Applied Intelligence (2025) 55:945
https://doi.org/10.1007/s10489-025-06785-8

Hybrid CNN-RWKV with high-frequency enhancement for real-world chinese-english scene text image super-resolution

Yanbin Liu[1] · Yu Zhu[1,2] · Hangyu Li[1] · Xiaofeng Ling[1]

Accepted: 13 July 2025 / Published online: 30 August 2025

Abstract

Existing scene text image super-resolution (STISR) methods primarily focus on the restoration of fixed-size English text images. Compared to English characters, Chinese characters present a greater variety of categories and more intricate stroke structures. In recent years, Transformer-based methods have achieved significant progress in image super-resolution task, but face the dilemma between global modeling and efficient computation. The emerging Receptance Weighted Key Value (RWKV) model can serve as a promising alternative to Transformer, enabling long-distance modeling with linear computational complexity. In this paper, we propose a Hybrid CNN-RWKV with High-Frequency Enhancement (HCR-HFE) model for STISR task. First, we design a recurrent bidirectional WKV (Re-Bi-WKV) attention which integrates bidirectional WKV (Bi-WKV) attention with a recurrent mechanism. Bi-WKV achieves global receptive field with linear complexity, while the recurrent mechanism establishes 2D image dependencies from different scanning directions. Additionally, a computationally efficient high-frequency enhancement module (HFEM) is incorporated to enhance high-frequency details, such as character edge information. Furthermore, we design a multi-scale large kernel convolutional (MLKC) block which integrates large kernel decomposition, gated aggregation and multi-scale mechanism to capture various-range dependencies with reduced computational cost. Finally, we introduce a multi-frequency channel attention (MFCA) which extends channel attention to the frequency domain, enabling the model to focus on critical features. Extensive experiments on real-world Chinese-English (Real-CE) dataset demonstrate that HCR-HFE outperforms previous methods in both quantitative metrics and visual results. Furthermore, HCR-HFE achieves excellent performance on natural image datasets, demonstrating its broad applicability.

Keywords Text image super-resolution · Recurrent bi-directional WKV attention · Multi-scale large kernel convolution · High-frequency enhancement · Multi-frequency channel attention

1 Introduction

Single image super-resolution (SISR) aims to reconstruct a high-resolution (HR) image from a degraded low-resolution (LR) input. Unlike general SISR methods, scene text image super-resolution (STISR) primarily focuses on text images, aiming to improve text recognition accuracy by reconstructing clear text content. Early STISR methods, which are primarily trained on synthetic datasets, exhibit limited generalization capability when applied to real-world scenarios.

To address this problem, Wang et al. [39] propose the first real-world text image dataset, termed TextZoom, which greatly facilitates the research of STISR. However, TextZoom only contains fixed-size English text image pairs (16×64, 32×128), where English character set is limited to alphanumeric symbols ('0' to '9', 'a' to 'z') as well as simple stroke structures. Consequently, models trained on TextZoom struggle to generalize to complex character systems, such as Chinese characters. To overcome this limitation, Ma et al. [28] build a novel Chinese-English text image dataset, namely Real-CE, which can support both the $2\times$ and $4\times$ STISR tasks. Compared to English characters,

✉ Yu Zhu
zhuyu@ecust.edu.cn

[1] School of Information Science and Engineering, East China University of Science and Technology, Shanghai 200237, China

[2] Shanghai Engineering Research Center of Internet of Things for Respiratory Medicine, Shanghai 200032, China

Chinese characters possess a greater variety of categories and intricate stroke structures. Many Chinese characters share similar glyph shapes, where even a slight deviation in strokes leads to completely different meanings.

Existing SR networks are primarily CNN-based [23, 49], Transformer-based [5, 20, 45], and Mamba-based [12, 41]. Due to the limitation of convolutional kernel size, CNN has a limited effective receptive field (ERF). Vision Transformer adopts standard self-attention (SA) mechanism to model long-range dependencies, but suffers from quadratic computational complexity. Consequently, Transformer-based models usually adopt local window attention [25], restricting the self-attention calculation within local windows. However, window attention comes at the expense of global receptive field, failing to essentially overcome the dilemma between global modeling and efficient computation. As an efficient alternative to Transformer, Mamba [11] utilizes state space models (SSMs) to capture long-range dependencies with linear computational complexity. Nevertheless, due to unidirectional sequence modeling property, SSM has a causal receptive field, which extends from the first token to the current token.

Receptance Weighted Key Value (RWKV) [32] model integrates the strengths of efficient parallel training from Transformer and efficient inference from RNN, enabling long-distance modeling with linear computational complexity. RWKV introduces two major improvements: (1) It proposes a WKV attention mechanism to achieve a causal receptive field with linear complexity. (2) It introduces a token shift layer to capture local context through the interaction between the current token and the previous one.

Given the modal gap between text and image, Vision-RWKV is proposed, applying the RWKV architecture to vision tasks. Vision-RWKV [8] enhances the vanilla RWKV in two ways: (1) It introduces bidirectional WKV attention as an alternative for the original causal WKV attention, thereby enabling a global receptive field. (2) It proposes a quad-directional token shift method, where the current token interacts with four tokens (up, down, left, and right). Vision-RWKV achieves comparable performance with ViT [7] but requires less computation, eliminating the necessity for window partition operation in ViT.

In this paper, we propose a Hybrid CNN-RWKV with High-Frequency Enhancement (HCR-HFE) model, which is the first work to apply RWKV architecture for STISR task. As the core component, the recurrent bidirectional WKV (Re-Bi-WKV) attention achieves a global ERF with linear computational complexity, effectively modeling 2D image dependencies from different scanning directions. Since the global features extracted by Re-Bi-WKV mainly contain low-frequency information, we design a parallel branch to enhance the high-frequency details. Inspired by the success of CNN-Transformer hybrid models [40, 46], we introduce the CNN-RWKV hybrid architecture, designing an additional multi-scale large kernel convolution block (MLKCB). To focus on learning critical features, we introduce a multi-frequency channel attention (MFCA), which generalizes channel attention to the frequency domain.

The main contributions of this paper are summarized as follows:

1) We propose a method termed HCR-HFE, which achieves superior performance on both Real-CE and general SR datasets. As the core component, Re-Bi-WKV achieves a global receptive field with linear complexity and introduces a recurrent attention mechanism that combines different scanning directions to effectively model 2D image dependencies.

2) We introduce a lightweight high-frequency enhancement module that significantly reduces computational overhead compared to Fourier transform-based approaches while maintaining seamless network integration.

3) MLKCB integrates large kernel decomposition, gated aggregation and multi-scale mechanism to capture various-range dependencies with reduced computational complexity, demonstrating particular effectiveness in suppressing blocking artifacts.

4) MFCA innovatively replaces standard global average pooling (GAP) operation with a discrete cosine transform (DCT)-based spectral analysis, leveraging multiple frequency components to preserve richer structural information.

2 Related work

2.1 Single Image Super Resolution (SISR)

CNN-based The pioneer work SRCNN [6] adopts three convolutional layers to learn the mappings from LR images to HR images. VDSR [18] introduces the global residual connection to void gradient vanishing. RCAN [50] introduces the channel attention mechanism to enhance the feature representation capability. Due to the limited size of convolutional kernels, CNN-based methods suffer from restricted receptive fields and are unable to effectively model global dependencies.

Transformer-based SwinIR [22] applies Swin Transformer [25] for image restoration task. DAT [5] alternately utilizes spatial self-attention and channel self-attention to aggregate global features. CRAFT [20] demonstrates that Transformer is adept at capturing low-frequency information but has limited capability in extracting high-frequency features. ART [45] alternately performs dense window attention and sparse

window attention, respectively for extracting local features and establishing global dependencies. HAT [4] combines window-based self-attention, channel attention, and overlapping cross-attention for SR task. CFAT [33] improves upon HAT by introducing additional triangular window attention to eliminate the boundary distortion problem. Despite their advancements, the above methods rely on window-based attention, which enhances computational efficiency at the cost of a global receptive field, failing to fundamentally address the trade-off between performance and efficiency.

Mamba-based MambaIR [13] is the first work to adapt state space model (SSM) for low-level image restoration task. DVMSR [19] incorporates Vision Mamba [53] and distillation strategy to effectively reduce computational complexity. FMSR [41] introduces a frequency selection module(FSM) into Mamba block, adaptively selecting the most critical frequency information through Fourier Transform. IRSRMamba [17] effectively captures global and local information by combining SSM with wavelet transform. However, due to the unidirectional sequence modeling nature of SSM, Mamba-based methods still face limitations in achieving excellent SR results.

2.2 Scene Text Image Super Resolution (STISR)

Wang et al. [39] propose the first real-world text image SR dataset, termed TextZoom. Additionally, a new model termed TSRN is proposed which incorporates a central alignment module and boundary-aware loss. TSRGAN [9] applies the generative adversarial architecture to the STISR task. TPGSR [27] employs a text recognition network to generate a categorical probability sequence as text prior (TP) which can guide the SR reconstruction process. TATT [26] designs a Transformer-based module termed TP Interpreter to fuse text prior and image features. PerMR [36] designs a dual-branch architecture, perceiving multiple representations from text recognition branch to facilitate SR reconstruction. DPMN [54] employs dual image-level priors, including text mask and graphic recognition result, to effectively extract both structural information and semantic information. MARCONet [21] employs a StyleGAN to generate corresponding structural priors for each character, which are then used to assist in text reconstruction. TCDM [31] adopts a text-conditional diffusion model to provide excellent text-to-image synthesis capability for the STISR task. Existing STISR methods are primarily designed for English text images, struggling to reconstruct structurally complex characters like Chinese characters. In this paper, we propose a novel architecture for real-world Chinese-English STISR task, achieving superior performance on the Real-CE dataset.

2.3 RWKV-based models

Due to the linear complexity and fast inference speed, RWKV-based models have been applied to various vision tasks. RWKV-SAM [43] adopts a hybrid CNN-RWKV architecture to achieve efficient image segmentation. Diffusion-RWKV [10] combines RWKV architecture with diffusion model for image generation task. Restore-RWKV [42] pioneers the adaptation of the RWKV architecture for efficient medical image restoration. PointRWKV [15] combines RWKV model with multi-scale framework for 3D point cloud learning task. Despite these advancements, the application of RWKV to SR task remains unexplored. To bridge this gap, we propose a hybrid CNN-RWKV model for Chinese-English STISR task, which effectively captures global dependencies with linear computational complexity while enhancing crucial high-frequency details.

3 The proposed method

3.1 Overall architecture

As shown in Fig. 1, the input is a 4-channel image, which is a concatenation of RGB image $I_{LR} \in \mathbb{R}^{H\times W\times 3}$ and its corresponding Canny edge map $M_{LR} \in \mathbb{R}^{H\times W\times 1}$. Canny edge map can be regarded as a prior label, enabling the SR network to distinguish between foreground and background, thus focusing more on the reconstruction of foreground region. The overall architecture is mainly divided into four parts, including shallow feature extraction, deep feature extraction, multi-layer feature fusion and image reconstruction.

Given the 4-channel input $I_{in} \in \mathbb{R}^{H\times W\times 4}$, the first step is to apply a 3×3 convolutional layer to extract shallow feature $F^0 \in \mathbb{R}^{H\times W\times C}$. Then, a series of residual groups (RGs) are stacked to gradually extract deep features. Specifically, each RG consists of a multi-scale large kernel convolution block (MLKCB), several Receptance Weighted Key Value blocks(RWKVBs) a multi-frequency channel attention (MFCA) and a 3×3 convolutional layer. Both MLKCB and RWKVB follow the 'LayerNorm $\rightarrow$ TokenMixer $\rightarrow$ LayerNorm $\rightarrow$ ChannelMixer' design style with two skip connections. Additionally, we introduce learnable scaling factors (denoted as w_1, w_2, s_1, s_2) that enhance the adaptability of residual connection by dynamically scaling the original input features. In the multi-layer feature fusion stage, the output features of each RG are concatenated along the channel dimension, followed by a 1×1 convolutional layer to restore the original number of channels. MFCA is then applied to adaptively adjust the importance

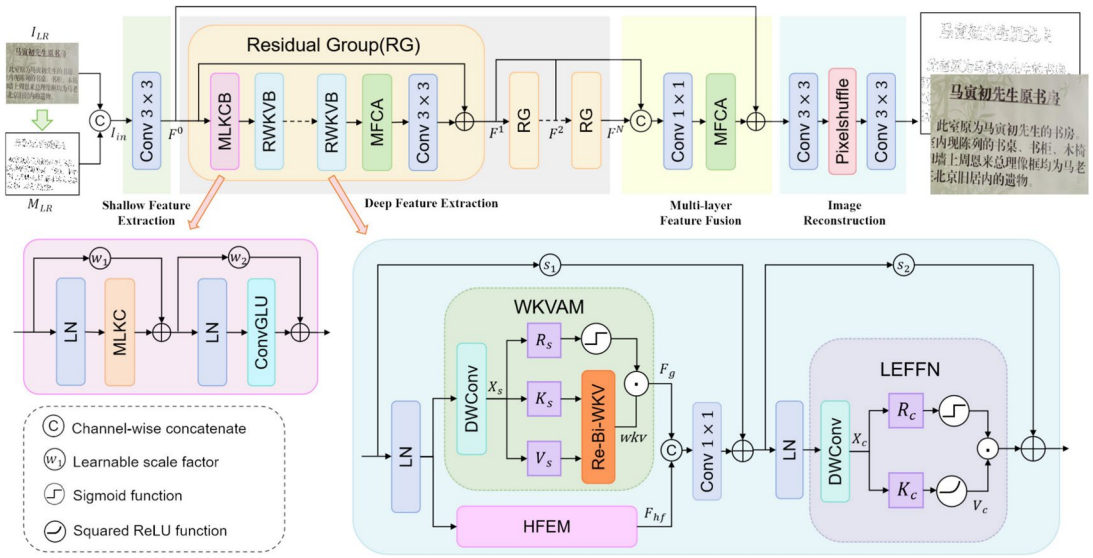


Fig. 1 The architecture of HCR-HFE

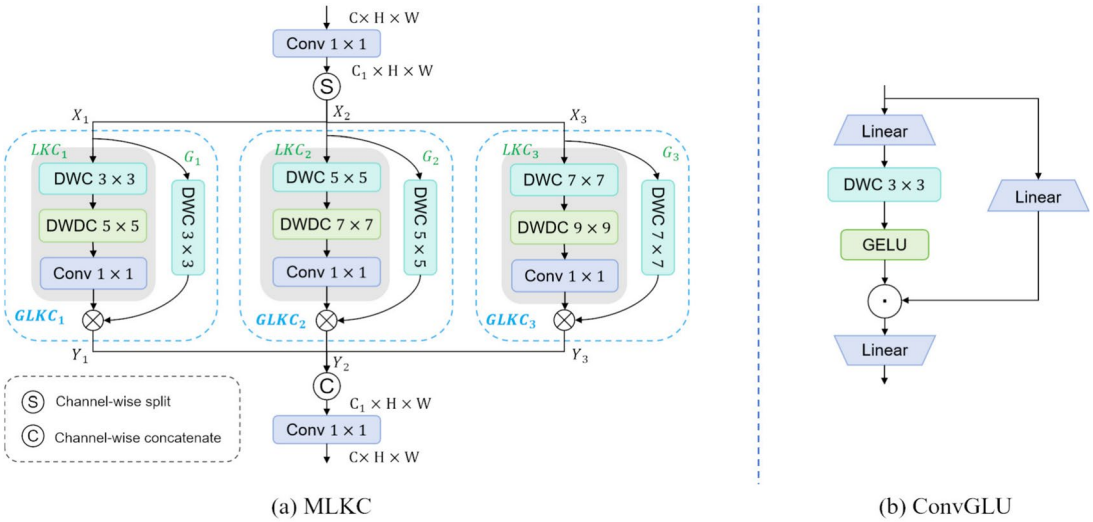


Fig. 2 The diagram of spatial mixer and channel mixer in MLKCB. (a) As the spatial mixer, MLKC integrates large kernel decomposition, gated aggregation and multi-scale mechanism. (b) As the channel mixer, ConvGLU introduces a 3×3 depth-wise convolution on top of gated linear unit (GLU) [34]

of each channel, and a global residual connection is adopted to fuse shallow feature F^0. For the reconstruction module, we upsample the fused feature to the target size through a pixel-shuffle layer. Additionally, a pair of convolutional layers are employed to aggregate features before and after the upsampling operation.

3.2 Multi-scale large kernel convolution block

As depicted in Fig. 1, MLKCB employs multi-scale large kernel convolution(MLKC) as token mixer and convolution gated linear unit (ConvGLU) as channel mixer. The designs of MLKC and ConvGLU are shown in Fig. 2. MLKC integrates large kernel decomposition, gated aggregation and multi-scale mechanism which will be introduced in detail below. ConvGLU introduces a 3×3 depth-wise convolution on top of gated linear unit (GLU) [34], which has been proven to outperform multi-layer perceptron(MLP) in various computer vision tasks.

Large kernel decomposition VAN [14] explores kernel decomposition strategy and proves that a large kernel convolution (*LKC*) can be decomposed into 3 parts: a $(2d-1) \times (2d-1)$ depth-wise convolution(*DWC*), $\lceil \frac{K}{d} \times \frac{K}{d} \rceil$ depth-wise dilation convolution(*DWDC*), and a point-wise convolution($Conv_{1\times1}$), where d denotes the

dilation rate. With the large kernel decomposition strategy, we can capture long-range dependencies with lower computational cost. The decomposition process is shown as follows:

$$LKC(X) = Conv_{1\times1}(DWDC(DWC(X))) \tag{1}$$

Gated aggregation During the large kernel decomposition process, depth-wise dilation convolution may introduce blocking artifacts for SR task. Based on this, we introduce a parallel gated branch to dynamically adjust the output of LKC. The Gated Large Kernel Convolution (GLKC) operation is formally defined as:

$$GLKC(X) = LKC(X) \otimes G(X) \tag{2}$$

where $G(\cdot)$ denotes the gated branch implemented via a depth-wise convolution operation, $LKC(\cdot)$ corresponds to the large kernel convolution branch, and $\otimes$ represents element-wise multiplication. The LKC branch has a large receptive field which is used to establish long-distance dependencies, while the gated branch can well preserve fine-grained local details. Multiplying the outputs of the two branches can effectively suppress blocking artifacts.

Multi-scale mechanism To aggregate multi-scale features, we introduce three parallel GLKC branches. Each branch employs a large kernel decomposition of a specific size:

$$(1)\quad LKC_1 : K = 9, d = 2 \rightarrow 3\times3DWC + 5\times5DWDC + 1\times1Conv$$
$$(2)\quad LKC_2 : K = 21, d = 3 \rightarrow 5\times5DWC + 7\times7DWDC + 1\times1Conv$$
$$(3)\quad LKC_3 : K = 35, d = 4 \rightarrow 7\times7DWC + 9\times9DWDC + 1\times1Conv$$

Figure 2(a) shows the structure of MLKC. Firstly, a 1×1 convolution is employed to adjust the channel dimension ($C \rightarrow C_1$). Then, split the input along the channel dimension to obtain X_1, X_2, X_3, which are respectively sent into the parallel GLKC branches to get Y_1, Y_2, Y_3. Subsequently, the three outputs are concatenated along the channel dimension, and another 1×1 convolutional layer is adopted to restore the original number of channels. The specific process is shown as follows:

$$\begin{gathered} X_1, X_2, X_3 = Split(Conv_{1\times1}^{C\rightarrow C_1}(X_{in})) \\ Y_1 = GLKC_1(X_1),\ \ Y_2 = GLKC_2(X_2),\ \ Y_3 = GLKC_3(X_3) \\ Y_{out} = Conv_{1\times1}^{C_1\rightarrow C}(Concat(Y_1, Y_2, Y_3)) \end{gathered} \tag{3}$$

where $GLKC_1(\cdot), GLKC_2(\cdot), GLKC_3(\cdot)$ represent three parallel GLKC branches, while $Split(\cdot)$ and $Concat(\cdot)$ respectively denote channel-wise splitting and concatenation operations.

3.3 RWKV block

3.3.1 Overall block design

As shown in Fig. 1, RWKVB designs a parallel dual-branch structure as token mixer, where WKV attention module (WKVAM) is utilized for extracting global features, and high-frequency enhancement module (HFEM) is employed to enhance high-frequency details. Then, channel concatenation is used to fuse the global feature $F_g \in \mathbb{R}^{H\times W\times C}$ extracted by WKVAM and high-frequency feature $F_{hf} \in \mathbb{R}^{H\times W\times C}$ extracted by HFEM. Following, the channel dimension of the fused feature is reduced from $2C$ to C through a 1×1 convolutional layer. Finally, local-enhancement feed-forward network (LEFFN) is employed to facilitate channel-wise feature fusion.

WKVAM and LEFFN respectively follow the designs of spatial mix module and channel mix module in Vision-RWKV [8]. As depicted in Fig. 3(a), both the uni-directional token shift layer (Uni-shift) in original RWKV [32] and the quad-directional token shift layer (Quad-shift) in Vision-RWKV [8] only perform token shifting operation from limited directions by simple linear interpolation. As we know, the locality of 2D images is inherently omni-directional, where adjacent tokens from all directions within a neighboring area are all relevant. Based on this observation, we employ a omni-directional convolution (Omin-Conv) in the first step of WKVAM and LEFFN to fuse adjacent tokens from all directions.

WKV attention module The input X_{in} first passes through a 3×3 depth-wise convolution (DWConv) to enhance local feature:

$$X_s = DWConv_{3\times3}(X_{in}) \tag{4}$$

Then, three parallel linear layers are applied to obtain receptance $R_s \in \mathbb{R}^{HW\times C}$, key $K_s \in \mathbb{R}^{HW\times C}$ and value $V_s \in \mathbb{R}^{HW\times C}$:

$$R_s = X_s W_R,\ \ K_s = X_s W_K,\ \ V_s = X_s W_V \tag{5}$$

where W_R, W_K and W_V denote three linear layers.

Subsequently, K_s and V_s are sent into recurrent bidirectional WKV (Re-Bi-WKV) to calculate the global attention result $wkv \in \mathbb{R}^{HW\times C}$. Re-Bi-WKV will be described in Section 3.3.2.

$$wkv = Re\text{-}Bi\text{-}WKV(K_s, V_s) \tag{6}$$

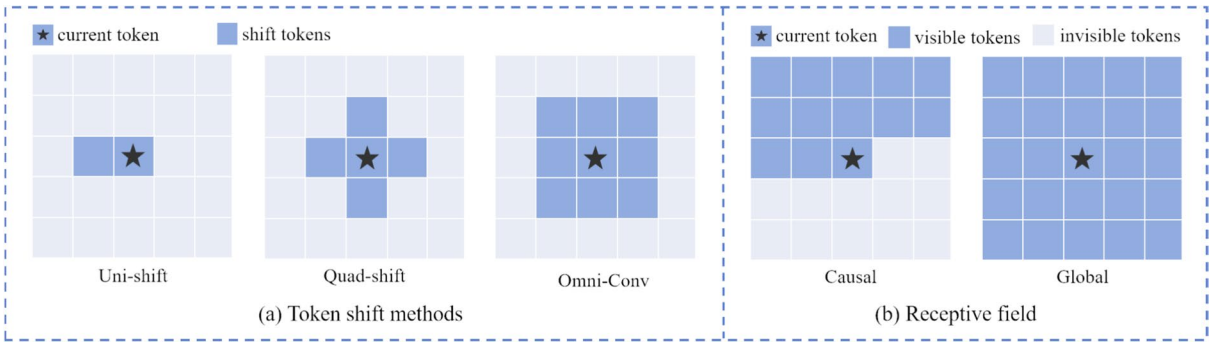


Fig. 3 (a) Illustrations of three token shift methods. Uni-shift captures local context by performing linear interpolation between the current token and its previous one. Quad-shift interacts with four tokens (up, down, left, and right) by linear interpolation. Omni-Conv integrates information from adjacent tokens in all directions. (b) Comparison of causal receptive field and global receptive field. The causal receptive field spans from the first token to the current token. In contrast, the global receptive field means that all tokens are visible

As a gating branch, $\sigma(R_s)$ is element-wise multiplied with *wkv* to adjust the global attention result. Then, a linear layer is introduced to obtain the final output:

$$X_{out} = (\sigma(R_s) \odot wkv)W_o \tag{7}$$

where $\sigma(\cdot)$ denotes the Sigmoid funtion, $\odot$ indicates element-wise multiply, and W_o represents a linear layer.

Local-enhancement feed-forward network Similar to WKVAM, the input feature X_{in} first passes through a 3×3 DWConv to enhance local context:

$$X_c = DWConv_{3\times3}(X_{in}) \tag{8}$$

Then, the receptance R_c and key K_c can be obtained as follows:

$$R_c = X_c W_R, \quad K_c = X_c W_K \tag{9}$$

It's worth noting that V_c is acquired from K_c, rather than directly from X_c:

$$V_c = ReLU^2(K_c)W_v \tag{10}$$

where $ReLU^2(\cdot)$ denotes the squared ReLU activation function with strong nonlinearity.

Notably, the transition process '$X_c \rightarrow K_c \rightarrow V_c$' consists of a linear layer, a squared ReLU activation function and another linear layer, which can effectively facilitate channel-wise feature fusion.

Finally, the same gating operation as in WKVAM is employed to obtain the final output:

$$X_{out} = (\sigma(R_c) \odot V_c)W_o \tag{11}$$

3.3.2 Recurrent bidirectional WKV

As shown in Fig. 4, recurrent bidirectional WKV (Re-Bi-WKV) integrates bidirectional WKV (Bi-WKV) attention and a recurrent attention mechanism. Bi-WKV achieves a global receptive field by performing a weighted summation of all tokens ($0 \rightarrow (T-1)$, i.e., from the first token to the last token). Additionally, the recurrent attention mechanism can effectively establish 2D image dependencies by combining horizontal scanning and vertical scanning.

Bidirectional WKV attention As depicted in Fig. 3(b), original WKV attention [32] achieves a causal receptive field that spans from the first token to the current token, making it suitable for modeling 1D causal sequences in the NLP domain. Bi-WKV attention [8] modifies the upper limit of the original WKV attention from t (the current token) to $T-1$ (the last token) in the summation formula, leading to a global receptive field where all tokens are visible. Given the inputs K_s and V_s, the Bi-WKV attention result of the t-th token (denoted as wkv_t) is computed as follows:

$$wkv_t = Bi\text{-}WKV(K_s, V_s)_t = \frac{\sum_{i=0,i\neq t}^{T-1} e^{-(|t-i|-1)/T\cdot w+k_i} v_i + e^{u+k_t} v_t}{\sum_{i=0,i\neq t}^{T-1} e^{-(|t-i|-1)/T\cdot w+k_i} + e^{u+k_t}} \tag{12}$$

where T represents the total number of tokens ($T = H \times W$). k_i and v_i respectively denote the i-th token of K_s and V_s. $-(|t-i|-1)/T$ indicates the relative position bias between the i-th token and the t-th token (current token). w is a learnable parameter, and u is a special case of w, giving a special attention to the current token.

For practical implementation, (12) can be reformulated into a recursive formula in an RNN form. By partitioning the summation term of the denominator and numerator in (12) at temporal boundary t, we can obtain 4 hidden states:

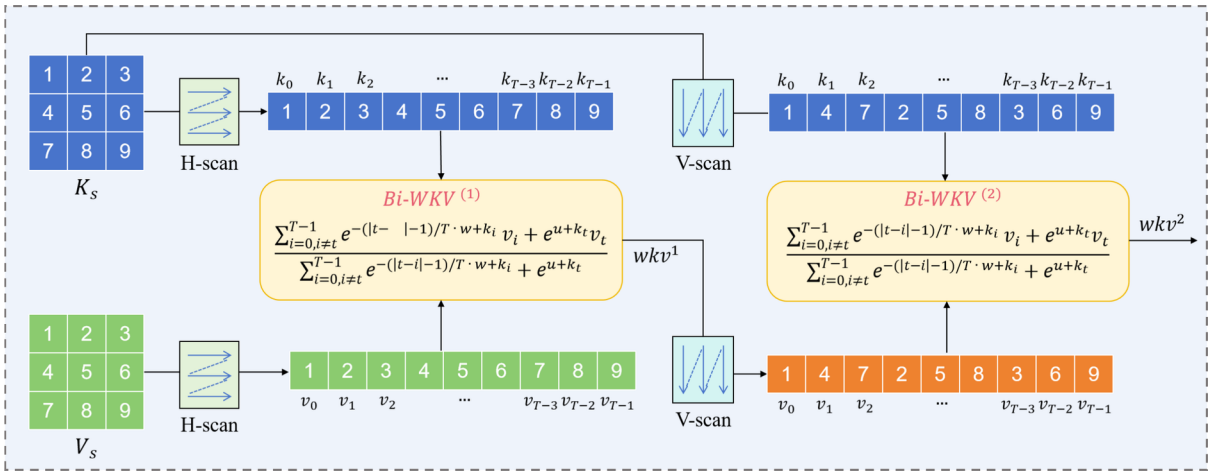


Fig. 4 The diagram of Re-Bi-WKV attention. Given the inputs K_s and V_s, we first perform a horizontal scanning to obtain a pair of 1D sequences, which are then processed through the first WKV attention computation, producing the initial attention output wkv^1. Subsequently, scan wkv^1 and K_s along the vertical direction to serve as the input for the second WKV attention calculation, ultimately yielding the final attention result wkv^2

Table 1 Comparison of computational complexity among Transformer, Mamba, and RWKV

Model	Transformer	Mamba	RWKV
Complexity	$\mathcal{O}(T^2C)$	$\mathcal{O}(TC^2)$	$\mathcal{O}(TC)$

$$
\begin{aligned}
a_{t-1} &= \sum_{i=0}^{t-1} e^{-(|t-i|-1)/T\cdot w+k_i} v_i \\
b_{t-1} &= \sum_{i=t+1}^{T-1} e^{-(|t-i|-1)/T\cdot w+k_i} v_i \\
c_{t-1} &= \sum_{i=0}^{t-1} e^{-(|t-i|-1)/T\cdot w+k_i} \\
d_{t-1} &= \sum_{i=t+1}^{T-1} e^{-(|t-i|-1)/T\cdot w+k_i}
\end{aligned}
\tag{13}
$$

Then, (12) can be rewritten as follows:

$$
wkv_t = Bi\text{-}WKV(K_s, V_s)_t = \frac{a_{t-1} + b_{t-1} + e^{u+k_t} v_t}{c_{t-1} + d_{t-1} + e^{u+k_t}} \tag{14}
$$

Each update step yields an attention result for a token (denoted by wkv_t), so the entire wkv matrix requires T steps. The computational complexity of Bi-WKV is shown as follows:

$$
\Omega(Bi\text{-}WKV) = 13 \times T \times C \tag{15}
$$

in which the coefficient 13 arises from the updates of 4 hidden states (a,b,c,d), the computation of exponentials, and the final calculation of wkv_t. T and C respectively denote the token length and the channel dimension. The complexity analysis confirms that Bi-WKV exhibits $\mathcal{O}(TC)$ linear scaling. Benifiting from Bi-WKV attention, RWKV outperforms Transformer and Mamba in terms of computational complexity, as quantitatively demonstrated in Table 1.

As shown in (12), the attention result wkv_t is a weighted sum of V_s along the token dimension ($0 \rightarrow (T-1)$, i.e., from the first token to the last token) with weighting coefficients being determined by the relative position bias and K_s. In other words, the attention result for each token is calculated from all other tokens, making Bi-WKV possess a global receptive field. Additionally, relative position bias assigns higher weights to tokens closer to the current token and lower weights to tokens farther away. This not only ensures a global receptive field but also implicitly leverages the prior knowledge that local tokens are generally more relevant. To conclude, Bi-WKV establishes global dependencies with linear computational complexity.

Recurrent attention mechanism The attention result of Bi-WKV (denoted by wkv_t) is related to the relative position bias $-(|t-i|-1)/T$, making Bi-WKV highly sensitive to the arrangement order of tokens. In a 2D image, the sequential order of tokens depends on the scanning direction, and sing-directional scanning cannot adequately model 2D image dependencies. To address this problem, we introduce a recurrent attention mechanism, which alternately applies Bi-WKV along different scanning directions in a recurrent manner:

$$
\begin{aligned}
wkv^1 &= Bi\text{-}WKV^{(1)}(H\text{-}scan(K_s), H\text{-}scan(V_s)) \\
wkv^2 &= Bi\text{-}WKV^{(2)}(V\text{-}scan(K_s), V\text{-}scan(wkv^1))
\end{aligned}
\tag{16}
$$

where $Bi\text{-}WKV^{(1)}(\cdot)$ and $Bi\text{-}WKV^{(2)}(\cdot)$ respectively denote the first and second Bi-WKV attention, wkv^1 and wkv^2 indicate the corresponding attention results. $H\text{-}scan(\cdot)$ and $V\text{-}scan(\cdot)$ represent horizontal scanning and vertical scanning.

During the $Bi\text{-}WKV^{(2)}(\cdot)$ calculation process, the key K_s stays the same, only with the sequential order altered by a different scanning direction, while the value is the previous attention result wkv^1. In other words, recurrent attention mechanism is introduced to complement and refine the previous attention result from another scanning direction.

3.3.3 High-frequency enhancement module

Traditional frequency separation techniques such as Fourier transform-based filtering (FTF) are computationally expensive and difficult to directly integrate into networks. In this paper, we separate high and low frequency information through simple pooling and differential operation, and adopt a dense residual block to enhance high frequency feature.

Given the input feature $F_{in} \in \mathbb{R}^{H\times W\times C}$, we first employ a 2×2 average pooling layer to halve the spatial dimension:

$$F_{pool} = avgpool_{2\times 2}(F_{in}) \tag{17}$$

where $F_{pool} \in \mathbb{R}^{\frac{H}{2}\times\frac{W}{2}\times C}$ and each pixel value in F_{pool} represents the average intensity of a specific 2×2 small region.

Then, bicubic interpolation is applied to upsample F_{pool}, resulting in $F_{bic} \in \mathbb{R}^{H\times W\times C}$:

$$F_{bic} = bicubic(F_{pool}) \tag{18}$$

F_{bic} shares the same spatial size with F_{in} and can be considered as the average smooth expression of F_{in} which corresponds to low-frequency components. Subtracting F_{bic} from F_{in} yields high-frequency feature F_{hf}:

$$F_{hf} = F_{in} - F_{bic} \tag{19}$$

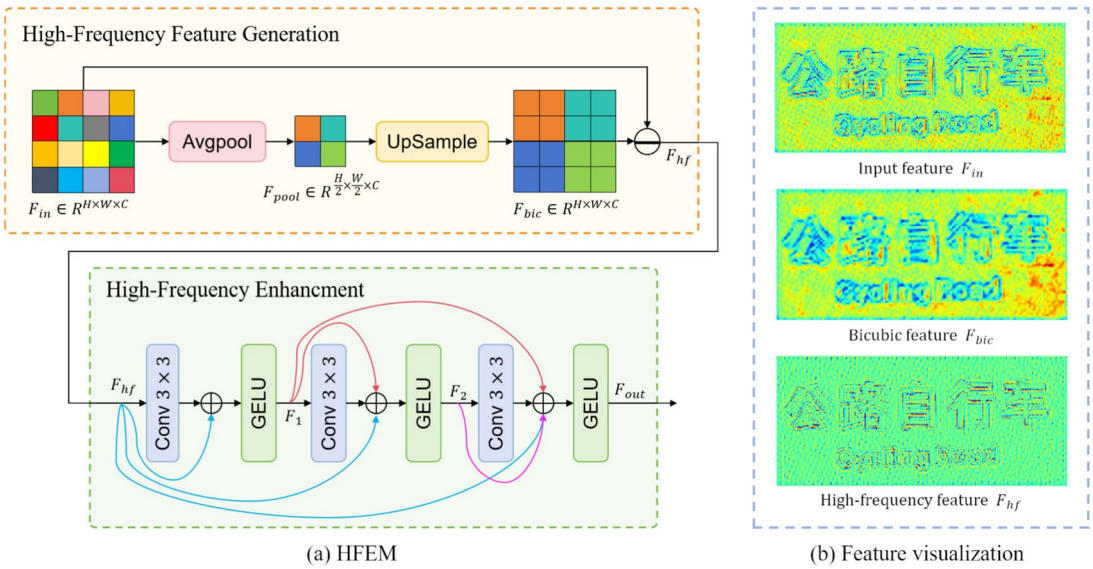
The feature maps of F_{in}, F_{bic} and F_{hf} are shown in Fig. 5(b). It can be observed that F_{bic} is the smooth representation of F_{in}, and F_{hf} retains the edges and details of an image. Therefore, we need to adopt small receptive fields to better focus on detail areas. Additionally, it has been shown that residual learning excels at exploring high-frequency features [2]. Based on the above viewpoints, we design a dense residual block with multiple small convolutional kernels to enhance high-frequency feature F_{hf}. The specific process is as follows:

$$\begin{aligned} F_1 &= GELU(Conv_{3\times 3}(F_{hf}) + F_{hf}) \\ F_2 &= GELU(Conv_{3\times 3}(F_1) + F_1 + F_{hf}) \\ F_{out} &= GELU(Conv_{3\times 3}(F_2) + F_2 + F_1 + F_{hf}) \end{aligned} \tag{20}$$

where $GELU(\cdot)$ denotes GELU activation function, and $Conv_{3\times 3}(\cdot)$ represents a 3×3 convolutional layer.

3.4 Multi-frequency channel attention

As shown in Fig. 6(a), the original channel attention employs global average pooling (GAP) to compress spatial information ($\mathbb{R}^{C\times H\times W} \rightarrow \mathbb{R}^{C\times 1\times 1}$), followed by a fully connected network (FCN) to compute channel-wise weight



(a) HFEM (b) Feature visualization

Fig. 5 (a) The diagram of high-frequency enhancement module(HFEM). HFEM primarily consists of two stages: High-frequency Feature Generation and High-Frequency Enhancement. (b) Visualization of input feature F_{in}, bicubic feature F_{bic} and high-frequency feature F_{hf}

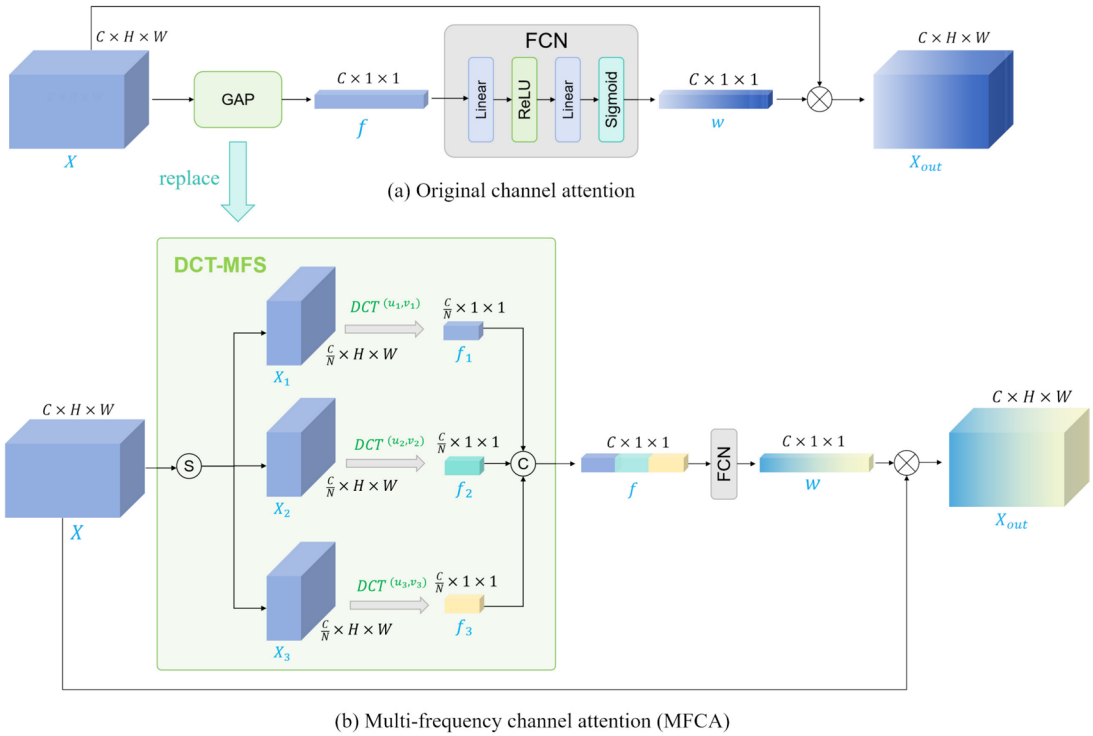


Fig. 6 The diagram of original channel attention and multi-frequency channel attention (MFCA). The original channel attention employs global average pooling (GAP) to compress spatial information, only preserving the lowest frequency components. In contrast, MFCA adopt a DCT-based multi-frequency selection (DCT-MFS) as an alternative to GAP, introducing more useful frequency information

$w \in \mathbb{R}^{C\times1\times1}$, which is then multiplied with the input feature X to assign specific weights to each channel:

$$\begin{aligned} w &= FCN(GAP(X)) \\ X_{out} &= w \otimes X \end{aligned} \tag{21}$$

where $\otimes$ denotes element-wise multiplication. During the multiplication process, the channel-wise weight w will be broadcasted along the spatial dimension.

In this paper, we design a DCT-based multi-frequency selection (DCT-MFS) as an alternative to GAP. 2D-DCT calculation formula is as follows:

$$F(u,v) = \sum_{x=0}^{H-1}\sum_{y=0}^{W-1} f(x,y)\cdot\cos\left(\frac{\pi u}{H}(x+\frac{1}{2})\right)\cos\left(\frac{\pi v}{W}(y+\frac{1}{2})\right) \tag{22}$$

where $F(u, v)$ denotes 2D-DCT frequency spectrum, while $f(x, y)$ represents the input feature map. Correspondingly, H and W respectively denote the height and width of $f(x, y)$.

It is well known that GAP is a special case of 2D-DCT, representing the lowest frequency information. The proof procedure is as follows:

$$\begin{aligned} F(0,0) &= \sum_{x=0}^{H-1}\sum_{y=0}^{W-1} f(x,y)\cdot\cos\left(\frac{\pi 0}{H}(x+\frac{1}{2})\right)\cos\left(\frac{\pi 0}{W}(y+\frac{1}{2})\right) = \sum_{x=0}^{H-1}\sum_{y=0}^{W-1} f(x,y) \\ GAP(f(x,y)) &= \frac{1}{HW}\sum_{x=0}^{H-1}\sum_{y=0}^{W-1} f(x,y) = \frac{1}{HW}F(0,0) \end{aligned} \tag{23}$$

where $F(0, 0)$ denotes the lowest frequency component of 2D-DCT. Significantly, it can be observed that $F(0, 0)$ exhibits a proportional relationship with the outcome of GAP operation.

To introduce more useful information, we generalize GAP to multiple frequency components, designing a multi-frequency channel attention (MFCA). As shown in Fig. 6(b), the input feature $X \in \mathbb{R}^{C\times H\times W}$ is split into N groups along the channel dimension (for simplicity, the diagram takes $N = 3$ as an example):

$$[X_1, X_2 \cdots X_N] = Split(X) \tag{24}$$

where $Split(\cdot)$ denotes the channel splitting operation, and $[X_1, X_2, \cdots, X_N]$ represent the resultant feature slices.

For each feature slice $X_i \in \mathbb{R}^{\frac{C}{N}\times H\times W}$, a specific 2D-DCT frequency component (u_i, v_i) is selected:

$$f_i = 2D\text{-}DCT^{(u_i,v_i)}(X_i), \quad i = 1, 2, \ldots, N \tag{25}$$

in which (u_i, v_i) represents the frequency indices corresponding to X^i, and $f_i \in \mathbb{R}^{\frac{C}{N}\times1\times1}$ denotes the i-th compressed vector.

Then, $[f_1, f_2, \cdots, f_N]$ are concatenated along the channel dimension to obtain the whole compression vector:

$$f = DCT\text{-}MFS(X) = Concat([f_1, f_2, \cdots, f_N]) \quad (26)$$

where $f \in \mathbb{R}^{C\times1\times1}$ denotes the multi-frequency vector, which is the final output of DCT-MFS.

Similar to the original channel attention, f passes through FCN to obtain channel weight, which is then multiplied with the input feature X to obtain the final output:

$$X_{out} = FCN(f) \otimes X \quad (27)$$

3.5 Loss function

In this paper, we adopt three types of loss, including pixel loss L_{pixel}, edge-perceptual loss L_{edge} and frequency distribution loss L_{freq}.

L_{pixel} is defined as the L_1 distance between SR image I_{SR} and HR image I_{HR}, which can force the SR image to have a high PSNR:

$$L_{pixel} = \|I_{HR} - I_{SR}\|_1 \quad (28)$$

L_{edge} measures the differences between the ground-truth edge map M_{HR} and the predicted edge map M_{SR} on the feature domain [28]. Specifically, the image features are weighted by edge features to enhance the text area:

$$L_{edge} = \|\phi(I_{HR}) \odot \phi(M_{HR}) - \phi(I_{SR}) \odot \phi(M_{SR})\|_1 \quad (29)$$

where $\phi(\cdot)$ represents the pre-trained VGG feature extractor [37], $\odot$ denotes element-wise multiply.

As we all know, each point in the frequency domain is calculated from all pixel points in the spatial domain, and thus the frequency spectrum contains rich global information. Studies [30] have shown that different frequency components (phase and magnitude) of an image possess distinct physical meanings, with phase being related to the brightness and color of the image, and magnitude affecting the shape and edge of the object. Based on this, we propose a frequency distribution loss L_{freq} that computes the distance between SR images and HR images in terms of amplitude spectra and phase spectra.

Firstly, VGG19 feature extractor is utilized to project I_{SR} and I_{HR} into feature perceptual space. Then, Fast Fourier Transform (FFT) is applied to convert the image features into the frequency domain, yielding corresponding spectral features (F_{SR} and F_{HR}). Subsequently, we compute the magnitude and phase of these spectral features and perform L_1 distance separately for magnitude and phase. The specific process is shown as follows:

$$\begin{gathered} F_{SR} = \mathcal{F}(\phi(I_{SR})), \quad F_{HR} = \mathcal{F}(\phi(I_{HR})) \\ A_{SR} = |F_{SR}|, \quad P_{SR} = \angle F_{SR}; \quad A_{HR} = |F_{HR}|, \quad P_{HR} = \angle F_{HR} \\ L_{freq} = \|A_{HR} - A_{SR}\|_1 + \|P_{HR} - P_{SR}\|_1 \end{gathered} \quad (30)$$

where $\phi(\cdot)$ represents the pre-trained VGG feature extractor, $\mathcal{F}(\cdot)$ denotes the 2D Fast Fourier Transform (2D-FFT). Besides, F_{SR} and F_{HR} denote the frequency spectrum features, with A_{HR}, A_{SR}, P_{HR}, P_{SR} being their corresponding magnitude and phase components.

4 Experiments and analysis

4.1 Experimental settings

4.1.1 Datasets

The details of the adopted datasets are shown in Table 2.

Real-CE dataset For STISR task, the proposed model is trained and tested on Real-CE [28] dataset, which contains a large number of Chinese-English LR-HR text image pairs. These image pairs are captured with different focal lengths (13mm, 26mm and 52mm), enabling the 2× (from 13mm to 26mm, from 26mm to 52mm) and 4× (from 13mm to 52mm) STISR tasks. Specifically, the dataset contains 2718 image pairs, 1935 of which are for training, and the rest are for testing. Notably, there are no domain shift concerns since the training and test data follow the same distribution.

Table 2 Details of the adopted datasets

Dataset	Task	Usage	Count	Characteristics
Real-CE	STISR	Training+Test	2718	Chinese-English text images
DIV2K	SISR	Training	800	Professional-grade 2K images
Flickr2K	SISR	Training	2650	Large-scale, diverse scenes
Set5	SISR	Test	5	Classic small test set
Set14	SISR	Test	14	Extended version of Set5
BSD100	SISR	Test	100	Structured urban environments
Urban100	SISR	Test	100	Diverse natural textures

Additionally, Real-CE also provides corresponding text labels for each LR-HR text image pair, allowing for the evaluation of STISR performance from the perspective of text recognition.

SISR datasets To validate the effectiveness of our method in SISR task, we employ DIV2K [38] and Flickr2K [24] as training data, and four common SISR benchmarks as test data, including Set5 [1], Set14 [44], BSD100 [29] and Urban100 [16]. For the training data DIV2K and Flickr2K, we perform data augmentation through horizontal flipping and random rotations (90°, 180°, and 270°), then crop images into 64×64 patches as input samples. Besides, the test sets provide diverse domain coverage: Urban100 specializes in urban architectural textures while BSD100 focuses on natural landscapes. Despite these inherent domain variations across test sets, the comprehensive scene diversity in our training data guarantees superior cross-domain generalization performance.

4.1.2 Evaluation metrics

For STISR task, we employ five evaluation metrics, including peak signal-to-noise ratio (PSNR), structural similarity index measure (SSIM), learned perceptual image patch similarity (LPIPS), word recognition accuracy (WRA) and normalized edit distance (NED). PSNR and SSIM measure the similarity between the reconstructed SR images and the ground-truth HR images in image space, while LPIPS evaluates perceptual quality in feature space. WRA and NED assess the SR performance from the perspective of text recognition. Specifically, WRA reflects recognition performance at the word level with binary outcomes (correct or incorrect), while NED demonstrates a finer-grained character-level recognition performance by calculating the sequence similarity between the recognition results and the text labels. For long text sequences, the finer-grained NED offers a more precise evaluation of recognition performance.

For SISR task, following previous works, we adopt PSNR and SSIM as evaluation metrics.

4.1.3 Implementation details

Training details Our HCR-HFE is implemented on Pytorch 2.0.1 with NVIDIA RTX A6000 GPU. During the training process, the input images are randomly cropped into 64×64 patches before passing through the model. We train HCR-HFE for 300K iterations with the batch size set to 16. The initial learning rate is set to 1×10^{-4} which is reduced by half at the milestone [10K, 50K, 100K, 150K]. We use Adam optimizer with $\beta_1 = 0.9, \beta_2 = 0.999$ for model training. When evaluating WRA and NED, we employ CRNN [35] as the text recognizer, utilizing the officially released PyTorch code and the pre-trained weights. Besides, text lines are cropped from the entire SR images and resized to match the input size of CRNN.

Model details For the network architecture, we stack 6 residual groups (RGs), with each RG containing 5 RWKB blocks. The number of channels (denoted by C) is configured as 128. In MFCA, N is set to 16, implying the selection of 16 frequency componnents.

4.2 Ablation study

For the ablation studies, we train and evaluate models on Real-CE dataset for the $\times 2$ SR task. Besides, we compare the model complexity under $\times 2$ SR setting, with the output size set to 320×320 for FLOPs calculation.

4.2.1 Effect of Re-Bi-WKV

In this session, we replace recurrent bidirectional WKV (Re-Bi-WKV) with unidirectional WKV (Uni-WKV) [32] and bidirectional WKV (Bi-WKV) [8]. Uni-WKV possesses a causal receptive field, only allowing it to see the current token and previous tokens. As shown in (12), Bi-WKV modifies the summation upper limit from t (the current token) to $T-1$ (the last token), leading to a global receptive field that all tokens are visible. Table 3 shows that Bi-WKV outperforms Uni-WKV across all metrics, due to its ability to capture information from a broader receptive field, thereby improving SR performance. Re-Bi-WKV builds upon Bi-WKV by introducing a recurrent mechanism, applying Bi-WKV along horizontal and vertical scanning directions in a recurrent manner. By integrating two scanning modes, Re-Bi-WKV can comprehensively model 2D image dependencies, leading to further performance improvements. In conclusion, transitioning from Uni-WKV to Bi-WKV and finally to Re-Bi-WKV demonstrates consistent performance improvements despite progressively increasing computational costs (FLOPs).

Table 3 Ablation study on different WKV attention methods

Attention	Params(M)	FLOPs(G)	NED↑	WRA↑	PSNR↑	SSIM↑	LPIPS↓
Uni-WKV	15.58	267	0.6964	34.68%	20.52	0.7385	0.1688
Bi-WKV	15.58	324	0.7010	35.12%	20.78	0.7422	0.1661
Re-Bi-WKV	15.58	382	**0.7042**	**35.43%**	**20.95**	**0.7442**	**0.1652**

Note: Bold values indicate the best performance

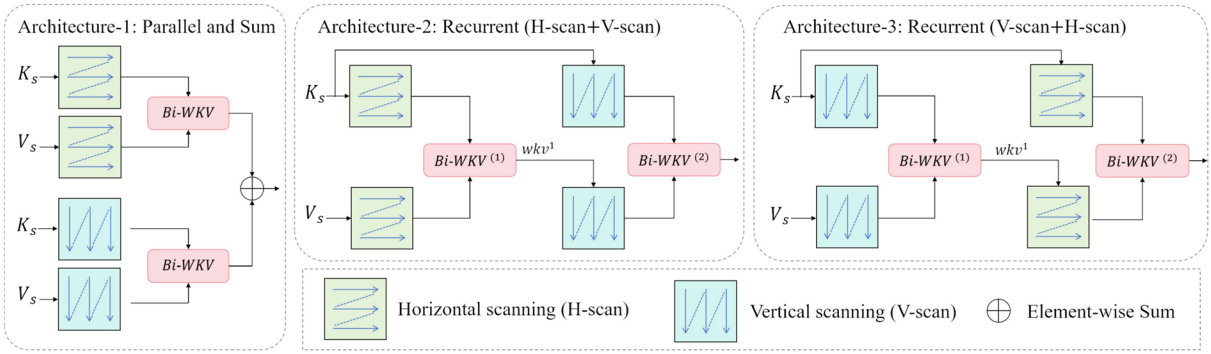


Fig. 7 The diagrams of three comparison architectures included in Table 4. Architecture-1 conducts parallel attention calculation from two scanning directions, and then sums up the two attention results. Architecture-2 employs the recurrent attention mechanism, applying Bi-WKV along horizontal and vertical scanning directions in a recurrent manner. Architecture-3 is similar to Architecture-2, but reverse the order of two scanning directions

Table 4 Ablation study on architecture designs for conducting WKV attention with different scanning modes

Architecture	NED↑	WRA↑	PSNR↑	SSIM↑	LPIPS↓
Parallel and Sum	0.7027	35.29%	20.88	0.7427	0.1661
Recurrent (H-scan+V-scan)	**0.7042**	**35.43%**	**20.95**	**0.7442**	**0.1652**
Recurrent (V-scan+H-scan)	0.7038	35.39%	**20.95**	0.7441	0.1655

Note: Bold values indicate the best performance

To further explore the optimal architecture for combining horizontal and vertical scanning, we evaluate three different designs, as depicted in Fig. 7. Architecture-1 employs a parallel attention strategy inspired by MambaIR [13]. Firstly, it scans from both horizontal and vertical direction to obtain two pairs of 1D token sequences, which are then computed in parallel to get two attention results. Subsequently, the two results are summed up to get the final output. Architecture-2 employs the recurrent attention mechanism as described in Section 3.3.2. Firstly, scan along the horizontal direction to obtain a pair of 1D sequences, then perform WKV attention calculation to acquire the first attention result wkv^1. Subsequently, scan wkv^1 along the vertical direction to serve as the input for the second WKV attention calculation, ultimately yielding the final attention result. Architecture-3 is similar to Architecture-2, but reverse the order of horizontal scanning (H-scan) and vertical scanning (V-scan). Table 4 demonstrates that recurrent attention mechanism outperforms parallel attention strategy, with Architecture-2 (H-scan followed by V-scan) performing slightly better than Architecture-3. Therefore, this paper adopts Architecture-2 for conducting WKV attention.

4.2.2 Effect of other proposed modules

Building on Re-Bi-WKV, we progressively integrate additional modules, which slightly increase the model complexity while further improving SR performance, as demonstrated in Table 5. Compared to Model-A, Model-B exhibits improvements in word recognition accuracy (WRA) by 0.16% and normalized edit distance (NED) by 0.0018. As depicted in Fig. 8(a), Model-A without high-frequency enhancement module(HFEM) shows poor detail recovery, whereas Model-B can effectively restore character edges. Figure 7(b) displays the Fourier spectrum, intuitively demonstrating the complementary roles of WKVAM and HFEM. Compared to the input feature F_{in}, global feature F_g output by WKVAM exhibits stronger energy concentration in the central region of spectrum map, while the spectrum of high-frequency feature F_{hf} spreads more broadly towards the surrounding area. The results suggest that WKVAM mainly extracts low-frequency information, while HFEM can activate more high-frequency components corresponding to character edge details, thereby improving text recognition performance. Notably, conventional frequency-domain enhancement techniques like Fourier Transform-based Filtering (FTF) suffers from high computational cost. The FTF pipeline involves three steps: (1) Perform Fast Fourier Transform (FFT) on the input feature map to convert it from the spatial domain to the frequency domain; (2) Design a frequency-domain filter to selectively enhance high-frequency components; (3) Apply the Inverse Fast Fourier Transform (IFFT) to the enhanced spectrum to reconstruct it in the spatial domain, obtaining the high-frequency-enhanced feature map. As quantitatively demonstrated

Table 5 Ablation study on HFEM, MLKCB, MFCA, L_{freq}

Model	Re-Bi-WKV	HFEM	MLKCB	MFCA	L_{freq}	Params(M)	FLOPs(G)	NED↑	WRA↑	PSNR↑	SSIM↑	LPIPS↓
model-A	✓					15.58	382	0.7042	35.43%	20.95	0.7442	0.1652
model-B	✓	✓				16.36	394	0.7060	35.59%	20.98	0.7447	0.1648
model-C	✓	✓	✓			17.84	428	0.7063	35.64%	21.02	0.7457	0.1644
model-D	✓	✓	✓	✓		19.21	456	0.7074	35.76%	21.05	0.7473	0.1638
model-E	✓	✓	✓	✓	✓	19.21	456	**0.7075**	**35.78%**	**21.18**	**0.7478**	**0.1625**

Note: Bold values indicate the best performance

in Table 6, our HFEM achieves superior computational efficiency (measured in parameters and FLOPs) while maintaining competitive performance metrics (NED, WRA, SSIM) compared to FTF.

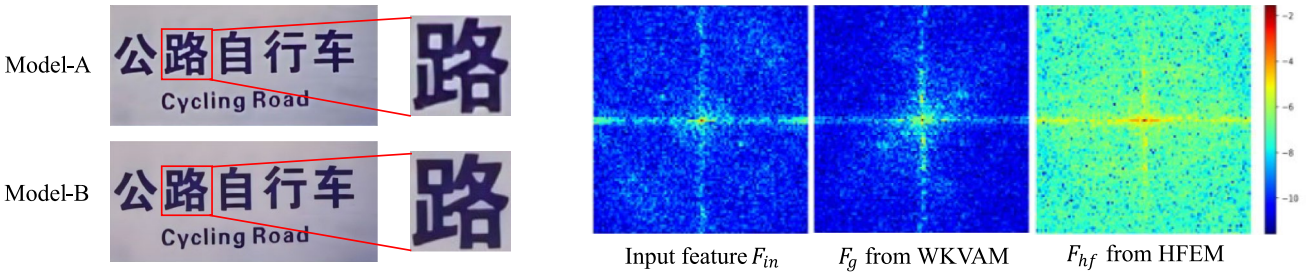
Model-C incorporates multi-scale large kernel convolution block (MLKCB) on top of Model-B, improving PSNR by 0.09dB. Considering that both long-range dependencies and local texture information are indispensable for SR reconstruction task, MLKCB is designed to encode features from coarse to fine scales.

Model-D builds upon Model-C by introducing multi-frequency channel attention (MFCA), which further improves WRA by 0.11% and NED by 0.0012. MFCA can enable the network to focus on learning critical information (i.e., character edges) and reduce the attention to irrelevant information, thereby effectively improving text recognition performance. Table 7 further explores the effects of different numbers of frequency components (denoted by K) in MFCA. Since low-frequency components typically carry more information, we arrange frequency components in Zigzag order and select the Top-K components. Here, $K = 1$ corresponds to GAP operation, using only the lowest-frequency component ((u,v)=(0,0)). $K \geq 2$ incorporates additional frequency components (prioritizing low-to-mid frequencies) following the Zigzag order. The selection of frequency components is inherently based on predefined DCT bases, meaning that increasing K does not affect the parameter count of the model. In MFCA, we perform channel splitting on the input features and then select specific frequency components for each group of feature slices, so it does not increase the FLOPs. As shown in Table 7, we set a group of K including 1, 4, 8, 16, or 32 to observe the performance change. The results indicate that $K = 16$ achieves the best trade-off across all metrics.

By introducing frequency distribution loss L_{freq}, Model-E excels in both PSNR and LPIPS, demonstrateing that computing distribution distance within the frequency domain can more effectively utilize global structural information, resulting in superior image fidelity and perceptual quality.

4.2.3 Effect of model hyperparameters

The number of RWKV blocks Table 8 explores the impact of varying the number of RWKV blocks (denoted as M) in each RG on both model complexity and SR performance. The results indicate that progressively increasing M from 3 to 5 leads to consistent improvements across all metrics, indicating enhanced model performance. However, when further increasing to 6 blocks, we observe diminishing returns (e.g., marginal PSNR gain of only 0.02dB) and even slight degradation in some metrics (e.g., NED, WRA and SSIM). This may be due to the increase in model parameters, which leads to



(a) SR reconstruction results (b) Fourier spectrum

Fig. 8 Effect of high-frequency enhancement module (HFEM). (a) Visualization of SR reconstruction results. Compared to Model-A, Model-B can effectively restore the character edge details. (b) Fourier spectrum of input feature F_{in}, global feature F_g and high-frequency feature F_{hf}. F_g exhibits strong energy concentration in the central region, corresponding to low-frequency components, while F_{hf} contains more high-frequency information

Table 6 Comparison of our HFEM with Fourier Transform-based Filtering (FTF)

Model	Params(M)	FLOPs(G)	NED↑	WRA↑	PSNR↑	SSIM↑	LPIPS↓
Model-B (HFEM)	16.36	394	**0.7060**	**35.59**	20.98	**0.7447**	0.1654
Replace with FTF	17.58	456	0.7055	35.58	**20.99**	0.7446	**0.1648**

Note: Bold values indicate the best performance

Table 7 Ablation study on the number of frequency component in MFCA

K	Params(M)	FLOPs(G)	NED↑	WRA↑	PSNR↑	SSIM↑	LPIPS↓
1	19.21	456	0.7058	35.61%	20.96	0.7454	0.1661
4	19.21	456	0.7068	35.70%	21.02	0.7466	0.1655
8	19.21	456	0.7072	35.75%	21.05	0.7471	0.1653
16	19.21	456	**0.7074**	**35.76%**	21.05	**0.7473**	**0.1652**
32	19.21	456	0.7068	35.69%	**21.06**	0.7469	0.1655

Note: Bold values indicate the best performance

Table 8 Ablation study on the number of RWKB blocks in each RG

RWKV blocks	Params(M)	FLOPs(G)	NED↑	WRA↑	PSNR↑	SSIM↑	LPIPS↓
3	13.52	321	0.7001	35.12%	20.67	0.7374	0.1712
4	15.75	376	0.7048	35.59%	20.98	0.7427	0.1661
5	19.21	456	**0.7075**	**35.78%**	21.18	**0.7478**	0.1635
6	22.68	535	0.7071	35.75%	**21.20**	0.7476	**0.1631**

Note: Bold values indicate the best performance

Table 9 Ablation study on chanel count

Channel Count	Params(M)	FLOPs(G)	NED↑	WRA↑	PSNR↑	SSIM↑	LPIPS↓
64	4.88	119	0.6987	35.12%	20.67	0.7374	0.1712
96	11.12	266	0.7039	35.59%	20.98	0.7427	0.1661
128	19.21	456	0.7075	**35.78%**	**21.18**	0.7478	0.1635
144	24.78	588	**0.7077**	35.72%	21.14	**0.7481**	**0.1628**

Note: Bold values indicate the best performance

overfitting problem. Considering the trade-off between model complexity and SR performance, M is configured as 5.

Chanel count Table 9 investigates the effect of channel count (denoted as C) on the model complexity and SR performance. As the channel count increases, the model parameters and FLOPs grow quadratically. In addition, SR performance metrics gradually tend to saturate when C reaches 128. Consequently, we select the channel count to be 128.

4.2.4 Multi-loss balance

For a multi-loss network, it is crucial to assign appropriate weights to each loss item. The overall training loss is formulated as:

Table 10 Ablation study on different loss balance coefficients

β	NED↑	WRA↑	PSNR↑	SSIM↑	LPIPS↓
0	0.7068	35.69%	20.05	0.7473	0.1652
0.0001	0.7069	35.71%	21.14	0.7475	0.1641
0.001	**0.7075**	**35.78%**	21.18	0.7478	**0.1625**
0.01	0.7067	35.67%	**21.19**	0.7477	0.1636
0.1	0.7063	35.64%	21.17	**0.7482**	0.1638

Note: Bold values indicate the best performance

$$L = L_{pixel} + \alpha L_{edge} + \beta L_{freq} \tag{31}$$

where L_{pixel}, L_{edge} and L_{freq} respectively denote pixel loss, edge-perceptual loss and frequency distribution loss. α, β indicate corresponding balance parameters.

For the SR task, L_{pixel} is the most fundamental and indispensable loss, which can be weighted as 1. Following the previous work [28], the weight coefficient of L_{edge} (denoted by α) is set to 1. In this session, we focus on exploring the optimal weight coefficient for the frequency distribution loss. Both SR images and HR images are normalized, leading to very small values for L_{pixel} and L_{edge}. In contrast, L_{freq} has a relatively high value, so its weight coefficient β is expected to be small. As shown in Table 10, we set a group of β from 0 to 0.1 to observe the performance change. The results indicate that β=0.001 achieves the best trade-off across all metrics, and this value is adopted in our final model.

4.3 Comparisons with state-of-the-arts

4.3.1 Evaluations on Real-CE dataset

We compare our model with various SR methods in terms of quantitative metrics, visual results and model complexity.

Quantitative metrics For quantitative comparisons, We select four advanced STISR methods including TSRN [39], TPGSR [27], TBSRN [3], TATT [26], as well as five SISR methods, namely EDSR [51], RCAN [50], ELAN [48], CFAT [33], MambaIR [13]. Tables 11 and 12 demonstrate that our model achieves optimal performance compared to CNN-based methods, Transformer-based methods and Mamba-based methods. Notably, SISR methods markedly outperform STISR methods on Real-CE dataset. This is owing to the fact that SISR methods take the entire images as input, whereas the STISR methods are specifically designed for cropped text images that only contain text lines, making adaptation to the Real-CE dataset more challenging. In addition, all five compared SISR methods perform poorly on the LPIPS metric, while the STISR

Table 11 Quantitative comparison with various methods for ×2 SR task on Real-CE test set

Methods	Type	×2				
		NED↑	WRA↑	PSNR↑	SSIM↑	LPIPS↓
TSRN [39]	CNN-based	0.4809	28.54%	18.99	0.5233	0.1677
TPGSR [27]	CNN-based	0.4913	30.07%	18.83	0.5562	0.1661
TBSRN [3]	CNN-based	0.5294	31.81%	19.01	0.5366	0.1652
TATT [26]	CNN+Transformer	0.5240	31.27%	19.06	0.5772	**0.1590**
EDSR [51]	CNN-based	0.6954	34.68%	20.74	0.7448	0.2258
RCAN [50]	CNN-based	0.7006	34.84%	20.98	0.7435	0.2173
ELAN [48]	Transformer-based	0.6992	35.08%	21.16	**0.7480**	0.2201
CFAT [33]	Transformer-based	0.7012	35.19%	21.09	0.7479	0.2163
MambaIR [13]	Mamba-based	0.7029	35.34%	21.14	0.7472	0.2147
Ours	CNN+RWKV	**0.7075**	**35.78%**	**21.18**	0.7478	0.1625

Note: Bold values indicate the best performance

Table 12 Quantitative comparison with various methods for ×4 SR task on Real-CE test set

Methods	Type	×4				
		NED↑	WRA↑	PSNR↑	SSIM↑	LPIPS↓
TSRN [39]	CNN-based	0.4159	23.16%	18.11	0.4850	0.1981
TPGSR [27]	CNN-based	0.4123	23.26%	18.07	0.4758	0.1843
TBSRN [3]	CNN-based	0.4444	25.27%	18.33	0.4826	**0.1715**
TATT [26]	CNN+Transformer	0.4342	23.30%	17.96	0.4904	0.1904
EDSR [51]	CNN-based	0.6330	28.82%	20.16	0.7195	0.2883
RCAN [50]	CNN-based	0.6321	28.79%	20.33	0.7232	0.2878
ELAN [48]	Transformer-based	0.6404	29.53%	20.39	**0.7299**	0.2892
CFAT [33]	Transformer-based	0.6436	29.70%	20.41	0.7248	0.2845
MambaIR [13]	Mamba-based	0.6478	30.01%	20.44	0.7254	0.2712
Ours	CNN+RWKV	**0.6587**	**30.81%**	**20.48**	0.7256	0.2045

Note: Bold values indicate the best performance

methods exhibit better performance. By introducing the edge-perceptual loss, our method has achieved significant improvement on LPIPS. In summary, our method can effectively balance various indicators, exhibiting excellent performance in terms of text legibility (NED, WRA), image fidelity (PSNR, SSIM), and perceptual quality (LPIPS). As the scale factor increases, more intricate details needs to be reconstructed, making the SR task more challenging, ultimately resulting in a noticeable decline in performance metrics for $\times 4$ SR task. Our method exhibits substantial improvements on both $\times 4$ and $\times 2$ SR tasks, especially in terms of WRA and NED, demonstrating superior text recognition performance.

Visual results Figures 9 and 10 respectively compare the visual results of our method with a Transformer-based method CFAT [33] and a Mamba-based method MambaIR [13] for $\times 2$ and $\times 4$ tasks. To highlight the contrast effect, a red rectangular frame is employed to mark the specific area on each image, and the display area is enlarged for better visualization. Additionally, we employ CRNN to recognize the text in the rectangular frames, and the incorrectly recognized characters are highlighted in red. As depicted in Fig. 9, the text regions restored by both ELAN and MambaIR lack sufficient clarity, posing challenges in accurately reconstructing intricate character strokes. As shown in the second row of Fig. 10, both results of CFAT and MambaIR exhibit artifacts near the character edges, leading to a deficiency in stroke clarity. In constrast, our method can effectively suppress artifacts and reconstruct

Fig. 9 Visual comparison with various methods for $\times 2$ SR task on Real-CE test set. The text below each image represents the recognition result by CRNN, with the incorrectly recognized characters highlighted in red

Fig. 10 Visual comparison with various methods for $\times 4$ SR task on Real-CE test set. The text below each image represents the recognition result by CRNN, with the incorrectly recognized characters highlighted in red

Table 13 Comparion of model complexity for $\times 4$ SR task. The output size is set to 640×640 for FLOPs calculation

Methods	RCAN	ELAN	CFAT	MambaIR	Ours
Params(M)	15.59	8.312	22.07	16.7	20.18
FLOPs(G)	407	284	587	439	496
ACC(%)	28.79	29.53	29.70	30.01	30.81
NED	0.6321	0.6404	0.6436	0.6478	0.6587

clear character strokes, thereby effectively improving text recognition performance.

Model complexity Table 13 compares the model parameters and FLOPs of various classic SR methods for the $\times 4$ task. It can be observed that our HCR-HFE outperforms other SOTA models with a trade-off between parameters and FLOPs. Notably, HCR-HFE requires fewer parameters and lower computational cost (FLOPs) than CFAT while delivering superior performance on the Real-CE dataset. Figure 11 presents a comparison of the inference speed between our HCR-HFE and several methods, including the Transformer-based CFAT [33], the Mamba-based MambaIR [13], and Ours-WSA. The latter is a variation of our HCR-HFE, where each WKV attention module (WKVAM) is replaced by a window self-attention (WSA) with a window size of 8×8. It's worth noting that the inference speed is measured in frames per second (FPS). As depicted in Fig. 11, our method significantly outperforms CFAT and MambaIR in terms of FPS for both $\times 2$ and $\times 4$ SR tasks. Additionally, replacing WKVAM with WSA results in a substantial drop in inference speed. The results demonstrate that RWKV has faster inference speed compared to both Transformer and Mamba. Notably, the inference speed of $\times 4$ SR task is significantly faster than that of $\times 2$ SR task. This is due to the fact that all input images for $\times 4$ task are captured with a focal length of 13mm, whereas the input for $\times 2$ task includes images captured at both 13mm and 26mm. Longer

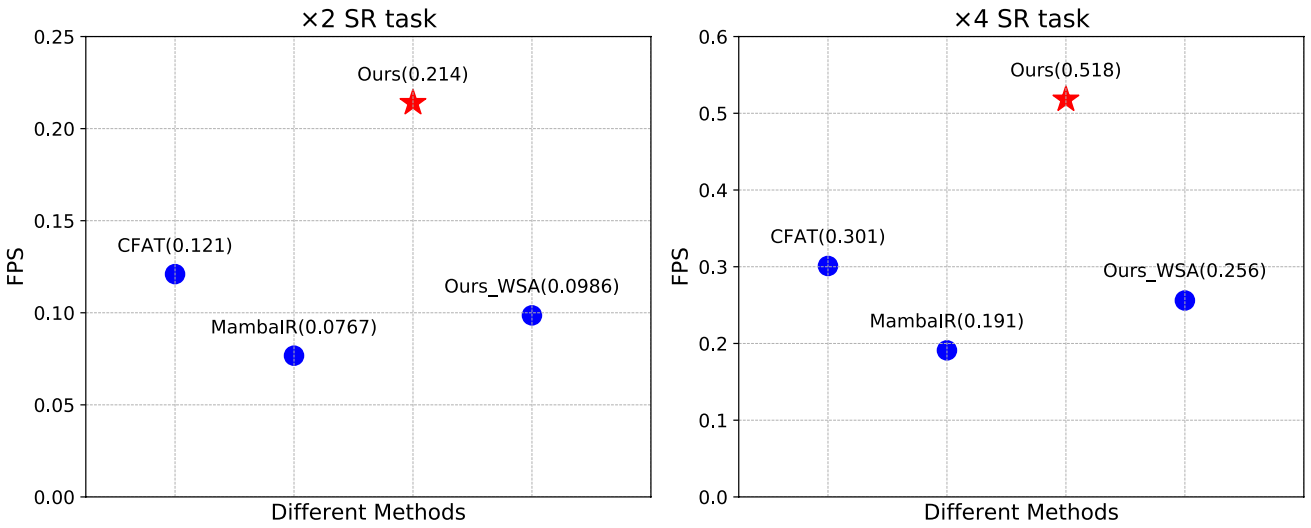


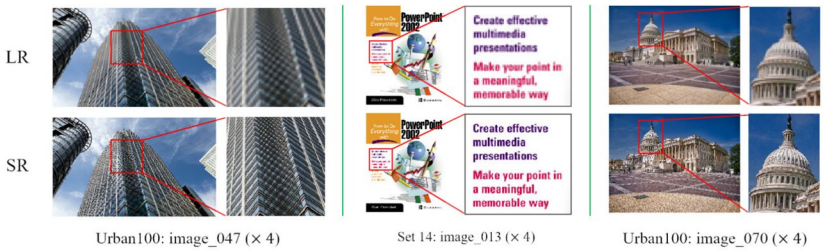
Fig. 11 Comparison of inference speed on Real-CE test set for both ×2 and ×4 SR tasks. The inference speed is measured in frames per second (FPS)

Table 14 PSNR/SSIM metrics on classical SISR datasets for ×4 task

Methods	Set5		Set14		BSD100		Urban100	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
EDSR [51]	32.46	0.8968	28.80	0.7876	27.71	0.7420	26.64	0.8033
RCAN [50]	32.63	0.9002	28.87	0.7889	27.77	0.7436	26.82	0.8087
ELAN [48]	32.75	0.9022	28.96	0.7914	27.83	0.7459	27.13	0.8167
SwinIR [22]	32.92	0.9044	29.09	0.7950	27.92	0.7489	27.45	0.8254
SRformer [52]	32.93	0.9041	29.08	0.7953	27.94	0.7502	27.68	0.8311
CVANet [47]	32.59	0.9001	28.92	0.7896	27.77	0.7437	26.96	0.8116
MambaIR [13]	33.03	0.9046	**29.20**	**0.7961**	27.98	0.7503	27.68	0.8287
Ours	**33.08**	**0.9055**	29.14	0.7955	**28.02**	**0.7510**	**27.90**	**0.8321**

Note: Bold values indicate the best performance

Fig. 12 Visual results on SISR datasets (×4)



focal lengths result in higher resolution images, which in turn slows down the inference speed.

4.3.2 Evaluations on SISR datasets

Our method is mainly designed for STISR task. To validate its effectiveness on general SISR task, we select seven classic SISR methods, including EDSR [51], RCAN [50], ELAN [48], SwinIR [22], SRformer [52], CVANet [47], and MambaIR [13], for quantitative comparison in terms of PSNR and SSIM with ×4 scale factor. Specifically, our model is trained on DIV2K [38] and Flickr2K [24] datasets, and then evaluated on four common benchmarks. Table 14 demonstrates that our method achieves competitive performance compared to previous methods, especially on Urban100 dataset [16]. Compared with traditional CNN-based method like EDSR, the proposed HCR-HFE achieves a 1.26 dB improvement in PSNR on Urban100, which is attributed to the global receptive field. Figure 12 illustrates the visual comparison of LR images and SR reconstruction images. Natural scene images contain many similar texture regions, like img_047 of Urban100 dataset. Our method possesses a global receptive field, enabling it to utilize surrounding similar regions to assist in the SR reconstruction of target regions, thereby effectively restoring fine texture details. For img_013 of Set14, our

HCR-HFE can reconstruct clear character strokes, leading to improved text recognition performance.

5 Discussion

The proposed HCR-HFE demonstrates superior performance across multiple evaluation metrics including text legibility (NED, WRA), image fidelity (PSNR, SSIM), and perceptual quality (LPIPS). Qualitative results further confirm these advantages through enhanced visual clarity. Notably, HCR-HFE achieves excellent performance on natural image datasets, demonstrating its broad applicability beyond text-specific scenarios. However, two main limitations should be noted. First, while our HCR-HFE demonstrates advantages in inference speed, it is not particularly competitive in terms of model parameters and FLOPs. Future work will focus on architectural optimizations to better balance reconstruction quality and computational complexity. Second, comprehensive subjective evaluation using mean opinion score (MOS) testing was deferred due to resource limitations, which will be addressed in the future work.

6 Conclusion

In this paper, we propose a Hybrid CNN-RWKV with High-Frequency Enhancement (HCR-HFE) method for real-world Chinese-English STISR task. Our method is primarily based on RWKV architecture, which can extract global features with linear computation complexity, eliminating the necessity of window partition operation in Transformer-based models. Additionally, we adopt a recurrent attention strategy which combines horizontal and vertical scanning to effectively model 2D image dependencies. Since the global features mainly contain low-frequency information, we design a high-frequency enhancement module (HFEM) to enhance the high-frequency details. Furthermore, a multi-scale large kernel convolutional block (MLKCB) is incorporated to establish various-range dependencies with a lower computational cost. Finally, we introduce a multi-frequency channel attention (MFCA) which enables the network to focus on learning critical features. Extensive experiments on Real-CE dataset demonstrate that HCR-HFE outperforms previous methods in both quantitative metrics and visual results. Moreover, HCR-HFE exhibits broad applicability, achieving excellent results on general SR datasets.

Acknowledgements This work was supported in part by the National Natural Science Foundation of China under Grant 62476088, and the Shanghai Automotive Industry Science and Technology Development Foundation under Grant 2304.

Author Contributions Yanbin Liu: Methodology, Software, Conceptualization, Writing-original draft, Writing-review & editing. Yu Zhu: Project administration, Supervision, Writing-review & editing. Hangyu Li: Formal analysis, Data curation, Validation. Xiaofeng Ling: Formal analysis, Supervision.

Data Availability The data supporting the findings of this study are openly available to the public. The most important benchmark dataset utilized in this research, namely Real-CE, can be accessed from https://drive.google.com/file/d/1d2pOgJ0e286OslzuGVsARfhW7FbQW0n-/view?usp=sharing. Furthermore, the general super-resolution datasets such as Set5, Set14, BSD100 and Urban100 are available at https://drive.google.com/drive/folders/1lsoyAjsUEyp7gm1t6vZI9j7jr9YzKzcF.

Declarations

Competing Interests The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Ethics approval This work does not involve experimental procedures with human subjects or animals.

References

1. Bevilacqua M, Roumy A, Guillemot C, et al (2012) Low-complexity single-image super-resolution based on nonnegative neighbor embedding
2. Chen G, Dai K, Yang K, et al (2024) Bracketing image restoration and enhancement with high-low frequency decomposition. arXiv:2404.13537
3. Chen J, Li B, Xue X (2021) Scene text telescope: text-focused scene image super-resolution. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 12026–12035
4. Chen X, Wang X, Zhou J, et al (2023a) Activating more pixels in image super-resolution transformer. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 22367–22377
5. Chen Z, Zhang Y, Gu J, et al (2023b) Dual aggregation transformer for image super-resolution. In: Proceedings of the IEEE/CVF international conference on computer vision, pp 12312–12321
6. Dong C, Loy CC, He K, et al (2014) Learning a deep convolutional network for image super-resolution. In: Computer vision–ECCV 2014: 13th European conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part IV 13, Springer, pp 184–199
7. Dosovitskiy A (2020) An image is worth 16x16 words: transformers for image recognition at scale. arXiv:2010.11929
8. Duan Y, Wang W, Chen Z, et al (2024) Vision-rwkv: efficient and scalable visual perception with rwkv-like architectures. arXiv:2403.02308
9. Fang C, Zhu Y, Liao L et al (2021) Tsrgan: real-world text image super-resolution based on adversarial learning and triplet attention. Neurocomputing 455:88–96
10. Fei Z, Fan M, Yu C, et al (2024) Diffusion-rwkv: scaling rwkv-like architectures for diffusion models. arXiv:2404.04478
11. Gu A, Dao T (2023) Mamba: linear-time sequence modeling with selective state spaces. arXiv:2312.00752
12. Guo H, Li J, Dai T, et al (2024) Mambair: a simple baseline for image restoration with state-space model. arXiv:2402.15648

13. Guo H, Li J, Dai T, et al (2025) Mambair: a simple baseline for image restoration with state-space model. In: European conference on computer vision, Springer, pp 222–241
14. Guo MH, Lu CZ, Liu ZN et al (2023) Visual attention network. Comput Vis Media 9(4):733–752
15. He Q, Zhang J, Peng J, et al (2024) Pointrwkv: efficient rwkv-like model for hierarchical point cloud learning. arXiv:2405.15214
16. Huang JB, Singh A, Ahuja N (2015) Single image super-resolution from transformed self-exemplars. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 5197–5206
17. Huang Y, Miyazaki T, Liu X, et al (2024) Irsrmamba: infrared image super-resolution via mamba-based wavelet transform feature modulation model. arXiv:2405.09873
18. Kim J, Lee JK, Lee KM (2016) Accurate image super-resolution using very deep convolutional networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 1646–1654
19. Lei X, ZHang W, Cao W (2024) Dvmsr: distillated vision mamba for efficient super-resolution. arXiv:2405.03008
20. Li A, Zhang L, Liu Y, et al (2023a) Feature modulation transformer: cross-refinement of global representation via high-frequency prior for image super-resolution. In: Proceedings of the IEEE/CVF international conference on computer vision, pp 12514–12524
21. Li X, Zuo W, Loy CC (2023b) Learning generative structure prior for blind text image super-resolution. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 10103–10113
22. Liang J, Cao J, Sun G, et al (2021) Swinir: image restoration using swin transformer. In: Proceedings of the IEEE/CVF international conference on computer vision, pp 1833–1844
23. Lim B, Son S, Kim H, et al (2017a) Enhanced deep residual networks for single image super-resolution. In: Proceedings of the IEEE conference on computer vision and pattern recognition workshops, pp 136–144
24. Lim B, Son S, Kim H, et al (2017b) Enhanced deep residual networks for single image super-resolution. In: Proceedings of the IEEE conference on computer vision and pattern recognition workshops, pp 136–144
25. Liu Z, Lin Y, Cao Y, et al (2021) Swin transformer: hierarchical vision transformer using shifted windows. In: Proceedings of the IEEE/CVF international conference on computer vision, pp 10012–10022
26. Ma J, Liang Z, Zhang L (2022) A text attention network for spatial deformation robust scene text image super-resolution. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 5911–5920
27. Ma J, Guo S, Zhang L (2023) Text prior guided scene text image super-resolution. IEEE Trans Image Process 32:1341–1353
28. Ma J, Liang Z, Xiang W, et al (2023b) A benchmark for chinese-english scene text image super-resolution. In: Proceedings of the IEEE/CVF international conference on computer vision, pp 19452–19461
29. Martin D, Fowlkes C, Tal D, et al (2001) A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In: Proceedings eighth IEEE international conference on computer vision. ICCV 2001, IEEE, pp 416–423
30. Ni Z, Wu J, Wang Z, et al (2024) Misalignment-robust frequency distribution loss for image transformation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 2910–2919
31. Noguchi C, Fukuda S, Yamanaka M (2024) Scene text image super-resolution based on text-conditional diffusion models. In: Proceedings of the IEEE/CVF winter conference on applications of computer vision, pp 1485–1495
32. Peng B, Alcaide E, Anthony Q, et al (2023) Rwkv: reinventing rnns for the transformer era. arXiv:2305.13048
33. Ray A, Kumar G, Kolekar MH (2024) Cfat: Unleashing triangular windows for image super-resolution. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 26120–26129
34. Shazeer N (2020) Variants improve transformer. arXiv:2002.05202
35. Shi B, Bai X, Yao C (2017) IEEE Trans Pattern Anal Mach Intell 39(11):2298–2304. https://doi.org/10.1109/TPAMI.2016.2646371
36. Shi Q, Zhu Y, Liu Y et al (2023) Perceiving multiple representations for scene text image super-resolution guided by text recognizer. Eng Appl Artif Intell 124:106551
37. Simonyan K, Zisserman A (2014) Very deep convolutional networks for large-scale image recognition. arXiv:1409.1556
38. Timofte R, Agustsson E, Van Gool L, et al (2017) Ntire 2017 challenge on single image super-resolution: Methods and results. In: Proceedings of the IEEE conference on computer vision and pattern recognition workshops, pp 114–125
39. Wang W, Xie E, Liu X, et al (2020) Scene text image super-resolution in the wild. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part X 16, Springer, pp 650–666
40. Xia C, Wang X, Lv F, et al (2024) Vit-comer: vision transformer with convolutional multi-scale feature interaction for dense predictions. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 5493–5502
41. Xiao Y, Yuan Q, Jiang K, et al (2024) Frequency-assisted mamba for remote sensing image super-resolution. arXiv:2405.04964
42. Yang Z, Zhang H, Zhao D, et al (2024) Restore-rwkv: efficient and effective medical image restoration with rwkv. arXiv:2407.11087
43. Yuan H, Li X, Qi L, et al (2024) Mamba or rwkv: exploring high-quality and high-efficiency segment anything model. arXiv:2406.19369
44. Zeyde R, Elad M, Protter M (2012) On single image scale-up using sparse-representations. In: Curves and Surfaces: 7th International Conference, Avignon, France, June 24-30, 2010, Revised Selected Papers 7, Springer, pp 711–730
45. Zhang J, ZhGluang Y, Gu J, et al (2022a) Accurate image restoration with attention retractable transformer. arXiv:2210.01427
46. Zhang J, Li X, Li J, et al (2023) Rethinking mobile block for efficient attention-based models. In: 2023 IEEE/CVF international conference on computer vision (ICCV), IEEE Computer Society, pp 1389–1400
47. Zhang W, Zhao W, Li J et al (2024) Cvanet: cascaded visual attention network for single image super-resolution. Neural Netw 170:622–634
48. Zhang X, Zeng H, Guo S, et al (2022b) Efficient long-range attention network for image super-resolution. In: European conference on computer vision, Springer, pp 649–667
49. Zhang Y, Li K, Li K, et al (2018a) Image super-resolution using very deep residual channel attention networks. In: Proceedings of the European conference on computer vision (ECCV), pp 286–301
50. Zhang Y, Li K, Li K, et al (2018b) Image super-resolution using very deep residual channel attention networks. In: Proceedings of the European conference on computer vision (ECCV), pp 286–301
51. Zhang Y, Li K, Li K, et al (2018c) Image super-resolution using very deep residual channel attention networks. In: Proceedings of the European conference on computer vision (ECCV), pp 286–301

52. Zhou Y, Li Z, Guo CL, et al (2023) Srformer: Permuted self-attention for single image super-resolution. In: Proceedings of the IEEE/CVF international conference on computer vision, pp 12780–12791
53. Zhu L, Liao B, Zhang Q, et al (2024) Vision mamba: efficient visual representation learning with bidirectional state space model. arXiv:2401.09417
54. Zhu S, Zhao Z, Fang P, et al (2023) Improving scene text image super-resolution via dual prior modulation network. In: Proceedings of the AAAI conference on artificial intelligence, pp 3843–3851