Iterative Predictor-Critic Code Decoding for Real-World Image Dehazing

Jiayi Fu[1] Siyu Liu[1] Zikun Liu[3] Chun-Le Guo[1,2]
Hyunhee Park[4] Ruiqi Wu[1] Guoqing Wang[5] Chongyi Li[1,2*]
[1]VCIP, CS, Nankai University [2]NKIARI, Shenzhen Futian
[3]Samsung R&D Institute China-Beijing [4]CIG, Samsung Electronics
[5]Donghai Laboratory, Zhoushan, Zhejiang
{fujiayi,liusiyu29}@mail.nankai.edu.cn, zikun.liu@samsung.com,
guochunle@nankai.edu.cn, inextg.park@samsung.com,
wuruiqi@mail.nankai.edu.cn, gqwang0420@hotmail.com, lichongyi@nankai.edu.cn

Abstract

*We propose a novel **I**terative **P**redictor-**C**ritic Code Decoding framework for real-world image dehazing, abbreviated as **IPC-Dehaze**, which leverages the high-quality codebook prior encapsulated in a pre-trained VQGAN. Apart from previous codebook-based methods that rely on one-shot decoding, our method utilizes high-quality codes obtained in the previous iteration to guide the prediction of the Code-Predictor in the subsequent iteration, improving code prediction accuracy and ensuring stable dehazing performance. Our idea stems from the observations that 1) the degradation of hazy images varies with haze density and scene depth, and 2) clear regions play crucial cues in restoring dense haze regions. However, it is non-trivial to progressively refine the obtained codes in subsequent iterations, owing to the difficulty in determining which codes should be retained or replaced at each iteration. Another key insight of our study is to propose Code-Critic to capture interrelations among codes. The Code-Critic is used to evaluate code correlations and then resample a set of codes with the highest mask scores, i.e., a higher score indicates that the code is more likely to be rejected, which helps retain more accurate codes and predict difficult ones. Extensive experiments demonstrate the superiority of our method over state-of-the-art methods in real-world dehazing. Our project page can be found at https://github.com/Jiayi-Fu/IPC-Dehaze.*

1. Introduction

In real-world scenarios, capturing hazy images is a common occurrence, wherein the visual output exhibits diminished clarity and sharpness. This phenomenon is primarily attributable to the presence of atmospheric particles that irregularly disperse and scatter light. The presence of haze can significantly impact the visual quality or accuracy of subsequent perception algorithms. Consequently, there has been a growing interest in image dehazing, aiming to restore clear images from their hazy counterparts.

Achieving a clean image from a hazy one is inherently challenging. Traditional methods address this ill-posed

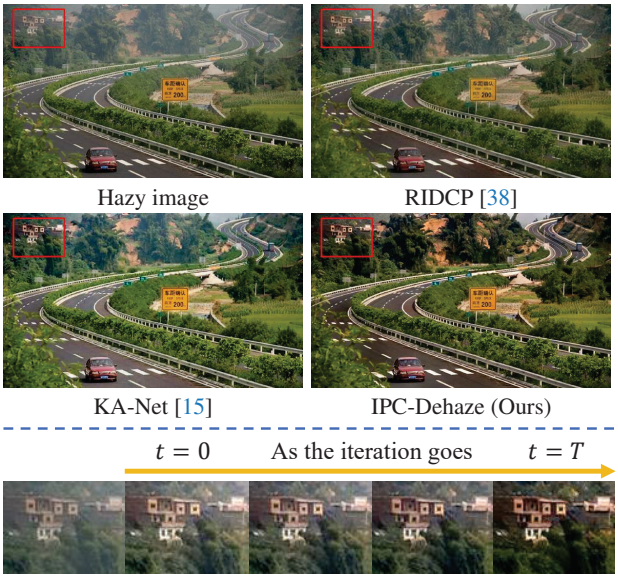


Figure 1. A comparison between the state-of-the-art real-world image dehazing methods and our IPC-Dehaze. In comparison, our result is sharper and clearer, with less color distortion and overexposure. The bottom images present the results of each iteration in our method, showing the continuous improvements with our core Predictor-Critic mechanism.

*Corresponding author

problem by leveraging various priors [2, 14, 19, 43], but they often struggle in real-world scenes due to ideal assumptions or limited generalization capabilities. The advent of deep learning has led to data-driven methods exhibiting remarkable performance in image dehazing. These methods either estimate the physical model-related transmission map [4, 30] or directly restore the clean image [12, 29]. However, they are commonly trained on images synthesized by ideal physical models, posing challenges when applied to complex real-world scenes.

To address this issue, some approaches focus on improving data synthesis [31, 40] or employ alternative training strategies like unsupervised learning [16, 41]. Recent work, RIDCP [38], highlights the importance of considering multiple degradation factors for synthesizing hazy images and demonstrates the effectiveness of high-quality codebook priors. In line with RIDCP, some works [10, 25] have been proposed for real-world dehazing. Despite the progress made by current methods, they still face some limitations. Specifically, they may overlook the spatial variations in degradation with haze density and scene depth, leading to over-recovery in thin haze areas or under-recovery in dense haze areas. Additionally, accurately recovering a challenging scene with a one-shot (i.e., only a single attempt or instance) mapping proves difficult, resulting in suboptimal performance.

Inspired by the physical phenomenon that areas with thin haze contain more information while areas with dense haze contain less, we found that dehazing efforts could benefit from focusing on relatively clear scenes first, before addressing more challenging areas. To achieve that, we follow previous methods [38] to utilize the high-quality codebook encapsulated in a pre-train VQGAN [13] as prior. Unlike previous codebook-based methods that perform a one-shot code prediction, we present a novel iterative decoding prediction framework, which performs dehazing in an easy-to-hard manner, improving the overall accuracy and stability of the dehazing process, as the comparison shown in Fig. 1.

In our method, an encoder first maps a hazy image into tokens. Then, according to the current tokens, a Predictor-Critic mechanism is used to alternately predict high-quality codes and evaluate which should be retained. As the iterations progress, the number of retained codes progressively grows until all codes are finalized. In particular, in the t-th iteration, we get the high-quality codes predicted in the previous iteration and a mask map that determines which codes are retained in this iteration. Next, we feed the mask map processed tokens into the Code-Predictor to predict all the high-quality codes in this iteration, and we also need to generate a mask map to benefit the next iteration. To achieve the iterative dehazing shown above, we fuse the tokens from clean and hazy images with a random mask. The processed tokens are fed to Code-Predictor to match the high-quality codebook, simulating the prediction of the t-th iteration during inference. Following the inference process, obtaining an effective and instructive mask map is important, so we propose Code-Critic to better reject the least likely codes and retain the good ones.

The **contributions** of this study are summarized as follows:

- We propose a novel iterative decoding dehazing framework. In comparison to one-shot methods, our method uses high-quality codes from previous iterations as cues to guide Code-Predictor in predicting the subsequent codes, implementing better iterative dehazing.
- We introduce Code-Critic to evaluate the interdependence in the output of the Code-Predictor, selecting which codes should be retained or rejected across iterative decoding steps to guide subsequent predictions. This module improves the coherence among selected codes and prevents error accumulation.
- With these novel designs, our approach surpasses state-of-the-art methods in real-world dehazing both qualitatively and quantitatively.

2. Related Work

Early single image dehazing methods [2, 14, 19, 43] recover images by estimating the parameters of imaging model [20]. However, hand-crafted priors suffer from poor generalization ability and cannot adapt to the various scenes. With the development of deep learning, data-driven methods have achieved remarkable results. One kind of approach [4, 23, 30] is inspired by the scattering model, but the performance is unsatisfactory when the scene does not conform to the ideal physical model. Another line [18, 29] is to directly obtain a clean image using a network. However, due to the domain gap between synthetic data and real data, the model trained on synthetic images often suffers from sub-optimal performance when applied to real-world hazy images.

There has been a remarkable amount of research focusing on real-world image dehazing. Real-world image dehazing methods can be roughly divided into three categories: 1) Domain Adaptation-based Method. Some methods employ domain adaptation to reduce the gap between the synthetic domain and the real domain [31, 32]. These supervised methods typically achieve excellent performance on synthetic datasets, but they often struggle to generalize well to real-world hazy images. 2) GAN-based Method. In contrast, GAN-based unsupervised dehazing methods learn the dehazing mapping from unpaired clean and hazy images. To produce clearer and more realistic images, Liu *et al*. [28] proposed a dehazing network based on a physical model and an enhancer network that supervises the mapping from the hazy domain to the clean domain. Yang *et al*. [40] introduced an unpaired image dehazing framework

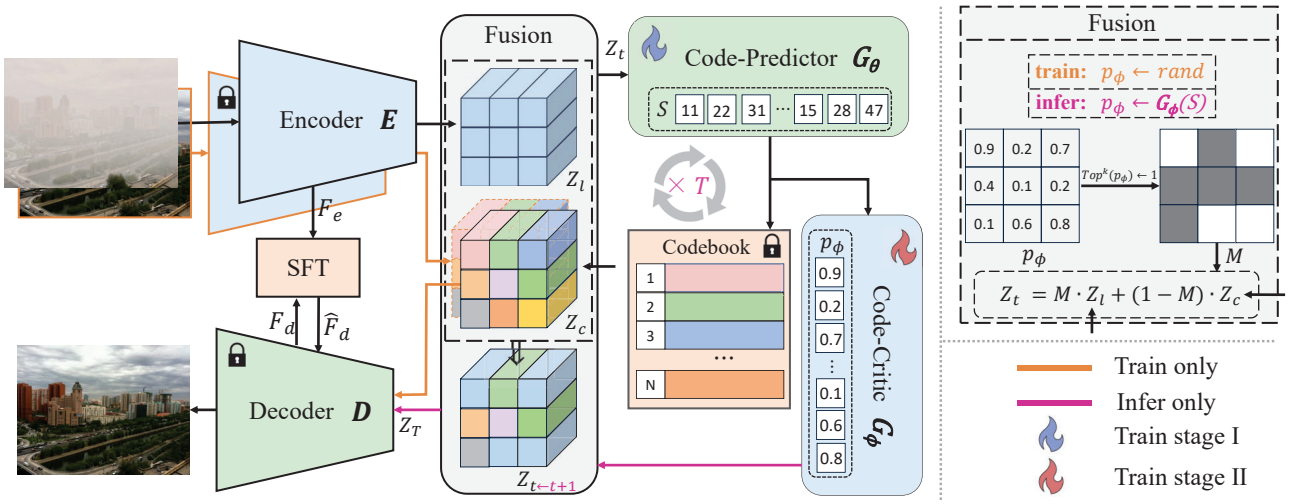


Figure 2. Overview of our IPC-Dehaze. *In the training phase*, we use fused tokens $Z_t = Z_l \odot M + Z_c \odot (1 - M)$ from the hazy and clean images, and predict the sequence codes S by Code-Predictor. We also train Code-Critic to evaluate each code in set S for potential rejection and resampling. *In the inference phase*, $Z_{t=0}$ is initially encoded as low-quality tokens Z_l. During the t-th iterative decoding step, the Code-Predictor takes Z_t as input, predicting the sequence codes S and the corresponding high-quality tokens Z_c. To retain the reliable codes and resample the others, the Code-Critic evaluates S and produces a mask map M by p_ϕ. This mask map M is then used to generate Z_{t+1} through a *Fusion* process. Following T iterations, Z_T is output to reconstruct the clean image by a decoder. The SFT refers to the Spital Feature Transform, which adjusts the feature within the encoder and decoder.

that utilizes self-enhancement to re-render blurred images with various haze densities. However, GAN-based methods often tend to produce unrealistic haze, which further compromises the overall dehazing performance. 3) Deep Prior-based Method. Some methods [10, 25] that introduce handcrafted priors in the network framework or loss functions still fail to overcome their inherent limitations. Some efforts have been made to utilize prior knowledge extracted from high-quality images to aid in image restoration. Some face restoration [17, 42] and super-resolution [7, 9] methods based on VQGAN [13] have demonstrated the superiority of high-quality priors. RIDCP [38] leverage a latent high-quality prior in image dehazing and achieve significant performance. However, the one-shot-based algorithm struggles with extremely dense haze and performs poorly in the inhomogeneous haze, failing to take full advantage of the information in the thin haze. To address these issues, our new framework introduces generative capability and high-quality priors and enhances the model's generalization through an iterative approach.

3. Methodology

In this work, we propose a novel image dehazing framework via iterative decoding, which is shown in Fig. 2. Firstly, to introduce robust high-quality priors, we pretrain a VQGAN [13] on high-quality datasets. In this stage, we can obtain a latent high-quality discrete codebook and the corresponding encoder and decoder. To match the correct code, we use the proposed Code-Predictor replacing the nearest neighbor matching (Stage I). To capture the global relationship of the code sequence, we introduce Code-Critic for selecting whether a code generated by the Code-Predictor is accepted or not during iterative decoding (Stage II). This ensures that the acceptance or rejection of a code should be not decided independently. We present the training process in Algorithm 1. During the inference phase, the process begins with tokens encoded from a hazy image. Over T iterations, the current token is fed into the Code-Predictor and the Code-Critic selects which high-quality codes will be retained for the next iteration. This process continues until all codes have been replaced with high-quality ones. We present the inference process in Algorithm 2.

3.1. Preliminary

Learning A Codebook via VQGAN. In order to alleviate the complexity of directly generating images in pixel space, VQVAE [33] proposes a vector quantized (VQ) autoencoder to learn discrete codebooks in the latent space. VQGAN [13] further enhances the perceptual quality of the reconstruction results by introducing adversarial loss and perceptual loss. Following VQGAN, the codebook learning consists of three parts:

i) The encoder E_H encodes the high-quality image patch $I_h \in \mathbb{R}^{H\times W\times 3}$ to latent features $Z_h = E_H(I_h) \in \mathbb{R}^{m\times n\times d}$, where the d denotes the dimension of code-

Algorithm 1 Training Process

Input: Z_l, Z_c, S_h, Code-Predictor G_θ, Code-Critic G_ϕ, mask schedule functions $\gamma(r)$, learning rate η, number of tokens in latent N.

repeat
  $r \sim \mathcal{U}_{(0,1)}$
  $M_t \leftarrow$ randomly sample($\lceil \gamma(r) \cdot N \rceil$)
  $Z_t \leftarrow Z_l \odot M_t + Z_c \odot (1 - M_t)$ {Confuse Z_l and Z_c based on M_t}
  $p_\theta \leftarrow G_\theta(Z_t)$
  $S \leftarrow \text{argmax}(p_\theta)$
  $\theta \leftarrow \theta - \eta \nabla_\theta \mathcal{L}_\theta$
  if training Code-Critic **then**
    $S \leftarrow$ sample from p_θ
    $M \leftarrow (S \neq S_h)$ {Build ground truth}
    $\phi \leftarrow \phi - \eta \nabla_\phi \mathcal{L}_\phi$
  end if
until convergence

Algorithm 2 Inference Process

Input: Encoder E_L, decoder D_H, Code-Predictor G_θ, Code-Critic G_ϕ, mask schedule functions $\gamma(r)$, codebook C, number of tokens in latent N, number of iterations T, low-quality images I_l

Output: clear images I_{rec}

$Z_l \leftarrow E_L(I_l)$
$M_1 \leftarrow 1$ {M_1 initialized to 1}
for $t \leftarrow 1$ **to** T **do**
  $Z_t \leftarrow Z_l \odot M_t + Z_c \odot (1 - M_t)$
  $p_\theta \leftarrow G_\theta(Z_t)$
  $S \leftarrow$ sample from p_θ
  $Z_c \leftarrow C(S)$ {Using S to select tokens from the codebook C}
  $p_\phi \leftarrow G_\phi(S)$
  $M_{t+1} \leftarrow$ sample$\lceil \gamma(\frac{t}{T}) \cdot N \rceil$ from p_ϕ
end for
$I_{rec} \leftarrow D_H(Z_c)$

book embedding vector.

ii) Each item in Z_h is replaced with the closest code in the codebook $C \in \mathbb{R}^{K \times d}$ with K codebook size to obtain the quantized feature $Z_c \in \mathbb{R}^{m \times n \times d}$ and sequence S_h, which is formulated by Eq. (1):

$$Z_c^{(i,j)} = \mathbf{q}(Z_l) = \arg\min_{c_k \in C} \|Z_h^{(i,j)} - c_k\|_2, \quad S_h^{(i,j)} = k \text{ such that } Z_c^{(i,j)} = c_k. \tag{1}$$

iii) Reconstructing the image I_h by the decoder D_H: $I_{rec} = D_H(Z_c)$.

Analysis. To reduce the ill-posedness of image dehazing, we pre-train a VQGAN to learn a latent discrete high-quality codebook, which improves the network's robustness against various degradations. Due to the domain gap between the hazy images and clean images, the nearest neighbor matching (Eq. (1)) is ineffective in accurately matching codes. This issue will be further discussed in the ablation study (see Fig. 7). Thus, we will introduce Code-Predictor to better match codes.

MaskGIT: An Iterative Generation Paradigm. MaskGIT [5] is an image generation approach that incorporates masked token modeling and parallel decoding, improving both performance and efficiency. Let $Y = [y_i]_{i=1}^N$ denote the latent tokens derived from the input image via an encoder, M represents the corresponding binary mask of Y, and $Y_{\bar{M}}$ represents Y masked by M. During training, MaskGIT trains a predictor that can predict the masked parts through $Y_{\bar{M}}$, aiming to maximize the posterior probability $P(y_i|Y_{\bar{M}})$. During inference, MaskGIT predicts all tokens, keeping only the most confident ones, and re-sampling the rest in the next iteration.

Analysis. In contrast to image generation, image restoration emphasizes the fidelity of the original scene. Thus, it makes sense to use the low-quality features of the image as a condition. Directly applying MaskGIT to image restoration is infeasible as it is designed for LQ-to-HQ mapping rather than HQ-to-HQ mapping; meanwhile, the mechanism for independently sampling tokens ignores their interdependence [22]. To address these issues, we employ Code-Critic at each iteration to identify the relationships between tokens. This approach effectively identifies which tokens to mask at each iteration.

3.2. Code-Predictor Training (Stage I)

In this stage, we will fix codebook C, decoder D_H, and pre-train encoder E_H (also referred to as E_L). When we have haze images I_l, we can obtain corresponding features Z_l with $E_L(I_l)$. We then randomly sample a binary mask matrix $M_t \in \mathbb{R}^{m \times n}$, where the mask ratio can be determined by $\lceil \gamma(r) \cdot (m \times n) \rceil$, where $\gamma(r)$ denotes the cosine function, and r is a random number sampled from a Uniform(0,1]. To obtain the input Z_t, we apply M to Z_l and Z_c, following the formulated by:

$$Z_t = Z_l \odot M_t + Z_c \odot (1 - M_t). \tag{2}$$

Then, Z_t is fed into Code-Predictor G_θ to forecast the probability $p_\theta \in \mathbb{R}^{(m \times n) \times K}$ over all codes in the codebook C. We adopt cross-entropy loss $\mathcal{L}_\theta$ to train G_θ:

$$\mathcal{L}_\theta = -\sum_{i=0}^{N} S_h^{(i)} \log p_\theta(S_h^{(i)}|Z_t), \tag{3}$$

where i is the index, N is the number of elements.

To align the features between hazy images and high-quality images, we introduce the SFT module [35, 42]:

$$\hat{F}_d = F_d + \alpha \odot F_d + \beta; \alpha, \beta = Conv(concat(F_e, F_d)). \tag{4}$$

Table 1. Quantitative comparison using NR-IQA on RTTS, Fattal, and URHI datasets. Red and blue indicate the best, second-best performers respectively. * indicate that due to image size constraints, we are unable to test the results under CLIPIQA and TOPIQ metrics.

Datasets	Metrics	Hazy images	MSBDN [12]	Dehamer [18]	DEA-Net [11]	DAD [31]	D4 [40]	PSD [10]	RIDCP [38]	KA-Net [15]	Ours
RTTS	MUSIQ↑	53.76	53.73	53.57	54.09	49.33	53.55	50.30	55.23	54.64	59.60
	PI↓	4.78	4.15	4.41	3.83	4.19	3.86	3.61	3.56	3.63	3.22
	MANIQA↑	0.311	0.311	0.310	0.314	0.221	0.297	0.256	0.251	0.259	0.327
	CLIPIQA↑	0.39	0.36	0.36	0.37	0.25	0.34	0.28	0.30	0.28	0.44
	Q-Align↑	3.04	3.04	3.10	3.11	2.84	2.97	2.63	3.24	3.09	3.49
	TOPIQ↑	0.400	0.406	0.401	0.407	0.335	0.402	0.353	0.412	0.394	0.500
Fattal	MUSIQ↑	63.61	63.67	64.40	63.33	58.17	63.92	60.96	65.48	64.09	66.22
	PI↓	3.18	2.47	2.48	2.66	3.02	2.44	2.83	2.37	2.81	2.41
	MANIQA↑	0.38	0.38	0.38	0.40	0.26	0.39	0.33	0.31	0.35	0.43
	CLIPIQA↑	0.51	0.53	0.51	0.50	0.42	0.55	0.48	0.42	0.50	0.59
	Q-Align↑	3.739	3.943	3.923	3.934	3.593	3.980	3.312	3.799	3.982	4.234
	TOPIQ↑	0.56	0.56	0.56	0.58	0.41	0.59	0.49	0.50	0.55	0.63
URHI*	MUSIQ↑	57.8	57.35	57.08	57.64	57.12	52.23	53.96	61.39	58.57	62.5
	PI↓	4.07	3.57	3.88	3.83	3.64	3.71	3.48	2.87	3.15	3.08
	MANIQA↑	0.355	0.348	0.350	0.355	0.344	0.262	0.294	0.314	0.306	0.364
	Q-Align↑	3.21	3.19	3.10	3.28	2.92	3.16	2.68	3.51	3.23	3.70

We jointly train low-quality image encoder E_L, Code-Predictor G_θ, and SFT. Further training details will be provided in the supplementary material.

3.3. Code-Critic Training (Stage II)

In training Stage I, we obtain high-quality restored images. However, during the inference stage, we only sample codes S from the output distribution p_θ of the Code-Predictor, without considering the relationships between the codes. To overcome this challenge, in Stage II training, we keep other model components fixed and introduce the Code-Critic G_ϕ to evaluate whether each code should be accepted.

We intuitively feed S into Code-Critic and output p_ϕ to check whether each code in S is consistent with the ones in S_h. If not, it's rejected, otherwise it's accepted. Based on this, we can create a label sequence $M = (S \neq S_h)$. During the training stage II, we use the binary cross-entropy loss to optimize the Code-Critic:

$$\mathcal{L}_\phi = -\sum_{i=0}^{N} M^{(i)} \log p_\phi(S^{(i)}) + (1-M^{(i)}) \log(1-p_\phi(S^{(i)})). \tag{5}$$

Since the Code-Critic is trained to accurately evaluate various cases of Code-Predictor, we introduce the sampling temperature to enhance the sampling diversity of Code-Predictor:

$$p_\theta^{(i)} = e^{\frac{p_\theta^{(i)}}{Temp}} / \sum_{j=0}^{K} e^{\frac{p_\theta^{(i)}}{Temp}}, \tag{6}$$

where $Temp$ is set to 2.

3.4. Iterative Decoding Via Predictor-Critic

The inference stage pipeline is illustrated in Algorithm 2. To start, the low-quality image I_l is encoded by E_L into Z_l, alongside an initial mask M_1. During the inference stage, the following process goes through a total of T iterations. At each iteration, we obtain Z_t using Eq. (2). Then, a code sequence S is sampled from the Code-Predictor G_θ. The accuracy and correlation of S are evaluated using the Code-Critic G_ϕ to determine which code in S should be rejected and resampled in the next iteration, as indicated by the binary mask M_{t+1}. The mask ratio is determined by $\lceil \gamma(r) \cdot N \rceil$. After completing T iterations, we can achieve clear images with $I_{rec} = D_H(Z_c)$.

4. Experiments

4.1. Experimental Settings

Training Datasets. During the codebook learning phase, we follow the Real ESRGAN [36] to generate paired data. We obtain images from DIV2K [1] and Flickr2K [27] and randomly crop and resize them into non-overlapping 512×512 patches. During the training phase (Stage I and Stage II), we generate paired images employing the synthetic data generation methodology proposed by RIDCP [38].

Testing Datasets. We qualitatively and quantitatively evaluate our model on the real-world datasets RTTS and URHI [24], which contain 4,322 and 4,809 images respectively, covering various scenes with different haze densities, resolutions, and degradation levels. Additionally, we further conduct comparisons on Fattal's dataset [14].

4.2. Network Architectures.

In our work, we utilize a VQGAN network similar to FeMaSR [7]. To accommodate images of different resolutions, we employ a $4\times$ RSTB [26] and a $2\times$ RSTB as the body for the Code-Predictor and Code-Critic, followed by a lin-

Figure 3. Visual comparison on RTTS. **Zoom in for best view**.

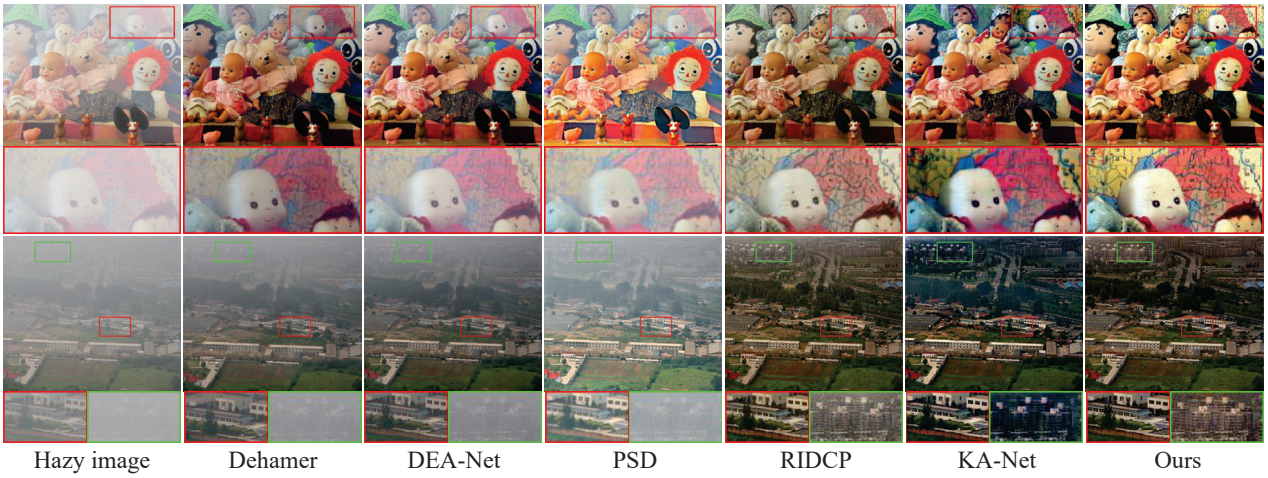


Figure 4. Visual comparison on Fattal. **Zoom in for best view**.

ear projection layer. More details about the network can be found in the supplemental material.

Our method is implemented with PyTorch on 4 NVIDIA RTX 3090 GPUs. We randomly resize and flip the input data, and crop it into 256×256 patches for data augmentation. We use Adam with parameters of $\beta_1 = 0.9$, $\beta_2 = 0.99$, and fix the learning rate to 1×10^{-4} for all stages of training. We pre-train VQGAN with a batch size of 32 and adopt a batch size of 16 for Stage I and Stage II. The networks are respectively trained with 400K, 100K, and 20K iterations during the three training stages.

4.3. Comparisons with State-of-the-art Methods

We compare our method with several state-of-the-art dehazing approaches through both quantitative and qualitative analyses. We present more experimental results and analysis in the supplementary material.

Quantitative Comparison. Due to the difficulty in obtaining ground truth hazy images from the real world, we quantitatively analyzed our method with some commonly used no-reference image quality assessment (IQA) metrics. To better assess the results, we apply IQA metrics focusing on different aspects, such as perceptual IQA (MUSIQ [21], PI [3], and MANIQA [39]), semantic IQA (TOPIQ [8]), and LLMs-based IQA (CLIPIQA [34], Q-Align [37]). All IQA metrics, except for PI, higher metric scores represent better image quality. We compare our method with the methods that have achieved outstanding results in benchmarks: MSDBN [12], Dehamer [18], and DEA-Net [11] as well as real-world image dehazing methods: DAD [31], PSD [10], D4 [40], RIDCP [38], and KA-Net [15].

For a fair comparison, we execute the official code for all the mentioned methods and evaluate them using IQA-Pytorch [6]. As seen from the Tab. 1, our method achieves first and second places in comparison to other state-of-the-art methods. The results suggest that our method performs better in dehazing capabilities, color fidelity, and image quality. Overall, our method obtained the best results

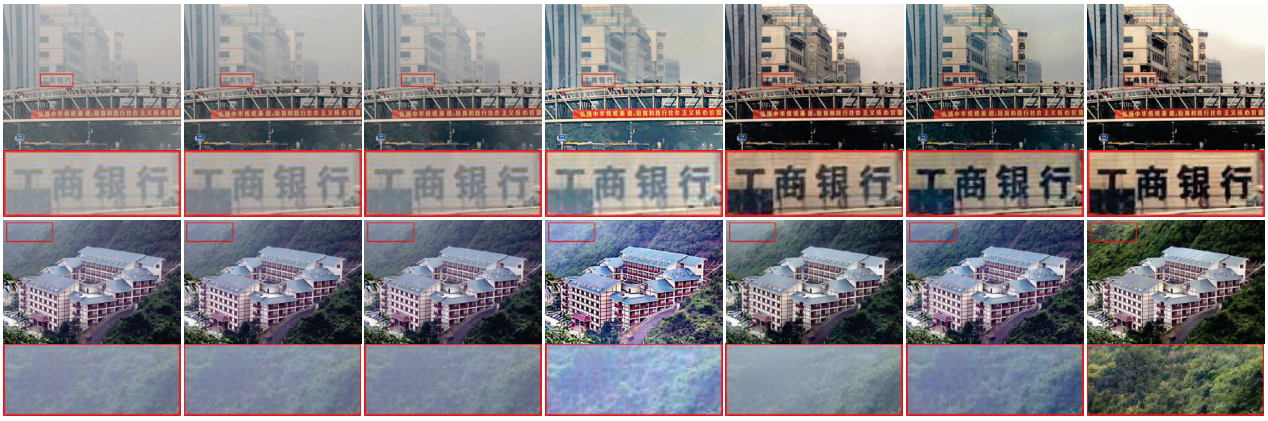
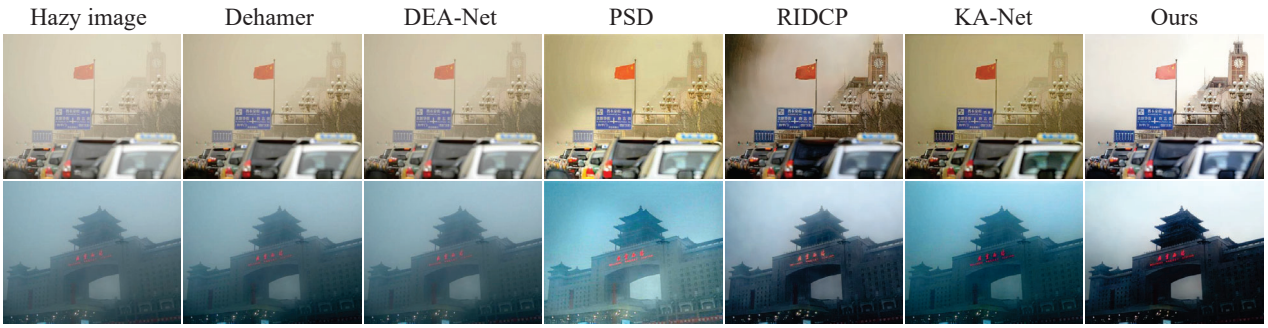
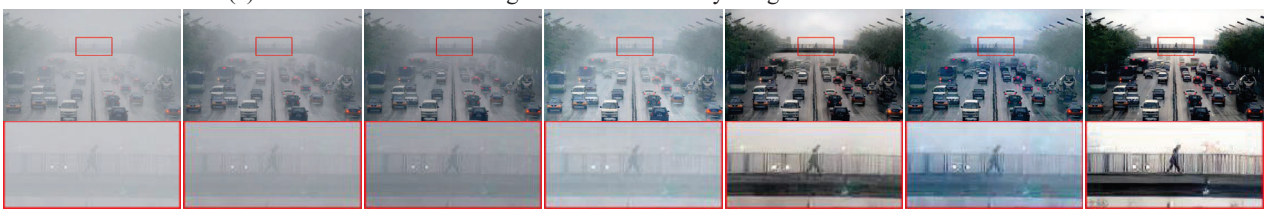


Figure 5. Visual comparison on URHI. **Zoom in for best view**.



(a) Results of SOTA dehazing methods on the hazy images with different color casts.



(b) Results of different SOTA dehazing methods on the dense hazy images.

Figure 6. Visual comparison on challenging scenarios. **Zoom in for best view**.

in quantitative metrics, further supporting its superiority in real-world image dehazing.

Qualitative Comparison. We conduct a qualitative comparison on the RTTS, Fattal, and URHI datasets, as shown in Figs. 3 to 5. Dehamer [18] and DEA-Net [11] show limited dehazing capabilities. PSD [10] is somewhat overexposed and color-shifting. RIDCP [38] and KA-Net [15] demonstrate effective performance, but the image quality is subpar and they struggle with dense haze areas. In comparison, the quality of our restoration significantly outperforms other methods in both thin and dense haze regions (such as within the red and green box in the bottom image in Fig. 4). The comparison reveals that our approach can generate high-quality, more natural, and cleaner images.

To further demonstrate the superiority of our approach, we additionally select some challenging scenarios for vision comparison. As shown in Fig. 6 (a), the captured hazy images may be color-biased due to the effects of low light, and particles such as sand, and dust. Our results can handle the color casts and have more natural colors and cleaner images. In Fig. 6 (b), our method still performs well even in the presence of dense haze with less noise compared to the suboptimal method, RIDCP.

4.4. Ablation Study and Analysis

To validate the significance of the Code-Predictor and Code-Critic, we conducted a series of experiments focused

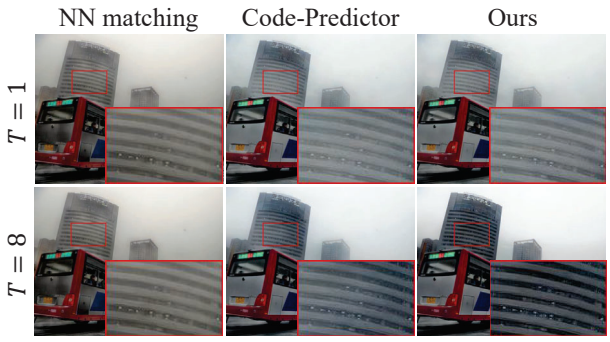


Figure 7. Ablation results of the proposed Code-Predictor. The first row displays the results with no iteration, while the second row shows the results with iteration. The first column shows the results of the NN-based code matching, the second column shows the results of Code-Predictor (without Code-Critic), and the third column shows our results. We observed that Code-Predictor allows for iterations (from $T = 1$ to $T = 8$) to bring about changes, and Code-Critic leads to better results.

Table 2. Quantitative ablation analysis of the Code-Predictor on RTTS when $T = 8$. Red indicates the best results.

Method	MUSIQ↑	PI↓	MANIQA↑	Q-Align↑	TOPIQ↑
NN Matching	58.19	3.25	0.303	3.25	0.458
w/o Code-Critic	57.74	3.32	0.303	3.36	0.462
Ours	59.60	3.22	0.327	3.49	0.500

on: (1) examining the effectiveness of iterative dehazing with Code-Predictor, and (2) investigating the enhancement of iterative dehazing through the integration of Code-Critic with Code-Predictor.

Effectiveness of Code-Predictor. First, we compare the Code-Predictor with the nearest neighbor (NN) matching method to validate the effectiveness of this module in our network. To eliminate interference, we did not introduce the Code-Critic in this ablation experiment. In order to apply iteration in the model based on nearest neighbor matching, we use the distance from token mapping to the codebook as the criterion for whether the generated code will be retained in the next iteration. As shown in Fig. 7, when using NN matching, increasing the number of iterations does not lead to better results (see the first column). It is obvious that during the training phase, what we learn is independent matching for each token without considering the relationships between tokens, so iteration does not change the code. However, Code-Predictor takes Z_l as a condition, which uses the high-quality code obtained from the previous iteration to guide the next iteration. Compared to the NN matching method, our Code-Predictor can select more likely codes during the iteration process, which is crucial for the iterative decoding prediction. Tab. 2 also verifies this conclusion from the point of view of quantitative analysis.

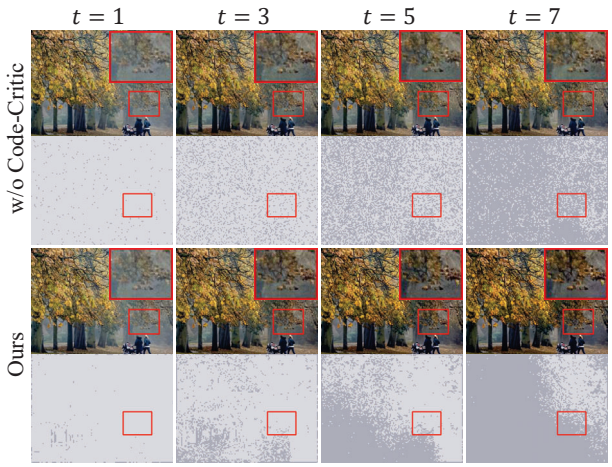


Figure 8. Ablation results of the proposed Code-Critic. The images from left to right display the results at different t When $T = 8$. The top image in each pair represents the results from sequence S, while the bottom image represents the mask map after S has been evaluated by Code-Critic. Without Code-Critic, the mask map tends to be random. Our method's mask map can first retain the code in thin haze areas, resulting in better outcomes.

However, without the Code-Critic, Code-Predictor cannot effectively determine which codes to keep in each iteration, resulting in limited improvements. We further discuss the role of Code-Critic below.

Effectiveness of Code-Critic. To further discuss the necessity of Code-Critic, we compare the sampling method based on code confidence and that based on Code-Critic. As shown in Algorithm 2, the results are shown in Figs. 7 and 8. It is evident that with the inclusion of the Code-Critic, the process of code selection exhibits a trend from close to distant, from simple to challenging. Moreover, significant improvements in image optimization are observed during iterations. Such a design leads to much cleaner and sharper results in comparison to the absence of the Code-Critic. Additionally, Tab. 2 further demonstrates the findings.

5. Conclusion

Inspired by the natural phenomenon that the degradation of haze images is closely related to the density of haze and scene depth, we propose a new method for real-world iterative dehazing. Our iterative process is different from the one-shot-based method. The one-shot-based method cannot effectively handle thin haze areas. In contrast, our process can first identify faint haze and restore regions with low difficulty. Then, it uses these restored parts as guidance to predict haze density and restore regions with higher difficulty. Extensive experiments demonstrate the superiority of our approach.

Acknowledgements. This work was supported in part by the National Natural Science Foundation of China (62306153, U23B2011, 62176130), the Fundamental Research Funds for the Central Universities (Nankai University, 070-63243143), the Natural Science Foundation of Tianjin, China (24JCJQJC00020), the Shenzhen Science and Technology Program (JCYJ20240813114237048) and the Key R&D Program of Zhejiang (2024SSYS0091). The computational devices of this work are supported by the Supercomputing Center of Nankai University (NKSC).

References

[1] Eirikur Agustsson and Radu Timofte. Ntire 2017 challenge on single image super-resolution: Dataset and study. In *CVPRW*, 2017. 5

[2] Dana Berman, Tali Treibitz, and Shai Avidan. Non-local image dehazing. In *CVPR*, 2016. 2

[3] Yochai Blau, Roey Mechrez, Radu Timofte, Tomer Michaeli, and Lihi Zelnik-Manor. The 2018 pirm challenge on perceptual image super-resolution. In *ECCVW*, pages 0–0, 2018. 6

[4] Bolun Cai, Xiangmin Xu, Kui Jia, Chunmei Qing, and Dacheng Tao. Dehazenet: An end-to-end system for single image haze removal. *IEEE TIP*, page 5187–5198, 2016. 2

[5] Huiwen Chang, Han Zhang, Lu Jiang, Ce Liu, and William T. Freeman. Maskgit: Masked generative image transformer. In *CVPR*, pages 11315–11325, 2022. 4

[6] Chaofeng Chen and Jiadi Mo. IQA-PyTorch: Pytorch toolbox for image quality assessment. [Online]. Available: https://github.com/chaofengc/IQA-PyTorch, 2022. 6

[7] Chaofeng Chen, Xinyu Shi, Yipeng Qin, Xiaoming Li, Xiaoguang Han, Tao Yang, and Shihui Guo. Real-world blind super-resolution via feature matching with implicit high-resolution priors. In *ACM MM*, pages 1329–1338, 2022. 3, 5

[8] Chaofeng Chen, Jiadi Mo, Jingwen Hou, Haoning Wu, Liang Liao, Wenxiu Sun, Qiong Yan, and Weisi Lin. Topiq: A top-down approach from semantics to distortions for image quality assessment. *IEEE TIP*, 33:2404–2418, 2024. 6

[9] Chaofeng Chen, Shangchen Zhou, Liang Liao, Haoning Wu, Wenxiu Sun, Qiong Yan, and Weisi Lin. Iterative token evaluation and refinement for real-world super-resolution. In *AAAI*, 2024. 3

[10] Zeyuan Chen, Yangchao Wang, Yang Yang, and Dong Liu. Psd: Principled synthetic-to-real dehazing guided by physical priors. In *CVPR*, pages 7180–7189, 2021. 2, 3, 5, 6, 7

[11] Zixuan Chen, Zewei He, and Zhe-Ming Lu. Dea-net: Single image dehazing based on detail-enhanced convolution and content-guided attention. *IEEE TIP*, 33:1002–1015, 2024. 5, 6, 7

[12] Hang Dong, Jinshan Pan, Lei Xiang, Zhe Hu, Xinyi Zhang, Fei Wang, and Ming-Hsuan Yang. Multi-scale boosted dehazing network with dense feature fusion. In *CVPR*, 2020. 2, 5, 6

[13] Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *CVPR*, 2021. 2, 3

[14] Raanan Fattal. Dehazing using color-lines. *ACM TOG*, page 1–14, 2014. 2, 5

[15] Yuxin Feng, Long Ma, Xiaozhe Meng, Fan Zhou, Risheng Liu, and Zhuo Su. Advancing real-world image dehazing: Perspective, modules, and training. *IEEE TPAMI*, 46(12): 9303–9320, 2024. 1, 5, 6, 7

[16] Alona Golts, Daniel Freedman, and Michael Elad. Unsupervised single image dehazing using dark channel prior loss. *IEEE TIP*, page 2692–2701, 2020. 2

[17] Yuchao Gu, Xintao Wang, Liangbin Xie, Chao Dong, Gen Li, Ying Shan, and Ming-Ming Cheng. Vqfr: Blind face restoration with vector-quantized dictionary and parallel decoder. In *ECCV*, pages 126–143. Springer, 2022. 3

[18] Chun-Le Guo, Qixin Yan, Saeed Anwar, Runmin Cong, Wenqi Ren, and Chongyi Li. Image dehazing transformer with transmission-aware 3d position embedding. In *CVPR*, pages 5812–5820, 2022. 2, 5, 6, 7

[19] Kaiming He, Jian Sun, and Xiaoou Tang. Single image haze removal using dark channel prior. *IEEE TPAMI*, 33(12): 2341–2353, 2010. 2

[20] R Hide. Optics of the atmosphere: Scattering by molecules and particles. *Physics Bulletin*, page 521–521, 1977. 2

[21] Junjie Ke, Qifei Wang, Yilin Wang, Peyman Milanfar, and Feng Yang. Musiq: Multi-scale image quality transformer. In *ICCV*, pages 5148–5157, 2021. 6

[22] José Lezama, Huiwen Chang, Lu Jiang, and Irfan Essa. Improved masked image generation with token-critic. In *ECCV*, pages 70–86, 2022. 4

[23] Boyi Li, Xiulian Peng, Zhangyang Wang, Jizheng Xu, and Dan Feng. Aod-net: All-in-one dehazing network. In *ICCV*, 2017. 2

[24] Boyi Li, Wenqi Ren, Dengpan Fu, Dacheng Tao, Dan Feng, Wenjun Zeng, and Zhangyang Wang. Benchmarking single-image dehazing and beyond. *IEEE TIP*, page 492–505, 2019. 5

[25] Lerenhan Li, Yunlong Dong, Wenqi Ren, Jinshan Pan, Changxin Gao, Nong Sang, and Ming-Hsuan Yang. Semi-supervised image dehazing. *IEEE TIP*, page 2766–2779, 2020. 2, 3

[26] Jingyun Liang, Jiezhang Cao, Guolei Sun, Kai Zhang, Luc Van Gool, and Radu Timofte. Swinir: Image restoration using swin transformer. In *ECCV*, pages 1833–1844, 2021. 5

[27] Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee. Enhanced deep residual networks for single image super-resolution. In *CVPRW*, 2017. 5

[28] Wei Liu, Xianxu Hou, Jiang Duan, and Guoping Qiu. End-to-end single image fog removal using enhanced cycle consistent adversarial networks. *IEEE TIP*, page 7819–7833, 2020. 2

[29] Xu Qin, Zhilin Wang, Yuanchao Bai, Xiaodong Xie, and Huizhu Jia. Ffa-net: Feature fusion attention network for single image dehazing. *AAAI*, page 11908–11915, 2020. 2

[30] Wenqi Ren, Si Liu, Hua Zhang, Jinshan Pan, Xiaochun Cao, and Ming-Hsuan Yang. Single image dehazing via

multi-scale convolutional neural networks. In *ECCV*, page 154–169, 2016. 2

[31] Yuanjie Shao, Lerenhan Li, Wenqi Ren, Changxin Gao, and Nong Sang. Domain adaptation for image dehazing. In *CVPR*, 2020. 2, 5, 6

[32] Pranjay Shyam, Kuk-Jin Yoon, and Kyung-Soo Kim. Towards domain invariant single image dehazing. *CVPR*, 2021. 2

[33] Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *NeurIPS*, 30, 2017. 3

[34] Jianyi Wang, Kelvin CK Chan, and Chen Change Loy. Exploring clip for assessing the look and feel of images. In *AAAI*, 2023. 6

[35] Xintao Wang, Ke Yu, Chao Dong, and Chen Change Loy. Recovering realistic texture in image super-resolution by deep spatial feature transform. In *CVPR*, 2018. 4

[36] Xintao Wang, Liangbin Xie, Chao Dong, and Ying Shan. Real-esrgan: Training real-world blind super-resolution with pure synthetic data. In *ICCVW*, 2021. 5

[37] Haoning Wu, Zicheng Zhang, Weixia Zhang, Chaofeng Chen, Liang Liao, Chunyi Li, Yixuan Gao, Annan Wang, Erli Zhang, Wenxiu Sun, et al. Q-align: teaching lmms for visual scoring via discrete text-defined levels. In *Proceedings of the 41st International Conference on Machine Learning*, pages 54015–54029, 2024. 6

[38] Rui-Qi Wu, Zheng-Peng Duan, Chun-Le Guo, Zhi Chai, and Chongyi Li. Ridcp: Revitalizing real image dehazing via high-quality codebook priors. In *CVPR*, pages 22282–22291, 2023. 1, 2, 3, 5, 6, 7

[39] Sidi Yang, Tianhe Wu, Shuwei Shi, Shanshan Lao, Yuan Gong, Mingdeng Cao, Jiahao Wang, and Yujiu Yang. Maniqa: Multi-dimension attention network for no-reference image quality assessment. In *CVPR*, pages 1191–1200, 2022. 6

[40] Yang Yang, Chaoyue Wang, Risheng Liu, Lin Zhang, Xiaojie Guo, and Dacheng Tao. Self-augmented unpaired image dehazing via density and depth decomposition. In *CVPR*, pages 2037–2046, 2022. 2, 5, 6

[41] Shiyu Zhao, Lin Zhang, Ying Shen, and Yicong Zhou. Refinednet: A weakly supervised refinement framework for single image dehazing. *IEEE TIP*, page 3391–3404, 2021. 2

[42] Shangchen Zhou, Kelvin Chan, Chongyi Li, and Chen Change Loy. Towards robust blind face restoration with codebook lookup transformer. *NeurIPS*, 35: 30599–30611, 2022. 3, 4

[43] Qingsong Zhu, Jiaming Mai, and Ling Shao. Single image dehazing using color attenuation prior. In *BMVC*, 2014. 2