Inf-Dehaze: Beyond GPU Memory Constraints for Ultra-High-Resolution Image Dehazing

Supplementary Material

1. Concept

GPU Memory refers to the high-speed memory integrated directly on the graphics card, used to store neural network weights and tensors during model inference and training. Owing to its proximity to GPU cores, it provides high bandwidth and fast data access. Consumer GPUs typically offer 4-32 GB of GPU memory, while professional computing GPUs may reach 48 GB or even up to 141 GB (e.g., NVIDIA H200). However, large-capacity GPU memory is extremely expensive, making memory reduction during inference both necessary and challenging.

DRAM (**D**ynamic **R**andom **A**ccess **M**emory) serves as the system's main memory, temporarily storing data and programs processed by the CPU. Common desktops are equipped with 8-128 GB of DRAM, whereas server systems can scale to over 1 TB. Although DRAM provides lower bandwidth and slower access compared to GPU memory, its cost per unit capacity is significantly lower. In this study, we strategically offload infrequently accessed variables (e.g., residual connections) from GPU memory to DRAM to conserve valuable GPU resources. For instance, processing an 8192×8192 image requires only about 4 GB of DRAM for residual data (less than 1 GB for typical 4K images), thus requiring only a small amount of DRAM. In extreme cases where both GPU memory and DRAM are severely constrained, the residual data can also be offloaded to secondary storage such as SSDs. To avoid ambiguity with CPU caches, we consistently refer to it as **DRAM** throughout this paper.

2. Module and Inference Framework Details

This section provides the implementation of the proposed Spatial-Frequency Cross Attention, Local Modeling Mixture-of-Experts, and Intra-Block Sparse Self-Attention modules. We also present the complete memory-efficient inference framework used in Inf-Dehaze.

2.1. Spatial-Frequency Cross Attention (SFCA)

SFCA integrates spatial-domain features with frequency-domain cues to better separate haze-related degradations from scene structures. Given an input feature map, we first transform it into the frequency domain and apply predefined masks to extract multiple frequency bands. The masked frequency components are transformed back into the spatial domain to form frequency responses.

For each frequency band j ($j \in \{1, 2, 3\}$), spatial queries $\mathbf{Q}_j^s$ and frequency keys/values $(\mathbf{K}_j^f, \mathbf{V}_j^f)$ are constructed. Cross-domain attention is computed as

$$\mathbf{Attn}_j = \mathrm{Softmax}\left(\frac{\mathbf{Q}_j^s(\mathbf{K}_j^f)^\top}{\sqrt{d}}\right)\mathbf{V}_j^f \tag{1}$$

where d denotes the attention head dimension. A weight predictor outputs fusion coefficients ω_j, and the final feature is obtained by

$$\mathbf{F}_{\text{out}} = \sum_{j=1}^{3} \omega_j \cdot \mathbf{Attn}_j \tag{2}$$

This module enables the network to leverage complementary spatial and frequency signals for more reliable haze removal.

2.2. Local Modeling Mixture-of-Experts (LMMoE)

LMMoE introduces a dedicated pathway for modeling localized spatial structures, complementing global transformer-based modeling. Given an input $\mathbf{X} \in \mathbb{R}^{B\times C\times H\times W}$, a gating network predicts a spatially adaptive distribution over N experts, and only the Top-k experts are activated at each location to reduce computational cost.

Each expert adopts a lightweight three-layer convolutional structure:

$$\mathbf{E}_i = \mathrm{Conv}_{1\times1}^{\text{out}}\left(\mathrm{DWConv}\left(\mathrm{Conv}_{1\times1}^{\text{in}}(\mathbf{X})\right)\right) \tag{3}$$

The output is computed as a weighted sum of the activated experts and a shared expert:

$$\mathbf{F}_{\text{out}} = \sum_{i\in\text{Top-}k} \alpha_i\mathbf{E}_i + \alpha_{\text{shared}}\mathbf{E}_{\text{shared}} \tag{4}$$

where α_i are the expert weights predicted by the gating network.

2.3. Intra-Block Sparse Self-Attention (IBSSA)

IBSSA enables scalable global modeling under extreme memory constraints by introducing structured sparsity into the attention computation. Queries and keys are first reordered using Locality Sensitive Hashing (LSH), which clusters tokens with high attention relevance near the diagonal of the similarity matrix. This reordering encourages a band-diagonal structure aligned with efficient attention kernels.

After sorting, the token sequence is divided into fixed-size diagonal blocks. Attention is computed within each

block, with optional interactions between adjacent blocks to mitigate potential information loss:

$$\text{Attention}(Q_B, K_B, V_B) = \text{Softmax}\left(\frac{Q_B K_B^\top}{\sqrt{d}}\right) V_B \quad (5)$$

The original token order is then restored to complete the computation. This sparse formulation significantly reduces memory usage while preserving long-range dependency modeling.

2.4. Memory-Efficient Inference Framework

Algorithm 1 outlines the core inference procedure of Inf-Dehaze, which combines asynchronous batch-based processing strategy with residual caching strategy to enable efficient ultra-high-resolution dehazing.

Algorithm 1 Memory-Efficient Inference Framework

Input: Image $\mathbf{I}$, patch size s, encoder E, bottleneck B, decoder D
Partition $\mathbf{I}$ into patches $\{\mathbf{P}_i\}_{i=1}^{N^2}$
Encoder Phase:
for each batch $\{\mathbf{P}_i\}_{i=1}^{b}$ **do**
  Compute features and residuals: $\{\mathbf{f}_i, \{\mathbf{r}_i^l\}\} = E(\{\mathbf{P}_i\})$
  Store all residuals $\{\mathbf{r}_i^l\}$ in DRAM
end for
Bottleneck Phase:
Concatenate features: $\mathbf{F} = \bigoplus_i \mathbf{f}_i$
Global modeling: $\mathbf{F}' = B(\mathbf{F})$
Split into enhanced features: $\{\mathbf{f}_i'\} = \text{split}(\mathbf{F}')$
Decoder Phase:
for each batch $\{\mathbf{f}_i'\}_{i=1}^{b}$ **do**
  Retrieve residuals $\{\mathbf{r}_i^l\}$ from DRAM
  Decode to obtain patches $\{\mathbf{P}_i'\} = D(\{\mathbf{f}_i'\}, \{\mathbf{r}_i^l\})$
end for
Merge all patches to obtain $\mathbf{I}_{\text{Dehaze}}$

3. Dataset Details

Our experiments are conducted on three representative datasets: the natural-scene dataset *4KID*, the remote-sensing dataset *8KDehaze*, and the real-world dataset *O-HAZE*. These datasets together cover a broad range of imaging scenarios, haze distributions, and spatial scales, providing a comprehensive benchmark for evaluating ultra-high-resolution image dehazing.

For each dataset, we randomly select 100 samples from the daytime subset of *4KID*, 100 samples from *8KDehaze*, and 5 samples from *O-HAZE* as the test set. The remaining images are used for training. To enhance generalization, all training samples are augmented through random cropping and random rotation.

Each dataset contains paired hazy and haze-free ground-truth images but differs significantly in haze characteristics and image structure. In *4KID* and *O-HAZE*, haze tends to be globally distributed, and its intensity correlates with scene depth and illumination, simulating natural atmospheric scattering in outdoor environments. In contrast, *8KDehaze* features ultra-high-resolution aerial and satellite images with irregular, cloud-like haze distributions. The haze is spatially uneven and often partial, leaving clear and hazy regions coexisting within a single frame. Such complex, high-resolution, and nonuniform degradation patterns introduce substantial challenges for dehazing models, particularly in maintaining global consistency while restoring fine-grained textures.

Table 1 summarizes the detailed dataset statistics, and Figure 1 presents representative examples from each dataset.

4. Implementation Details

Experimental Setup. Our experiments were primarily conducted on a server equipped with two NVIDIA H200 GPUs (141 GB memory each), dual Intel Xeon Gold 6330 CPUs, and 256 GB of DDR4 DRAM. The software environment consisted of Linux, Python 3.10, and CUDA 12.4. To evaluate the adaptability of our model under resource-limited conditions, we additionally tested Inf-Dehaze on an RTX 4090 (24 GB), an RTX 4050 Laptop GPU (6 GB), and even on a CPU-only setting (albeit significantly slower). Despite employing the residual caching strategy, Inf-Dehaze requires only an additional 4 GB of DRAM, imposing no substantial demands on system memory. Unlike many recent Mamba-based models, our approach supports both Windows and Linux for training and inference. Furthermore, since our work aims to serve as a solid baseline for future research, we adopt a simple Swin Transformer layer for the encoder-decoder blocks by default. This design choice allows future studies to easily replace it with more advanced CNN-, Transformer-, Mamba-, or other neural-network-based modules to further improve performance.

Model Hyperparameter Setup. For the proposed modules in Inf-Dehaze, the SFCA module first reduces the input dimension to 32 via a linear layer to improve computational efficiency. Three frequency bands are extracted via filters guided by two downsampled spatial masks at 1/16 and 1/32 of the input resolution. The WeightNet is a lightweight CNN consisting of a convolution layer (reducing dimension to 4), a ReLU activation, another convolution(reducing dimension to 1), and a pooling layer. Finally, SFCA restores the output to the original dimension through a linear projection and incorporates a residual connection for stability. The LMMoE module employs a gating mechanism based on a convolution followed by a softmax function. Each expert first expands the channel dimension by a factor of four using a convolutional layer, applies local modeling using depth-wise separable convolutions with kernel sizes of 3×3 or 5×5,

Figure 1. Typical samples from *4KID*, *8KDehaze*, and *O-HAZE*

Table 1. Overview of datasets used in the experiments.

Dataset	Quantity	Image Size	Source	Content
4KID	15606	3840 × 2160	Video Frames	Urban Streets
8KDehaze	10000	8192 × 8192	Aerial Images	Urban, Mountains, Desert,Coastlines
O-HAZE	45	1286 × 947 to 5436 × 3612	Real-world Experiment	Parks, Suburban

and then reduces the dimension back via another convolution. LMMoE contains three experts with 3×3 kernels and three with 5×5 kernels, among which only two are activated during inference. Additionally, it includes one shared expert with a 3×3 kernel, with $\alpha_{\text{shared}} = 0.1$.

In the overall architecture of Inf-Dehaze, the model adopts asynchronous batch-based tile processing strategy, where each tile has a spatial size of 256. The batch size is set to 1 for images with resolutions below 1024, to 4 for resolutions between 1024 and 4096, and to 16 for resolutions above 8192. This parameter can be dynamically adjusted to balance speed and efficiency. The linear embedding layer expands the channel dimension to 256 to enrich feature representation. The encoder and decoder follow a symmetric design (except for the SFCA module), each consisting of four stages. The encoder includes $[2, 3, 4, 2]$ Transformer layers across the four stages, with $[4, 4, 8, 16]$ attention heads, while the decoder mirrors this configuration. The bottleneck contains 4 Transformer layers, enabling efficient global information exchange.

Training Setup. During training, we mix the training subsets of all three datasets and train the model for 500 epochs. All methods are optimized using the standard L1 loss. To further enhance the local modeling capability of the LMMoE module in Inf-Dehaze, we introduce an additional auxiliary loss. The initial learning rate is set to 0.001 and decayed using the cosine annealing schedule. The batch size is fixed at 2. Overall, a single epoch takes approximately 30 minutes and requires about 131 GB of GPU memory. We observe that dataset I/O is the primary bottleneck in training speed; faster storage and higher CPU performance significantly reduce loading time and can speed up training by several times.

Figure 2. More dehazed results on *4KID*, *8KDehaze* and *O-HAZE*

Due to the substantial memory requirements, we additionally design an alternative training strategy. Analogous to the common practice in large language models, pretraining on shorter sequences before scaling to longer ones, our alternative strategy performs 500 epochs of pretraining at a lower resolution (1024×1024). We then freeze the encoder and decoder and fine-tune only the bottleneck module for 100 epochs on higher-resolution images, using an initial learning rate of 0.00001. This approach keeps the memory requirement below 40 GB, making it suitable for common compute-oriented GPUs such as the A100, while incurring only a negligible performance drop (-0.08 dB). With FP16 precision, the entire training process can even be completed on a single RTX 4090. It is worth noting that the model exhibits slower convergence when handling ultra-high-resolution images. To achieve strong performance, extensive training is required, with particular emphasis on thoroughly optimizing the encoder and decoder during the pretraining stage.

Ablation Study Setup. In the ablation study on inference strategies, we verify that the residual caching strategy substantially reduces GPU memory consumption when processing high-resolution images, while introducing only a modest increase in latency. However, the residual caching strategy is tightly coupled with the asynchronous batch-based inference mechanism. When adopting a sequential inference scheme used in DehazeXL, residual caching fails to

deliver the expected benefits. This occurs because, although residual caching reduces memory usage within the bottleneck module, the encoder still produces a large number of intermediate residuals during sequential processing. Since these residuals cannot be cached promptly, memory usage in the encoder increases considerably, thereby offering only marginal reductions in overall GPU memory consumption.

In the speed evaluation, the default configuration achieves a favorable balance among performance, inference speed, and memory usage. The memory-optimized variant, Inf-Dehaze$_{\text{memory}}$, sets the inference batch size to 4 to further reduce memory consumption, though at the cost of notably increased latency. In contrast, Inf-Dehaze$_{\text{fast}}$ attains the best inference speed among all compared methods, while maintaining competitive memory usage and only a slight performance drop relative to the standard Inf-Dehaze. This variant replaces the encoder-decoder modules with ConvNeXt blocks [2]. The encoder and decoder each consist of four stages, with two ConvNeXt blocks per stage. The asynchronous batch size is increased to 64, and the residual caching strategy is disabled to maximize inference throughput for latency-sensitive applications. These results further demonstrate the potential of Inf-Dehaze as a versatile backbone for future research.

5. More Qualitative Results

We provide additional visual comparisons to illustrate the dehazing performance of Inf-Dehaze across diverse scenarios. For clarity, we compare our method with several representative approaches. As shown in Figure 2, the examples cover a wide range of conditions, including light and heavy haze, uniform and non-uniform distributions, natural and remote-sensing imagery, as well as both synthetic and real-world datasets.

Notably, DiffDehaze [3] leverages prior knowledge to improve perceptual quality in some heavily degraded regions. However, due to the inherent characteristics of diffusion models, it prioritizes perceptual realism at the expense of pixel fidelity, resulting in significantly lower PSNR and SSIM scores [4]. Its inability to effectively utilize global information also leads to strong color inconsistencies between adjacent regions, producing pronounced blocking artifacts. Patch-based inference methods similarly exhibit varying degrees of blockiness.

Among direct inference approaches, 4KDehazing [5] produces globally consistent results without blocking artifacts, but its pixel-level accuracy remains limited, and it struggles with the highly non-uniform haze commonly seen in remote-sensing imagery. DehazeXL benefits from global modeling and achieves relatively uniform dehazing, yet its weak global-local information integration results in structured artifacts and suboptimal performance on real-world data.

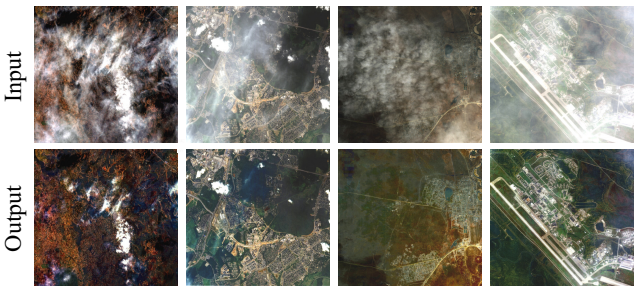

Figure 3. Challenging experiments on real-world UHR images.

In contrast, Inf-Dehaze delivers high-quality and spatially consistent dehazing across datasets and scene types, even under strict memory constraints. On the real-world *O-HAZE* dataset, it achieves the most thorough haze removal. In challenging cases where image degradation is affected by complex illumination effects, Inf-Dehaze produces the most faithful color restoration—slightly darker than the ground truth but more realistic given the scene context.

Furthermore, to rigorously validate the dehazing capability of Inf-Dehaze on real-world ultra-high-resolution remote sensing images, we constructed a test dataset comprising $30,000 \times 30,000$ pixel images. As illustrated in Figure 3, Inf-Dehaze achieves consistent and robust dehazing performance in challenging scenarios characterized by uneven and diverse haze distributions, effectively enhancing the quality of remote sensing images."

Overall, the extensive experiments demonstrate the capability of Inf-Dehaze to handle both synthetic and real-world high-resolution dehazing. Future work could further enhance its potential by developing large-scale, high-quality real-world haze datasets.

6. Motivation

Inf-Dehaze is designed to deliver high-quality dehazing for ultra-high-resolution images under constrained computational resources. This challenge arises in two practical scenarios: (1) performing 8K image dehazing on edge devices such as the NVIDIA Jetson Nano with only 4 GB of GPU memory, which is critical for robotics and autonomous driving; and (2) processing extremely large images (e.g., 30,000×30,000) on more capable servers equipped with GPUs such as the NVIDIA A100 (80 GB), which is essential in remote sensing and medical imaging. Existing patch-based inference methods often suffer from blocking artifacts. Among direct-inference approaches, 4KDehazing offers fast runtime but limited restoration quality, while DehazeXL achieves stronger dehazing through global modeling but suffers from an inefficient bottleneck design, resulting in high memory consumption and noticeable structural artifacts. Consequently, there remains no memory-efficient solution for high-resolution image dehazing.

To address these issues, we adopt a stream-style architec-

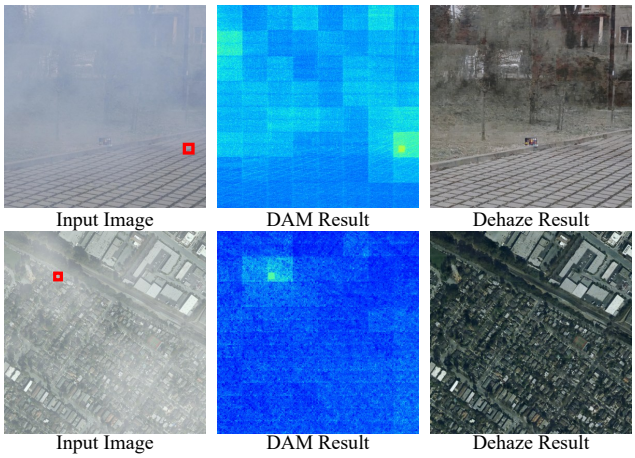

Figure 4. Attribution maps of Inf-Dehaze, The redbox indicates the region for attribution.

ture based on batch-wise encoding, a bottleneck stage, and batch-wise decoding, carefully optimizing memory usage across all components. We observe that the bottleneck consumes more memory than the encoder-decoder, largely due to residual tensors that are stored but not actively used in computation. This motivates our residual caching strategy, which temporarily stores these residuals in DRAM to reduce GPU memory usage. However, as shown in our ablation studies, caching only in the bottleneck fails to reduce overall memory consumption because residuals accumulated during sequential encoding still reside on the GPU. To resolve this, we introduce an asynchronous batch-based processing strategy that caches encoder residuals to DRAM in parallel with ongoing computation, significantly reducing memory usage during inference.

In addition, to achieve high-quality restoration, we design the SFCA module to enhance feature extraction in the encoder. The LMMoE module improves local modeling and suppresses blocking artifacts, while IBSSA provides efficient global information integration, enabling uniform and high-fidelity dehazing across the entire image.

Compared with existing dehazing methods such as DehazeXL, our model achieves remarkably high memory efficiency and is the first to process extremely large images (e.g., 30,000×30,000 remote-sensing images) on a consumer-level GPU (RTX4090). This efficiency stems from three main factors. First, we employ a residual caching strategy tightly integrated with an asynchronous batch-based processing strategy. Second, we leverage the highly efficient and sparse IBSSA module together with the locally modeling LMMoE. Third, we carefully tune the model's parameters to avoid imbalanced or inefficient GPU usage and to enhance its global modeling capability.

7. Dehazing Attribution Maps of Inf-Dehaze

Dehazing Attribution Map (DAM) [1] is a visual attribution tool designed to analyze how effectively a model leverages global information when reconstructing a specified region. As shown in the Figure 4, Inf-Dehaze makes efficient use of image-wide context, enabling high-quality dehazing. Benefiting from the LMMoE and IBSSA modules, the proposed method effectively integrates both high-frequency local information from neighboring regions and broad global cues. Interestingly, in natural scenes where haze distributions follow relatively regular patterns, Inf-Dehaze tends to rely more on global information. In contrast, for remote-sensing images with highly irregular and stochastic haze distributions, the model places greater emphasis on local learning. It also exhibits heightened attention to edge and texture structures, such as tile seams or road boundaries.

8. Broader Impacts

Image dehazing is a fundamental task in image restoration, aiming to remove haze-induced degradation from captured imagery. In computer vision, effective dehazing can substantially benefit downstream applications such as object detection, segmentation, and tracking, and it plays an important role in domains including autonomous driving, surveillance, remote sensing, and medical imaging. This work introduces a dehazing model specifically designed for ultra-high-resolution images, achieving strong performance with competitive computational efficiency and inference speed. Unlike content-generative models that can create novel, realistic imagery—raising concerns about deepfakes and misinformation—image dehazing is a restoration task that recovers latent scene information without introducing new content, and thus carries minimal societal risk.

References

[1] Jiuchen Chen, Xinyu Yan, Qizhi Xu, and Kaiqi Li. Tokenize image patches: Global context fusion for effective haze removal in large images. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 2258–2268, 2025. 6

[2] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11976–11986, 2022. 5

[3] Ruiyi Wang, Yushuo Zheng, Zicheng Zhang, Chunyi Li, Shuaicheng Liu, Guangtao Zhai, and Xiaohong Liu. Learning hazing to dehazing: Towards realistic haze generation for real-world image dehazing. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 23091–23100, 2025. 5

[4] Ruihan Yang and Stephan Mandt. Lossy image compression with conditional diffusion models. *Advances in Neural Information Processing Systems*, 36:64971–64995, 2023. 5

[5] Zhuoran Zheng, Wenqi Ren, Xiaochun Cao, Xiaobin Hu, Tao Wang, Fenglong Song, and Xiuyi Jia. Ultra-high-definition image dehazing via multi-guided bilateral learning. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16180–16189. IEEE, 2021. 5