	$F_\beta^w \uparrow$	$MAE \downarrow$
EVP	.8466	.9225	.7521	.03033
∇HFC	.847	.9238	.7526	.03073
$\nabla HFC + f_{emb}$.8512	.9245	.7584	.02965

D. Ablation Studies

We conducted an ablation study on our PDA model, utilizing the BDD100K dataset for validation purposes. This rigorous examination of our design decisions is undertaken due to the deliberate inclusion of only one $DiffVP$ block within the model, a simplification intended to facilitate a more comprehensive understanding of its performance characteristics and details of which can be referred to in Table III. **PDA v/s PDA(w/o $DiffIP$ & w/o f_{emb})** PDA is better than PDA(w/o $DiffIP$ & w/o f_{emb}) and shallow vision encoder by 0.73% on an average on S_α , E_ϕ and F_β^w with best MAE scores. In this setup, the image is directly passed through the $VPTune$ block without any pre-processing on it. **PDA v/s PDA(w/o $DiffIP$)** PDA is better than PDA(w/o

DiffIP) by 0.6% on an average on BDD100K dataset with the best MAE scores which tells that $(0.73\% - 0.6\%) = 0.1\%$ gap is closed by f_{emb} and is evident from the above ablation. **PDA v/s PDA(w/o f_{emb})** As shown in Table III, PDA performance is better than in PDA(w/o f_{emb}) by 0.44 % and has better MAE scores. It is evident that the performance gain by having *DiffIP* is about $(0.73\% - 0.44\%) = 0.29\%$. Therefore, PDA with both *DiffIP* and f_{emb} is the best setting for training rather than PDA(w/o *DiffIP*) and PDA(w/o f_{emb}). And by jointly learning *DiffIP* and f_{emb} there is a performance gain of additional $(0.73\% - (0.29\%+0.1\%)) = 0.34\%$ and while the individual contribution of *DiffIP* and f_{emb} is 0.29% and 0.1%. We also conduct further ablation on the effectiveness of our ∇HFC Image processing block as compared to EVP HFC block as shown in Table IV. **HFC vs ∇HFC** PDA with only ∇HFC the performance gain is 0.07 % with comparable MAE scores. So ∇HFC will give a slight performance gain in the PDA model as compared to the fixed HFC block in the case of EVP. **HFC vs $\nabla HFC + f_{emb}$** PDA with ∇HFC and f_{emb} jointly increase the performance by 0.43% as compared to EVP with HFC block. To summarize, using PDA with f_{emb} and *DiffIP* containing ∇HFC gives the performance benefit and is better than the state-of-the-art EVP and SAM-Adapter models.

VI. CONCLUSION

In this paper, we have introduced DiffPrompter, a novel differentiable visual and latent prompting mechanism for adaptors aimed at addressing semantic segmentation tasks. Our study concludes that the integration of the ∇HFC image processing block alongside our series (SDA) architectures exhibits exceptional performance in vehicle segmentation across datasets featuring adverse weather conditions. Furthermore, our ablation study validates the efficacy of concurrently learning image and latent embeddings, directly correlating with enhanced performance in segmentation tasks and achieving state-of-the-art results in out-of-distribution generalization.

REFERENCES

- [1] T. Chen, L. Zhu, C. Ding, R. Cao, S. Zhang, Y. Wang, Z. Li, L. Sun, P. Mao, and Y. Zang, "Sam fails to segment anything?—sam-adapter: Adapting sam in underperformed scenes: Camouflage, shadow, and more," *arXiv preprint arXiv:2304.09148*, 2023. 1, 2
- [2] W. Liu, X. Shen, C.-M. Pun, and X. Cun, "Explicit visual prompting for low-level structure segmentations," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 19434–19445. 1, 2, 3
- [3] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 3213–3223. 1
- [4] F. Yu, H. Chen, X. Wang, W. Xian, Y. Chen, F. Liu, V. Madhavan, and T. Darrell, "Bdd100k: A diverse driving dataset for heterogeneous multitask learning," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 2636–2645. 4
- [5] C. Sakaridis, D. Dai, and L. V. Gool, "Guided curriculum model adaptation and uncertainty-aware evaluation for semantic nighttime image segmentation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 7374–7383. 1, 4

- [6] X. Hu, X. Xu, and Y. Shi, "How to efficiently adapt large segmentation model(sam) to medical images," 2023. 1
- [7] S. Kalwar, D. Patel, A. Aanegola, K. R. Konda, S. Garg, and K. M. Krishna, "Gdip: Gated differentiable image processing for object detection in adverse conditions," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 7083–7089. 2, 3
- [8] W. Liu, X. Shen, C.-M. Pun, and X. Cun, "Explicit visual prompting for universal foreground segmentations," *arXiv preprint arXiv:2305.18476*, 2023. 2
- [9] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3431–3440. 2
- [10] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention—MICCAI 2015: 18th International Conference, Munich, Germany, October 5–9, 2015, Proceedings, Part III 18*. Springer, 2015, pp. 234–241. 2
- [11] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 4, pp. 834–848, 2017. 2
- [12] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2881–2890. 2
- [13] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7794–7803. 2
- [14] E. Xie, W. Wang, Z. Yu, A. Anandkumar, J. M. Alvarez, and P. Luo, "Segformer: Simple and efficient design for semantic segmentation with transformers," in *Neural Information Processing Systems (NeurIPS)*, 2021. 2
- [15] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo *et al.*, "Segment anything," *arXiv preprint arXiv:2304.02643*, 2023. 2
- [16] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell *et al.*, "Language models are few-shot learners," *Advances in neural information processing systems*, vol. 33, pp. 1877–1901, 2020. 2
- [17] M. Jia, L. Tang, B.-C. Chen, C. Cardie, S. Belongie, B. Hariharan, and S.-N. Lim, "Visual prompt tuning," in *European Conference on Computer Vision*. Springer, 2022, pp. 709–727. 2
- [18] M. Sandler, A. Zhmoginov, M. Vladymyrov, and A. Jackson, "Fine-tuning image transformers using learnable memory," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022, pp. 12 155–12 164. 2
- [19] W. Liu, G. Ren, R. Yu, S. Guo, J. Zhu, and L. Zhang, "Image-adaptive yolo for object detection in adverse weather conditions," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2022. 2
- [20] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018. 2
- [21] C. Sakaridis, D. Dai, and L. Van Gool, "Acde: The adverse conditions dataset with correspondences for semantic driving scene understanding," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 10 765–10 775. 4
- [22] O. Zendel, K. Honauer, M. Murschitz, D. Steininger, and G. F. Dominguez, "Wilddash - creating hazard-aware benchmarks," in *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018. 4
- [23] O. Zendel, M. Schörghuber, B. Rainer, M. Murschitz, and C. Beleznaï, "Unifying panoptic segmentation for autonomous driving," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022, pp. 21 351–21 360. 4
- [24] D.-P. Fan, M.-M. Cheng, Y. Liu, T. Li, and A. Borji, "Structure-measure: A new way to evaluate foreground maps," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 4548–4557. 4
- [25] D.-P. Fan, C. Gong, Y. Cao, B. Ren, M.-M. Cheng, and A. Borji, "Enhanced-alignment measure for binary foreground map evaluation," in *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*. International Joint Conferences on Artificial Intelligence Organization, 7 2018, pp. 698–704. [Online]. Available: <https://doi.org/10.24963/ijcai.2018/97> 5