SelfPromer: Self-Prompt Dehazing Transformers with Depth-Consistency

Cong Wang1, Jinshan Pan2, Wanyu Lin1, Jiangxin Dong2, Wei Wang3, Xiao-Ming Wu1

1Department of Computing, The Hong Kong Polytechnic University
2School of Computer Science and Engineering, Nanjing University of Science and Technology
3International School of Information Science and Engineering, Dalian University of Technology

Abstract
This work presents an effective depth-consistency Self-Prompt Transformer, terms as SelfPromer, for image dehazing. It is motivated by an observation that the estimated depths of an image with haze residuals and its clear counterpart vary. Enforcing the depth consistency of dehazed images with clear ones, therefore, is essential for dehazing. For this purpose, we develop a prompt based on the features of depth differences between the hazy input images and corresponding clear counterparts that can guide dehazing models for better restoration. Specifically, we first apply deep features extracted from the input images to the depth difference features for generating the prompt that contains the haze residual information in the input. Then we propose a prompt embedding module that is designed to perceive the haze residuals, by linearly adding the prompt to the deep features. Further, we develop an effective prompt attention module to pay more attention to haze residuals for better removal. By incorporating the prompt, prompt embedding, and prompt attention into an encoder-decoder network based on VQGAN, we can achieve better perception quality. As the depths of clear images are not available at inference, and the dehazed images with one-time feed-forward execution may still contain a portion of haze residuals, we propose a new continuous self-prompt inference that can iteratively correct the dehazing model towards better haze-free image generation. Extensive experiments show that our SelfPromer performs favorably against the state-of-the-art approaches on both synthetic and real-world datasets in terms of perception metrics including NIQE, PI, and PIQE. The source codes will be made available at https://github.com/supersupercong/SelfPromer.

Introduction
Recent years have witnessed advanced progress in image dehazing due to the development of deep dehazing models. Mathematically, the haze process is usually modeled by an atmospheric light scattering model (He, Sun, and Tang 2011) formulated as:

$$I(x) = J(x)T(x) + (1 - T(x))A,$$

where I and J denote a hazy and haze-free image, respectively, and A denotes the global atmospheric light, $x$ denotes the pixel index, and the transmission map $T$ is usually modeled as $T(x) = e^{-\beta d(x)}$ with the scene depth $d(x)$, and the scattering coefficient $\beta$ reflects the haze density.

Most existing works develop various variations of deep Convolutional Neural Networks (CNNs) for image dehazing (Liu et al. 2019a; Dong et al. 2020; Dong and Pan 2020; Chen et al. 2021; Jin et al. 2022, 2023; Wang et al. 2023). They typically compute a sequence of features from the hazy input images and directly reconstruct the clear ones based on the features, which have achieved state-of-the-art results on benchmarks (Li et al. 2019) in terms of PSNRs and SSIMs. However, as dehazing is ill-posed, very small errors in the estimated features may degrade the performance. Existing works propose to use deep CNNs as image priors and then restore the clear images iteratively. However, they cannot effectively correct the errors or remove the haze residuals in the dehazed images as these models are fixed in the iterative process (Liu et al. 2019b). It is noteworthy that the human visual system generally possesses an intrinsic correction mechanism that aids in ensuring optimal results for a task. This phenomenon has been a key inspiration behind the development of a novel dehazing approach incorporating a correction mechanism that guides deep models toward better haze-free results generation.

Specifically, if a dehazed result exists haze residuals, a correction mechanism can localize these regions and guide the relevant task toward removing them. Notably, NLP-based text prompt learning has shown promise in guiding the models by correcting the predictions (Liu et al. 2023). However, text-based prompts may not be appropriate for tasks that require solely visual inputs without accompanying text. Recent works (Herzig et al. 2022; Gan et al. 2023) attempted to address this issue by introducing text-free prompts into vision tasks. For instance, PromptonomyViT (Herzig et al. 2022) evaluates the adaptation of multi-task prompts such as depth, normal, and segmentation to improve the performance of the video Transformers. Nevertheless, these prompts may not be suitable for image dehazing tasks, as they could not capture the haze-related content.

To better guide the deep model for better image dehazing, this work develops an effective self-prompt dehazing Transformer. Specifically, it explores the depth consistency of hazy images and their corresponding clear ones as a prompt. In particular, our study is motivated by the substan-
The difference map (e) is derived by comparing the estimated depth of hazy images to their clear counterparts. The clear counterpart in the image dehazing task.

We propose a new continuous self-prompt inference approach to iteratively correct the deep models toward better haze-free image generation.

### The Proposed SelfPromer

Our SelfPromer comprises two branches: the prompt branch and the self-prompt dehazing Transformer branch. The prompt branch generates a prompt by using the deep depth difference and deep feature extracted from the hazy input. The other branch exploits the generated prompt to guide the deep model for image dehazing. We incorporate a prompt embedding module and prompt attention module to perceive and pay more attention to haze residuals for better removal. The proposed modules are formulated into an encoder-decoder architecture based on VQGAN for better perception quality (Zhou et al. 2022; Chen et al. 2022).

### Overall Framework

Fig. 2 illustrates our method at the training stage. Given a hazy images I, we first utilize trainable encoder $Enc(\cdot)$ to extract features:

$$F_{Enc} = Enc(I).$$

Then, we compute the depth difference of the hazy image $I$ and its corresponding clear image $J$ in feature space:

$$D_1 = DE(I); D_2 = DE(J),$$

$$F_{D_1} = Enc_{pre}(D_1); F_{D_2} = Enc_{pre}(D_2),$$

$$F_{diff} = |F_{D_1} - F_{D_2}|,$$

where $DE(\cdot)$ denotes the depth estimator\(^1\) (Ranftl et al. 2022). $Enc_{pre}(\cdot)$ denotes the pre-trained VQGAN encoder which is frozen when training our dehazing models.

\(^1\)We chose DPT, NextVITL_384 to balance accuracy, speed, and model size: https://github.com/isl-org/MiDaS.
Next, we exploit $F_{\text{Diff}}$ to build the Prompt, and develop a prompt embedding module and a prompt attention module in Transformers, i.e., PTB to better generate haze-aware features:

$$
\text{Prompt} = F_{\text{Diff}} \cdot F_{\text{Enc}}, \quad \# \text{Prompt} \quad (4a)
$$

$$
F_{\text{ProEmbed}} = \text{Prompt} + F_{\text{Enc}}, \quad \# \text{Prompt Embedding} \quad (4b)
$$

$$
F_{\text{PTB}} = PTB(\text{Prompt}, F_{\text{ProEmbed}}), \quad \# \text{Prompt Transformer} \quad (4c)
$$

where $F_{\text{ProEmbed}}$ means the features of prompt embedding.

The generated feature $F_{\text{PTB}}$ is further matched with the learned haze-free Codebook at the pre-trained VQGAN stage by the Lookup method (Esser, Rombach, and Ommer 2021; Zhou et al. 2022):

$$
F_{\text{enc}} = \text{Lookup}(F_{\text{PTB}}, \text{Codebook}). \quad (5)
$$

Finally, we reconstruct the dehazing images $\tilde{J}$ from the matched features $F_{\text{enc}}$ by decoder of pre-trained VQGAN $D_{\text{rec}}^{\text{frozen}}$ with residual learning (Chen et al. 2022) by mutual deformable fusion module MDFM:

$$
\tilde{J} = D_{\text{rec}}^{\text{frozen}}(F_{\text{enc}}) + MDFM(F_{\text{enc}}, F_{\text{PTB}}), \quad (6)
$$

where $F_{\text{enc}}$ means the encoder features at s scale, while $F_{\text{PTB}}$ denotes the s× upsampling features of PTB. We conduct the residual learning with MDFM in {1, 1/2, 1/4, 1/8} scales between the encoder and decoder like FeMaSR (Chen et al. 2022). Here, $F_{\text{PTB}}$ denotes the $F_{\text{enc}}$ in Eq. (2).

**Loss Functions.** We use pixel reconstruction loss $\mathcal{L}_{\text{rec}}$, codebook loss $\mathcal{L}_{\text{code}}$, perception loss $\mathcal{L}_{\text{per}}$, and adversarial loss $\mathcal{L}_{\text{adv}}$ to measure the error between the dehazed images $\tilde{J}$ and the corresponding ground truth $J$:

$$
\mathcal{L} = \mathcal{L}_{\text{rec}} + \lambda_{\text{code}}\mathcal{L}_{\text{code}} + \lambda_{\text{per}}\mathcal{L}_{\text{per}} + \lambda_{\text{adv}}\mathcal{L}_{\text{adv}}, \quad (7)
$$

where

$$
\mathcal{L}_{\text{rec}} = ||\tilde{J} - J||_1 + \lambda_{\text{SSIM}}(1 - \text{SSIM}(\tilde{J}, J)), \quad (8a)
$$

$$
\mathcal{L}_{\text{code}} = ||z_{\text{enc}} - z_{\text{dec}}||_2, \quad (8b)
$$

$$
\mathcal{L}_{\text{per}} = ||\Phi(J) - \Phi(\tilde{J})||_2^2, \quad (8c)
$$

$$
\mathcal{L}_{\text{adv}} = E_j[\log D(J)] + E_{\tilde{J}}[1 - \log D(\tilde{J})], \quad (8d)
$$

Figure 2: SelfPromer at training stage. Our method comprises two branches: prompt branch and self-prompt dehazing Transformer branch. The prompt branch generates a prompt by using the deep depth difference and deep feature extracted from the hazy input. The other branch exploits the generated prompt to guide the deep model for image dehazing. We incorporate a prompt embedding module and prompt attention module to perceive and pay more attention to the haze residuals for better removal. The proposed modules are formulated into an encoder-decoder architecture based on VQGAN for better perception quality. MDFM is detailed in Eq. (11). The inference is illustrated in Fig. 3.
where $SSIM(\cdot)$ denotes the structural similarity (Wang et al. 2004) for better structure generation. $z_q$ is the haze-free codebook features by feeding haze-free images $J$ to pre-trained VQGAN while $\hat{z}_q$ is the reconstructed codebook features. $\Phi(\cdot)$ denotes the feature extractor of VGG19 (Simonyan and Zisserman 2015). $D$ is the discriminator (Zhu et al. 2017). $\lambda_{\text{code}}, \lambda_{\text{adv}}, \lambda_{\text{adv}}$, and $\lambda_{\text{sim}}$ are weights.

For inference, we propose a new self-prompt inference approach as our training stage involves the depth of clear images to participate in forming the prompt while clear images are not available at testing.

**Self-Prompt Transformers**

The self-prompt Transformer contains the prompt generated by the prompt branch, a prompt embedding module, and a prompt attention module which is contained in the prompt Transformer block. In the following, we introduce the definition of the prompt, prompt embedding module, and prompt attention module, and prompt Transformer block in detail.

**Prompt** (Definition). The prompt is based on the estimated depth difference between the input image and its clear counterpart. It is defined in Eq. (4a) which can better contain haze residual features as $F_{\text{D}_{\text{diff}}}$ with higher response value reveals inconsistent parts which potentially correspond to the haze residuals in the input hazy image.

**Prompt Embedding**. Existing Transformers (Zheng et al. 2022) usually use the position embedding method (Fig. 4(a)) to represent the positional correlation, which does not contain haze-related information so that it may not effectively perceive the haze residual information well. Moreover, image restoration requires processing different input sizes at inference while the position embedding is defined with fixed parameters at training (Zheng et al. 2022). Hence, position embedding may be not a good choice for image dehazing. To overcome these problems, we propose prompt embedding which is defined in Eq. (4b). By linearly adding the extracted features $F_{\text{Enc}}$ with Prompt, the embedded feature $F_{\text{ProEmbed}}$ perceives the haze residual features as Prompt extracts the haze residual features. Note that as $F_{\text{ProEmbed}}$ has the same size as $F_{\text{Enc}}$, it does not require fixed sizes like position embedding.

**Prompt Attention**. Existing Transformers usually extract Query $Q$, Key $K$, and Value $V$ from input features to estimate scaled-dot-product attention shown in Fig. 4(c). Although Transformers are effective for feature representation, the standard operation may be not suitable for image dehazing. To ensure the Transformers pay more attention to haze residuals for better removal, we propose prompt attention $\text{ProAtt}(\cdot)$ by linearly adding the query with Prompt:

$$Q = Q + \text{Prompt}, \quad (9a)$$

$$\text{ProAtt}(Q, K, V) = \text{Softmax} \left( \frac{QK^T}{\sqrt{d_{\text{head}}}} \right) V, \quad (9b)$$

where $d_{\text{head}}$ means the number of the head. We set $d_{\text{head}}$ as 8 in this paper. Fig. 4(d) illustrates the proposed prompt attention. Note that as $Q$ in attention is to achieve the similarity relation for expected inputs (Ding et al. 2021), our prompt attention by linearly adding the prompt Prompt with the

Query $Q$ can pay more attention to haze residuals for better removal.

**Prompt Transformer Block**. According to the above attention design, our prompt Transformer block (PTB) can be sequentially computed as:

$$Q, K, V = LN(X^{l-1}), \quad (10a)$$

$$X^{l} = \text{ProAtt}(Q, K, V) + X^{l-1}, \quad (10b)$$

$$X^{l} = \text{MLP}(LN(X^{l})) + X^{l}, \quad (10c)$$

where $X^{l-1}$ and $X^{l}$ mean the input and output of the $l$th prompt Transformer block. Specially, $X^{0}$ is the $F_{\text{ProEmbed}}$. $LN$ and $\text{MLP}$ denote the layer normalization and multilayer perceptron. The PTB is shown in the right part of Fig. 2.

It is worth noting that our prompt embedding and prompt attention are flexible as we can manually set $F_{\text{D}_{\text{diff}}} = 0$, the network thus automatically degrade to the model without prompt, which will be exploited to form our continuous self-prompt inference.

**Mutual Deformable Fusion Module**

As VQGAN is less effective for preserving details (Gu et al. 2022; Chen et al. 2022), motivated by the deformable models (Dai et al. 2017; Zhu et al. 2019) that can better fuse features, we propose a mutual deformable fusion module (MDFM) by fusing features mutually to adaptively learn more suitable offsets for better feature representation:

$$\text{off}_1 = \text{Conv}(C[F_{\text{Enc}}, F_{\text{PTB}}]); \text{off}_2 = \text{Conv}(C[F_{\text{PTB}}, F_{\text{Enc}}]), \quad (11a)$$

$$Y_1 = \text{DMC}(F_{\text{Enc}}, \text{off}_1); Y_2 = \text{DMC}(F_{\text{PTB}}, \text{off}_2), \quad (11b)$$

$$F_{\text{MDFM}} = \text{Conv}(C[Y_1, Y_2]), \quad (11c)$$

where $\text{Conv}(\cdot), C[\cdot]$, and $\text{DMC}(\cdot)$ respectively denote the $1 \times 1$ convolution, concatenation, and deformable convolution. $\text{off}_k (k = 1, 2)$ denotes the estimated offset.
image generation: The iterative inference is conducted to correct the deep participation in building the prompt to conduct prompt dehazing. Our trained network and obtain clearer dehazed results which depth difference attention through linear addition. By setting feature-level, but these images are unavailable at inference. Addition-

Our model requires the depth of clear images during training.

Continuous Self-Prompt Inference

Our model requires the depth of clear images during training, but these images are unavailable at inference. Additionally, dehazed images generated by a one-time feed-forward execution may still contain some haze residuals. To address these issues, we propose a continuous self-prompt inference approach that leverages prompt embedding and prompt attention through linear addition. By setting feature-level depth difference $F_{\text{Diff}}$ to zero, we can feed hazy images to our trained network and obtain clearer dehazed results which participate in building the prompt to conduct prompt dehazing. The iterative inference is conducted to correct the deep models to ensure the deep models toward better haze-free image generation:

$$J_i^{\text{w/o prompt}} = \mathcal{N}^{\text{w/o prompt}} (J_{i-1}^{\text{w/o prompt}}), \text{ set } F_{\text{Diff}} = 0, \text{ # Step 1 } (12a)$$

$$\text{Prompt} = F_{\text{Diff}} \cdot F_{\text{Enc}}; F_{\text{Enc}} = \mathcal{E}nc(J_i^{\text{w/o prompt}}), \text{ # Step 2 } (12b)$$

$$J_i^{\text{w/o prompt}} = \mathcal{N}^{\text{prompt}} (J_i^{\text{w/o prompt}}, \text{ Prompt}), \text{ # Step 3 } (12c)$$

$$J_i^{\text{w/o prompt}} = J_i^{\text{prompt}}, (i = 1, 2, \cdots), \text{ # Step 4 } (12d)$$

where $\mathcal{N}^{\text{w/o prompt}}$ denotes our trained network without prompt by setting $F_{\text{Diff}}$ as zero, while $\mathcal{N}^{\text{prompt}}$ means our trained network with prompt. $F_{\text{Diff}} = \mathcal{E}nc_{\text{pre}} (DE(J_i^{\text{w/o prompt}})) - \mathcal{E}nc_{\text{pre}} (DE(J_i^{\text{prompt}}))$. $J_0^{\text{w/o prompt}}$ denotes the original hazy images, while $J_i^{\text{w/o prompt}}$ in Eq. (12a) is regarded as the image with haze residuals and $J_i^{\text{w/o prompt}}$ in Eq. (12c) is regarded as the clear counterpart of $J_i^{\text{w/o prompt}}$. $J_i^{\text{prompt}}$ means the $i$th prompt dehazing results.

According to Eq. (12), the inference is a continuous self-promp

scheme, i.e., we get the clear images from the hazy image itself by feeding it to $\mathcal{N}^{\text{w/o prompt}}$ to participate in producing the prompt and the inference is continuously conducted. Fig. 3 better illustrates the inference process.

Fig. 5 shows our continuous self-prompt at 2nd and 3rd prompts outperforms the baseline which uses ground-truth (GT) to participate in forming the prompt like the process of the training stage.

Figure 5: Continuous self-prompt inference vs. GT guidance (Baseline) on the SOTS-indoor dataset. GT guidance means we use the GT image to participate in forming the prompt at inference like the process of the training stage, which serves as the baseline.

**Experiments**

In this section, we evaluate the effectiveness of our method against state-of-the-art ones (SOTAs) on commonly used benchmarks and illustrate the effectiveness of the key components in the proposed method.

**Implementation Details.** We use 10 PTBs, i.e., $l = 10$, in our model. The details about the VQGAN are presented in the supplementary materials. We crop an image patch of $256 \times 256$ pixels. The batch size is 10. We use ADAM (Kingma and Ba 2015) with default parameters as the optimizer. The initial learning rate is 0.0001 and is divided by 2 at 160K, 320K, and 400K iterations. The model training terminates after 500K iterations. The weight parameters $\lambda_{\text{code}}, \lambda_{\text{per}}, \lambda_{\text{adv}},$ and $\lambda_{\text{sim}}$ are empirically set as 1, 1, 0.1, and 0.5. Our implementation is based on the PyTorch.

**Synthetic Datasets.** Following the protocol of (Yang et al. 2022), we use the RESIDE ITS (Li et al. 2019) as our training dataset and the SOTS-indoor and SOTS-outdoor (Li et al. 2019) as the testing datasets.

**Real-world Datasets.** Li et al. (2019) collect large-scale real-world hazy images, called UnannotatedHazyImages. We use these images as a real-world hazy dataset.

**Evaluation Metrics.** As we mainly aim to recover images with better perception quality, we use widely-used NIQE (Mittal, Soundararajan, and Bovik 2013), PI (Ma et al. 2017), and PIQE (N. et al. 2015) to measure restoration quality. Since the distortion metrics PSNR and SSIM (Wang et al. 2004) cannot model the perception quality well, we use them for reference only. Notice that all metrics are re-computed for fairness. We use the grayscale image to compute the PSNR and SSIM. We compute NIQE and PI by the provided metrics at https://pypi.org/project/pyiqa/. The PIQE is computed via https://github.com/buyizhiyou/NRVQA.

**Main Results**

**Results on Synthetic Datasets.** Tab. 1 and Tab. 2 respectively report the comparison results with SOTAs on the SOTS-indoor and SOTS-outdoor datasets (Li et al. 2019). Our method achieves better performance in terms of NIQE, PI, and PIQE, indicating the generated results by our method possess higher perception quality. Fig. 6 and Fig. 7 show that our method restores much clearer images while the evaluated approaches generate the results with haze residual or artifacts. As we train the network with a one-time feed-forward process, PSNRs and SSIMs are naturally decreased (SelfPromer vs. SelfPromer$_3$ in Tabs. 1 and 2) when inference is conducted iteratively. We argue distortion metrics including PSNRs and SSIMs are not good measures for image dehazing as Figs. 6 and 7 have shown methods with higher PSNR and SSIMs cannot recover perceptual results, e.g., Dhamer (Guo et al. 2022) and D4 (Yang et al. 2022), while our method with better perception metrics can generate more realistic results.

**Results on Real-world Datasets.** Tab. 3 summarises the comparison results on the real-world datasets (Li et al. 2019), where our method performs better than the evaluated methods. Fig. 8 illustrates that our method generates an image with vivid color and finer details.
### Table 1: Comparisons on SOTS-indoor dataset. Our method achieves better performance in terms of NIQE, PI, and PIQE. The best results are marked in bold. $\downarrow$ (↑) denotes lower (higher) is better. **SelfPromer*$_i$ means the $i$th prompt results.

<table>
<thead>
<tr>
<th>Methods</th>
<th>GridNet</th>
<th>PFDN</th>
<th>UHD</th>
<th>PSD</th>
<th>Uformer</th>
<th>Restormer</th>
<th>D4</th>
<th>Dehamer</th>
<th>*<em>SelfPromer</em>$_1$</th>
<th>*<em>SelfPromer</em>$_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PIQE $\downarrow$</td>
<td>28.924</td>
<td>32.157</td>
<td>39.204</td>
<td>35.174</td>
<td>29.806</td>
<td>29.384</td>
<td>30.480</td>
<td>32.446</td>
<td>30.596</td>
</tr>
<tr>
<td>Distortion</td>
<td>PSNR $\uparrow$</td>
<td>32.306</td>
<td>33.243</td>
<td>16.920</td>
<td>13.934</td>
<td>33.947</td>
<td>36.979</td>
<td>19.142</td>
<td>36.600</td>
<td>35.960</td>
</tr>
<tr>
<td></td>
<td>SSIM $\uparrow$</td>
<td>0.9840</td>
<td>0.9827</td>
<td>0.7831</td>
<td>0.7160</td>
<td>0.9846</td>
<td>0.9900</td>
<td>0.8520</td>
<td>0.9865</td>
<td>0.9877</td>
</tr>
</tbody>
</table>

Table 1: Comparisons on SOTS-indoor dataset. Our method achieves better performance in terms of NIQE, PI, and PIQE. The best results are marked in bold. $\downarrow$ (↑) denotes lower (higher) is better. **SelfPromer*$_i$ means the $i$th prompt results.

![Image](attachment:image1.png)

Figure 6: Visual comparisons on SOTS-indoor. SelfPromer generates clearer results, even than the GT image.

### Table 2: Comparisons on SOTS-outdoor. SelfPromer achieves better perception metrics including NIQE, PI, and PIQE, suggesting that the proposed method has a better generalization ability to unseen images for more natural results generation.

<table>
<thead>
<tr>
<th>Methods</th>
<th>GridNet</th>
<th>PFDN</th>
<th>UHD</th>
<th>PSD</th>
<th>Uformer</th>
<th>Restormer</th>
<th>D4</th>
<th>Dehamer</th>
<th>*<em>SelfPromer</em>$_1$</th>
<th>*<em>SelfPromer</em>$_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Perception</td>
<td>NIQE $\downarrow$</td>
<td>2.844</td>
<td>2.843</td>
<td>3.756</td>
<td>2.884</td>
<td>2.903</td>
<td>2.956</td>
<td>2.917</td>
<td>3.164</td>
<td><strong>2.646</strong></td>
</tr>
<tr>
<td></td>
<td>PI $\downarrow$</td>
<td>2.070</td>
<td>2.326</td>
<td>3.381</td>
<td>2.392</td>
<td>2.241</td>
<td>2.254</td>
<td>2.137</td>
<td>2.251</td>
<td><strong>2.003</strong></td>
</tr>
<tr>
<td></td>
<td>SSIM $\uparrow$</td>
<td>0.8016</td>
<td>0.8532</td>
<td>0.6074</td>
<td>0.7488</td>
<td>0.8798</td>
<td>0.8634</td>
<td>0.9540</td>
<td>0.8926</td>
<td>0.8771</td>
</tr>
</tbody>
</table>

Table 2: Comparisons on SOTS-outdoor. SelfPromer achieves better perception metrics including NIQE, PI, and PIQE, suggesting that the proposed method has a better generalization ability to unseen images for more natural results generation.

![Image](attachment:image2.png)

Figure 7: Visual comparisons on SOTS-outdoor. SelfPromer is able to generate more natural results. Note that our method produces more consistent colors in the sky region, while the others generate inconsistent colors and the D4 (Yang et al. 2022) leaves extensive haze.

### Analysis and Discussion

We further analyze the effectiveness of the proposed method and understand how it works on image dehazing. The results in this section are obtained from the SOTS-indoor dataset if not further mentioned. Our results are from the $i$th prompt inference for fair comparisons, i.e., $i = 1$ in Eq. (12) if not further specifically mentioned.

**Effectiveness of prompt.** Initially, we assess the effect of the prompt on image dehazing. Notably, various prospective prompt candidates exist, such as image-level depth difference as the input of the VQGAN encoder or concatenation between deep features extracted from the input and depth features as the input of the Transformers. Our proposed prompt is compared with these candidates, as illustrated in Tab. 4(b) and 4(c), demonstrating that none of these candidates outperforms our proposed prompt.

Note that our method without prompt leads to a similar model with CodeFormer (Zhou et al. 2022) which directly inserts regular Transformers into VQGAN. Tab. 4 shows prompt help yield superior perception quality than the model without prompt (Tab. 4(a)). The efficacy of our model with the prompt is further affirmed by Fig. 9, indicating that the
Table 3: Comparisons on real-world dataset. SelfPromer achieves better performance, indicating that our method is more robust to real-world scenarios for realistic results generation.

<table>
<thead>
<tr>
<th>Methods</th>
<th>GridNet</th>
<th>PFDN</th>
<th>UHD</th>
<th>PSD</th>
<th>Uformer</th>
<th>Restormer</th>
<th>D4</th>
<th>Dehamer</th>
<th>SelfPromer</th>
<th>SelfPromer3</th>
</tr>
</thead>
</table>

Figure 8: Visual comparisons on real-world dataset. Our SelfPromer is able to generate much clearer results.

Table 4: Effect of the proposed prompt. Feature-level depth difference is a better prompt formalization, while the concatenation in image-level and feature-level between the input image and its depth is not as well as ours.

<table>
<thead>
<tr>
<th>Experiments</th>
<th>NIQE ↓</th>
<th>PI ↓</th>
<th>PIQE ↓</th>
</tr>
</thead>
<tbody>
<tr>
<td>(a) Without the prompt</td>
<td>4.258</td>
<td>3.937</td>
<td>31.904</td>
</tr>
<tr>
<td>(b) Image-level depth difference</td>
<td>4.901</td>
<td>4.343</td>
<td>32.141</td>
</tr>
<tr>
<td>(c) Concat of image and depth features</td>
<td>4.362</td>
<td>4.077</td>
<td>34.107</td>
</tr>
<tr>
<td>(d) Feature-level depth difference (Ours)</td>
<td>4.252</td>
<td>3.926</td>
<td>30.596</td>
</tr>
</tbody>
</table>

Figure 9: Visual comparisons of the model without prompt (b) and with prompt (c) on real-world scenarios.

Table 5: Effectiveness of prompt embedding and attention.

<table>
<thead>
<tr>
<th>Experiments</th>
<th>NIQE ↓</th>
<th>PI ↓</th>
<th>PIQE ↓</th>
</tr>
</thead>
<tbody>
<tr>
<td>(a) Without embedding</td>
<td>4.410</td>
<td>4.113</td>
<td>32.193</td>
</tr>
<tr>
<td>(b) Position embedding</td>
<td>4.267</td>
<td>3.992</td>
<td>31.877</td>
</tr>
<tr>
<td>(c) Regular attention</td>
<td>4.300</td>
<td>4.102</td>
<td>31.486</td>
</tr>
<tr>
<td>(d) Proposed (Ours)</td>
<td><strong>4.252</strong></td>
<td>3.926</td>
<td><strong>30.596</strong></td>
</tr>
</tbody>
</table>

Figure 10: Effectiveness of continuous self-prompt (Ours) vs. recurrent dehazing (Recur.). ‘Ours w/o prompt’ means the results of Eq. (12a).

Effect of the number of steps in continuous self-prompt. The inference stage involves several steps to generate the prompt for better image dehazing. We thus examine the effect of the number of steps in the continuous self-prompt. Fig. 10 reveals that the optimal performance is achieved with
a number of steps equal to 3 in the continuous self-prompts (i.e., \( i = 3 \) in Eq. (12)), in terms of NIQE. Notably, additional prompts do not improve the dehazing performance any further. One real-world example in Fig. 11 demonstrates that our continuous self-prompt method can gradually enhance dehazing quality.

**Continuous self-prompt vs. recurrent dehazing.** We use the continuous self-prompt approach to restore clear images progressively at inference. To determine whether a recurrent method that is training our model without prompt achieves similar or better results, we compare our proposed method with it in Fig. 10, demonstrating that the recurrent method is not as good as our continuous self-prompt.

**Continuous self-prompt vs. GT guidance.** Fig. 5 compares the NIQE performance of ground truth (GT) guidance with that of the continuous self-prompt algorithm. Results show that while GT guidance performs better than the 1\(^{st} \) prompt, it falls short of the effectiveness of the 2\(^{nd} \) and 3\(^{rd} \) prompts. This is likely due to GT guidance’s limited ability to handle haze residuals which may still exist in the dehazed images, which are addressed by the self-prompt’s ability to exploit residual haze information to progressively improve dehazing quality over time. Moreover, as GT is not available in the real world, these findings may further support the use of self-prompt as a more practical alternative.

**Depth-consistency.** Fig. 12 shows heat maps of depth differences obtained by the continuous self-prompt inference with different prompt steps. The results demonstrate both image-level and feature-level depth differences decrease as the number of prompt steps increases, indicating the depths obtained with the prompt, i.e., Eq. (12c), become increasingly consistent with those obtained without it, i.e., Eq. (12a).

**Applications to Low-Light Image Enhancement**

Furthermore, we extend the application of our method, SelfPromer, to the domain of low-light image enhancement. To assess its performance, we conduct a comparative analysis with current state-of-the-art methods, SNR (Xu et al. 2022) and LLFlow (Wang et al. 2022). All these methods are trained using the widely adopted LOL dataset (Wei et al. 2018). Fig. 13 shows several visual examples sourced from real-world benchmarks (Lee, Lee, and Kim 2013; Guo, Li, and Ling 2017). These illustrative results effectively underscore our approach is capable of generating results that are notably more true-to-life, with colors that appear more natural. On the contrary, existing state-of-the-art methods tend to yield results that suffer from under-/over-exposure issues.

**Conclusion**

We have proposed a simple yet effective self-prompt Transformer for image dehazing by exploring the prompt built on the estimated depth difference between the image with haze residuals and its clear counterpart. We have shown that the proposed prompt can guide the deep model for better image dehazing. To generate better dehazing images at the inference stage, we have proposed continuous self-prompt inference, where the proposed prompt strategy can remove haze progressively. We have shown that our method generates results with better perception quality in terms of NIQE, PI, and PIQE.

**References**

Dong, J.; and Pan, J. 2020. Physics-Based Feature Dehazing Networks. In ECCV, 188–204.