WARP: Wide Attention with Rich Projections for Image Super-Resolution

Jiwon Kim[1] and Kyoung Mu Lee[1,2]

[1]Dept. of ECE, [2]ASRI & IPAI, Seoul National University, Republic of Korea
{j.kim, kyoungmu}@snu.ac.kr

Abstract. State-of-the-art super-resolution (SR) transformers stack 16–30+ blocks of small shifted-window attention. Information between distant positions must therefore traverse multiple attention layers, each bounded by the local window—an indirect path that grows with network depth. We show that *direct* wide attention over large position-fixed windows (64×64 at training, up to 128×128 at inference) can match or surpass these deep narrow-window designs with only 12 blocks. A key enabler is 2D Rotary Position Embeddings (RoPE), which encode relative positions through query-key rotations rather than additive bias matrices, enabling memory-efficient attention over thousands of tokens. To fully exploit this large spatial context, we replace the standard linear QKV projection with a *rich* nonlinear projection module that produces more expressive features for each attention operation; sharing a single module across all blocks keeps the model at 20M parameters while providing the representational capacity that would otherwise require 115M. The shared module jointly produces spatial and channel QKV through a unified output. RoPE further enables resolution-adaptive inference via YaRN-style scaling [22] with an inverted temperature correction tailored for SR. With only 20M parameters, our model, WARP, achieves PSNR competitive with or superior to state-of-the-art methods on standard benchmarks.

Keywords: Super-resolution · Vision transformer · Wide attention · Rotary position embedding

1 Introduction

Single-image super-resolution (SISR) reconstructs a high-resolution (HR) image from its low-resolution (LR) counterpart. The field has evolved from CNN-based architectures [9, 14, 17, 35] to transformer-based designs [5, 6, 16, 18] that leverage self-attention for long-range dependencies. A dominant paradigm has emerged: *narrow windows and deep networks*. SwinIR [16] introduced 8×8 shifted windows for SR, and subsequent methods extended the receptive field via various cross-window mechanisms [5, 18, 32, 36], achieving strong results within the shifted-window paradigm (Fig. 1a).

However, these methods share a fundamental limitation: information between distant positions must propagate *indirectly* through multiple attention layers,

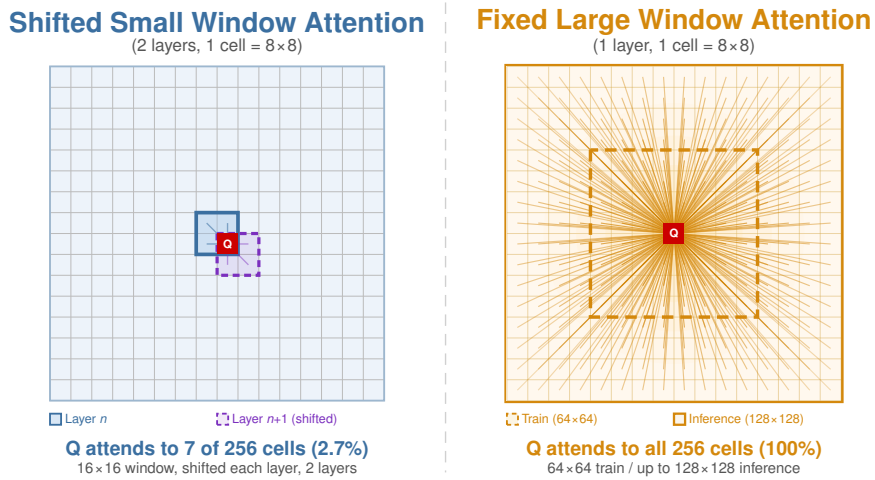

Fig. 1: Shifted small window attention vs. fixed large window attention. Existing methods **(a)** use small shifted windows (e.g., 16×16 pixels); after 2 layers, a query in cell Q reaches only 7 of 256 cells. Our method **(b)** uses a large position-fixed window (64×64 pixels at training, expandable up to 128×128 pixels at inference via fine-tuning), where Q attends to all cells directly in a single layer.

each bounded by the local window. After two layers of 16×16 shifted windows, a query still reaches only 7 of 256 cells (Fig. 1a)—yet natural images abound in self-similar patterns that would benefit from direct long-range communication in a single step.

This motivates *wide attention with a shallow network*. A practical barrier has been the reliance on additive relative position biases (RPB), which require materializing the full $N \times N$ bias matrix in memory, negating the $O(N)$ memory benefit of efficient attention kernels [8] and making large windows memory-prohibitive. We observe that 2D Rotary Position Embeddings (RoPE) [26] sidestep this barrier: by encoding positions through query-key rotations before the dot product, RoPE enables memory-efficient attention with $O(N)$ memory over thousands of tokens. With RoPE, we use position-fixed windows of 64×64 (4,096 tokens) with only 12 blocks, and after fine-tuning, extrapolate to 128×128 via YaRN scaling [22] with an inverted temperature adapted for SR.

To fully exploit this large spatial context, each block needs richer feature extraction than a standard linear projection provides. We replace the linear QKV projections—inherited unchanged from the original transformer [28] across all SR methods—with a rich nonlinear module that produces more expressive Q, K, V features. Since such a module is parameter-heavy, independent projections for 12 blocks would require 115M parameters; we share a single module across all blocks, keeping the model at 20M parameters while retaining the representational capacity that makes wide attention effective. The shared module jointly produces QKV for both spatial and channel attention.

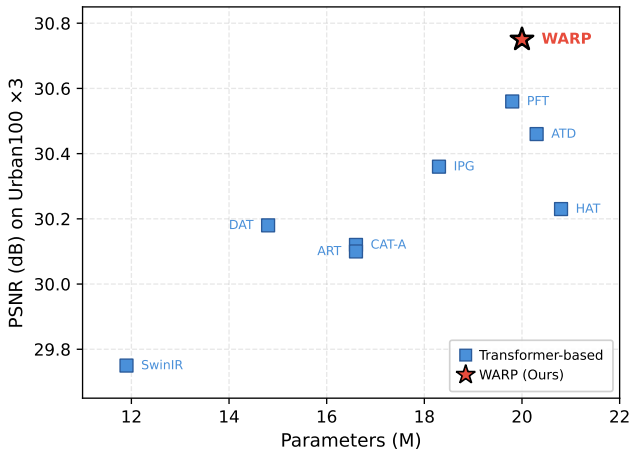

Fig. 2: PSNR vs. parameters on Urban100 ×3. WARP achieves the highest PSNR among all methods at a comparable parameter count to recent transformer-based approaches.

Our model, **W**ide **A**ttention with **R**ich **P**rojections (**WARP**), achieves performance competitive with or superior to state-of-the-art methods with 20M parameters trained on DF2K. For example, on Urban100 ×3, WARP reaches 30.75 dB PSNR, surpassing the previous best (PFT [18], 30.56 dB) by +0.19 dB—a substantial margin in this saturating benchmark (Fig. 2).

Our contributions are:

- Wide attention (64×64 training, up to 128×128 inference) with 2D RoPE and only 12 blocks matches or surpasses methods using 16–30 blocks with shifted windows.
- A rich nonlinear QKV projection shared across all blocks reduces parameters from 115M to 20M while jointly producing spatial and channel QKV.
- We adapt YaRN [22]—an LLM context-extension technique—to 2D vision transformers, finding that SR requires *inverted* temperature scaling ($\tau<1$) rather than the flattening used in LLMs.
- WARP (20M params) achieves the best or tied-best PSNR on all five benchmarks at ×4 and ×3, and on four of five at ×2.

2 Related Work

2.1 Deep Learning for Super-Resolution

CNN-based SR progressed from early architectures [9, 14] through residual designs [17] to channel attention [35]. The transformer era began with IPT [3] (global attention, ImageNet pretraining), and SwinIR [16] established the dominant paradigm of 8×8 shifted windows. Subsequent work focused on overcoming

window locality: overlapping cross-attention [5], alternating spatial-channel attention [6], permuted attention [36], adaptive token dictionaries [32], hierarchical token reduction [33], and progressive focused attention across 30 blocks [18]. All share a design philosophy of small windows combined with depth and cross-window mechanisms. Our work directly attends to the full 64×64 spatial context in each of 12 blocks, replacing indirect multi-hop propagation with a single attention step.

2.2 Position Encoding and Context Extension

RoPE [26] encodes relative positions through query-key rotations without explicit bias terms, extended to 2D for vision by [12]. Unlike RPB, which requires materializing an $N \times N$ bias matrix [8], RoPE enables $O(N)$-memory attention (Sec. 3.2). LLM context-extension methods—positional interpolation [4], YaRN [22]—have not previously been applied to vision transformers for SR; we adapt them to extrapolate from 64×64 training patches to 128×128 inference windows.

2.3 Parameter Sharing

Cross-layer sharing has a history in SR: DRCN [15] applied a single convolutional filter recursively; ELAN [34] reused attention scores across adjacent layers. We extend this to the entire nonlinear QKV projection across all blocks, enabled by position-fixed windows that ensure a homogeneous input domain. State space models [10, 11] and diffusion models [23] are orthogonal to our transformer-focused approach.

3 Method

3.1 Overall Architecture

Our network follows the standard encoder–reconstruction paradigm. Given a low-resolution input $\mathbf{I}_{\text{LR}} \in \mathbb{R}^{H \times W \times 3}$, a 3×3 convolution extracts shallow features $\mathbf{F}_0 \in \mathbb{R}^{H \times W \times C}$. These features pass through L transformer blocks with a global residual connection:

$$\mathbf{I}_{\text{SR}} = \text{PixelShuffle}\big(\text{Conv}_{3\times3}(\mathbf{F}_0 + \mathbf{F}_L)\big) + \text{Upsample}(\mathbf{I}_{\text{LR}}), \tag{1}$$

where PixelShuffle [25] rearranges channels into spatial dimensions and Upsample denotes bilinear interpolation. The final 3×3 convolution weights are initialized to zero so that $\mathbf{I}_{\text{SR}} = \text{Upsample}(\mathbf{I}_{\text{LR}})$ at initialization, ensuring the network starts as a bilinear upsampler.

Each transformer block consists of a dual attention module and a SwiGLU feed-forward network, both with pre-LayerNorm and residual connections:

$$\mathbf{F}' = \mathbf{F} + \text{DualAttn}(\text{LN}(\mathbf{F})), \tag{2}$$

$$\mathbf{F}'' = \mathbf{F}' + \text{FFN}_{\text{SwiGLU}}(\text{LN}(\mathbf{F}')). \tag{3}$$

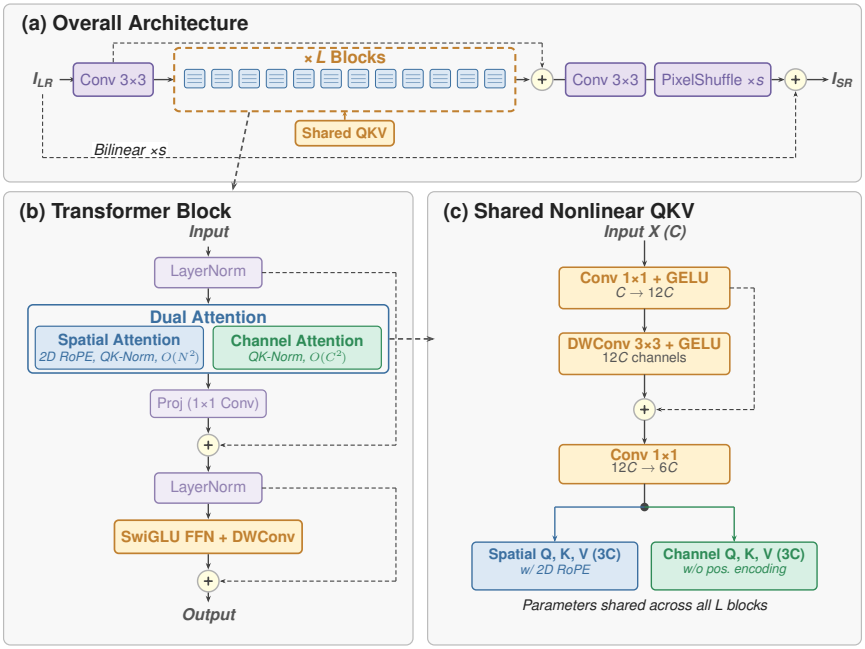

Fig. 3: Overview of WARP. (a) Overall network with shallow feature extraction, L stacked transformer blocks sharing a single nonlinear QKV projection, and PixelShuffle reconstruction. (b) Transformer block: dual attention (spatial + channel) with 2D RoPE operates on large position-fixed windows. (c) Rich nonlinear QKV projection module shared across all blocks.

A key design feature is that all L blocks share a single nonlinear QKV projection module (Fig. 3), which produces a $6C$-dimensional output split into spatial and channel QKV triplets (Sec. 3.5). Each block maintains its own QK-normalization layers and output projection for block-specific adaptation.

3.2 Wide Attention with 2D RoPE

To avoid the indirect path of deep narrow-window stacking, WARP uses *wide attention*: rather than small local windows (e.g., 8×8), we use large position-fixed windows that cover the entire 64×64 training patch. Every token attends to all $N = H \times W = 4{,}096$ tokens within the window, providing full spatial context without requiring window shifting or cross-window communication.

Position encoding in large fixed windows is critical not only for generalization but also for memory efficiency. Existing SR transformers use learned relative position biases (RPB), adding an $N \times N$ bias matrix $\mathbf{B}$ to the attention scores: $\text{softmax}((\mathbf{QK}^\top + \mathbf{B})/\sqrt{d_h})$. This forces materialization of the full bias matrix, since the dense bias must be loaded from memory for every tile—imposing a provable $\Omega(N^2)$ memory-access lower bound that negates the I/O

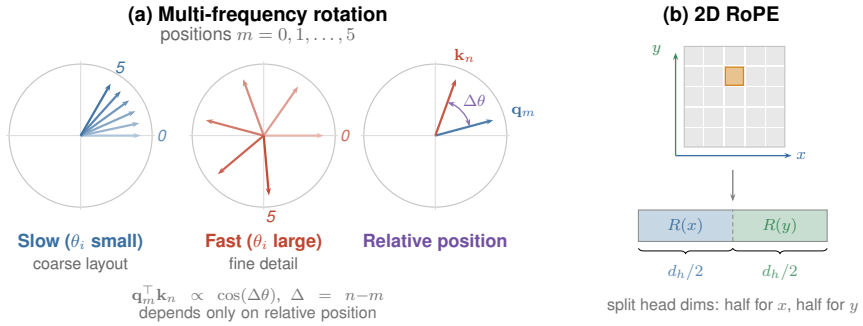

Fig. 4: Rotary Position Embedding (RoPE). (a) RoPE encodes position by rotating query and key vectors. Slow-rotating dimensions capture coarse layout; fast-rotating ones resolve fine detail. The dot product depends only on relative displacement. (b) For 2D images, each head is split in half: one half encodes the horizontal coordinate x, the other the vertical y. At inference, we apply YaRN scaling [22] to extrapolate beyond the training window size (Sec. 3.2).

savings of memory-efficient kernels [8]. For 64×64 windows ($N = 4{,}096$), this means $\sim$16.8M entries per head, making large windows impractical under RPB.

We adopt 2D Rotary Position Embeddings (RoPE) [26], which rotate Q and K *independently* by position-dependent angles before the dot product. For each pair of adjacent dimensions i, a vector at position m is rotated by angle $m\theta_i$:

$$R_{m,i} = \begin{bmatrix} \cos m\theta_i & -\sin m\theta_i \\ \sin m\theta_i & \cos m\theta_i \end{bmatrix}, \quad \theta_i = b^{-2i/(d_h/2)}, \tag{4}$$

where b is the base frequency (we use b=10,000), $d_h/2$ is the per-axis dimension in 2D RoPE, and the frequencies θ_i decrease geometrically (Fig. 4a). The dot product $(R_m\mathbf{q})^\top(R_n\mathbf{k}) = \mathbf{q}^\top R_{n-m}\mathbf{k}$ depends only on the relative displacement $n-m$, so no additive $N\times N$ bias matrix is needed—making RoPE fully compatible with memory-efficient SDPA at $O(N)$ memory. For 2D spatial positions (Fig. 4b), we split each head dimension in half:

$$\mathrm{RoPE}(\mathbf{q}, x, y) = R(x) \cdot \mathbf{q}_{1:d_h/2} \,\|\, R(y) \cdot \mathbf{q}_{d_h/2+1:d_h}, \tag{5}$$

where $R(\cdot)$ is the rotary matrix from Eq. (4) and $\|$ denotes concatenation.

Resolution-Adaptive Inference via YaRN Scaling. At inference, input images are typically larger than the 64×64 training patches. We partition the feature map into overlapping tiles and apply attention independently within each tile, averaging the outputs in the overlap regions. Crucially, the QKV projection, depthwise convolutions, and FFN operate on the *full* feature map; only the attention computation is tiled. With 16-pixel tile overlap, these global operations propagate information across the entire image.

RoPE degrades for windows larger than the training size: high-frequency components encounter unseen rotation angles, and the increased token count shifts the attention distribution. We adapt YaRN [22] to 2D RoPE to address both issues. First, NTK-aware frequency scaling adjusts each frequency band: given extrapolation ratio $s = T/T_0$ (tile size T, training size $T_0 = 64$),

$$\theta_i' = \theta_i \cdot s^{-i/(d_h/4-1)}, \tag{6}$$

where $d_h/4$ is the number of frequency indices per spatial axis (each axis receives $d_h/2$ dimensions in 2D RoPE, forming $d_h/4$ rotation pairs). At inference, the lower-frequency bands additionally receive YaRN's interpolation (a T_0/T factor), completing the YaRN scheme alongside this NTK ramp.

Inverted Temperature for SR. YaRN also prescribes a temperature correction $\sqrt{0.1 \ln s + 1}$ that *flattens* attention to accommodate longer sequences. For SR, however, doubling the window adds spatially *redundant* tokens (e.g., homogeneous background) while the relevant texture remains localized; attention must become *sharper*, not flatter (Tab. 2 confirms this empirically). We therefore apply an inverted temperature $\tau<1$:

$$\mathrm{Attn}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \mathrm{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\tau\sqrt{d_h}}\right)\mathbf{V}. \tag{7}$$

We determine τ per scale via a sweep on DF2K validation patches; larger tiles require lower τ: $\tau = 0.80, 0.83, 0.89$ for $\times 2, \times 3, \times 4$. Since lower upscaling factors yield larger LR feature maps, we allocate larger tiles to $\times 2$ and smaller ones to $\times 4$: T=128, 112, 96 for $\times 2, \times 3, \times 4$, respectively, with 16-pixel overlap to ensure boundary pixels retain sufficient local context. A fine-tuning phase adapts the model to these extrapolated sizes (Sec. 4.2). The position-fixed window layout enables cross-block QKV sharing (Sec. 3.4).

3.3 Rich Nonlinear QKV Projection

With only 12 blocks operating on up to $128 \times 128 = 16{,}384$ tokens, each block must maximize information transfer. Standard SR transformers project input features to Q, K, V via a single linear operation: $[\mathbf{Q}, \mathbf{K}, \mathbf{V}] = \mathbf{X}\mathbf{W}_{QKV}$. Here $\mathbf{X} \in \mathbb{R}^{N \times C}$ contains N tokens with C channels, and $\mathbf{W}_{QKV} \in \mathbb{R}^{C \times 3C}$ maps each token to a concatenated Q, K, V triplet that is then split along the channel dimension. We replace this with a nonlinear module that maps $\mathbf{X} \in \mathbb{R}^{H \times W \times C}$ to a $6C$-dimensional output through a two-layer network with spatial mixing:

$$\mathbf{H} = \mathrm{GELU}\big(\mathrm{Conv}_{1\times1}(\mathbf{X})\big), \quad \mathbf{H} \in \mathbb{R}^{H \times W \times E}, \tag{8}$$

$$\mathbf{H}' = \mathrm{GELU}\big(\mathrm{DWConv}_{3\times3}(\mathbf{H})\big) + \mathbf{H}, \tag{9}$$

$$\mathbf{Z} = \mathrm{Conv}_{1\times1}(\mathbf{H}'), \quad \mathbf{Z} \in \mathbb{R}^{H \times W \times 6C}, \tag{10}$$

where $E = r \cdot C$ is the expanded hidden dimension with expansion ratio r (we use $r = 12$). The first 1×1 convolution expands the feature dimension; GELU

provides nonlinear feature selection; the 3×3 depthwise convolution introduces local spatial context with a residual connection; and the final 1×1 convolution compresses to $6C$ dimensions, producing QKV for both spatial and channel attention (Sec. 3.5).

The GELU nonlinearity enables data-dependent feature selection—different inputs activate different channels of the expanded representation, producing more diverse Q, K, V patterns than a linear projection. The depthwise convolution captures local spatial context before attention, complementing the long-range spatial mixing from attention itself.

3.4 Shared Projection Design

The nonlinear projection is parameter-heavy: for our base model ($C = 320$, $L = 12$, $r = 12$), giving each block its own projection would inflate the model from 20M to 115M parameters. We share a *single* projection module across all blocks, enabled by the position-fixed window design (Sec. 3.2): since all blocks observe the same unshifted feature map, a single projection can serve all of them.

The expanded hidden dimension ($E = 12C$) provides a rich intermediate representation from which different blocks can extract different aspects. Each block retains its own per-head QK-normalization (LN on Q and K) and output projection (1×1 convolution with C^2 parameters). These lightweight per-block layers let each block rescale and recombine the shared Q, K, V features differently. The projection module itself runs at every block, but its $\sim$8.6M parameters are stored once and shared across all L blocks, so the parameter cost is amortized.

The shared design reduces parameters by $5.75\times$ (20M vs. 115M), making the rich nonlinear projection practical. Conceptually, this resembles weight tying in recurrent architectures, but applied only to the projection—each block retains distinct attention patterns through its own output projection and normalization.

3.5 Dual Spatial-Channel Attention

The shared projection produces a $6C$-dimensional output $\mathbf{Z}$, split into spatial and channel QKV triplets:

$$[\mathbf{Q}_s, \mathbf{K}_s, \mathbf{V}_s, \mathbf{Q}_c, \mathbf{K}_c, \mathbf{V}_c] = \text{split}(\mathbf{Z}), \quad \text{each} \in \mathbb{R}^{H \times W \times C}, \tag{11}$$

where subscripts s and c denote spatial and channel attention, respectively.

Spatial Attention. After per-head QK-normalization, 2D RoPE (Sec. 3.2) is applied to $\mathbf{Q}_s$ and $\mathbf{K}_s$:

$$\mathbf{O}_{\text{spatial}} = \text{softmax}\left(\frac{\text{RoPE}(\tilde{\mathbf{Q}}_s) \cdot \text{RoPE}(\tilde{\mathbf{K}}_s)^\top}{\tau\sqrt{d_h}}\right)\mathbf{V}_s, \tag{12}$$

where $\tilde{\mathbf{Q}}_s = \text{LN}(\mathbf{Q}_s)$, $\tilde{\mathbf{K}}_s = \text{LN}(\mathbf{K}_s)$, $d_h = C/n_h$ is the head dimension, n_h is the number of heads, and τ is the temperature factor from Eq. (7) ($\tau = 1$ during base training, $\tau < 1$ at inference to sharpen attention over larger windows).

Channel Attention. Channel attention transposes the operation to compute affinities between channels rather than spatial positions:

$$\mathbf{A}_{\text{ch}} = \text{softmax}\left(\frac{\tilde{\mathbf{Q}}_c^\top \tilde{\mathbf{K}}_c}{\sqrt{N}}\right), \quad \mathbf{O}_{\text{channel}} = \mathbf{V}_c \mathbf{A}_{\text{ch}}, \tag{13}$$

where $\mathbf{A}_{\text{ch}} \in \mathbb{R}^{d_h \times d_h}$ captures inter-channel relationships, scaled by $\sqrt{N}$, with N denoting the number of spatial positions.

Merging. The outputs are combined by element-wise addition:

$$\mathbf{O} = \text{Conv}_{1\times1}(\mathbf{O}_{\text{spatial}} + \mathbf{O}_{\text{channel}})\,. \tag{14}$$

Rather than alternating spatial and channel attention across separate blocks, our approach computes both in every block from the single shared projection. The spatial and channel QKV are simply carved from different dimensions of the same $6C$ output, requiring no separate projection or pathway.

3.6 Feed-Forward Network

Following each dual attention module, we apply a SwiGLU [24] feed-forward network with depthwise convolution:

$$\text{FFN}(\mathbf{X}) = \mathbf{W}_{\text{out}}\big(\sigma(\text{DWConv}(\mathbf{H})) + \mathbf{H}\big), \quad \mathbf{H} = \sigma(\mathbf{W}_{\text{gate}}\mathbf{X}) \odot \mathbf{W}_{\text{up}}\mathbf{X}, \tag{15}$$

where σ is the SiLU activation and $\odot$ denotes element-wise multiplication. The expansion ratio is $8/3$, following the standard SwiGLU convention.

4 Experimental Results

4.1 Datasets and Metrics

Training Data. We train on DF2K, combining 800 images from DIV2K [1] and 2,650 from Flickr2K [17] (3,450 total).

Evaluation Benchmarks. We evaluate on five standard benchmarks: Set5 [2], Set14 [30], BSD100 [19], Urban100 [13], and Manga109 [20].

Protocol. All metrics are computed on the Y channel (luminance) of YCbCr, with a border crop equal to the upscaling factor. We report PSNR (dB) and SSIM [29].

4.2 Implementation Details

WARP uses $C = 320$, $L = 12$ blocks, $n_h = 10$ attention heads, and expansion ratio $r = 12$ for the shared nonlinear QKV projection, totaling 20.0M parameters. Training proceeds in two phases.

Phase 1: Base Training (500K iterations). LR images are generated via bicubic downsampling. Training patches of size 64×64 (LR) / 256×256 (HR for $\times 4$) are randomly cropped with horizontal flips and 90° rotations. We minimize L1 loss with Adam ($\beta_1 = 0.9$, $\beta_2 = 0.99$, no weight decay) and a Warmup-Stable-Decay (WSD) schedule: 1K warmup to 2×10^{-4}, stable phase, and 10% cosine decay to 10^{-6}. EMA with decay 0.999 is maintained. Batch size is 32.

Phase 2: YaRN Fine-Tuning (50K iterations). Starting from Phase 1 EMA weights, we fine-tune at the per-scale tile size T with YaRN-scaled RoPE and inverted temperature τ, adapting the model to extrapolated window sizes. At inference, edge tiles smaller than T use YaRN scaling and temperature adapted to their own size (reducing to the training setting as the tile approaches T_0), rather than the full-tile values. LR is 1×10^{-4}, with L1 loss and EMA decay 0.999. Per-scale settings: T=128, τ=0.80 ($\times 2$); T=112, τ=0.83 ($\times 3$); T=96, τ=0.89 ($\times 4$).

4.3 Comparison with State-of-the-Art

Tab. 1 compares WARP with state-of-the-art methods across $\times 2$, $\times 3$, and $\times 4$ SR.

At $\times 2$, WARP achieves the best PSNR on Set5, Urban100, and Manga109, ties for best on BSD100, and only narrowly trails PFT on Set14 (34.99 vs. 35.00). For $\times 3$, WARP leads on all five benchmarks—+0.09 dB on Set5, +0.19 dB on Urban100, and +0.12 dB on Manga109 over PFT. For $\times 4$, it achieves the best or tied-best PSNR on all five benchmarks, with gains on Urban100 (+0.06 dB) and Manga109 (+0.10 dB). The largest improvements appear on Urban100 and Manga109—benchmarks with large images containing repetitive structures where direct long-range attention captures self-similar patterns in a single layer that small windows can only reach through multi-hop propagation. On smaller benchmarks (Set5, Set14, BSD100), where local context is often sufficient, our method remains competitive, but the margin narrows. This pattern is consistent across all three upscaling factors, confirming that wide attention with resolution-adaptive inference generalizes robustly across the SR operating range.

With 20M parameters and only 12 transformer blocks, WARP achieves these results through a structurally distinct design, demonstrating that wide attention with rich shared projections is a viable alternative to the dominant narrow-and-deep paradigm. Notably, this is achieved with a standard self-attention mechanism—without shifted windows, cross-attention layers, or auxiliary training losses—suggesting that investing capacity in the input projection can substitute for additional architectural complexity.

4.4 Visual Comparisons

Fig. 5 presents $\times 4$ visual comparisons. In `img033`, competing methods merge many thin siding stripes into a few broad bands, while WARP preserves every distinct line. In `barbara`, narrow-window methods alias the fine woven fabric into

Table 1: Quantitative comparison for ×2, ×3, and ×4 SR. Best and second best results are highlighted. PSNR (dB) / SSIM.

Method	Train	Scale	Params	Set5	Set14	BSD100	Urban100	Manga109
EDSR [17]	DIV2K	×2	42.6M	38.11 / 0.9602	33.92 / 0.9195	32.32 / 0.9013	32.93 / 0.9351	39.10 / 0.9773
RCAN [35]	DIV2K	×2	15.4M	38.27 / 0.9614	34.12 / 0.9216	32.41 / 0.9027	33.34 / 0.9384	39.44 / 0.9786
HAN [21]	DIV2K	×2	63.6M	38.27 / 0.9614	34.16 / 0.9217	32.41 / 0.9027	33.35 / 0.9385	39.46 / 0.9785
SwinIR [16]	DF2K	×2	11.8M	38.42 / 0.9623	34.46 / 0.9250	32.53 / 0.9041	33.81 / 0.9433	39.92 / 0.9797
CAT-A [7]	DF2K	×2	16.5M	38.51 / 0.9626	34.78 / 0.9265	32.59 / 0.9047	34.26 / 0.9440	40.10 / 0.9805
ART [31]	DF2K	×2	16.4M	38.56 / 0.9629	34.59 / 0.9267	32.58 / 0.9048	34.30 / 0.9452	40.24 / 0.9808
DAT [6]	DF2K	×2	14.8M	38.58 / 0.9629	34.81 / 0.9272	32.61 / 0.9055	34.37 / 0.9458	40.18 / 0.9809
HAT [5]	DF2K	×2	20.6M	38.63 / 0.9630	34.86 / 0.9274	32.62 / 0.9053	34.45 / 0.9466	40.26 / 0.9809
IPG [27]	DF2K	×2	18.1M	38.61 / 0.9632	34.73 / 0.9270	32.60 / 0.9052	34.48 / 0.9464	40.24 / 0.9810
ATD [32]	DF2K	×2	20.1M	38.61 / 0.9629	34.95 / 0.9276	32.65 / 0.9056	34.70 / 0.9476	40.37 / 0.9810
PFT [18]	DF2K	×2	19.6M	38.68 / 0.9635	35.00 / 0.9280	32.67 / 0.9058	34.90 / 0.9490	40.49 / 0.9815
WARP (Ours)	DF2K	×2	20.0M	38.71 / 0.9633	34.99 / 0.9278	32.67 / 0.9058	34.99 / 0.9494	40.52 / 0.9813
EDSR [17]	DIV2K	×3	43.0M	34.65 / 0.9280	30.52 / 0.8462	29.25 / 0.8093	28.80 / 0.8653	34.17 / 0.9476
RCAN [35]	DIV2K	×3	15.6M	34.74 / 0.9299	30.65 / 0.8482	29.32 / 0.8111	29.09 / 0.8702	34.44 / 0.9499
HAN [21]	DIV2K	×3	64.2M	34.75 / 0.9299	30.67 / 0.8483	29.32 / 0.8110	29.10 / 0.8705	34.48 / 0.9500
SwinIR [16]	DF2K	×3	11.9M	34.97 / 0.9318	30.93 / 0.8534	29.46 / 0.8145	29.75 / 0.8826	35.12 / 0.9537
CAT-A [7]	DF2K	×3	16.6M	35.06 / 0.9326	31.04 / 0.8538	29.52 / 0.8160	30.12 / 0.8862	35.38 / 0.9546
ART [31]	DF2K	×3	16.6M	35.07 / 0.9325	31.02 / 0.8541	29.51 / 0.8159	30.10 / 0.8871	35.39 / 0.9548
DAT [6]	DF2K	×3	14.8M	35.07 / 0.9324	31.06 / 0.8547	29.54 / 0.8170	30.18 / 0.8886	35.43 / 0.9550
HAT [5]	DF2K	×3	20.8M	35.07 / 0.9329	31.08 / 0.8555	29.54 / 0.8167	30.23 / 0.8896	35.53 / 0.9552
IPG [27]	DF2K	×3	18.3M	35.10 / 0.9332	31.10 / 0.8554	29.53 / 0.8168	30.36 / 0.8901	35.53 / 0.9554
ATD [32]	DF2K	×3	20.3M	35.11 / 0.9330	31.13 / 0.8556	29.57 / 0.8176	30.46 / 0.8917	35.63 / 0.9558
PFT [18]	DF2K	×3	19.8M	35.15 / 0.9333	31.16 / 0.8561	29.58 / 0.8178	30.56 / 0.8931	35.67 / 0.9560
WARP (Ours)	DF2K	×3	20.0M	35.24 / 0.9337	31.16 / 0.8560	29.59 / 0.8179	30.75 / 0.8952	35.79 / 0.9564
EDSR [17]	DIV2K	×4	43.0M	32.46 / 0.8968	28.80 / 0.7876	27.71 / 0.7420	26.64 / 0.8033	31.02 / 0.9148
RCAN [35]	DIV2K	×4	15.6M	32.63 / 0.9002	28.87 / 0.7889	27.77 / 0.7436	26.82 / 0.8087	31.22 / 0.9173
HAN [21]	DIV2K	×4	64.2M	32.64 / 0.9002	28.90 / 0.7890	27.80 / 0.7442	26.85 / 0.8094	31.42 / 0.9177
SwinIR [16]	DF2K	×4	11.9M	32.92 / 0.9044	29.09 / 0.7950	27.92 / 0.7489	27.45 / 0.8254	32.03 / 0.9260
CAT-A [7]	DF2K	×4	16.6M	33.08 / 0.9052	29.18 / 0.7960	27.99 / 0.7510	27.89 / 0.8339	32.39 / 0.9285
ART [31]	DF2K	×4	16.6M	33.04 / 0.9051	29.16 / 0.7958	27.97 / 0.7510	27.77 / 0.8321	32.31 / 0.9283
DAT [6]	DF2K	×4	14.8M	33.08 / 0.9055	29.23 / 0.7973	28.00 / 0.7518	27.87 / 0.8343	32.51 / 0.9291
HAT [5]	DF2K	×4	20.8M	33.04 / 0.9056	29.23 / 0.7973	28.00 / 0.7517	27.97 / 0.8368	32.48 / 0.9292
IPG [27]	DF2K	×4	17.0M	33.15 / 0.9062	29.24 / 0.7973	27.99 / 0.7519	28.13 / 0.8392	32.53 / 0.9300
ATD [32]	DF2K	×4	20.3M	33.10 / 0.9058	29.24 / 0.7974	28.01 / 0.7526	28.17 / 0.8404	32.62 / 0.9306
PFT [18]	DF2K	×4	19.8M	33.15 / 0.9065	29.29 / 0.7978	28.02 / 0.7527	28.20 / 0.8412	32.63 / 0.9306
WARP (Ours)	DF2K	×4	20.0M	33.18 / 0.9054	29.30 / 0.7987	28.04 / 0.7524	28.26 / 0.8431	32.73 / 0.9312

coarser diagonal bands, while WARP more faithfully recovers the regular weave. In `UchuKigeki`, the repeating diamond pattern degenerates into rounded blobs; WARP reconstructs sharp four-pointed shapes. The text in `MukoukizuN` stays legible only in our result. The straight edges of the star-lattice in `MomoyamaHa` curve into amorphous blobs under narrow-window methods, while WARP recovers the angular geometry. In each case, the distinguishing patterns span distances well beyond local window boundaries, illustrating the practical benefit of wide attention for structurally complex images.

4.5 Ablation Study

We ablate key design choices on ×2 SR, where the extrapolation ratio is largest ($s = 2.0$: training 64×64, inference 128×128). YaRN scaling and temperature experiments use 1,000 DF2K patches.

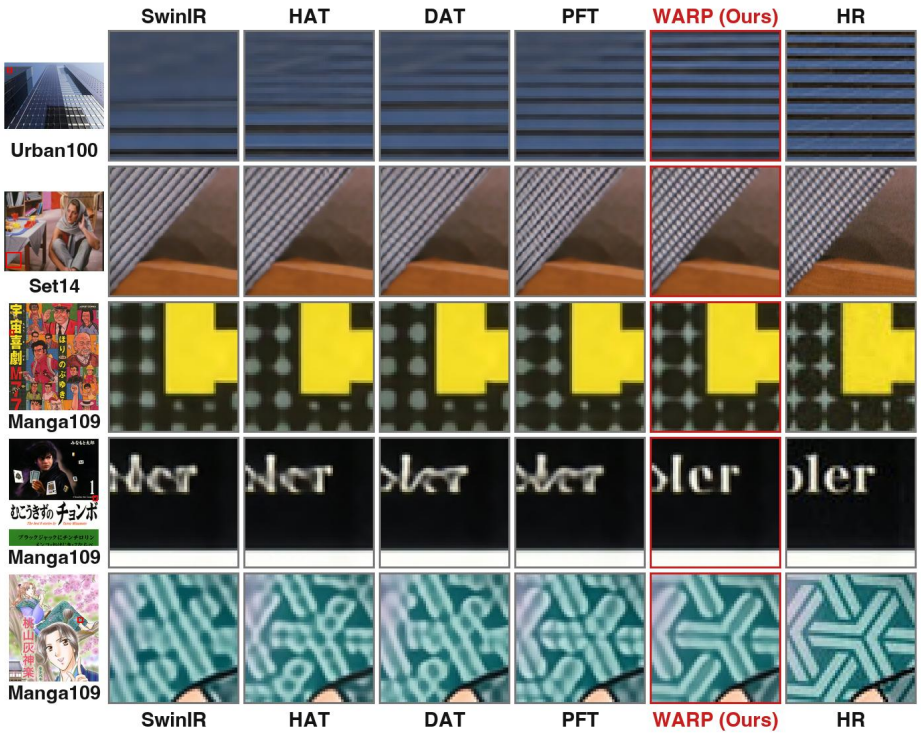

Fig. 5: Visual comparison on challenging ×4 SR examples spanning Urban100, Set14, and Manga109. WARP's wide attention recovers long-range structural patterns—periodic stripes, woven fabric, lattices, and text—that small-window methods blur or alias. Rows (top to bottom): `img033` (Urban100), `barbara` (Set14), `UchuKigeki`, `MukoukizuN`, `MomoyamaHa` (Manga109).

YaRN Scaling Analysis. Tab. 2 dissects YaRN scaling for resolution extrapolation. Naïvely doubling the tile size degrades PSNR (e.g., −0.13 dB on Urban100). YaRN's prescribed temperature ($\sqrt{0.1 \ln s + 1} \approx 1.03$) *flattens* softmax—useful in LLMs but harmful in SR, where larger windows mostly add redundant tokens—costing a further −0.14 dB on Urban100. Our inverted temperature τ=0.80 instead *sharpens* attention, gaining +0.23 dB on Urban100 *without fine-tuning* (surpassing even the training-resolution baseline), with fine-tuning adding +0.04 dB.

Temperature Selection. Sweeping τ on 1,000 DF2K patches (500K model, *before* fine-tuning), PSNR peaks at τ=0.80 with a broad plateau (±0.05 in τ yields <0.05 dB); the same protocol selects τ=0.83 for ×3 (s=1.75) and τ=0.89 for ×4 (s=1.5).

Wider Attention Windows. Fig. 6 shows wider inference tiles consistently improve PSNR on Urban100 (+0.44 dB from tile 48 to 128, most gain below s=1.5);

Table 2: YaRN scaling for resolution extrapolation (×2 SR, tile 64 → 128, $s = 2.0$, 500K pretrained). DF2K: 1,000 patches; Urban100: full benchmark.

Configuration	DF2K	Urban100
500K, tile = 64 (training res.)	38.64	34.85
500K, tile = 128 (naïve extrap.)	38.58	34.72
+ YaRN (standard $\tau \approx 1.03$)	38.54	34.58
+ YaRN (our τ=0.80)	38.67	34.95
+ YaRN (τ=0.80) + 50K FT	**38.70**	**34.99**

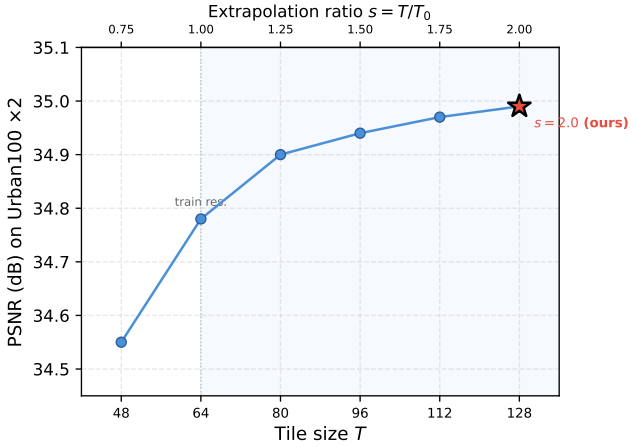

Fig. 6: Effect of inference tile size on Urban100 PSNR (×2 SR, fine-tuned model, YaRN scaling). Train tile = 64. Wider windows monotonically improve PSNR, with diminishing returns beyond s=1.5.

sub-training resolution (s=0.75) also works without retraining, confirming 2D RoPE generalizes to shorter and longer sequences alike. To confirm this reflects window size itself and not YaRN extrapolation, we further train four WARP variants *from scratch* (C=320, ×4, 200K iters) that differ *only* in window size (Tab. 3): PSNR rises strictly monotonically on all five benchmarks, with gains tracking long-range repetition—modest on smooth Set5 (+0.48 dB) but large on texture-rich Urban100/Manga109 (+1.24/+1.11 dB)—isolating window size, not other design choices, as the source of the improvement. The wide window stays tractable: 2D RoPE keeps attention in the $O(N)$ FlashAttention regime (no $N \times N$ bias matrix), and window-independent QKV/FFN computation dominates the budget, so compute grows only modestly as the window widens. Local Attribution Maps (supplementary) further confirm WARP's markedly wider effective receptive field than narrow-window baselines.

QKV Projection Design. Tab. 4 isolates each design choice with iso-parameter variants (C=96, ∼0.9M): *linear* replaces the nonlinear projection with a stan-

Table 3: From-scratch window-size ablation (×4 SR, C=320, 200K iters, training-tile inference). Four variants differ *only* in window size; all else is identical. PSNR (dB) rises monotonically on every benchmark, isolating window size as the causal driver of WARP's gains (largest on texture-rich Urban100/Manga109).

Window	Set5	Set14	B100	Urban100	Manga109
16×16	32.65	28.93	27.74	26.84	31.40
32×32	32.96	29.12	27.90	27.49	32.05
48×48	33.05	29.18	27.97	27.83	32.31
64×64	**33.13**	**29.27**	**28.01**	**28.08**	**32.51**

Table 4: Ablation of QKV projection design (C=96, ×4 SR, Set5, 100K iters). Iso-parameter variants match the baseline (∼0.9M) by adjusting expansion ratio r or depth L.

Configuration	Params	L	PSNR	Δ
Full model (shared + nonlinear + dual)	∼0.9M	4	32.48	—
− Nonlinear (linear QKV)	∼0.9M	4	32.18	−0.30
− Nonlinear (linear QKV, L=9)	∼0.9M	9	32.47	−0.01
− Shared (independent, r=2)	∼0.9M	4	32.41	−0.07
− Dual (spatial-only, r=14)	∼0.9M	4	32.43	−0.05

dard linear one (compensating with depth), *independent* uses per-block rather than shared projections, and *spatial-only* removes the channel-attention branch. At equal depth, replacing the nonlinear projection with a linear one drops PSNR by 0.30 dB. Recovering this gap requires increasing depth to L=9 (−0.01 dB) at the cost of 2.25× more $O(N^2)$ attention passes, showing that nonlinear projections trade cheap pointwise ops for expensive attention blocks. Sharing outperforms independent projections by 0.07 dB at matched capacity, and dual attention adds 0.05 dB over spatial-only. The iso-parameter comparison is key: at this C=96 scale, the full design matches or exceeds each variant at matched model size, confirming that the gains stem from architectural choices rather than increased capacity. We repeat this ablation at the full C=320 scale on five benchmarks (Tab. 5). Removing the nonlinear projection lowers PSNR by 0.02–0.04 dB on the natural-image benchmarks (Set5/Set14/B100) and by 0.16/0.14 dB on the texture-rich Urban100/Manga109, consistent with the small-scale ablation (Tab. 4) and the mid-scale (C=192) result in the supplementary; the rich-projection benefit persists, concentrated on the texture-heavy benchmarks. At this scale a non-shared projection is impractical, inflating the model from 20M to ∼115M parameters. By contrast, the spatial-only (−dual) variant, which reinvests the channel-branch budget into additional depth (L=16 vs 12), is competitive with the full model at C=320 in this single-run comparison—within 0.04 dB on Set5/Set14/B100 and higher by 0.14 dB on Urban100/Manga109. We retain dual attention—the configuration used for all main results (Tab. 1) and

Table 5: Full-scale QKV ablation (C=320, ×4 SR, trained from scratch for 200K iters, iso-parameter; non-tiled full-LR inference). PSNR (dB) on five benchmarks; *single run per variant*. The independent (non-shared) variant is omitted as it inflates the model to ∼115M parameters.

Configuration	Params	Set5	Set14	B100	Urban100	Manga109
Full (L=12)	20.0M	33.13	29.23	28.01	27.81	32.22
– Nonlinear (L=20)	19.4M	33.10	29.19	27.99	27.65	32.08
– Dual (L=16)	20.0M	33.16	29.23	28.03	27.95	32.36

beneficial at the smaller scale (Tab. 4)—and leave a fuller characterization of the depth-versus-channel-branch trade-off at scale to future work. These full-scale ablations (Tabs. 3 and 5) train from scratch for 200K iterations, so their absolute PSNR sits below the 500K-plus-fine-tuned deployed model of Tab. 1.

Inference Time. We measure WARP on an RTX 4090 at 512^2 output following [18]; baseline latencies are quoted from its supplementary. At the training tile size, WARP runs in 1148/653/366 ms (×2/×3/×4), vs. HAT 1078/799/725, PFT 1594/1158/852, ATD 1394/1038/867, and IPG 2320/1651/1060 ms. With YaRN tiles, latency rises to 3664/1053/590 ms; WARP stays fastest at ×4 (590 ms vs. HAT 725, PFT 852), trading speed for quality at the larger ×2/×3 tiles.

5 Conclusion

Direct attention over wide spatial context, with a rich shared projection, is a viable alternative to deep narrow-window stacking. With only 12 blocks, WARP matches or surpasses 30+ block methods. The key ingredients—2D RoPE for memory-efficient large windows (keeping wide attention compatible with FlashAttention [8]), a shared nonlinear projection for rich QKV features, and YaRN scaling with inverted temperature that lets a single model serve multiple inference resolutions—are each individually simple, yet their combination produces a structurally distinct and competitive design point in the SR landscape. More broadly, these results suggest that the depth–width tradeoff in SR transformers remains underexplored, and shallower models with richer per-block representations offer a complementary path to the prevailing strategy of deeper stacking.

Limitations and Future Work. The ×2 YaRN latency highlights the cost of quadratic attention at large tile sizes; sparse patterns [18] or mixture-of-experts gating could mitigate this. Learning τ end-to-end (rather than via a per-scale sweep) and extending the shared projection to other dense prediction tasks (e.g., denoising, deblurring) are natural further directions.

References

1. Agustsson, E., Timofte, R.: NTIRE 2017 challenge on single image super-resolution: Dataset and study. In: CVPRW. pp. 126–135 (2017)
2. Bevilacqua, M., Roumy, A., Guillemot, C., Alberi-Morel, M.L.: Low-complexity single-image super-resolution based on nonnegative neighbor embedding. In: BMVC. pp. 1–10 (2012)
3. Chen, H., Wang, Y., Guo, T., Xu, C., Deng, Y., Liu, Z., Ma, S., Xu, C., Xu, C., Gao, W.: Pre-trained image processing transformer. In: CVPR. pp. 12299–12310 (2021)
4. Chen, S., Wong, S., Chen, L., Tian, Y.: Extending context window of large language models via positional interpolation. arXiv preprint arXiv:2306.15595 (2023)
5. Chen, X., Wang, X., Zhou, J., Qiao, Y., Dong, C.: Activating more pixels in image super-resolution transformer. In: CVPR. pp. 22367–22377 (2023)
6. Chen, Z., Zhang, Y., Gu, J., Kong, L., Yang, X., Yu, F.: Dual aggregation transformer for image super-resolution. In: ICCV. pp. 12312–12321 (2023)
7. Chen, Z., Zhang, Y., Gu, J., Zhang, Y., Kong, L., Yuan, X.: Cross aggregation transformer for image restoration. In: NeurIPS (2022)
8. Dao, T., Fu, D., Ermon, S., Rudra, A., Ré, C.: FlashAttention: Fast and memory-efficient exact attention with IO-awareness. In: NeurIPS (2022)
9. Dong, C., Loy, C.C., He, K., Tang, X.: Learning a deep convolutional network for image super-resolution. In: ECCV. pp. 184–199 (2014)
10. Guo, H., Guo, Y., Zha, Y., Zhang, Y., Li, W., Dai, T., Xia, S.T., Li, Y.: MambaIRv2: Attentive state space restoration. In: CVPR (2025)
11. Guo, H., Li, J., Dai, T., Ouyang, Z., Ren, X., Xia, S.T.: MambaIR: A simple baseline for image restoration with state-space model. In: ECCV (2024)
12. Heo, B., Park, S., Han, D., Yun, S.: Rotary position embedding for vision transformer. In: ECCV (2024)
13. Huang, J.B., Singh, A., Ahuja, N.: Single image super-resolution from transformed self-exemplars. In: CVPR. pp. 5197–5206 (2015)
14. Kim, J., Lee, J.K., Lee, K.M.: Accurate image super-resolution using very deep convolutional networks. In: CVPR. pp. 1646–1654 (2016)
15. Kim, J., Lee, J.K., Lee, K.M.: Deeply-recursive convolutional network for image super-resolution. In: CVPR. pp. 1637–1645 (2016)
16. Liang, J., Cao, J., Sun, G., Zhang, K., Van Gool, L., Timofte, R.: SwinIR: Image restoration using Swin transformer. In: ICCV Workshops. pp. 1833–1844 (2021)
17. Lim, B., Son, S., Kim, H., Nah, S., Lee, K.M.: Enhanced deep residual networks for single image super-resolution. In: CVPRW. pp. 136–144 (2017)
18. Long, W., Zhou, X., Zhang, L., Gu, S.: Progressive focused transformer for single image super-resolution. In: CVPR (2025)
19. Martin, D., Fowlkes, C., Tal, D., Malik, J.: A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In: ICCV. pp. 416–423 (2001)
20. Matsui, Y., Ito, K., Aramaki, Y., Fujimoto, A., Ogawa, T., Yamasaki, T., Aizawa, K.: Sketch-based manga retrieval using Manga109 dataset. Multimedia Tools and Applications **76**(20), 21811–21838 (2017)
21. Niu, B., Wen, W., Ren, W., Zhang, X., Yang, L., Wang, S., Zhang, K., Cao, X., Shen, H.: Single image super-resolution via a holistic attention network. In: ECCV (2020)

22. Peng, B., Quesnelle, J., Fan, H., Shippole, E.: YaRN: Efficient context window extension of large language models. In: ICLR (2024)
23. Saharia, C., Ho, J., Chan, W., Salimans, T., Fleet, D.J., Norouzi, M.: Image super-resolution via iterative refinement. IEEE TPAMI **45**(4), 4713–4726 (2023)
24. Shazeer, N.: GLU variants improve transformer. arXiv preprint arXiv:2002.05202 (2020)
25. Shi, W., Caballero, J., Huszár, F., Totz, J., Aitken, A.P., Bishop, R., Rueckert, D., Wang, Z.: Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In: CVPR. pp. 1874–1883 (2016)
26. Su, J., Lu, Y., Pan, S., Murtadha, A., Wen, B., Liu, Y.: RoFormer: Enhanced transformer with rotary position embedding. Neurocomputing **568**, 127063 (2024)
27. Tian, Y., Chen, H., Xu, C., Wang, Y.: Image processing GNN: Breaking rigidity in super-resolution. In: CVPR (2024)
28. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: NeurIPS. pp. 5998–6008 (2017)
29. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. IEEE TIP **13**(4), 600–612 (2004)
30. Zeyde, R., Elad, M., Protter, M.: On single image scale-up using sparse-representations. In: International Conference on Curves and Surfaces. pp. 711–730 (2012)
31. Zhang, J., Zhang, Y., Gu, J., Zhang, Y., Kong, L., Yuan, X.: Accurate image restoration with attention retractable transformer. In: ICLR (2023)
32. Zhang, L., Li, Y., Zhou, X., Zhao, X., Gu, S.: Transcending the limit of local window: Advanced super-resolution transformer with adaptive token dictionary. In: CVPR (2024)
33. Zhang, X., Zhang, Y., Yu, F.: HiT-SR: Hierarchical transformer for efficient image super-resolution. In: ECCV (2024)
34. Zhang, X., Zeng, H., Guo, S., Zhang, L.: Efficient long-range attention network for image super-resolution. In: ECCV. pp. 649–667 (2022)
35. Zhang, Y., Li, K., Li, K., Wang, L., Zhong, B., Fu, Y.: Image super-resolution using very deep residual channel attention networks. In: ECCV. pp. 286–301 (2018)
36. Zhou, Y., Li, Z., Guo, C.L., Bai, S., Cheng, M.M., Hou, Q.: SRFormer: Permuted self-attention for single image super-resolution. In: ICCV. pp. 12780–12791 (2023)