# Ultra-High-Definition Dynamic Multi-Exposure Image Fusion via Infinite Pixel Learning

**Xingchi Chen**[1,2*], **Zhuoran Zheng**[1,2*], **Xuerui Li**[3], **Yuying Chen**[1], **Shu Wang**[4], **Wenqi Ren**[1†]

[1]Shenzhen Campus of Sun Yat-sen University
[2]Jimei University
[3]The State University of New York at Buffalo
[4]Fuzhou University
chenxch66@mail2.sysu.edu.cn, zhengzhr@mail.sysu.edu.cn,
xueruili@buffalo.edu, elvin.chenyuying@gmail.com,
shu@fzu.edu.cn, renwq3@mail.sysu.edu.cn

## Abstract

With the continuous improvement of device imaging resolution, the popularity of Ultra-High-Definition (UHD) images is increasing. Unfortunately, existing methods for fusing multi-exposure images in dynamic scenes are designed for low-resolution images, which makes them inefficient for generating high-quality UHD images on a resource-constrained device. To alleviate the limitations of extremely long-sequence inputs, inspired by the Large Language Model (LLM) for processing infinitely long texts, we propose a novel learning paradigm to achieve UHD multi-exposure dynamic scene image fusion on a single consumer-grade GPU, named Infinite Pixel Learning (IPL). The design of our approach comes from three key components: The first step is to slice the input sequences to relieve the pressure generated by the model processing the data stream; Second, we develop an attention cache technique, which is similar to KV cache for infinite data stream processing; Finally, we design a method for attention cache compression to alleviate the storage burden of the cache on the device. In addition, we provide a new UHD benchmark to evaluate the effectiveness of our method. Extensive experimental results show that our method maintains high-quality visual performance while fusing UHD dynamic multi-exposure images in real-time (>40fps) on a single consumer-grade GPU.

## Introduction

Using multiple images with different exposures to generate a clear image is a common technique. In recent years, due to the emergence of sophisticated imaging sensors and displays, the popularity of Ultra-High-Definition (UHD) images is rapidly increasing. However, existing multi-exposure image fusion (MEF) algorithms focus on generating a single low-resolution image and cannot run multiple UHD images with different exposures on resource-constrained devices.

To date, many methods have been developed to enable running UHD images on consumer GPUs, such as UHD image dehazing (Zheng et al. 2021), UHD deblurring (Deng et al. 2021), and UHD deraining (Chen et al. 2024). However, most of these methods focus on processing a single image with 4K ($3840 \times 2160$) resolution or higher on a single consumer GPU, which is usually inefficient for processing multiple heterogeneous UHD images. In this research field, among existing multi-exposure fusion algorithms, such as traditional algorithms (Zhang and Cham 2010; Ma and Wang 2015; Li et al. 2020), only a few of them can process UHD images in real time. Unfortunately, the images obtained by manually extracting features are prone to artifacts or ghosting. In addition, among deep learning-based methods (Yan et al. 2019; Xu, Ma, and Zhang 2020; Prabhakar et al. 2021; Zhen Liu and Liu 2022; Tel et al. 2023) only AHDRNet (Yan et al. 2019), FSHDR (Prabhakar et al.

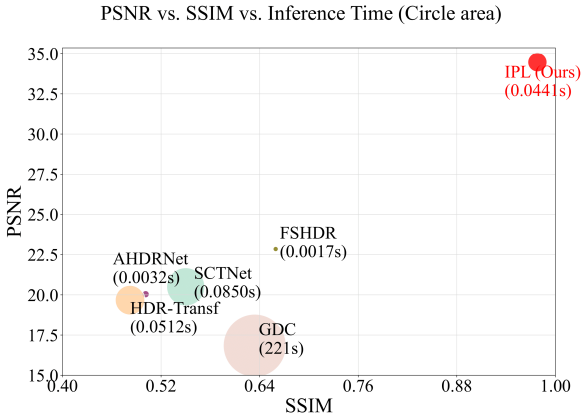


Figure 1: Model performance and efficiency comparison between our proposed IPL and other MEF methods on our proposed dataset. Since most methods are unable to process UHD images directly, the calculation is performed based on the maximum resolution (Zheng et al. 2021) that these algorithms can handle on a single GPU. Our method has approximately **46%** higher PSNR and **48%** higher SSIM than the second-best method, and the inference speed reaches real-time (>40fps), achieving an optimal trade-off between performance and efficiency.

*These authors contributed equally.
†Corresponding author

2021), can process UHD images in real-time, but they generate UHD images with low quality. In this paper, we attempt to develop an efficient deep network that balances speed and accuracy (see Figure 1).

Inspired by Large Language Models (LLMs) for processing extremely long sequential inputs (Han et al. 2024), we highlight the importance of chunking and cache acceleration. To boost the inference speed of the model, LLMs leverage KV caches, which store precomputed key/value vectors and reuse them for token generation, reducing redundant computations (Ge et al. 2024). Building on this foundation, ChunkAttention (Ye et al. 2024) enhances self-attention speed using fragmentation, shared KV caches, and batched attention to reduce memory operations. FastGen (Ge et al. 2024) enables adaptive KV cache construction through lightweight attention analysis. Obviously, in UHD image processing, adopting chunk-cache techniques is crucial for accelerating speed while ensuring accuracy. Additionally, it is worth noting that KV cache memory consumption increases rapidly with model size and generation length, placing a significant burden on device memory. Multiple studies (Yang et al. 2024; Li et al. 2024; Liu et al. 2024) aim to optimize memory usage and computational efficiency of LLMs, which is crucial for resource-constrained environments and real-time applications. Therefore, we develop a new machine learning paradigm based on the chunk-cache-quantization pipeline, i.e. infinite pixel learning.

Specifically, in the Dimensional Attention Enhancement Module (DAEM), we first chunk the input data along the channel, width, and height dimensions, and then apply the cyclic scanner to extract and enhance important details in each dimension. Second, we develop an attention cache technique, which is similar to a KV cache for infinite data stream processing. The attention cache stores the extracted features to prevent redundant calculations and accelerate inference. Finally, we employ quantization compression to reduce the storage burden of the cache on the device. Next, we propose the Dimensional Rolling Transformation Module (DRTM) to address the loss of overall features during the cyclic scanner process. Inspired by, yet distinct from, MLP-Mixer (Tolstikhin et al. 2021), DRTM encodes the feature maps of the image from the perspective of channel, width, and height. By arranging feature maps from different views, DRTM effectively models long-range dependencies. Furthermore, we provide a UHD benchmark to evaluate the effectiveness of our method.

Our main contributions are summarized as follows:

- We develop a novel learning paradigm named **Infinite Pixel Learning (IPL)**, which processes infinite continuous data streams using three key components: slice cyclic scanner, attention cache technique, and quantization compression. IPL enables full-resolution UHD image inference on a single consumer-grade GPU while effectively avoiding ghosting artifacts.
- We design an efficient network that contains two main modules: one for extracting the local features of the image and the other for capturing the global features of the image through dimensional scrolling transformation.
- We introduce 4K-DMEF, the first UHD MEF benchmark dataset for dynamic scenes. Our method, tested on both this UHD dataset and other non-UHD datasets, achieves a balance between performance and efficiency.

## Related Works

**Dynamic Multi-Exposure Image Fusion.** Multi-exposure image fusion technology tackles detail loss in highlights and shadows of single-exposure images, preserving detail across all exposures. However, ghosting artifacts from moving objects or camera misalignment remains a key challenge in dynamic scenes (Woo, Ryu, and Kim 2021; Tan et al. 2023). To avoid ghosting artifacts, AHDRNet (Yan et al. 2019) uses the attention module to guide merging according to the reference image. FSHDR (Prabhakar et al. 2021) proposed pioneering work to achieve efficient ghosting removal using zero-shot and few-shot learning strategies. HDR-Transformer (Zhen Liu and Liu 2022) and SCTNet (Tel et al. 2023) utilize transformers to capture key factors for ghosting removal. Although several of the above studies have made significant progress, they have primarily focused on low-resolution images. When applied to UHD images, these methods often encounter memory overflow issues and are unable to perform full-resolution inference on a single consumer-grade GPU.

**Ultra-High-Definition Image Processing.** In recent years, advancements in sensors and displays have rapidly increased the popularity of UHD images. Consequently, many advanced methods have been developed to handle UHD images on consumer GPUs. Existing UHD image processing methods are primarily employed in various applications such as dehazing (Zheng et al. 2021), deblurring (Deng et al. 2021), deraining (Chen et al. 2024; Xiao et al. 2023) and low-light enhancement (Li et al. 2023; Wang et al. 2023). These methods are designed to improve image quality in specific challenging conditions. Additionally, UHDFormer (Wang et al. 2024) is the first universal UHD image restoration Transformer, advancing the development of UHD image processing technology. However, most methods address processing a single 4K ($3840\times2160$) or higher resolution image on a consumer GPU, which is often inefficient for handling multiple heterogeneous UHD images. Therefore, we focus on fusing UHD dynamic multi-exposure images on a single consumer GPU, marking the first approach to address this task specifically for UHD images.

## Methodology

### Overall Architecture

An overview of IPL is shown in Figure 2. Given three images with different exposures $\mathbf{X_i} \in \mathbb{R}^{3\times \mathrm{W}\times \mathrm{H}}$, $\mathbf{i}\in \{1,2,3\}$, we first concatenate them and map them to the feature space through a DownSampler to obtain the low-level features $\mathbf{F_0} \in \mathbb{R}^{\mathrm{C}\times \mathrm{W}\times \mathrm{H}}$, where C, W, and H represent channel, width, and height, respectively. Then, the multiple stacked Feature Integration Blocks (FIBs) are used to generate finer deep features $\mathbf{F_i}$ from $\mathbf{F_0}$ for image fusion, where a Feature Integration Block consists of a Dimensional Attention Enhancement Module (DAEM), and a Dimensional Rolling

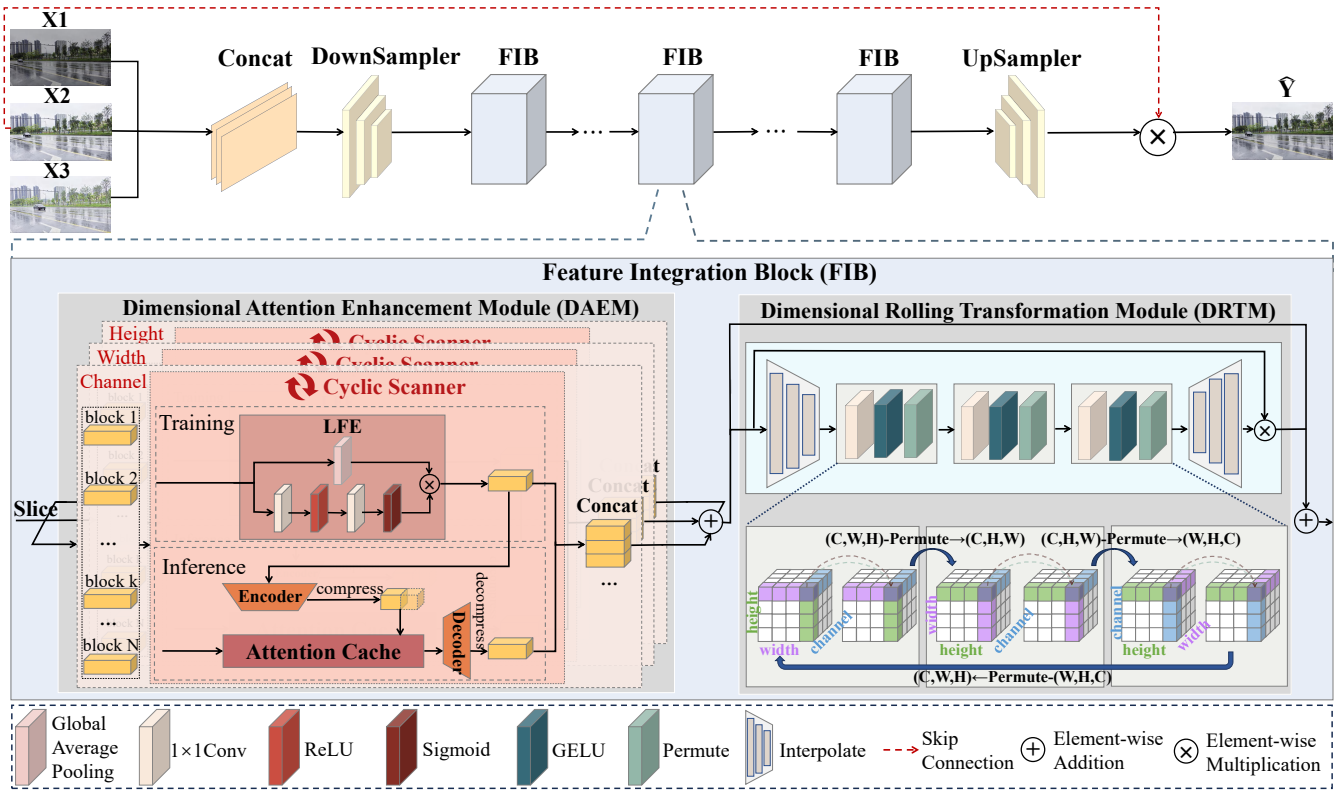


Figure 2: The overall architecture of IPL, which extracts features using a series of Feature Integration Blocks (FIBs). The FIB mainly contains a Dimensional Attention Enhancement Module (DAEM) and a Dimensional Rolling Transformation Module (DRTM). DAEM has three key components: Slice Cyclic Scanner, Attention Cache Technique, and Quantization Compression, forming a chunk-cache-quantization pipeline to process infinite input pixels efficiently. DRTM associates feature from different views by permuting feature maps to compensate for global features.

Transformation Module (DRTM). Finally, the sum of the final features $\mathbf{F_f}$ is fed to an UpSampler and then multiplied with $\mathbf{X_2}$ to obtain the fused image $\hat{\mathbf{Y}}$. DownSampler and UpSampler are both composed of a sub-pixel convolutional layer (He et al. 2016) and a $3\times$ 3 convolutional layer.

## Dimensional Attention Enhancement Module

DAEM aims to capture the local features of images. Inspired by LLMs for processing extremely long sequential inputs, we develop a new machine learning paradigm based on the chunk-cache-quantization pipeline, i.e. infinite pixel learning. Specifically, DAEM includes a Slice Cyclic Scanner, Attention Cache Technique, and Quantization Compression.

**Slice Cyclic Scanner.** To alleviate the load on the model when processing the data stream, we follow ChunkAttention (Ye et al. 2024) and chunk the input data along the three dimensions: channel, width, and height. Next, to capture the local features, we apply the cyclic scanner to the blocks in each dimension. The outputs of the three dimensions are added to produce the final output of the DAEM. Within the cyclic scanner, we build the local feature extractor (LFE). Specifically, in LFE, we simultaneously feed the input blocks into both global average pooling and a series of convolutions to extract deep features. These deep features are then processed through a sigmoid function to generate attention weights for the input features. Subsequently, the input features are then adjusted according to these weights via element-wise multiplication, enabling dynamic channel importance adjustment. The circulation continues until all blocks have been feature extracted by LFE. LFE can be described by the following equations:

$$\mathbf{F_m} = \mathrm{GAP}(\mathbf{F_{in}}), \tag{1}$$

$$\mathbf{F_w} = \mathrm{Sigmoid}(\mathrm{Conv}(\mathrm{RELU}(\mathrm{Conv}(\mathbf{F_{in}})))), \tag{2}$$

$$\mathbf{F_{LFE}} = \mathbf{F_m} \odot \mathbf{F_w}, \tag{3}$$

where $\mathbf{F_{in}}$ is the input features, $\mathbf{F_m}$ is the averaged features, $\mathbf{F_w}$ is the channel weights, and $\mathbf{F_{LFE}}$ indicates the final output features of LFE. $\mathrm{GAP}(\cdot)$ denotes global average pooling, $\mathrm{Sigmoid}(\cdot)$ and $\mathrm{ReLU}(\cdot)$ are the Sigmoid and ReLU functions, respectively, $\mathrm{Conv}(\cdot)$ denotes a $1 \times 1$ convolution, and $\odot$ represents the element-wise product.

**Attention Cache Technique.** To boost the inference speed of the model, LLMs usually employ the KV cache mechanism to trade storage space for reduced inference time (Ge et al. 2024). Inspired by this approach, we developed a similar technique called attention cache. In LFE, deep features

are processed through a sigmoid function to generate attention weights for the input features. Consequently, the attention cache is used to cache local features extracted by LFE, which helps to avoid redundant calculations and accelerates inference. The technique can be expressed as:

$$\mathbf{F_k} = \begin{cases} \mathrm{CR}(\mathbf{k}); & \text{if } \mathrm{LFE}(\mathbf{k}) \text{ in AC}, \\ \mathrm{LFE}(\mathbf{k});\ \mathrm{CW}(\mathbf{k}); & \text{otherwise}, \end{cases} \tag{4}$$

where $\mathbf{F_k}$ denotes the final local features of the block $\mathbf{k}$, AC is the attention cache, and $\mathrm{LFE}(\cdot)$ represents LFE. $\mathrm{CR}(\cdot)$ and $\mathrm{CW}(\cdot)$ denote the read and write operations in the attention cache, respectively. The write operation will write both the $\mathbf{k}$ value and corresponding $\mathrm{LFE}(\mathbf{k})$ into the attention cache.

During training, the normal computational intermediate results (feature maps) of the network are computed through convolutional operations. During inference, the convolutional operation is stored as a cache, which is read directly without a computational process, so the whole network can achieve accelerated inference.

**Quantization Compression.** With model training and inference proceeding, the memory consumption of the attention cache increases rapidly, significantly intensifying the burden of device memory. To alleviate this problem, we use quantization compression to reduce storage memory.

Specifically, we build an encoder to compress the float tensor into the quantized tensor, for a float tensor $\mathbf{t}$ the process can be written as:

$$\mathbf{s} = \frac{\mathbf{t_{max}} - \mathbf{t_{min}}}{\mathbf{q_{max}} - \mathbf{q_{min}}}, \tag{5}$$

$$\mathbf{zp} = \mathrm{MAX}(\mathbf{q_{min}}, \mathrm{MIN}(\mathbf{q_{max}}, \mathrm{Round}(\mathbf{q_{min}} - \frac{\mathbf{t_{min}}}{\mathbf{s}}))), \tag{6}$$

$$\mathbf{t_q} = \mathrm{MAX}(\mathbf{q_{min}}, \mathrm{MIN}(\mathbf{q_{max}}, \mathrm{Round}(\frac{\mathbf{t}}{\mathbf{s} + \mathbf{zp}}))), \tag{7}$$

where $\mathbf{s}$ is the scaling factor, $\mathbf{zp}$ is the zero point, and $\mathbf{t_q}$ is the quantized tensor. $\mathbf{t_{max}}$ and $\mathbf{t_{min}}$ are the maximum and minimum values in the tensor $\mathbf{t}$, respectively. $\mathbf{q_{max}}$ and $\mathbf{q_{min}}$ denote the maximum and minimum values of the quantization range, respectively. $\mathrm{MAX}(\cdot)$ and $\mathrm{MIN}(\cdot)$ take the maximum and minimum values of the comparison, respectively. $\mathrm{Round}(\cdot)$ converts a floating-point number to the nearest integer.

Correspondingly, we build a decoder to decompress the quantized tensor, the process can be written as:

$$\mathbf{t_d} = \mathbf{s} \times (\mathbf{t_q} - \mathbf{zp}), \tag{8}$$

where $\mathbf{t_d}$ denote dequantized tensor.

Furthermore, the process of extracting local features can be written as:

$$\mathbf{F_k} = \begin{cases} \mathrm{Decoder}(\mathrm{CR}(\mathbf{k})); & \text{if } \mathrm{LFE}(\mathbf{k}) \text{ in AC}, \\ \mathrm{LFE}(\mathbf{k});\ \mathrm{Encoder}(\mathrm{CW}(\mathbf{k})); & \text{otherwise}, \end{cases} \tag{9}$$

where $\mathrm{Encoder}(\cdot)$ denotes quantization compression operation, and $\mathrm{Decoder}(\cdot)$ represents decompression operation.

## Dimensional Rolling Transformation Module

To address the loss of overall features during cyclic scanners, we propose DRTM. Inspired by, yet distinct from, MLP-Mixer (Tolstikhin et al. 2021), DRTM encodes the feature maps of the image from the perspective of channel, width, and height. To capture long-range dependencies of features, we ingeniously establish relationships across both spatial and channel dimensions using simple dimension transformation operations. This approach associates and fuses the information encoded in each feature map through a dimensional rolling transformation operation.

The architecture of the DRTM and details of dimension transformation operations are shown in Figure 2. Specifically, we first adjust the resolution of the input features and then perform some dimension transformation operations on them. Given the input feature $\mathbf{F_{in}}$, this procedure can be written as:

$$\mathbf{F'_t} = \mathrm{IP}(\mathbf{F_{in}}), \tag{10}$$

$$\mathbf{F_t} = \underbrace{\mathrm{Permute}(\mathrm{GELU}(\mathrm{Conv}(\mathbf{F'_t})))}_{\times 3}, \tag{11}$$

where $\mathbf{F'_t}$ and $\mathbf{F_t}$ are intermediate results, $\mathrm{IP}(\cdot)$ corresponds to the interpolation operation, $\mathrm{Conv}(\cdot)$ is a $1 \times 1$ convolution, $\mathrm{GELU}(\cdot)$ represents GELU function, $\mathrm{Permute}(\cdot)$ denotes the dimension transformation operation and $\times 3$ denotes three operations in sequence.

Afterward, we use interpolation to adjust the feature $\mathbf{F_t}$ to its original resolution to estimate the attention map and adaptively modulate the input $\mathbf{F_{in}}$ according to the estimated attention via element-wise product. This process can be written as:

$$\mathbf{F_G} = \mathrm{IP}(\mathbf{F_t}) \odot \mathbf{F_{in}}, \tag{12}$$

where $\mathbf{F_G}$ are the final output features and $\odot$ represents element-wise product.

## Feature Integration Block

In general, our FIB pipeline can be written as:

$$\mathbf{F_{DAEM}} = \mathrm{DAEM}(\hat{\mathbf{F}}_{\mathbf{in}}), \tag{13}$$

$$\mathbf{F_{out}} = \mathbf{F_{DAEM}} + \mathrm{DRTM}(\mathbf{F_{DAEM}}), \tag{14}$$

where $\hat{\mathbf{F}}_{\mathbf{in}}$ is the intermediate features, $\mathbf{F_{DAEM}}$ represents output of DAEM, and $\mathbf{F_{out}}$ denotes the final output features.

## Loss Function

To optimize the weights and biases of the network, we utilize the L1 loss in the RGB color space as the fundamental reconstruction loss. This approach ensures that the network accurately captures and reproduces the fine details and color information of the input images. L1 loss can be written as:

$$\mathrm{L} = \|\mathbf{Y} - \hat{\mathbf{Y}}\|_1, \tag{15}$$

where $\mathbf{Y}$ and $\hat{\mathbf{Y}}$ denote the GT and output, respectively.

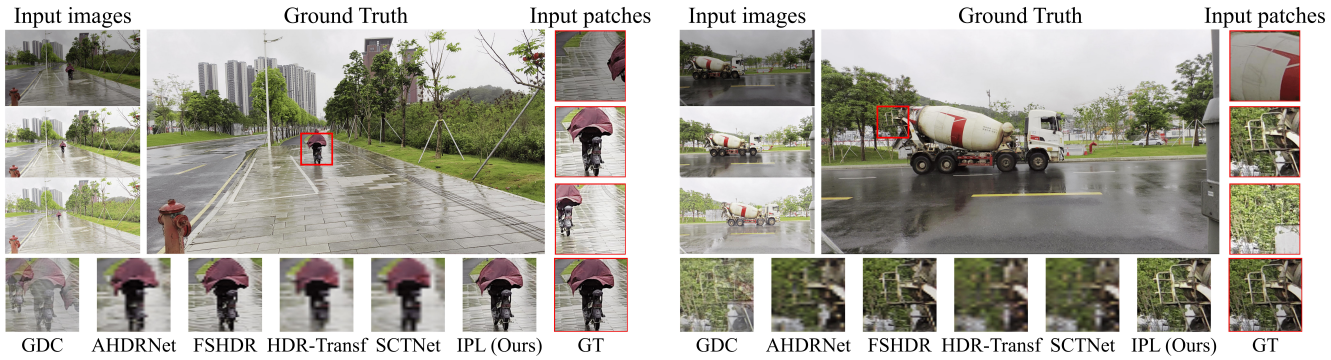


Figure 3: Qualitative comparison on our proposed 4K-DMEF dataset. All methods are trained using our training set on a single RTX 4090 GPU. Our method, IPL, outperforms all SOTA methods in processing UHD multi-exposure image inputs. It effectively avoids ghosting and blurring, achieving full-resolution inference with remarkable clarity and precision.

| Methods | MR | PSNR↑ | SSIM↑ | LPIPS↓ | TIME↓ | MACs↓ | Params↓ |
|---|---|---|---|---|---|---|---|
| GDC (Zhang and Cham 2010) | 3840 × 2160 | 16.84 | 0.6340 | 0.3584 | 221.00s | 242.20M | - |
| AHDRNet (Yan et al. 2019) | 256 × 256 | 20.04 | 0.5012 | 0.4906 | 0.0032s | 162.77M | 1.2398M |
| FSHDR (Prabhakar et al. 2021) | 512 × 512 | 22.85 | 0.6595 | 0.3520 | **0.0017s** | 957.92M | 1.8284M |
| HDR-Transf (Zhen Liu and Liu 2022) | 200 × 200 | 19.67 | 0.4822 | 0.5202 | 0.0512s | 95.94M | 1.1978M |
| SCTNet (Tel et al. 2023) | 200 × 200 | 20.01 | 0.5302 | 0.5204 | 0.0850s | 78.67M | **0.9615M** |
| IPL (Ours) | 256 × 256 | **33.29** | **0.9776** | **0.0427** | 0.0441s | **66.95M** | 0.9747M |

Table 1: Comparison of quantitative results on our 4K-DMEF datasets. MR denotes the maximum resolution each algorithm can handle on a single RTX 4090 GPU. TIME represents the inference time required for each algorithm to fuse a single image. The best and second-best values are highlighted in bold and underlined, respectively.

| Methods | PSNR | SSIM |
|---|---|---|
| GDC (Zhang and Cham 2010) | 27.87 | 0.7286 |
| AHDRNet (Yan et al. 2019) | 42.29 | 0.9882 |
| FSHDR (Prabhakar et al. 2021) | 41.79 | 0.9900 |
| HDR-Transf (Zhen Liu and Liu 2022) | 42.18 | 0.9884 |
| SCTNet (Tel et al. 2023) | 42.49 | 0.9887 |
| IPL (Ours) | 42.13 | 0.9902 |

Table 2: Comparison of quantitative results on Kalantari.

## Experiments

In this section, we evaluate the proposed method by conducting comprehensive experiments on both our UHD dataset and several public non-UHD datasets. We compare our method against five state-of-the-art multi-exposure fusion (MEF) methods of GDC (Zhang and Cham 2010), AHDRNet (Yan et al. 2019), FSHDR (Prabhakar et al. 2021), HDR-Transformer (Zhen Liu and Liu 2022), and SCTNet (Tel et al. 2023). In addition, we conduct ablation studies to show the effectiveness of each module within our network.

### Datasets

**Our UHD Dynamic Multi-Exposure Image Dataset.** To train and evaluate the proposed network as well as the comparison methods, we build a benchmark dataset named 4K-DMEF. Specifically, we record videos at a resolution of 4K (3840 × 2160) with a mobile phone, capturing dynamic elements such as moving people and cars. From these videos, we extract three frames from a video, which we label as samples 1, 2, and 3 of dynamic scenes. To synthesize different exposure levels, we employ the LECARM (Ren et al. 2019) method. Using LECARM, we composite each of the three frames into three different exposure levels: low, medium, and high. For consistency and accuracy in our dataset, we use the original image of sample 2 as the ground truth reference. Ultimately, we collected data for 110 UHD dynamic scenes, dividing them into 80 scenes for training and 30 for testing.

**Non-UHD Dataset.** We also conduct experiments using two public non-UHD dynamic multi-exposure image datasets: Kalantari Dataset (Kalantari and Ramamoorthi 2017) and Mobile-HDR Dataset (Liu et al. 2023). Kalantari Dataset (Kalantari and Ramamoorthi 2017) is a conventional benchmark widely used in previous works (Yan et al. 2019; Prabhakar et al. 2021; Zhen Liu and Liu 2022; Tel et al. 2023). It has a resolution of 1500 × 1000 pixels and includes 74 training scenes and 15 test scenes. Mobile-HDR Dataset (Liu et al. 2023) is a recent dataset captured using mobile phone cameras. We divide this dataset into 85 training scenes and 30 test scenes, each with a resolution of 2000 × 1500 pixels.

### Implementation Details

We conduct our experiments using PyTorch on a single NVIDIA GeForce RTX 4090 GPU. To optimize the network, we employ the AdamW optimizer with a learning

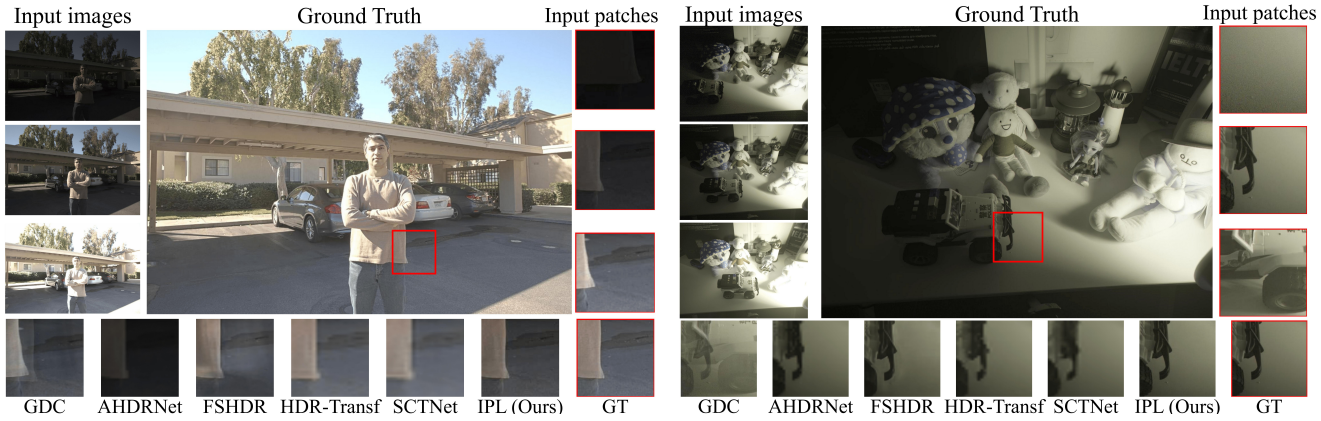


Figure 4: Visual comparison on the public non-UHD dataset. In the first comparison (left) with the Kalantari dataset (Kalantari and Ramamoorthi 2017), our IPL method shows competitive performance. In the second comparison (right) with the Mobile-HDR dataset (Liu et al. 2023), our IPL method excels in detail restoration and performs well in relatively low-light conditions.

| Methods | PSNR | SSIM |
|---|---|---|
| GDC (Zhang and Cham 2010) | 28.02 | 0.6148 |
| AHDRNet (Yan et al. 2019) | 36.64 | 0.9803 |
| FSHDR (Prabhakar et al. 2021) | 32.56 | 0.9706 |
| HDR-Transf (Zhen Liu and Liu 2022) | 36.67 | 0.9809 |
| SCTNet (Tel et al. 2023) | 36.20 | 0.9788 |
| IPL (Ours) | 36.45 | 0.9902 |

Table 3: Comparison of quantitative results on Mobile-HDR.

| Group | DAEM | | | DRTM | PSNR↑ | SSIM↑ | LPIPS↓ |
|---|---|---|---|---|---|---|---|
| | C | W | H | | | | |
| (a) | | | | ✓ | 8.92 | 0.5553 | 0.4190 |
| (b) | ✓ | | | ✓ | 32.96 | 0.9776 | 0.0408 |
| (c) | | ✓ | | ✓ | 34.44 | 0.9776 | 0.0403 |
| (d) | | | ✓ | ✓ | 34.61 | 0.9784 | 0.0404 |
| (e) | ✓ | ✓ | | ✓ | 33.15 | 0.9778 | 0.0404 |
| (f) | ✓ | | ✓ | ✓ | 32.99 | 0.9778 | 0.0412 |
| (g) | | ✓ | ✓ | ✓ | 34.51 | 0.9782 | 0.0401 |
| (h) | ✓ | ✓ | ✓ | | 32.82 | 0.9775 | 0.0489 |
| (i) | ✓ | ✓ | ✓ | ✓ | **34.78** | **0.9805** | **0.0397** |

Table 4: Ablation study on the key components of DEAM and DRTM. The comparison is conducted on our 4K-DMEF.

rate $2 \times 10^{-4}$. The network undergoes training for 1200 epochs with a batch size of 4. Additionally, the number of Feature Integration Blocks (FIBs) is 8, and the number of feature channels is 48. It is important to note that many existing methods are unable to perform full-resolution inference directly on UHD images. Inspired by Zheng (Zheng et al. 2021), for these methods, we first downsample the input UHD images using bilinear interpolation to the maximum resolution that these algorithms can handle on a single GPU. After performing inference, we upsample the output back to 4K resolution.

## Quantitative Results

We employ three well-known image quality assessment metrics, namely PSNR, SSIM (Wang et al. 2004), and

| Attention Cache | PSNR↑ | SSIM↑ | TIME↓ |
|---|---|---|---|
| | 34.78 | 0.9805 | 0.1475s |
| ✓ | 33.29 | 0.9776 | **0.0441s** |

Table 5: Ablation study on the key components of Attention Cache. The comparison is conducted on our 4K-DMEF dataset. TIME denotes the time taken to generate an image.

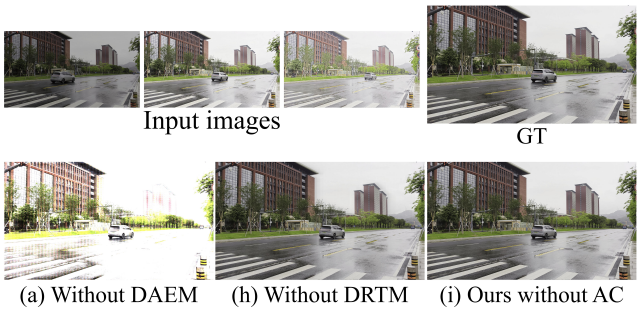


Figure 5: Visualization results of ablation Experiments. As shown, without DAEM, the model suffers severe performance degradation, failing to capture overall features and resulting in a slightly different color tone from the GT.

LPIPS (Zhang et al. 2018), to quantify the performance of different methods. Also, we record the highest resolution that each algorithm can handle on a single RTX 4090 GPU, referred to as Maximum Resolution, and the time it takes to generate a fused image, referred to as Inference Time.

In Table 1, we present the quantitative comparison results of various methods evaluated on our proposed 4K-DMEF dataset. It can be observed that our IPL method achieves approximately 46% and 48% higher PSNR and SSIM, respectively, compared to the second-best method, FSHDR. Notably, a lower LPIPS value indicates better perceptual quality. Our IPL method achieves a significant improvement, with an LPIPS value about 88% better than FSHDR.

| Model | PSNR↑ | SSIM↑ | LPIPS↓ |
|---|---|---|---|
| AHDRNet | 20.04 | 0.5012 | 0.4906 |
| AHDRNet with cache | **22.81** | **0.6583** | **0.3529** |

Table 6: Comparison between the AHDRNet and AHDRNet with the cache.

In terms of inference time, although our method is not the fastest, it still achieves real-time performance at over 40 fps. Tables 2 and 3 show that our IPL achieves performance comparable to the state-of-the-art methods on non-UHD datasets. Combined with the results shown in Figure 4, these findings indicate that our IPL method maintains high-quality results and performs competitively across different resolutions of multi-exposure image fusion.

## Qualitative Results

In addition to the quantitative evaluations, we provide qualitative comparisons to further illustrate the effectiveness of the proposed method. Figure 3 presents the visual comparisons, providing a detailed view of the results produced by our method compared to other MEF approaches.

As shown in Figure 3, the fusion results generated by the traditional GDC (Zhang and Cham 2010) method frequently exhibit ghosting artifacts. Additionally, the fusion results from other neural network-based methods tend to be blurred to varying degrees. This blurring occurs because these methods are unable to directly process UHD images with different exposures on a single GPU, leading to a loss of pixel information during the necessary downsampling process. Obviously, our IPL approach is the only method that can efficiently perform full 4K resolution inference on a single GPU, avoiding common issues such as ghosting and blurring. It effectively fuses UHD dynamic multi-exposure images, overcoming the challenges that other methods typically face.

## Ablation study

We perform three ablation studies to demonstrate the effectiveness of each IPL component, evaluating each in a fair setting. For these experiments, we use the same architecture and hyperparameters, varying only one component in each ablation. The evaluation of these ablation experiments is conducted on our 4K-DMEF dataset.

**Effectiveness of Dimensional Attention Enhancement Module.** Based on the results in Table 4, it is obvious that, without the DAEM, the model experiences the most severe performance degradation (see Figure 5). This aligns with our hypothesis, as the primary mechanisms for handling extremely long pixel input are crucial factors contributing to the outstanding performance of IPL. From the results of group (b) to (h), it is evident that all three dimensions of the slice contain crucial feature information, particularly in the width and height dimensions, which are indispensable for the final image fusion.

**Effectiveness of Dimensional Rolling Transformation Module.** By comparing the results of groups (a), (h), and (i), it is evident that DRTM plays a crucial role in the final fusion outcome. The removal of DRTM leads to a performance decrease of nearly 2 dB in terms of PSNR, highlighting the critical role and effectiveness of our design.

| Metrics | QC | CNN-W | CNN-C |
|---|---|---|---|
| Inference Time | **0.0441s** | 0.0690s | 0.1978s |

Table 7: Comparison of compression methods. QC refers to quantization compression, CNN-W denotes using CNN to compress the width dimension, and CNN-C denotes using CNN to compress the channel dimension.

**Effectiveness of Attention Cache.** Based on the results in Table 5, although performance is slightly reduced due to quantization compression in the attention cache, the inference time improves significantly. Specifically, it decreases from 0.1475 seconds to **0.0441** seconds, the processing time has reduced by 70%, enabling real-time inference.

# Discussion

**Discussion of Chunk-Cache Pipeline.** To investigate the necessity and effectiveness of the chunk-cache pipeline, we conduct it to the comparison method AHDRNet (Yan et al. 2019) and conduct experiments on our proposed 4K-DMEF dataset. Specifically, we divide the input images into blocks of size $64 \times 64$ in both width and height dimensions. We then cache the intermediate results of each chunk, making them readily available for inference when needed. We show the result in Table 6.

**Discussion of Compression Methods.** We explore two compression methods: quantization and CNN compression. As shown in Table 7, quantization reduces memory and computation by converting weights to low-precision integers, achieving an average inference time of 0.0441s per image. For CNN compression, we perform two experiments: width compression (CNN-W) and channel compression (CNN-C). In CNN-W, three $1 \times 1$ convolutional layers with ReLU reduce the width from 64 to 16, while in CNN-C, a single $3 \times 3$ convolutional layer reduces the channels to 1. The average inference times for CNN-W and CNN-C are 0.0690s and 0.1978s, both notably slower than quantization.

# Conclusion

In this paper, we propose an advanced ultra-high-definition dynamic multi-exposure image fusion method via infinite pixel learning. Our approach features a chunk-cache-quantization pipeline that efficiently captures dimensional local features, avoids redundant computation, and accelerates inference on resource-limited devices. We also introduce a dimensional rolling transformation operation to associate and fuse different views of the feature map. Additionally, we provide a new UHD benchmark to evaluate the effectiveness of our method. Quantitative and qualitative results show that our proposed algorithm, can reach real-time (>40fps) for a single UHD image and generate satisfactory visual results on real-world UHD images.

## Acknowledgments

This work was supported by the National Natural Science Foundation of China (No. 62322216, 62172409, 62311530686) and Shenzhen Science and Technology Program (No. KQTD20221101093559018).

## References

Chen, H.; Chen, X.; Wu, C.; Zheng, Z.; Pan, J.; and Fu, X. 2024. Towards Ultra-High-Definition Image Deraining: A Benchmark and An Efficient Method. arXiv:2405.17074.

Deng, S.; Ren, W.; Yan, Y.; Wang, T.; Song, F.; and Cao, X. 2021. Multi-Scale Separable Network for Ultra-High-Definition Video Deblurring. In *ICCV*.

Ge, S.; Zhang, Y.; Liu, L.; Zhang, M.; Han, J.; and Gao, J. 2024. Model Tells You What to Discard: Adaptive KV Cache Compression for LLMs. In *ICLR*.

Han, C.; Wang, Q.; Xiong, W.; Chen, Y.; Ji, H.; and Wang, S. 2024. LM-Infinite: Simple On-the-Fly Length Generalization for Large Language Models. In *ICLR*.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep Residual Learning for Image Recognition. In *CVPR*.

Kalantari, N. K.; and Ramamoorthi, R. 2017. Deep high dynamic range imaging of dynamic scenes. *ACM TOG*, 36: 1 – 12.

Li, C.; Guo, C.-L.; man zhou; Liang, Z.; Zhou, S.; Feng, R.; and Loy, C. C. 2023. Embedding Fourier for Ultra-High-Definition Low-Light Image Enhancement. In *ICLR*.

Li, H.; Ma, K.; Yong, H.; and Zhang, L. 2020. Fast Multi-Scale Structural Patch Decomposition for Multi-Exposure Image Fusion. *IEEE TIP*, 29: 5805–5816.

Li, Y.; Yu, Y.; Liang, C.; He, P.; Karampatziakis, N.; Chen, W.; and Zhao, T. 2024. LoftQ: LoRA-Fine-Tuning-Aware Quantization for Large Language Models. In *ICLR*.

Liu, J.; Gong, R.; Wei, X.; Dong, Z.; Cai, J.; and Zhuang, B. 2024. QLLM: Accurate and Efficient Low-Bitwidth Quantization for Large Language Models. In *ICLR*.

Liu, S.; Zhang, X.; Sun, L.; Liang, Z.; Zeng, H.; and Zhang, L. 2023. Joint HDR Denoising and Fusion: A Real-World Mobile HDR Image Dataset. In *CVPR*.

Ma, K.; and Wang, Z. 2015. Multi-Exposure Image Fusion: A Patch-Wise Approach. In *IEEE International Conference on Image Processing*.

Prabhakar, K. R.; Senthil, G.; Agrawal, S.; Babu, R. V.; and Gorthi, R. K. S. S. 2021. Labeled from Unlabeled: Exploiting Unlabeled Data for Few-shot Deep HDR Deghosting. In *CVPR*.

Ren, Y.; Ying, Z.; Li, T. H.; and Li, G. 2019. LECARM: Low-Light Image Enhancement Using the Camera Response Model. *IEEE Transactions on Circuits and Systems for Video Technology*, 29(4): 968–981.

Tan, X.; Chen, H.; Zhang, R.; Wang, Q.; Kan, Y.; Zheng, J.; Jin, Y.; and Chen, E. 2023. Deep Multi-Exposure Image Fusion for Dynamic Scenes. *IEEE TIP*, 32: 5310–5325.

Tel, S.; Wu, Z.; Zhang, Y.; Heyrman, B.; Demonceaux, C.; Timofte, R.; and Ginhac, D. 2023. Alignment-Free HDR Deghosting with Semantics Consistent Transformer. In *ICCV*.

Tolstikhin, I.; Houlsby, N.; Kolesnikov, A.; Beyer, L.; Zhai, X.; Unterthiner, T.; Yung, J.; Steiner, A.; Keysers, D.; Uszkoreit, J.; Lucic, M.; and Dosovitskiy, A. 2021. MLP-Mixer: An all-MLP Architecture for Vision. In *NeurIPS*.

Wang, C.; Pan, J.; Wang, W.; Fu, G.; Liang, S.; Wang, M.; Wu, X.-M.; and Liu, J. 2024. Correlation Matching Transformation Transformers for UHD Image Restoration. In *AAAI*.

Wang, T.; Zhang, K.; Shen, T.; Luo, W.; Stenger, B.; and Lu, T. 2023. Ultra-High-Definition Low-Light Image Enhancement: A Benchmark and Transformer-Based Method. In *AAAI*.

Wang, Z.; Bovik, A.; Sheikh, H.; and Simoncelli, E. 2004. Image Quality Assessment: from Error Visibility to Structural Similarity. *IEEE TIP*, 13(4): 600–612.

Woo, S.-M.; Ryu, J.-H.; and Kim, J.-O. 2021. Ghost-Free Deep High-Dynamic-Range Imaging Using Focus Pixels for Complex Motion Scenes. *IEEE TIP*, 30: 5001–5016.

Xiao, J.; Fu, X.; Liu, A.; Wu, F.; and Zha, Z.-J. 2023. Image De-Raining Transformer. *IEEE TPAMI*, 45(11): 12978–12995.

Xu, H.; Ma, J.; and Zhang, X.-P. 2020. MEF-GAN: Multi-Exposure Image Fusion via Generative Adversarial Networks. *IEEE TIP*, 29: 7203–7216.

Yan, Q.; Gong, D.; Shi, Q.; van den Hengel, A.; Shen, C.; Reid, I.; and Zhang, Y. 2019. Attention-Guided Network for Ghost-Free High Dynamic Range Imaging. In *CVPR*.

Yang, D.; Han, X.; Gao, Y.; Hu, Y.; Zhang, S.; and Zhao, H. 2024. PyramidInfer: Pyramid KV Cache Compression for High-throughput LLM Inference. In *ACL ARR*.

Ye, L.; Tao, Z.; Huang, Y.; and Li, Y. 2024. ChunkAttention: Efficient Attention on KV Cache with Chunking Sharing and Batching. In *ICLR*.

Zhang, R.; Isola, P.; Efros, A. A.; Shechtman, E.; and Wang, O. 2018. The Unreasonable Effectiveness of Deep Features as a Perceptual Metric. In *CVPR*.

Zhang, W.; and Cham, W.-K. 2010. Gradient-Directed Composition of Multi-Exposure Images. In *CVPR*.

Zhen Liu, B. Z., Yinglong Wang; and Liu, S. 2022. Ghost-Free High Dynamic Range Imaging with Context-Aware Transformer. In *ECCV*.

Zheng, Z.; Ren, W.; Cao, X.; Hu, X.; Wang, T.; Song, F.; and Jia, X. 2021. Ultra-High-Definition Image Dehazing via Multi-Guided Bilateral Learning. In *CVPR*.