基于预测器-评价器迭代解码的真实世界图像去雾方法

傅佳怡[1] 刘思语[1] 刘子坤[3] 郭春乐[1,2] Hyunhee Park[4] 武睿祺[1] 王国庆[5] 李重仪[1,2*]

[1] VCIP，CS，南开大学 [2] 南开国际深圳研究院，深圳福田

[3] 三星电子中国研究院-北京 [4]CIG，三星电子 [5] 东海实验室，舟山，浙江

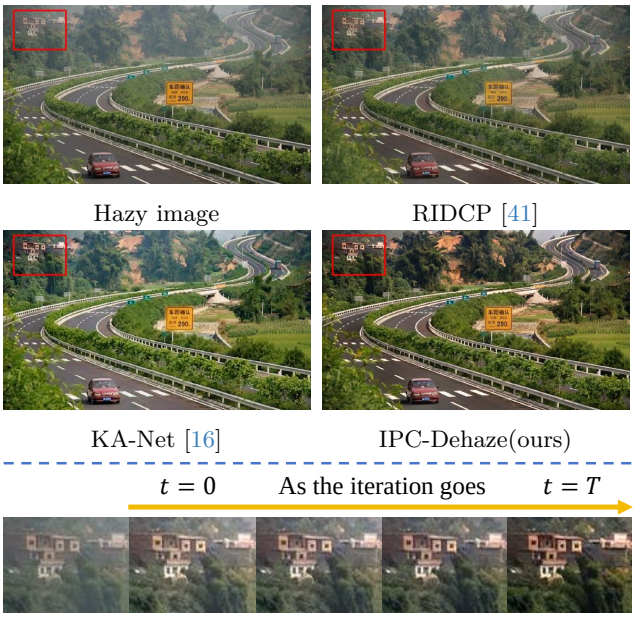
{fujiayi,liusiyu29,wuruiqi}@mail.nankai.edu.cn, {zikun.liu,inextg.park}@samsung.com,

{guochunle,lichongyi}@nankai.edu.cn, gqwang0420@hotmail.com

[代码] [网站]

Abstract

我们提出了一种基于预测器-评价器迭代解码（***I**terative **P**redictor-**C**ritic Code Decoding*）的真实世界图像去雾框架，简称 ***IPC-Dehaze***，该方法利用了预训练 *VQGAN* 中封装的高质量 *codebook* 先验。与以往基于 *codebook* 的方法依赖单次解码不同，我们的方法通过迭代利用前一次生成的高质量 *code* 来引导后续迭代中的 *Code-Preidictor,*，从而提升 *code* 预测精度并保证去雾效果的稳定性。我们的核心思想基于以下两个发现：*1)* 雾天图像的退化程度受雾密度和场景深度的影响；*2)* 清晰区域为恢复浓雾区域提供了关键信息。然而，由于难以判断每次迭代中哪些 *code* 需要保留或替换，在后续迭代中想要逐步优化它们并不容易。本研究的另一关键创新是提出 *Code-Critic* 模块来捕捉 *code* 之间的关联性。*Code-Critic* 能够用于评估 *code* 的相关性，然后重新采样一组有最高掩码分数的 *code*，即分数越高表示该 *code* 被剔除的可能性越大，从而保留更准确的 *code* 并预测难以获取的 *code*。大量实验表明，本方法在真实世界去雾任务中优于现有方法。



IPC-Dehaze 的迭代结果

图 1. 现有最先进的真实世界图像去雾方法与我们的 IPC-Dehaze 对比。相较之下，我们的结果更为清晰锐利，且色彩失真和过度曝光现象更少。底部图像展示了本方法每次迭代的结果，验证了我们方法的核心：预测器-评价器机制能够持续提升去雾性能。

1. 引言

在真实世界场景中，拍摄到含雾的图像是很常见的，其视觉效果表现为低清晰度和低锐度。这一现象主要源自大气颗粒不规则地进行光的色散和散射。雾的存在会严重影响图像的视觉质量或后续感知算法的准确性。因此，对于图像去雾技术的关注日益增长，旨在从带雾的原始图像中恢复出清晰的图像。

从含雾图像中恢复清晰图像本质上极具挑战性。传统方法通过利用多种先验信息来解决这一不适定问题 [2, 15, 20, 46]，但由于理想化的假设或泛化能力的

*通讯作者

限制，它们在真实世界场景中表现欠佳。深度学习的出现推动数据驱动的去雾方法表现出卓越的性能。这些方法或者估计物理模型相关的透射图 [4, 32]，或者直接恢复出清晰的图像 [12, 31]。然而，这类方法通常基于理想物理模型合成的图像进行训练，导致其在复杂真实世界场景中的应用面临挑战。

为解决这一问题，部分研究关注于改进数据合成方式 [33, 43] 或采用无监督学习等替代训练策略 [17, 44]。最近的工作，RIDCP [41] 强调了在合成含雾图像时考虑多重退化因素的重要性，并验证了高质量 codebook 先验的有效性。沿袭 RIDCP 的方向，一些工作 [10, 27] 已经给出了针对真实世界去雾的解决方案。尽管现有方法取得了一定的进展，但仍存在着一些局限性。具体来说，它们可能忽略退化程度随雾密度和场景深度的空间变化，导致在薄雾区域过度恢复或在浓雾区域恢复不足。此外，仅通过单次操作（即仅尝试一次去雾）来准确恢复具有挑战性场景的图像会比较困难，从而导致性能欠佳。

受物理现象的启发，我们知道薄雾区域包含更多的信息而浓雾区域的信息较少，我们发现去雾应优先关注相对清晰的区域，再逐步处理更有挑战性的区域。为此，我们沿用先前方法 [41] 中预训练的 VQGAN[14] 所封装的高质量 codebook 作为先验。与以往基于 codebook 的单次预测方法不同，我们提出了一种新型迭代解码框架，通过由易到难的方式逐步优化去雾效果，显著提升了去雾的准确性和稳定性。如 Fig. 1所示。

在我们的方法中，首先通过编码器将含雾图像映射为 token。然后根据当前的 token，采用预测器-评价器机制交替进行高质量 code 的预测，并评估出哪些 code 应该被保留。随着迭代的进行，所保留的 code 数量也在逐步增加，直到最终确定所有 code。举例说明，在第 t 次迭代中，我们获取了前一次迭代所预测的高质量 code 以及决定本次迭代需保留的 code 的掩码图。然后，我们将经过掩码图处理后的 token 输入 Code-Preidictor，预测本次迭代的所有高质量 code，并同样生成用于下一阶段的掩码图。为实现上述迭代去雾过程，我们通过随机掩码融合清晰与含雾图像的 token，处理后的 token 输入 Code-Preidictor 以匹配高质量 codebook，用来模拟推理阶段第 t 次迭代的预测过程。推理过程中，生成有效且有指导意义的掩码图至关重要，因此我们提出 Code-Critic 以更精准地剔除最不可靠的 code 并保留优质 code。

我们**研究的贡献**总结如下：

- 我们提出了一种新型迭代解码去雾框架。与单次解码方法相比，我们的方法利用先前迭代获得的高质量 code 作为线索，引导 Code-Preidictor 去预测后续的 code，从而实现更优的迭代去雾。
- 我们引入了 Code-Critic 来评估 Code-Preidictor 输出的 code 的相关性，在迭代解码步骤中选择应保留或剔除的 code 以引导后续预测。该模块提升了所选 code 一致性并防止误差累积。
- 通过这些新设计，我们的方法在真实世界去雾任务中，在定性及定量评估上均超越了现有最先进方法。

2. 相关工作

早期的单幅图像去雾方法 [2, 15, 20, 46] 通过估计成像模型的参数来恢复图像 [21]。然而，手工设计的先验知识泛化能力较差，难以适应多样化的场景。随着深度学习的发展，数据驱动的方法取得了显著的成果。其中一类方法 [4, 25, 32] 基于散射模型设计，但当场景不符合理想物理模型时，其性能不尽如人意。另一类方法 [19, 31] 则直接利用网络生成清晰的图像。但由于合成数据与真实数据间域的分布差异，基于合成图像训练的模型在真实世界含雾图像上的表现往往欠佳。

近年来，针对真实世界图像去雾已经开展了大量的研究。现有方法可大致分为三类：1) 基于域适应的方法。部分研究采用域适应技术缩小合成域与真实域之间的差距 [33, 34]。这些监督学习方法在合成数据集上表现优异，但难以泛化到真实世界含雾图像。2) 基于 GAN 的方法。相比之下，基于 GAN 的无监督去雾方法是从非配对的清晰与含雾图像中学习去雾映射。为生成更清晰逼真的图像，Liu 等人 [30] 提出了一个基于物理模型的去雾网络，以及一个能够监督含雾域到清晰域的映射过程的增强网络。Yang 等人 [43] 引入了利用自增强技术对含不同雾密度的模糊图像进行再渲染的无配对去雾框架。然而，基于 GAN 的方法往往容易生成不真实的雾，进一步降低了整体去雾性能。3) 基于先验的方法。部分方法 [10, 27] 在网络框架或损失函数中引入手工先验，但仍无法克服其固有局限。近期一些研究尝试从高质量图像中提取先验知识来辅助图像复原。一些基于 VQGAN[14] 的人脸修复 [18, 45] 与超分辨率 [7, 9] 方法已证明高质量先验的优越性。RIDCP [41]

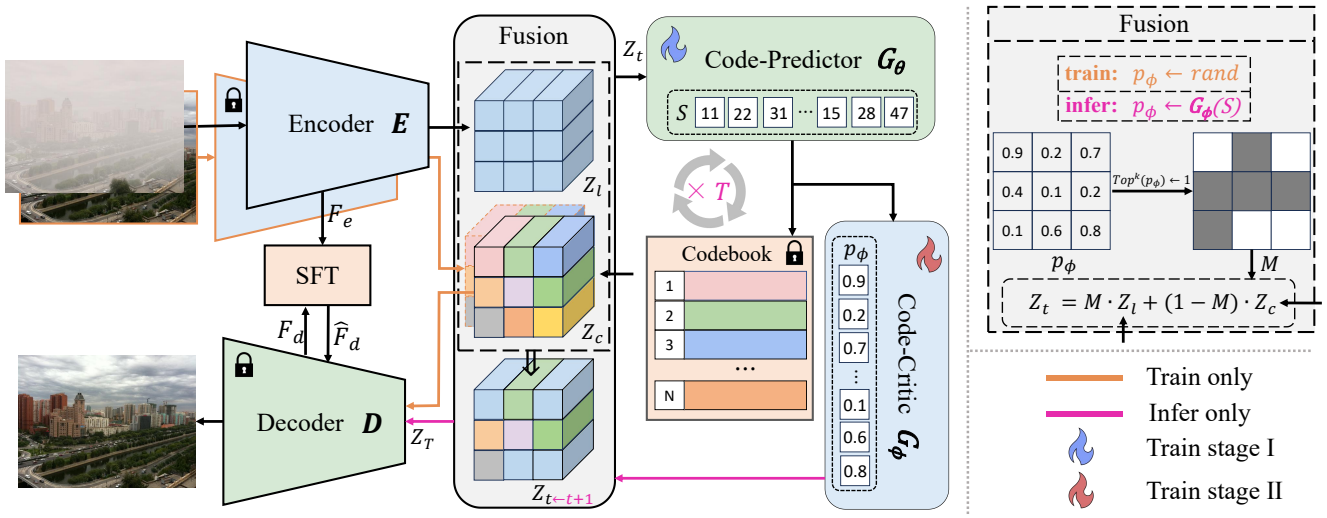


图 2. IPC-Dehaze 框架概述。在训练阶段，我们使用含雾图像与清晰图像的融合 token $Z_t = Z_l \odot M + Z_c \odot (1-M)$，通过 Code-Preidictor 预测序列 code S。同时，我们还训练了 Code-Critic 对集合 S 中的每个 code 进行评估，以决定是否需要剔除与重采样。在推理阶段，初始时刻 $Z_{t=0}$ 被编码为低质量 token Z_l。在第 t 次迭代解码步骤中，Code-Preidictor 以 Z_t 为输入，预测序列 code S 及对应的高质量 token Z_c。为保留可靠 code 并重采样其余 code，Code-Critic 评估序列 code S 并通过 p_ϕ 生成掩码图 M。该掩码图 M 随后通过 *Fusion* 过程用于生成 Z_{t+1}。经过 T 次迭代后，输出 Z_T 并通过解码器重建清晰图像。SFT 指代空间特征变换，用于对齐编码器与解码器内部特征。

在图像去雾中利用了潜在高质量先验并取得显著效果，但其基于单次解码的算法对极浓雾或非均匀雾处理不佳，未能充分利用薄雾区域的信息。针对这些问题，我们的新框架引入了生成能力与高质量先验知识，并通过迭代方法增强了模型的泛化能力。

3. 方法

本文提出了一种基于迭代解码的新型图像去雾框架，其架构如 Fig. 2所示。首先，为引入鲁棒的高质量先验，我们基于高质量数据集预训练了一个 VQGAN [14]。在此阶段，我们可以获得一个潜在的高质量离散 codebook 及对应的编码器与解码器。为匹配正确的 code，我们提出用 Code-Preidictor 替代传统的最近邻匹配方法（阶段 I）。为捕捉 code 序列的全局关联性，我们引入 Code-Critic 以在迭代解码过程中决定是否接受选 Code-Preidictor 所生成的 code（阶段 II），这样确保 code 接受或剔除决策并非由 Code-Preidictor 独立确定。训练流程详见 Algorithm 1。

在推理阶段中，以含雾图像编码得到的 token 作为初始输入。经过 T 次迭代，当前 token 被输入 Code-Preidictor，Code-Critic 则选择哪些高质量 code 将被保留至下一迭代。此过程持续至所有 code 均被高质量 code 所替换。推理流程详见 Algorithm 2。

3.1. 预备知识

通过 VQGAN 学习一个 codebook。为减轻直接在像素空间中生成图像的复杂性，VQVAE[36] 提出通过矢量量化（VQ）自编码器在隐空间学习离散 codebook。VQGAN[14] 通过引入对抗损失与感知损失，进一步提升重建结果的感知质量。遵循 VQGAN，codebook 的学习由三部分组成：

i) 编码器 E_H 将高质量图像块 $I_h \in \mathbb{R}^{H\times W\times 3}$ 编码为隐特征 $Z_h = E_H(I_h) \in \mathbb{R}^{m\times n\times d}$，其中 d 表示 codebook 嵌入向量的维度。

ii) 将 Z_h 中的每个元素替换为 codebook $C \in \mathbb{R}^{K\times d}$ 中最接近的 code，其中 code 数量为 K，从而获得量化特征 $Z_c \in \mathbb{R}^{m\times n\times d}$ 和序列 S_h，该过程由 Eq. (1) 表示：

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

iii) 通过解码器 D_H 重建图像 I_h: $I_{rec} = D_H(Z_c)$.

分析。为降低图像去雾问题的不适定性，我们预

Algorithm 1 训练过程

输入：Z_l，Z_c，S_h，Code-Preidictor G_θ，Code-Critic G_ϕ，掩码调度函数 $\gamma(r)$，学习率 ηN，潜在特征中的 token 数目。

repeat
  $r \sim \mathcal{U}_{(0,1)}$
  $M_t \leftarrow$ 随机采样($\lceil \gamma(r) \cdot N \rceil$)
  $Z_t \leftarrow Z_l \odot M_t + Z_c \odot (1 - M_t)$ {基于 M_t 混淆 Z_l 和 Z_c}
  $p_\theta \leftarrow G_\theta(Z_t)$
  $S \leftarrow \text{argmax}(p_\theta)$
  $\theta \leftarrow \theta - \eta \nabla_\theta \mathcal{L}_\theta$
  if training Code-Critic **then**
    $S \leftarrow$ sample from p_θ
    $M \leftarrow (S \neq S_h)$ {构造 GT}
    $\phi \leftarrow \phi - \eta \nabla_\phi \mathcal{L}_\phi$
  end if
until convergence

Algorithm 2 推理过程

输入：编码器 E_L，解码器 D_H，Code-Preidictor G_θ，Code-Critic G_ϕ，掩码调度函数 $\gamma(r)$，codebook C，潜在特征中的 token 数目 N，迭代次数 T，低质量图像 I_l

输出: 清晰图像 I_{rec}

$Z_l \leftarrow E_L(I_l)$
$M_1 \leftarrow 1$ {M_1 初始化为 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
  $Z_c \leftarrow C(S)$ {用 S 从 codebook C 中索引 token}
  $p_\phi \leftarrow G_\phi(S)$
  $M_{t+1} \leftarrow \text{sample}\lceil \gamma(\frac{t}{T}) \cdot N \rceil$ from p_ϕ
end for
$I_{rec} \leftarrow D_H(Z_c)$

训练了一个 VQGAN 以学习潜在的离散高质量 codebook，从而提升网络对各类退化的鲁棒性。由于含雾图像与清晰图像之间存在域差异，最近邻匹配（Eq. (1)）在准确匹配 code 时效果不佳。这一问题将在消融实验中进一步讨论（见 Fig. 7）。因此，我们引入了 Code-Preidictor 以实现更优的 code 匹配。

MaskGIT：一种迭代生成范式. MaskGIT [5] 是一种结合掩码 token 建模与并行解码的图像生成方法，在提升性能的同时也提高了效率。设 $Y = [y_i]_{i=1}^N$ 表示通过编码器从输入图像中提取的潜在 token，M 表示 Y 对应的二值掩码，$Y_{\bar{M}}$ 表示由掩码 M 处理过后的 Y。在训练阶段，MaskGIT 训练一个预测器，可通过 $Y_{\bar{M}}$ 预测被掩码的部分，目标是最大化后验概率 $P(y_i|Y_{\bar{M}})$。在推理阶段，MaskGIT 预测所有 token，仅保留置信度最高的 token，并在下一次迭代中对其余 token 进行重采样。

分析。与图像生成不同，图像复原更注重原始场景的保真度。因此，利用图像的低质量特征作为条件是有意义的。直接将 MaskGIT 应用于图像复原是不可行的，因为其设计初衷是低质量到高质量（LQ-to-HQ）映射，而非高质量到高质量（HQ-to-HQ）映射；同时，独立采样 token 的机制忽略了 token 之间的相关性 [24]。为解决这些问题，我们在每次迭代中引入 Code-Critic 以识别 token 之间的关系。该方法能够有效确定每次迭代中需要掩码的 token。

3.2. Code-Preidictor 的训练 (阶段 I)

在这个阶段，我们将固定 codebook C、解码器 D_H 和预训练编码器 E_H（也称为 E_L）。当我们有含雾图像 I_l 时，我们可以通过 $E_L(I_l)$ 获得对应的特征 Z_l。接着，我们随机采样一个二值掩码矩阵 $M_t \in \mathbb{R}^{m \times n}$，其中掩码比例由 $\lceil \gamma(r) \cdot (m \times n) \rceil$ 决定，同时 $\gamma(r)$ 表示余弦函数，r 是从区间 (0,1] 上的均匀分布中采样的随机数。为获得输入 Z_t，我们按照以下公式将 M 应用于 Z_l 和 Z_c：

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

然后，将 Z_t 输入 Code-Preidictor G_θ 来预测每一个 code 在 codebook C 中的概率 $p_\theta \in \mathbb{R}^{(m \times n) \times K}$。我们使用交叉熵损失 $\mathcal{L}_\theta$ 来训练 G_θ：

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

其中 i 是索引，N 是元素的数量。

表 1. 在 RTTS、Fattal 和 URHI 数据集上使用 NR-IQA 进行定量比较。红色和蓝色分别表示最佳和次佳表现。* 表示由于图像尺寸限制，我们无法在 CLIPIQA 和 TOPIQ 指标下测试结果。

数据集	指标	带雾图像	MSBDN [12]	Dehamer [19]	DEA-Net [11]	DAD [33]	D4 [43]	PSD [10]	RIDCP [41]	KA-Net [16]	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

为了对齐加雾图像和去雾图像间的特征，我们引入了 SFT 模块 [38, 45]：

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

我们联合训练低质量图像编码器 E_L，Code-Preidictor G_θ 和 SFT 模块。更多训练细节在补充材料中提供。

3.3. Code-Critic 的训练 (阶段 II)

在训练阶段 I 中，我们获得了高质量的复原图像。然而，在推理阶段，我们仅从 Code-Preidictor 的输出分布 p_θ 中采样 code S，而并未考虑 code 之间的联系。为解决这一问题，在阶段 II 的训练中，我们固定其他模型组件，引入 Code-Critic G_ϕ 以评估每个 code 是否应被接受。

我们直观地将 S 输入 Code-Critic 中，并输出 p_ϕ 以检查 S 中的每个 code 是否与 S_h 中的 code 一致。如果不一致，则该 code 被剔除；否则接受该 code。基于此，我们可以建立一个标签序列 $M = (S \neq S_h)$。在训练阶段 II 中，我们使用二元交叉熵损失来优化 Code-Critic：

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

由于 Code-Critic 是为了准确评估 Code-Preidictor 的各种情况而训练的，我们引入了采样温度来提高 Code-Preidictor 的采样多样性：

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

其中 $Temp$ 设置为 2。

3.4. 通过预测器-评价器进行迭代解码

推理阶段的流程如 Algorithm 2所示。首先，低质量图像 I_l 通过 E_L 编码为 Z_l，并生成初始掩码 M_1。在推理阶段，整个过程共进行 T 次迭代。在每次迭代中，我们根据 Eq. (2) 获得 Z_t。然后，从 Code-Preidictor G_θ 中采样一个 code 序列 S。使用 Code-Critic G_ϕ 评估 S 的准确性和相关性，以确定 S 中哪些 code 应在下一次迭代中被剔除并重新采样，这一过程由二值掩码 M_{t+1} 决定。掩码比例由 $\lceil \gamma(r) \cdot N \rceil$ 确定。完成 T 次迭代后，我们可以通过 $I_{rec} = D_H(Z_c)$ 重建清晰图像。

图 3. 与 RTTS 的视觉效果对比。**放大以便观看**。

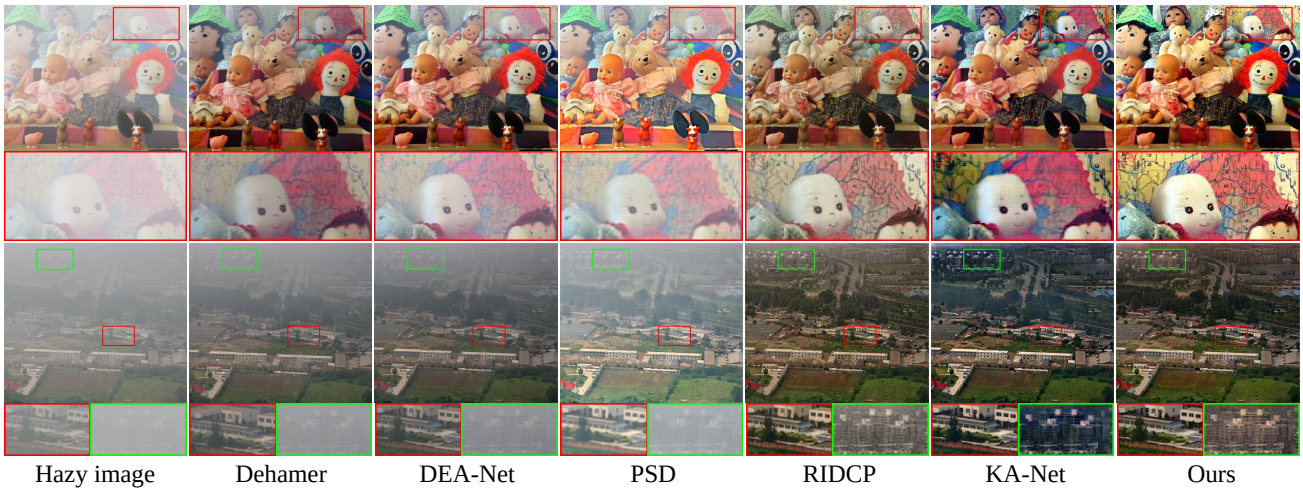


图 4. 与 Fattal 的视觉效果对比。**放大以便观看**。

4. 实验

4.1. 实验设置

生成训练集。在 codebook 学习阶段，我们沿用 Real ESRGAN [39] 的方法来生成成对的数据。我们从 DIV2K [1] 和 Flickr2K [29] 中产生数据，并将其随机裁剪为不重叠的 512×512 图像块。在训练阶段 (阶段 I 和阶段 II)，我们采用 RIDCP [41] 提出的合成数据生成方式来产生配对图像。

生成测试集。我们在真实世界数据集 RTTS 和 URHI [26] 上对我们的模型进行定性和定量评估，这两个数据集分别包含 4,322 和 4,809 张图像，涵盖了不同雾密度、分辨率和退化程度的多种场景。此外，我们还在 Fattal 的数据集上 [15] 进行了进一步的对比实验。

4.2. 网络架构。

在我们的工作中，我们采用了一个类似于 FeMaSR [7] 的 VQGAN 结构。为适应不同分辨率的图像，我们采用了 4×RSTB [28] 和 2×RSTB 分别作为 Code-Preidictor 和 Code-Critic 的主干，并在其后添加一个线性投影层。更多关于网络的细节参见补充材料。

我们的方法基于 PyTorch 框架，在 4 块 NVIDIA RTX 3090 GPU 上实现。我们对输入数据进行随机缩放和翻转，并裁剪为 256×256 大小的图像块以进行数据增强。我们在全部训练阶段中采用 Adam 优化器，参数设置为 $\beta_1 = 0.9$，$\beta_2 = 0.99$，并将学习率固定为 1×10^{-4}。我们用 32 的批量大小预训练 VQGAN，在阶段 I 和阶段 II 中将批量大小设置为 16 进行训练。在三个训练阶段中，网络分别经过 40 万，10 万和 2 万次

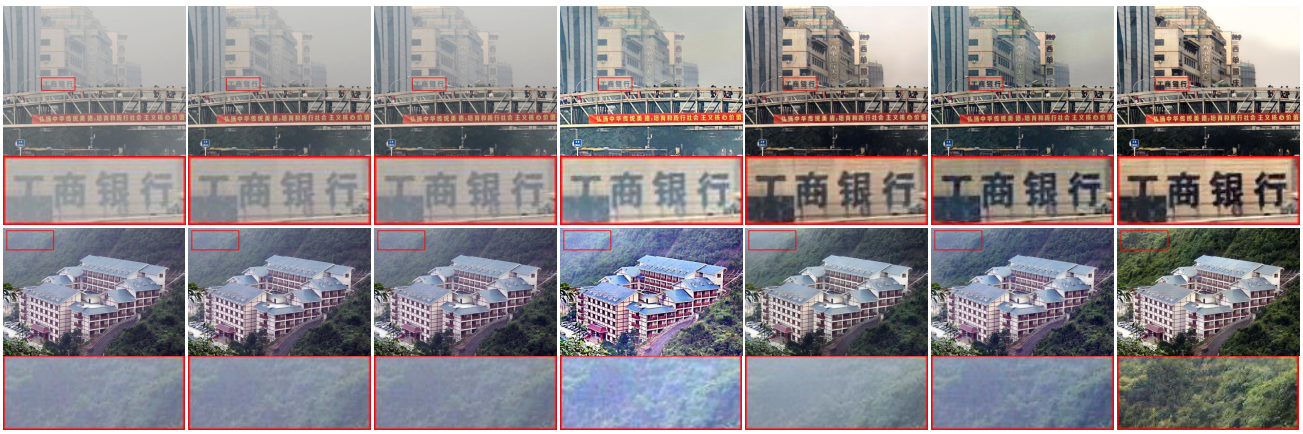
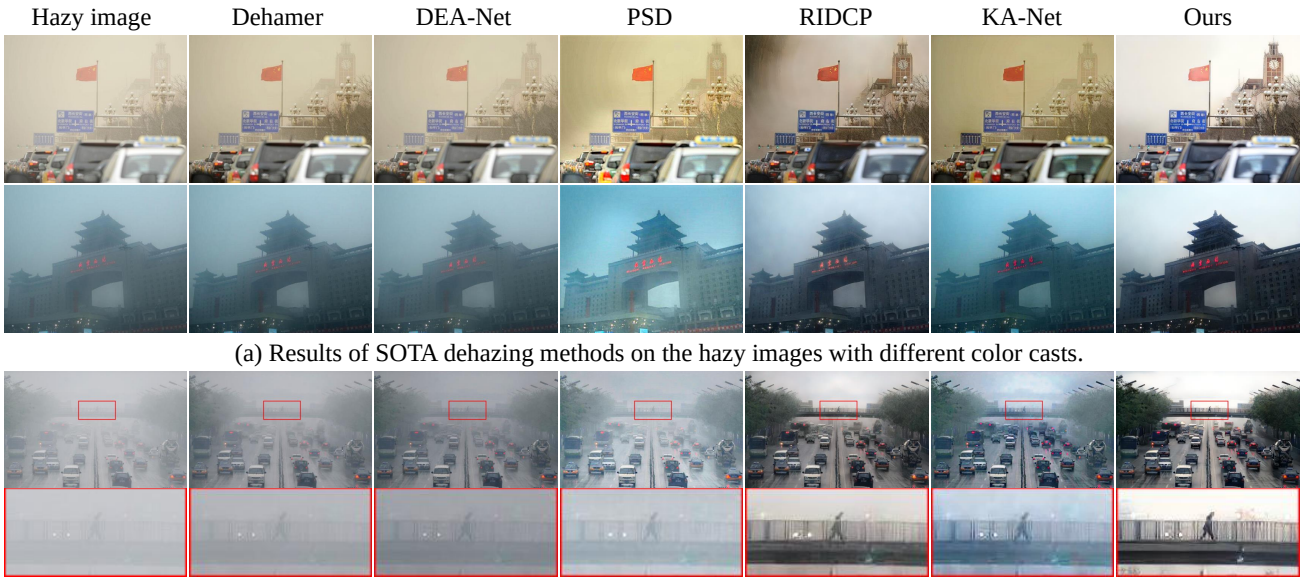


图 5. 与 URHI 的视觉效果对比。**放大以便观看**。



(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.

图 6. 高难度场景的视觉效果对比。**放大以便观看**。

迭代。

4.3. 与最先进的方法对比

我们通过定量定性分析，将我们的方法与几种最先进的去雾方法进行比较。我们在补充材料中展示了更多实验结果和分析。

定量对比。由于从真实世界中获取真实含雾图像较为困难，我们使用一些常用的无参考图像质量评估（IQA）指标对我们的方法进行了定量分析。为了更好地评估结果，我们应用了多种关注角度不同的 IQA 指标，例如感知 IQA（MUSIQ [23]、PI [3] 和 MANIQA [42]），语义 IQA（TOPIQ [8]）以及基于大语言模型的 IQA（CLIPIQA [37]、Q-Align [40]）。除了 PI 以外的其余 IQA 指标中，更高的分数表示更好的图像质量。我们将我们的方法与在基准测试中表现突出的方法进行比较，包括：MSDBN [12]、Dehamer [19]、DEA-Net [11]，以及真实世界去雾方法：DAD [33]、PSD [10]、D4 [43]、RIDCP [41] 和 KA-Net [16]。

为确保公平比较，我们运行了所有提及方法的官方代码，并使用 IQA-Pytorch [6] 对它们进行评估。如

Tab. 1所示，我们的方法在与其他最先进方法的对比中取得了第一和第二名的成绩。结果表明，我们的方法在去雾能力、色彩保真度和图像质量方面表现更优。总的来说，我们的方法在定量指标上取得了最佳结果，进一步证明了其在真实世界图像去雾任务中的优越性。

定性比较。我们在 RTTS、Fattal 和 URHI 数据集上进行了定性比较，如 Figs. 3 to 5所示。Dehamer [19] 和 DEA-Net [11] 展示出的的去雾能力有限。PSD [10] 存在一定程度的过曝和色彩偏移问题。RIDCP [41] 和 KA-Net [16] 表现较为有效，但图像质量仍有不足，且在浓雾区域表现欠佳。相比之下，我们的方法在薄雾和浓雾区域（如 Fig. 4中底部图像的红色和绿色框内区域）均显著优于其他方法。对比结果表明，我们的方法能够生成更高质量、更自然、更清晰的图像。

为了进一步证明我们方法的优越性，我们额外选择了一些具有挑战性的场景进行视觉效果比较。如 Fig. 6 (a) 所示，受低光照以及沙尘等颗粒物的影响，拍摄的带雾图像可能存在颜色偏差。我们的结果能够有效去除色偏，呈现更自然的色彩，并生成更干净的图像。在 Fig. 6 (b) 中，即使在浓雾环境下，我们的方法仍能取得不错的效果，相较于次优方法 RIDCP，我们的方法拥有更少的噪声。

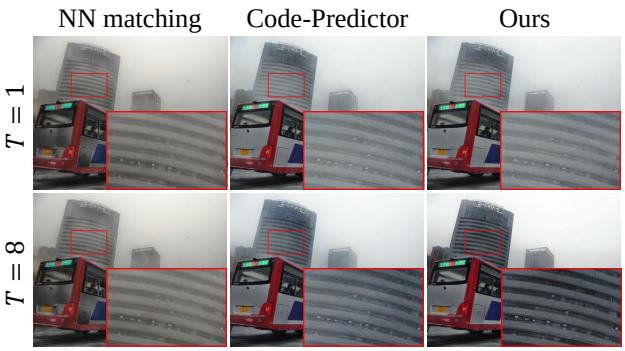


图 7. Code-Preidictor 的消融实验结果。第一行显示了无迭代情况下的实验结果，第二行则展示了添加迭代的结果。第一列展示了基于最近邻的 code 匹配结果，第二列展示了 Code-Preidictor 的结果（不使用 Code-Critic），第三列则为我们的最终结果。我们能观察到添加迭代的 Code-Preidictor（从 $T = 1$ 到 $T = 8$）能够不断产生变化，而 Code-Preidictor 的引入进一步提升了预测效果。

表 2. 当 $T = 8$ 时，对于 Code-Preidictor 在 RTTS 上的定量消融实验分析。红色是最好的结果。

方法	MUSIQ↑	PI↓	MANIQA↑	Q-Align↑	TOPIQ↑
最近邻匹配	58.19	3.25	0.303	3.25	0.458
无 Code-Critic	57.74	3.32	0.303	3.36	0.462
我们的方法	59.60	3.22	0.327	3.49	0.500

4.4. 消融实验和分析

为了验证 Code-Preidictor 和 Code-Critic 的重要性，我们进行了以下两方面的实验：（1）评估 Code-Preidictor 在迭代去雾中的有效性；（2）研究通过结合 Code-Critic 和 Code-Preidictor 在迭代去雾中的提升。

Code-Preidictor 的有效性 首先，我们将 Code-Preidictor 与最近邻（NN）匹配方法进行对比，以验证该模块在网络中的作用。为避免干扰，本次消融实验未引入 Code-Critic。为了在基于最近邻匹配的方法上实现迭代，我们采用从 token 映射到 codebook 的距离作为判定标准，以决定生成的 code 是否在下一次迭代中保留。如 Fig. 7所示，在最近邻匹配的情况下，增加迭代次数并不会带来更好的结果（见第一列）。显然在训练阶段，模型所学习到的是各个 token 的独立匹配，而并没有考虑 token 之间的关系，因此迭代不会改变 code。然而，Code-Preidictor 将 Z_l 视为条件，利用上一次迭代所得的高质量 code 来引导下一次迭代。

与最近邻匹配方法相比，我们的 Code-Preidictor 在迭代过程中能够选择更有可能的 code，这对迭代解码预测至关重要。Tab. 2 也从定量分析的角度验证了这一结论。然而，在缺少 Code-Critic 的情况下，Code-Preidictor 无法有效判断每次迭代应保留哪些 code，因此提升效果仍然有限。关于 Code-Critic 的作用，我们将在下文进一步讨论。

Code-Critic 的有效性。为了进一步探讨 Code-Critic 的必要性，我们对比了基于 code 置信度和基于 Code-Critic 的采样方法。如 Algorithm 2所示，其结果见 Figs. 7 and 8。很明显在引入 Code-Critic 后，code 的选择过程呈现出由近及远、由简单到复杂的趋势。同时，在迭代过程中，图像优化效果也得到了显著提升。相较于不使用 Code-Critic 的情况，这种设计会获得更加清晰锐利的结果。此外 Tab. 2也进一步验证了这一发现。

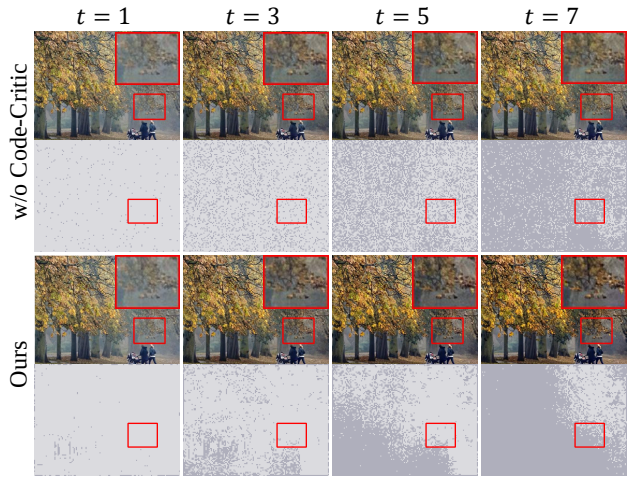


图 8. 所提出的 Code-Critic 的消融实验结果。从左至右的图片展示了在 $T = 8$ 时，不同 t 下的结果。每组图像中，上方图像展示序列 S 的结果，而下方图像展示 S 经 Code-Critic 评估后的掩码图。若不使用 Code-Critic，掩码图将呈现随机分布。而我们的方法能够优先保留薄雾区域的 code，从而获得更优的去雾效果。

5. 结论

受自然现象的启发，带雾图像的退化程度很大程度上与雾的密度及场景深度密切相关，所以我们提出了一种用于真实世界迭代去雾新方法。我们的迭代过程与单次去雾方法不同。单次去雾方法不能有效去除薄雾，而我们的迭代过程能够首先识别出稀薄雾区域并优先恢复低难度区域，随后利用已恢复区域作为引导，去预测雾密度并继续恢复高难度区域。大量实验表明验证了我们方法的优越性。

致谢。本工作的部分研究得到了国家自然科学基金（62306153，U23B2011，62176130）、中央高校基本科研业务费（南开大学，070-63243143）、天津市自然科学基金（24JCJQJC00020）、深圳市科技计划（JCYJ20240813114237048）、浙江省重点研发计划（2024SSYS0091）的资助。本工作的算力由南开大学超算中心（NKSC）提供。

Supplementary Material

Abstract

This supplementary material presents the network architectures, objective functions, further ablation experiments, and more visual results. Specifically, in ablation experiments, we include ablation results on the real-world URHI dataset to validate the effectiveness of the Code-Critic in Appendix C.1. Besides, throughout the paper, we typically set $T = 8$. To justify this setting, we discuss the effect of the number of iterations in Appendix C.2.

A. Network Architectures

The structural details of IPC-Dehaze are shown in Tab. 3. In VQGAN, we use a network structure similar to FeMaSR [7] and set the codebook size K to 1024 and embedding dim to 256. To achieve code matching and code evaluation at different resolutions, we use RSTB [28] as the backbone and add a linear projection layer.

Layers	Configuration	Output Size
Conv_in	$c_{in}=3$ $c_{out}=64$ $ksz=4$	$(h, w, 64)$
Block1	$c_{in}=64$ $c_{out}=128$ $ksz=3$ $stride=2$	$(h/2, w/2, 128)$
Block2	$c_{in}=128$ $c_{out}=256$ $ksz=3$ $stride=2$	$(h/4, w/4, 256)$
Before_quant	$c_{in}=256$ $c_{out}=256$ $ksz=1$ $stride=1$	$(h/4, w/4, 256)$
RSTB_block1	$\begin{bmatrix} ws=8 \\ d=256 \\ head=8 \\ depth=6 \end{bmatrix} \times 4$	$(h/4, w/4, 256)$
Norm1	d=256	$(h/4, w/4, 256)$
Linear1	$f_{in}=256$ $f_{out}=1024$	$(h/4, w/4, 1024)$
RSTB_block2	$\begin{bmatrix} ws=8 \\ d=256 \\ head=8 \\ depth=6 \end{bmatrix} \times 2$	$(h/4, w/4, 256)$
Norm2	d=256	$(h/4, w/4, 256)$
Linear2	$f_{in}=256$ $f_{out}=1$	$(h/4, w/4, 1)$
Codebook	$K=1024$ $d=256$	$(h/4, w/4, 256)$
After_quant	$c_{in}=256$ $c_{out}=256$ $ksz=3$ $stride=1$	$(h/4, w/4, 256)$
Upsample	ratio=2	$(h/2, w/2, 256)$
Block3	$c_{in}=256$ $c_{out}=128$ $ksz=3$ $stride=1$	$(h/2, w/2, 128)$
SFT_block1	$c_{in}=128$ $c_{out}=128$ $ksz=3$ $stride=1$	$(h/2, w/2, 128)$
Upsample	ratio=2	$(h, w, 128)$
Block4	$c_{in}=128$ $c_{out}=64$ $ksz=3$ $stride=1$	$(h, w, 64)$
SFT_block2	$c_{in}=64$ $c_{out}=64$ $ksz=3$ $stride=1$	$(h, w, 64)$
Conv_out	$c_{in}=64$ $c_{out}=3$ $ksz=3$ $stride=1$	$(h, w, 3)$

表 3. Architecture details of the IPC-Dehaze. Blue, green, brown, and yellow represent the layers of VQGAN, Code-Predictor, Code-Critic, and SFT, respectively. c_{in}, c_{out}, and ksz are the input channel, output channel, and kernel size, respectively. The ws is the window size and d is the embedding dim. f_{in} is the number of input features and f_{out} is the number of output features. The input of the network is an RGB image $\in \mathbb{R}^{h\times w\times 3}$.

B. Objective Functions

B.1. Pretrain: VQGAN Training

Following VQGAN, we adopt pixel-level reconstruction loss $\mathcal{L}_1$ and code-level loss $\mathcal{L}_{code}$ to train the Encoder E_H, Decoder D_H, and Codebook C:

$$\mathcal{L}_1 = \|I_h - I_{rec}\|_1, \tag{7}$$

$$\begin{aligned}\mathcal{L}_{code} = &\|sg(z_c) - Z_h^{(i,j)}\|_2^2 + \beta\|z_c - sg(Z_h^{(i,j)})\|_2^2 \\ &+ \lambda_g\|\mathrm{CONV}(Z_h^{(i,j)}) - \Phi(I_h)\|_2^2,\end{aligned} \tag{8}$$

where $sg(\cdot)$ is the stop-gradient operation, $\beta = 0.25$ and $\lambda_g = 0.1$ in this training. The $\mathrm{CONV}(\cdot)$ and $\Phi(\cdot)$ denote a convolution layer and a pre-trained VGG19 [35], respectively.

To restore better texture, we use perceptual loss $\mathcal{L}_{per}$ [22], and adversarial loss $\mathcal{L}_{adv}$ [13] as part of the loss function:

$$\mathcal{L}_{per} = \|\Phi(I_h) - \Phi(I_{rec})\|_1, \tag{9}$$

$$\mathcal{L}_{adv} = \log D(I_h) + \log(1 - D(I_{rec})). \tag{10}$$

In the pre-training stage, the loss is expressed as:

$$\mathcal{L}_{VQGAN} = \mathcal{L}_1 + \mathcal{L}_{code} + \mathcal{L}_{per} + \lambda_{adv}\mathcal{L}_{adv}, \tag{11}$$

where $\lambda_{adv} = 0.1$.

B.2. Stage I: Code-Predictor Training

In this training stage, we use $\mathcal{L}_1$, $\mathcal{L}_{per}$, and $\mathcal{L}_{adv}$ to train the Encoder E_L. In addition, we use the cross-entropy loss $\mathcal{L}_\theta$ to train the Code-Predictor. The total loss is expressed as:

$$\mathcal{L}_\theta = -\sum_{i=0}^{N} S_h^{(i)} \log p_\theta(S^{(i)}|Z_t), \tag{12}$$

$$\mathcal{L}_{total} = \mathcal{L}_1 + \mathcal{L}_{per} + \lambda_{adv}\mathcal{L}_{adv} + \mathcal{L}_\theta, \tag{13}$$

where $\lambda_{adv} = 0.1$, S_h is the code sequence from the clean image, and S is the code sequence predicted by Code-Predictor. Z_t denotes the fused tokens and p_θ denotes the output distribution of Code-Predictor.

B.3. Stage II: Code-Critic Training

In the training stage II, we keep all other modules fixed, exclusively train the Code-Critic, and utilize only binary cross-entropy loss:

$$\mathcal{L}_\phi = -\sum_{i=0}^{N} M^{(i)} \log p_\phi(S^{(i)}) + (1-M^{(i)}) \log(1-p_\phi(S^{(i)})), \tag{14}$$

where $M = (S_h \neq S)$ and p_ϕ denotes the output of Code-Critic.

Since the Code-Critic module is only trained to make an accurate evaluation of Code-Predictor's diverse cases, we introduce the sampling temperature when Code-Predictor samples the code sequence S. The partial code is shown in Fig. 9.

```
# Fuse the hq_feats and lq_feats with mask
input_feats=hq_feats*~mask+lq_feats*mask
# Get the logits with [b, h*w, K].
logits = net_predictor.transformer(input_feats)
# Add sampling temperature
logits/=Tem
probs = F.softmax(logits, -1)
# Samples the id.
sampled_ids = torch.multinomial(probs, 1)
# Evaluate the sampled_ids
masked_logits = net_critic(sampled_ids,h,w)
```

图 9. Partial code for Code-Critic Trainin.

C. Ablation Experiments

C.1. 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. We conduct quantitative experience on two real-world datasets RTTS and URHI [26], which both contain over 4,000 images. We present the results in Tab. 4.

表 4. Quantitative ablation analysis of the Code-Predictor. Red indicates the best results. In this experiment, we set $T = 8$.

(a) Results on RTTS.

Method	MUSIQ↑	PI↓	MANIQA↑	CLIPIQA↑	Q-Align↑	TOPIQ↑
NN Matching Based	58.19	3.25	0.303	0.391	3.25	0.458
Ours (w/o Code-Critic)	57.74	3.32	0.303	0.412	3.36	0.462
Ours	59.60	3.22	0.327	0.44	3.49	0.500

(b) Results on URHI.

Method	MUSIQ↑	PI↓	MANIQA↑	Q-Align↑
NN Matching Based	62.06	3.01	0.350	3.48
Ours (w/o Code-Critic)	60.51	3.13	0.343	3.55
Ours	62.50	3.08	0.364	3.70

C.2. Iteration Number

We investigate the impact of varying the iteration number T on inference performance, as summarized in Tab. 5. To balance performance and efficiency, we set $T = 8$ as the default. As shown in Tab. 5, reducing T slightly degrades performance but still outperforms other methods. For faster inference, a smaller T can be selected without the need for network retraining.

表 5. Quantitative analysis of the different iteration numbers. Red indicates the best results.

T	MUSIQ↑	PI↓	MANIQA↑	CLIPIQA↑	Q-Align↑	TOPIQ↑
3	58.40	3.33	0.314	0.425	3.43	0.478
4	58.71	3.31	0.318	0.431	3.45	0.484
6	58.97	3.30	0.321	0.437	3.47	0.490
8	59.60	3.22	0.327	0.444	3.489	0.500
10	59.23	3.28	0.325	0.442	3.488	0.496

D. Visual Results

D.1. Visualisation of Iterative Decoding

Fig. 10 visualizes the iterative process, and it can be seen that during the iterative process, our results get better and better in terms of dehazing, image quality, and visual effects.

D.2. More Visual Comparisons

Figs. 11 to 13 show visual comparisons on the RTTS, URHI [26] and Fattal [15] datasets. We compare our method with the methods that have achieved outstanding results in benchmarks: MSDBN [12], Dehamer [19], and DEA-Net [11] as well as real-world image dehazing methods: DAD [33], PSD [10], D4 [43], RIDCP [41], and KA-Net [16]. The results show that our method can achieve more natural, high-quality images, especially in dense hazy areas, which is a significant improvement compared to other methods.

D.3. Failure Case

In this part, we discuss the failure case of our method. To be specific, our method exhibits a progressive approach to dehazing, processing denser haze regions first and gradually moving to thinner regions during iterative dehazing. This gives us a distinct advantage in handling depth-continuous scenes, as high-quality codes from earlier iterations can serve as cues for the Code-Predictor to refine subsequent predictions. However, the performance is limited in depth-discontinuous scenes, as illustrated in Fig. 14. While our method performs exceptionally well in depth-continuous regions (outside the red box), it struggles as the other methods in depth-discontinuous regions (inside the red box, such as trees or rocks). Such depth-discontinuous scenarios present significant challenges for all existing dehazing methods.

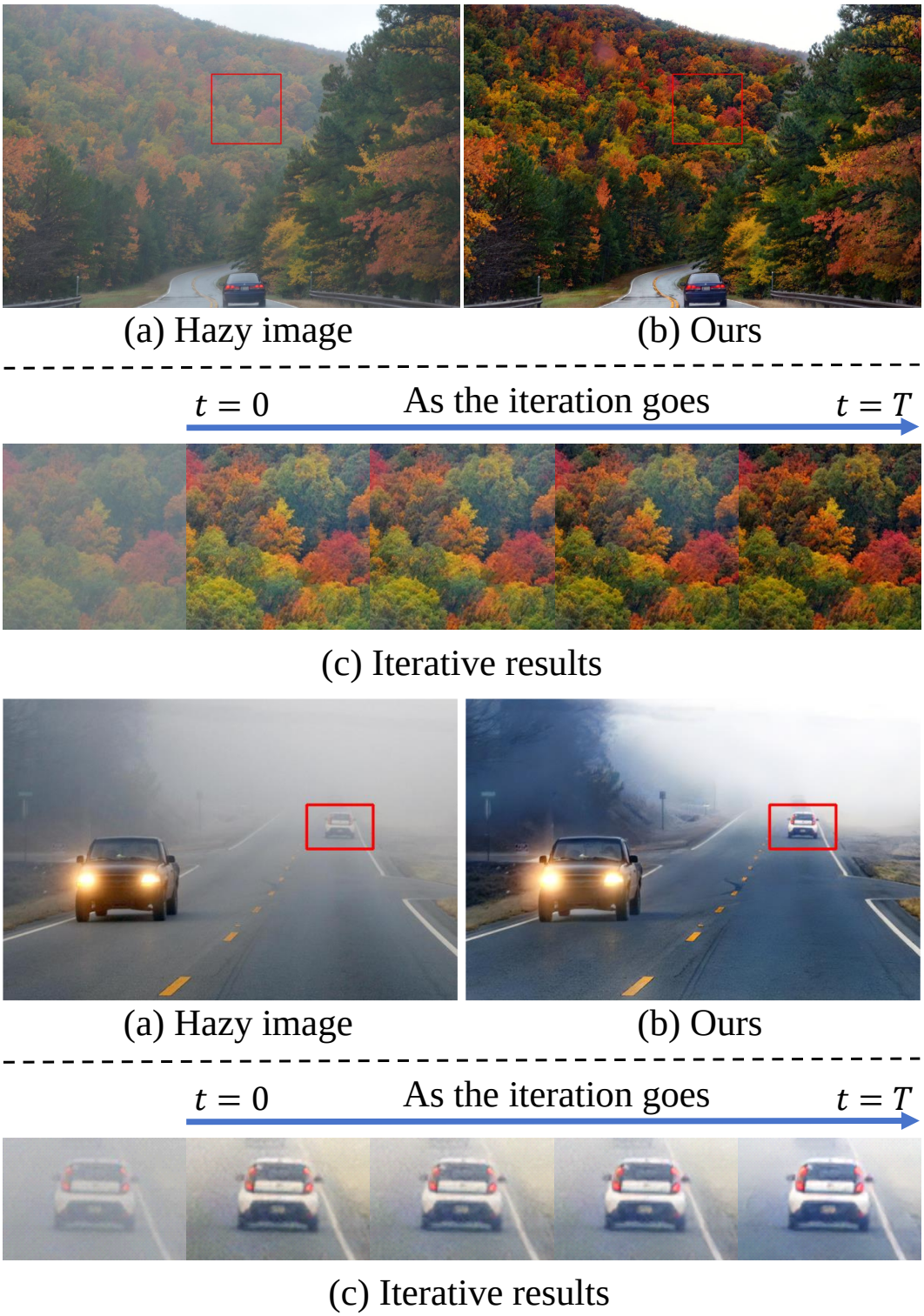


图 10. The top-left shows the input image, while the top-right displays our result. Below, the images from left to right depict the intermediate results of the iterative decoding process when $T = 8$.

图 11. Visual comparison on RTTS. **Zoom in for best view**.

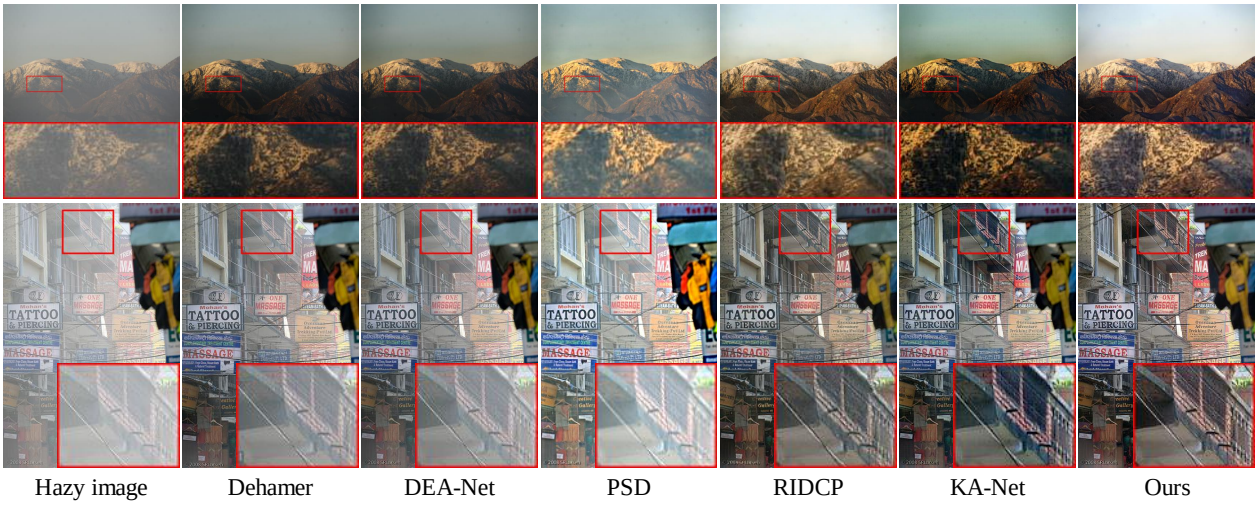


图 12. Visual comparison on Fattal. **Zoom in for best view**.

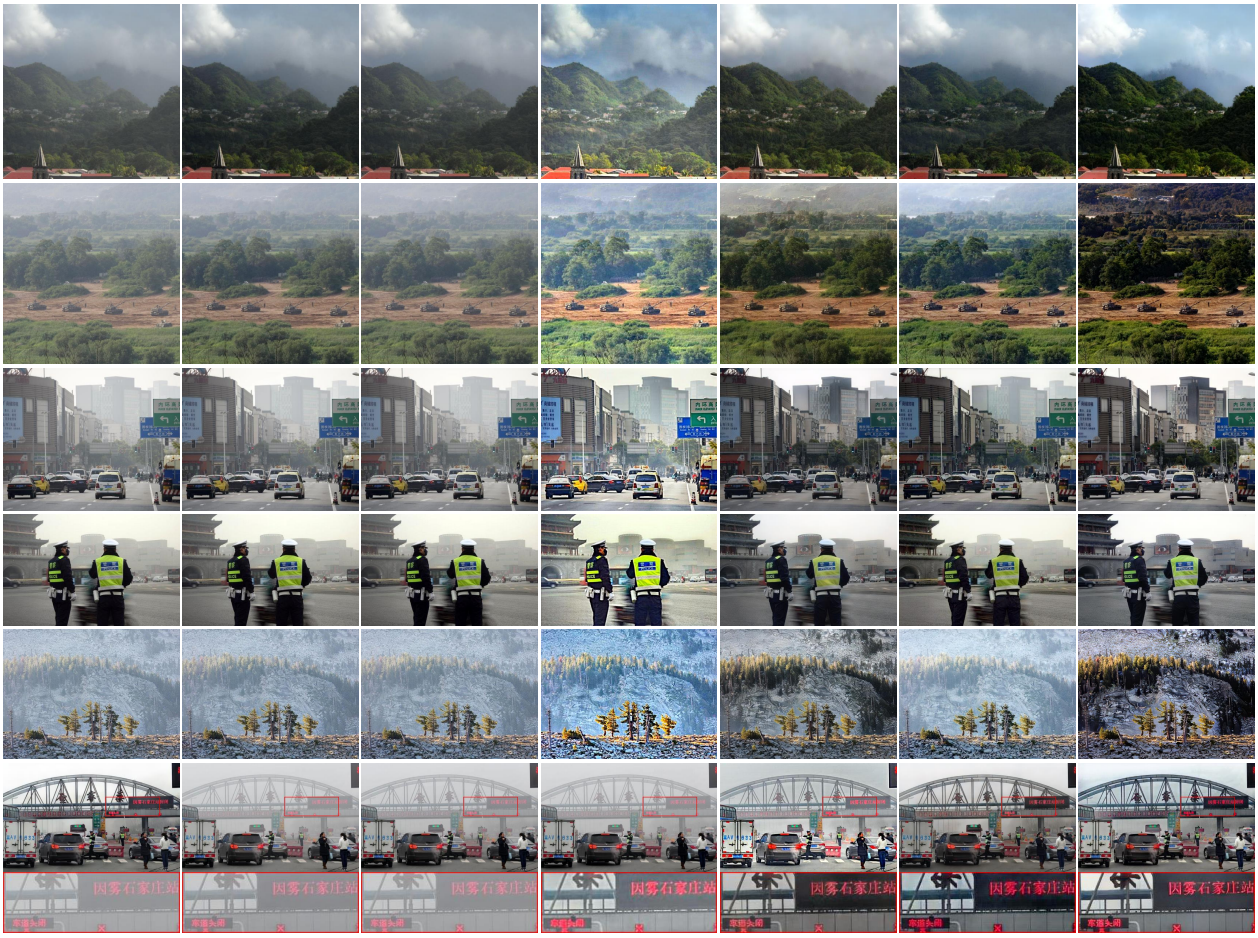


图 13. Visual comparison on URHI. **Zoom in for best view**.

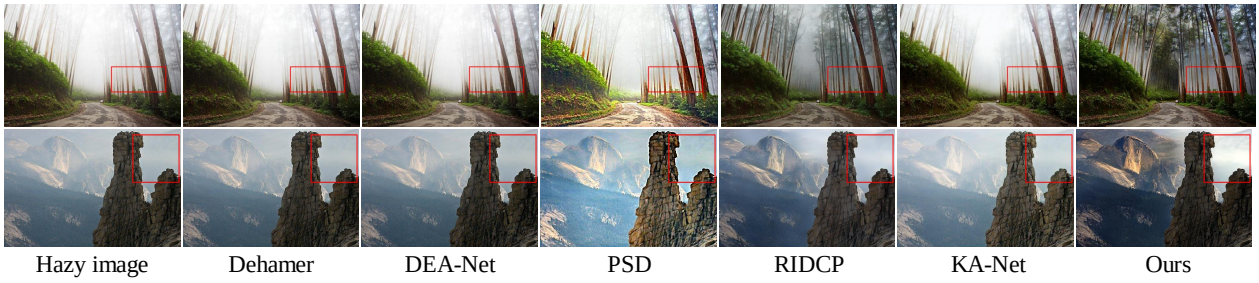


图 14. Failure Case. As shown in the red boxes, depth-discontinuous regions present significant challenges for existing dehazing methods. While our method performs well in the depth-continuous regions outside the red box, it struggles inside the red box, where trees and rocks disrupt depth continuity, rendering our method as ineffective as other approaches.

References

[1] Eirikur Agustsson and Radu Timofte. Ntire 2017 challenge on single image super-resolution: Dataset and study. In *CVPRW*, 2017. 6

[2] Dana Berman, Tali Treibitz, and Shai Avidan. Non-local image dehazing. In *CVPR*, 2016. 1, 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. 7

[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. 7

[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. 2, 6, 10

[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. 7

[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. 2

[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, 5, 7, 8, 12

[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, 7, 8, 12

[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, 7, 12

[13] Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *CVPR*, pages 12873–12883, 2021. 10

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

[15] Raanan Fattal. Dehazing using color-lines. *ACM TOG*, page 1–14, 2014. 1, 2, 6, 12

[16] 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, 7, 8, 12

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

[18] 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. 2

[19] 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, 7, 8, 12

[20] Kaiming He, Jian Sun, and Xiaoou Tang. Single image haze removal using dark channel prior. *IEEE TPAMI*, 33(12):2341–2353, 2010. 1, 2

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

[22] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *ECCV*, pages 694–711. Springer, 2016. 10

[23] Junjie Ke, Qifei Wang, Yilin Wang, Peyman Milanfar, and Feng Yang. Musiq: Multi-scale image quality transformer. In *ICCV*, pages 5148–5157, 2021. 7

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

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

[26] 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. 6, 11, 12

[27] 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

[28] 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. 6, 10

[29] Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee. Enhanced deep residual networks for single image super-resolution. In *CVPRW*, 2017. 6

[30] 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

[31] 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

[32] 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

[33] Yuanjie Shao, Lerenhan Li, Wenqi Ren, Changxin Gao, and Nong Sang. Domain adaptation for image dehazing. In *CVPR*, 2020. 2, 5, 7, 12

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

[35] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *ICLR*, 2015. 10

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

[37] Jianyi Wang, Kelvin CK Chan, and Chen Change Loy. Exploring clip for assessing the look and feel of images. In *AAAI*, 2023. 7

[38] 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. 5

[39] Xintao Wang, Liangbin Xie, Chao Dong, and Ying Shan. Real-esrgan: Training real-world blind super-resolution with pure synthetic data. In *ICCVW*, 2021. 6

[40] 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. 7

[41] 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, 5, 6, 7, 8, 12

[42] 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. 7

[43] 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, 7, 12

[44] 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

[45] Shangchen Zhou, Kelvin Chan, Chongyi Li, and Chen Change Loy. Towards robust blind face restoration with codebook lookup transformer. *NeurIPS*, 35: 30599–30611, 2022. 2, 5

[46] Qingsong Zhu, Jiaming Mai, and Ling Shao. Single image dehazing using color attenuation prior. In *BMVC*, 2014. 1, 2