N-Gram in Swin Transformers for Efficient Lightweight Image Super-Resolution

Haram Choi[1] Jeongmin Lee[2] Jihoon Yang[1]*
[1]Department of Computer Science & Engineering, Sogang University [2]LG Innotek

Abstract

While some studies have proven that Swin Transformer (Swin) with window self-attention (WSA) is suitable for single image super-resolution (SR), the plain WSA ignores the broad regions when reconstructing high-resolution images due to a limited receptive field. In addition, many deep learning SR methods suffer from intensive computations. To address these problems, we introduce the N-Gram context to the low-level vision with Transformers for the first time. We define N-Gram as neighboring local windows in Swin, which differs from text analysis that views N-Gram as consecutive characters or words. N-Grams interact with each other by sliding-WSA, expanding the regions seen to restore degraded pixels. Using the N-Gram context, we propose NGswin, an efficient SR network with SCDP bottleneck taking multi-scale outputs of the hierarchical encoder. Experimental results show that NGswin achieves competitive performance while maintaining an efficient structure when compared with previous leading methods. Moreover, we also improve other Swin-based SR methods with the N-Gram context, thereby building an enhanced model: SwinIR-NG. Our improved SwinIR-NG outperforms the current best lightweight SR approaches and establishes state-of-the-art results. Codes are available at https://github.com/rami0205/NGramSwin.

1. Introduction

The goal of single image super-resolution (SR) is to reconstruct high-resolution (HR) images from low-resolution (LR) images. Many deep learning-based methods have worked in this field. In particular, several image restoration studies [16, 27, 55, 61, 63, 66] have adapted the window self-attention (WSA) proposed by Swin Transformer (Swin) [32] as it integrates long-range dependency of Vision Transformer [14] and locality of conventional convolution. However, two critical problems remain in these works. First, the receptive field of the plain WSA is limited within a small local window [52,56,58]. It prevents the models from utilizing the texture or pattern of neighbor windows to recover degraded pixels, producing the distorted images. Second, recent state-of-the-art SR [9,27,61,66] and lightweight SR [6, 15, 35, 63] networks require intensive computations. Reducing operations is essential for real-world applications if the parameters are kept around a certain level (*e.g*., 1M, 4MB sizes), because the primary consumption of semiconductor energy (concerning time) for neural networks is related to Mult-Adds operations [17,47].

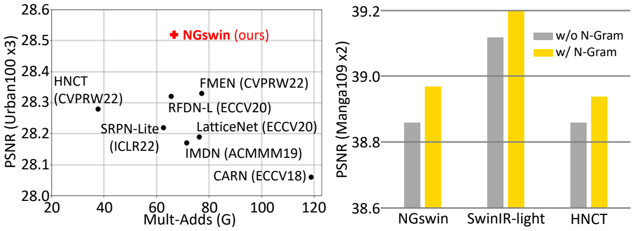


Figure 1. Two tracks of this paper using the N-Gram context. **(Left)** NGswin outperforms previous leading SR methods with an efficient structure. **(Right)** Our proposed N-Gram context improves different Swin Transformer-based SR models.

To overcome these problems, we define the N-Gram context as the interaction of neighbor local windows. Neighbor uni-Gram embeddings interact with each other by *sliding-WSA* to produce the N-Gram context features before window partitioning. The uni-Gram embeddings result from a channel-reducing group convolution [10] to decrease the complexity of N-Gram interaction (see Fig. 3c). Our N-Gram context efficiently expands the receptive field of WSA for recovery tasks. This work introduces N-Gram to low-level vision with Transformers for the first time, inspired by the following facts: N-Gram language models treat the extended context beyond each separate word to understand text statistically [8]. Since images have heavy spatial redundancy, some degraded pixels can be recovered from contextual information of neighbor pixels [18].

As shown in Fig. 1, our work progresses in two tracks. **Mainly**, to solve the problem of the intensive operations in SR, we propose an efficient N-Gram Swin Transformer (**NGswin**). As illustrated in Fig. 3a, NGswin consists of five components: a shallow module, three hierarchical encoder stages (with *patch-merging*) that contain NSTBs (N-Gram Swin Transformer Blocks), SCDP Bot-

*Corresponding author.

tleneck (pixel-Shuffle, Concatenation, Depth-wise convolution, Point-wise projection), a small decoder stage with NSTBs, and a reconstruction module. NSTBs employ our N-Gram context and the scaled-cosine attention proposed by Swin V2 [31]. SCDP bottleneck, which takes multi-scale outputs of the encoder, is a variant of bottleneck from U-Net [46]. Experimental results demonstrate that the components above contribute to the efficient and competitive performance of NGswin. **Secondly**, focusing on improved performances, we apply the N-Gram context to other Swin-based SR models, such as SwinIR-light [27] and HNCT [16]. Notably, **SwinIR-NG** (improved SwinIR-light with N-Gram) establishes state-of-the-art lightweight SR.

The main contributions of this paper are summarized as:

(1) We introduce the N-Gram context to the low-level vision with Transformer for the first time. It enables the SR networks to expand the receptive field to recover each degraded pixel by *sliding-WSA*. For efficient calculation of N-Gram WSA, we produce uni-Gram embeddings by a channel-reducing group convolution.

(2) We propose an efficient SR network, NGswin. It exploits the hierarchical encoder (with *patch-merging*), an asymmetrically small decoder, and SCDP bottleneck. These elements are critical for competitive performance in the efficient SR on ×2, ×3, and ×4 tasks.

(3) The N-Gram context improves other Swin Transformer methods. The improved SwinIR-NG achieves state-of-the-art results on lightweight SR.

2. Related Work

Efficient SR. Many single image super-resolution (SR) studies have increased network efficiency. CARN [2] introduced cascading residual blocks. IMDN [23] used information multi-distillation and selective feature fusion. LatticeNet [37] utilized the lattice filter that varies Fast Fourier Transformation. ESRT [35] combined convolutional neural networks (CNN) and channel-reducing Transformers [53]. SwinIR-light [27] and HNCT [16] appended CNN to Swin Transformer [32]. SRPN-Lite [65] applied the network pruning technique [45] on EDSR-baseline [28], a CNN-based lightweight SR model. Most recently, ELAN-light [63] utilized group-wise multi-scale self-attention.

N-Gram. In language model (LM), N-Gram is a sequence of consecutive characters or words. The size N is typically set to 2 or 3 [39]. The N-Gram LM that considers a longer span of context in sentences was operating well statistically in the past. Even some deep learning LMs still adopted N-Gram. Sent2Vec [43] used N-Gram embeddings to learn sentence embedding by averaging word-embedding. To learn the sentence representation better, [33] computed the word N-Gram context by recurrent neural networks (RNN) and passed it to the attention layer. ZEN [12, 49] trained a BERT-styled [11] N-Gram encoder for all possible N-Gram pairs from the Chinese or Arabic lexicon to convey salient pairs to the character encoder. Meanwhile, some high-level vision studies also adopted this concept. The Pixel N-grams approach [26] saw N-Gram in pixel level and a single (horizontal or vertical) direction. View N-gram Network [20] regarded consecutive (along time steps) multi-view images of a 3D object as an N-Gram. In contrast, our N-Gram considers bi-directional 2D information in local window level, given a single image for low-level vision.

Swin Transformer. Swin Transformer [32] (SwinV1) proposed window self-attention (WSA) that computes self-attention within non-overlapping local windows, to avoid quadratic time-complexity to the resolution of feature map. SwinV1 also placed a shifted window scheme in consecutive layers, capturing interaction across windows. Some studies [16, 27] utilized effective SwinV1 for SR. The revised version [31] (SwinV2) modified SwinV1. For advanced model capacity with milder optimization, SwinV2 introduced residual post-normalization and scaled-cosine attention (in Eq. (2), (3)) instead of pre-normalization configuration and scaled-dot-product attention.

3. Methodology

3.1. Problem Verification

The plain window self-attention (WSA) suffers from limited receptive field, as criticized in many recent studies [13, 52, 56, 58, 61, 62, 66]. We observe this issue by visualizing feature maps after self-attention. The similar pixel values from self-attention of the deeper layer tend to recover into a homogeneous pattern or texture. In (h) of Fig. 2, however, the patterns in the red box and its neighbors differ (***problem α***), causing the distortions in (e). *Problem α* stems from ***problem β***: The plain WSA (*i.e.*, *w/o* N-Gram) of the shallow layer is limited to only a local window. It cannot utilize surrounding patterns to infer the recovery pattern of each window. In (f) and (g), the distinctive colors across

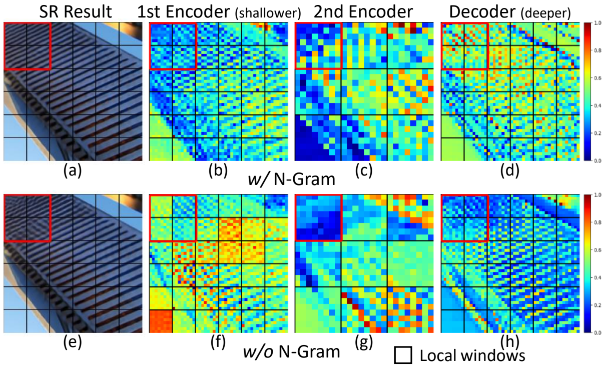
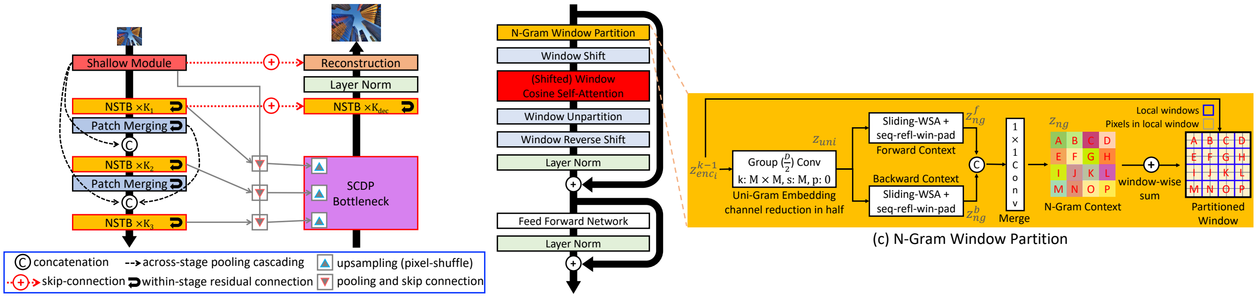


Figure 2. The feature maps after window self-attention in each intermediary layer of NGswin with and without N-Gram.



(a) Overall architecture of NGswin

(b) NSTB (proposed)

(c) N-Gram Window Partition

Figure 3. Overall architecture of NGswin and NSTB (N-Gram Swin Transformer Block). **(a)** We adopt an asymmetric U-Net architecture. SCDP Bottleneck (pixel-Shuffle, Concatenation, Depth-wise convolution, and Point-wise projection), a variant of the U-Net bottleneck, takes multi-scale outputs of encoder stages, including the shallow module. **(b)** Our proposed N-Gram method is implemented in NSTB. We also employ scaled-cosine attention and post-normalization. **(c)** k, s, p, and M are kernel size, stride, padding, and local window size, respectively. The dimensionality reduction through uni-Gram embedding makes *sliding-WSA* efficient. The bi-directional contexts share *sliding-WSA* weights. For window-wise sum, a value in z_{ng} is equally added to M^2 pixels in one local window at the same position.

Sentence1: We use laptop pc for office work.

→N-Gram (forward): (SOS we), (we use), (use laptop), (laptop pc), (for office), (office work), (work EOS)

→N-Gram (backward): (SOS we), (we use), (use laptop), (laptop pc), (pc for), (for office), (office work), (work EOS)

→N-Gram (bi-directional): (SOS we use), (we use laptop), (use laptop pc), (laptop pc for), (pc for office),
(for office work), (office, work, EOS)

*SOS: Start Of Sentence
*EOS: End Of Sentence

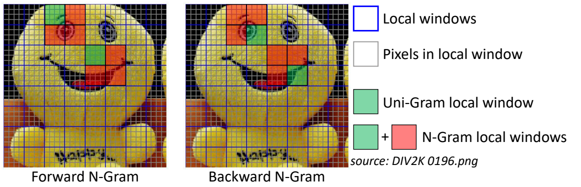
Sentence2: Office work makes me tired.

→N-Gram (forward): (SOS office), (office work), (work makes), (makes me), (me tired), (tired EOS)

→N-Gram (backward): (SOS office), (office work), (work makes), (makes me), (me tired), (tired EOS)

→N-Gram (bi-directional): (SOS office work), (office work makes), (work makes me), (makes me tired),
(me tired EOS)

(a) N-Gram in text.



source: DIV2K 0196.png

(b) N-Gram in image (proposed).

Figure 4. N-Gram in text and image ($N = 2$). **(a)** The underlined words are the target words and the non-underlined words are neighbors of the target words. **(b)** Each local window is defined as uni-Gram. The lower-right (or upper-left) local windows are defined as forward (or backward) N-Gram neighbors.

the adjacent windows reveal *problem* β. This seems to be resolved in the deeper layers, but it fails to overcome *problem* α. To address this, we propose the N-Gram context to compensate this vulnerability. Our N-Gram attention can consider broad regions (*i.e.*, surrounding patterns) beyond a window. In (a)-(d), the semantically relevant areas yield similar attention results, producing more accurate details. This crucial advantage solves both *problem* α and β.

3.2. Definition of N-Gram in Image

N-Gram in text. As shown in Fig. 4a, the N-Gram language model views the consecutive forward, backward, or bi-directional words as the N-Gram of the target word. The words are independent of each other for uni-Gram (*i.e.*, word-embedding), but they interact with each other by averaging word-embeddings [43], RNN [33], or attention [12] when considering N-Gram. In contrast, an N-Gram composed of a particular word pair (*e.g.*, "office work") never interacts with the other N-Gram combinations when producing an N-Gram feature.

N-Gram in image. N-Gram in an image should have the properties above. Accordingly, we define a uni-Gram as a non-overlapping local window in Swin Transformer, within which the pixels interact with each other by self-attention (SA). N-Gram is defined as the larger window, including neighbors of each uni-Gram. To sum up, pixels of each uni-Gram and uni-Grams of each N-Gram in image correspond to characters of each word and words of each N-Gram in text, respectively. As depicted in Fig. 4b, setting N-Gram size N to 2 indicates a bi-Gram that combines a local window (green area) and its neighbor windows (red areas) at lower-right (forward) or upper-left (backward). The N-Gram interactions will be explained in Sec. 3.4.

3.3. Overall Architecture of NGswin

As illustrated in Fig. 3a, we adopt U-Net [46] architecture: hierarchical encoder stages, a bottleneck layer, a decoder stage, and skip-connection from the encoder to the decoder at the same resolution[1]. However, our network's encoder and decoder are asymmetric, which indicates a significantly smaller decoder [18, 44].

Encoder. Given a low-resolution (LR) image $I_{LR} \in \mathbb{R}^{3\times H\times W}$, a shallow module (a 3×3 convolution) extracts $z_s \in \mathbb{R}^{HW\times D}$, where H, W, and D stand for height, width, and network dimension (channels), respectively. z_s is passed through three encoder stages, each composed of $\mathcal{K}_i$ N-Gram Swin Transformer Blocks (NSTB, Sec. 3.4)

[1] In this paper, "resolution" indicates height and width of feature maps, excluding network dimension (channel).

Table 1. Comparison of computational complexity with state-of-the-art networks. Our NGswin is much more efficient. Mult-Adds is evaluated on a 1280 × 720 HR image.

Scale	NGswin	SwinIR-light [27][2]	ESRT [35]	DiVANet [6]	ELAN-light [63]
x2	**140.4G**	243.7G	191.4G	189.0G	168.4G
x3	**66.6G**	109.5G	96.4G	89.0G	75.7G
x4	**36.4G**	61.7G	67.7G	57.0G	43.2G

and a 2 × 2 *patch-merging* except the last stage. We set $\{\mathcal{K}_1, \mathcal{K}_2, \mathcal{K}_3\}$ to $\{6, 4, 4\}$ by default. The mapping function $\mathcal{F}^k_{enc_i}$ of k-th ($1 \leq k \leq \mathcal{K}_i$) NSTB in i-th ($1 \leq i \leq 3$) encoder stage is formulated as:

$$z^k_{enc_i} = \mathcal{F}^k_{enc_i}(z^{k-1}_{enc_i}),\ z^k_{enc_i} \in \mathbb{R}^{HW/(2^{i-1})^2 \times D}, \quad (1)$$

where $z^0_{enc_i}$ equals $z_{enc_{i-1}}$, which results from downsampling $z^{\mathcal{K}_{i-1}}_{enc_{i-1}}$ ($z_{enc_0} = z_s$). In other words, the first NSTB in the 2nd or 3rd stage takes the output of *patch-merging* in the previous stage as input. The *patch-merging* follows Swin Transformer [31, 32], except that the network dimension is decreased from $4D$ to D instead of $2D$. Since *patch-merging* halves the resolutions, NGswin consumes much fewer attention computations than state-of-the-art attention-based lightweight SR methods, as revealed in Tab. 1.

Pooling Cascading. Following the global cascading in CARN [2], we employ a cascading mechanism (ⓒ marks and the dashed lines in Fig. 3a) across the stages, including the shallow module. Unlike CARN, we place 2×2 max-poolings before concatenating the intermediary features because the first and second stages halve the resolutions of features. This dense connectivity [21] reflects the flow of the information and gradient in the previous layers, which helps the network to learn meaningful representations.

Bottleneck. All outputs from the shallow module and the last NSTBs of each encoder stage are taken by SCDP bottleneck. The bottleneck layer maps them into $z_{scdp} \in \mathbb{R}^{HW \times D}$. A detailed explanation is in Sec. 3.5.

Decoder. z_{scdp} is fed into a single decoder stage, which is asymmetrically smaller than the encoder [18, 44]. This means fewer stages and NSTBs in our decoder, which highly enhances the efficiency as shown in Fig. 5. It contains $\mathcal{K}_{dec}$ (by default, 6) NSTBs and a final layer-norm (LN) [5] that allows stable learning. The decoder NSTB architecture is the same as the encoder NSTB. As done in U-Net [46], the input to the decoder is residually connected [19] with $z^{\mathcal{K}_1}_{enc_1}$ of the first encoder stage. z_s and decoder output $z_{dec} \in \mathbb{R}^{HW \times D}$ are added with a global skip-connection [3, 24, 27]. This boosts optimization and allows the reconstruction module to utilize both locality and long-range dependency.

Reconstruction. Following [2, 24, 27, 28], the reconstruction module contains a convolution that adjusts dimension and a pixel-shuffler [48]. Unlike previous works, we additionally place a convolution that produces the SR image $I_{SR} \in \mathbb{R}^{3 \times rH \times rW}$, where r is a scale factor (*e.g.*, ×4). The illustration is in the supplementary Sec. A.3.

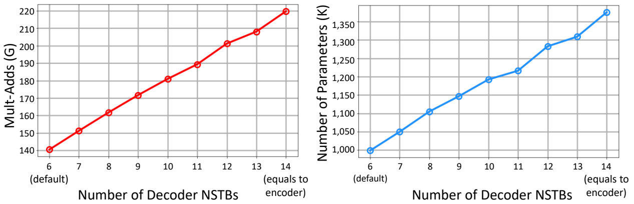


Figure 5. Efficiency of asymmetrically small decoder in NGswin. Mult-Adds is evaluated on ×2 task with a 1280×720 image.

3.4. N-Gram Swin Transformer Block (NSTB)

As illustrated in Fig. 3b, our NSTB adopts scaled-cosine attention and post-normalization proposed in SwinV2 [31]. For scaled-cosine window self-attention (WSA), we use the following formula:

$$WSA(Q, K, V) = Softmax(cos(Q, K)/\tau + B)V, \quad (2)$$

where $Q, K, V \in \mathbb{R}^{M^2 \times D}$ are the query, key, and value matrices; $B \in \mathbb{R}^{M^2 \times M^2}$ is the relative position bias between each pixel within a local window; τ is a learnable scalar set to larger than 0.01 [31]. M is the window size set to 8 by default, and the corresponding M^2 indicates the number of pixels in a local window. For the given matrices $Q = \left[q_{ij}\right]$ and $K = \left[k_{ij}\right]$, the cosine similarity is calculated as:

$$cos(Q, K) = \left[q_{ij}/\|q_i\|_2\right] \left[k_{ij}/\|k_i\|_2\right]^T \quad (3)$$

In window partitioning (the top of Fig. 3b), we implement the N-Gram context algorithm through four steps (see Fig. 3c). This algorithm is identically applied to other Swin Transformer models (SwinIR-light [27], HNCT [16]), focusing only on better performances. As denoted by Eq. (1), the input to k-th NSTB in i-th encoder stage is $z^{k-1}_{enc_i} \in \mathbb{R}^{hw \times D}$, where $h = H/2^{i-1}$ and $w = W/2^{i-1}$.

First, the input is mapped into the uni-Gram ($N = 1$) embedding $z_{uni} \in \mathbb{R}^{\frac{D}{2} \times w_h \times w_w}$ by an $M \times M$ channel-reducing group convolution [10] (stride: M, groups: $\frac{D}{2}$). w_h ($=\frac{h}{M}$) and w_w ($=\frac{w}{M}$) represent the number of windows in height and width. It is worth noting that the reduction of channel and resolution by uni-Gram embedding makes N-Gram WSA (in the next step) more efficient. Considering $\Omega(WSA) = 4hwD^2 + 2M^2hwD$ [32], halved D and M^2 times reduced hw highly decrease computations.

Second, the N^2 pixels in each N-Gram ($N > 1$) of z_{uni} interact with each other by WSA (by Eq. (2) with $M{=}N$ and $D{=}\frac{D}{2}$) to obtain the forward N-Gram feature $z^f_{ng} \in \mathbb{R}^{w_h \times w_w \times \frac{D}{2}}$. As shown in Fig. 6, we implement *sliding-WSA* as sliding-window convolution operated in CNN. As an $N \times N$ window slides through z_{uni},

[2] We correct Mult-Adds [27] underestimated on a 1024×720 HR image.

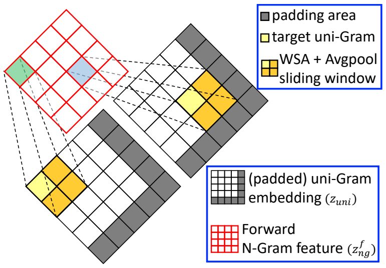


Figure 6. Sliding-WSA. A window operates SA and avg-pool to get the forward N-Gram feature, as the window slides through the uni-Gram embedding. z^b_{ng} can be obtained by upper-left padding.

Algorithm 1 SCDP Bottleneck Pseudo-code, PyTorch-like

```
# zi: output list of last NSTBs in three encoder stages
# zs: output of shallow module

x = list()
for i in range(3): # pixel-"S"huffle
    x_ = zi[i] + down(zs, i) # before shuffling
    x.append(PixelShuffle(x_, 2**i))
x = torch.cat(x, dim=-1) # "C"oncatenation
x = Rearrange(x, ’(h w) d -> d h w’) # ignores batch
x = GELU(depth_wise(x)) # "D"epth-wise convolution
x = Rearrange(x, ’d h w -> (h w) d’)
x = LayerNorm(point_wise(x)) # "P"oint-wise projection

def down(z, exp): # downsizing zs
    z = Rearrange(z, ’(h w) d -> d h w’)
    for e in range(exp): # iterative max-poolings
        z = MaxPool2D(z) # 2x2 pool
    z = LeakyReLU(z)
    return Rearrange(z, ’d h w -> (h w) d’)
```

scaled-cosine self-attention and $N\times N$ average-pooling are computed. But the scaled-dot-product attention is used for SwinIR-light and HNCT, following their own methods. We use *seq-refl-win-pad* instead of trivial zero padding for $(N-1)$ size of paddings, as explained in the supplementary Sec. A.1. Subsequently, we can obtain the backward N-Gram feature z^b_{ng} by reversed *seq-refl-win-pad* (*i.e.*, upper-left side padding). The computations of bi-directional N-Gram features share the *sliding-WSA* weights. Since the image is two-dimensional data, our N-Gram can be seen from max quad-directions (lower-right, lower-left, upper-right, and upper-left), unlike text that can be seen from max bi-directions. However, Tab. 5 demonstrates that the trade-off between performance and efficiency is optimized at the bi-directions.

Third, after the concatenation of z^f_{ng} and z^b_{ng}, a 1×1 convolution merges it to produce the N-Gram context z_{ng}.

Finally, $z_{ng}\in\mathbb{R}^{D\times w_h\times w_w}$ is added window-wise to the partitioned windows (size: $M^2\times D\times w_h\times w_w$) from $z^{k-1}_{enc_i}$. In Fig. 3c, one value in z_{ng} is equally added to M^2 pixels in one local window at the same position (marked as the same character) — *i.e.*, the average correlations from self-attention within each N-Gram serve as bias terms of each pixel. After the four steps, NSTB follows the sequence in Fig. 3b. The window-shifts are operated in the even numbered blocks, same as in Swin Transformer.

NSTBs and *patch-merging* within a stage are residually connected [19] from a previous layer to the next (the rounded arrows of Fig. 3a), rather than dense connections [21]. The connection to a *patch-merging*, however, is excepted from the third encoder stage and the decoder stage. For more, see the supplementary Sec. A.2.

3.5. SCDP Bottlneck

Many SR models [27, 42, 64] commonly never used the hierarchical encoder, which downsamples the resolutions of features after one stage. As demonstrated in Tab. 6, the hierarchical networks are inferior to the less (or non) hierarchical architectures. However, our encoder is constructed hierarchically by *patch-merging*. Thus, only passing the output $z^{\mathcal{K}_3}_{enc_3}$ of the last NSTB in the final encoder stage to the bottleneck makes the recovery task more challenging. The use of SCDP bottleneck, though, can convey rich representations of multi-scale features to the decoder and maintain the efficiency of NGswin. Algorithm 1 provides the pseudo-code of SCDP pipeline. It contains dimensionality rearrangements, non-linear activation functions, and LN omitted in the main text below.

SCDP stands for pixel-<u>**S**</u>huffle, <u>**C**</u>oncatenation, <u>**D**</u>epth-wise convolution, and <u>**P**</u>oint-wise projection. In contrast with the bottleneck of standard U-Net that takes the output of the last encoder layer [46, 55, 59], SCDP bottleneck takes multi-scale outputs from the encoder. **First**, as depicted in the blue-edged triangles in Fig. 3a, for obtaining z'_{enc_i} the pixel-shuffle [48] layers upsample the outputs $z^{\mathcal{K}_i}_{enc_i}$ of the last NSTB in each encoder stage into the resolution of I_{LR}, $H\times W$. Before upsizing, z_s is iteratively max-pooled into the resolution of each $z^{\mathcal{K}_i}_{enc_i}$, then added to $z^{\mathcal{K}_i}_{enc_i}$. This process gives multi-scale information to the bottleneck. **Second**, all z'_{enc_i} are concatenated in channel (network dimension) space. **Third**, the output passes through a 3×3 depthwise convolutional layer for learning spatial representations in each channel space. **Finally**, a point-wise linear projection is applied to match dimension D. As a result, we get z_{scdp} and then add it to $z^{\mathcal{K}_1}_{enc_1}$ to pass it to the decoder.

4. Experiments

4.1. Experimental Setup

Training. We used 800 HR-LR (high- and low-resolution) image pairs from DIV2K [1] dataset. LR images were randomly cropped into 64×64 size patches augmented by random horizontal flip and rotation (90°, 180°, 270°), as in the recent works [6, 16, 27, 63]. We minimized L_1 pixel-loss between I_{SR} and the ground truth I_{HR}: $\mathcal{L}=\|I_{HR}-I_{SR}\|_1$, with Adam [25] or AdamW [34] optimizer. NGswin, SwinIR-NG, and HNCT-NG (improved with N-Gram) were

Table 2. Comparison of efficient super-resolution results. D2K stands for the DIV2K dataset we used to train NGswin. DF2K indicates a merged dataset of D2K and Flickr2K [51] containing 800 + 2,650 HR-LR image pairs. 291 images dataset is from [4, 57]. Mult-Adds is evaluated on a 1280 × 720 HR image. The best, second best, and third best performances are in red, blue, and underline.

Method	Training Dataset	Scale	Mult-Adds	#Params	Set5		Set14		BSD100		Urban100		Manga109	
					PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
EDSR-baseline [28]	D2K	×2	316.3G	1,370K	37.99	0.9604	33.57	0.9175	32.16	0.8994	31.98	0.9272	38.54	0.9769
MemNet [50]	291	×2	2,662.4G	677K	37.78	0.9597	33.28	0.9142	32.08	0.8978	31.31	0.9195	-	-
CARN [2]	D2K+291	×2	222.8G	1,592K	37.76	0.9590	33.52	0.9166	32.09	0.8978	31.92	0.9256	38.36	0.9765
IMDN [23]	D2K	×2	158.8G	694K	38.00	0.9605	33.63	0.9177	32.19	0.8996	32.17	0.9283	38.88	0.9774
LatticeNet [37]	D2K	×2	169.5G	756K	38.06	0.9607	33.70	0.9187	32.20	0.8999	32.25	0.9288	38.94	0.9774
RFDN-L [30]	D2K	×2	145.8G	626K	38.08	0.9606	33.67	0.9190	32.18	0.8996	32.24	0.9290	38.95	0.9773
SRPN-Lite [65]	DF2K	×2	139.9G	609K	38.10	0.9608	33.70	0.9189	32.25	0.9005	32.26	0.9294	-	-
HNCT [16]	D2K	×2	82.4G	357K	38.08	0.9608	33.65	0.9182	32.22	0.9001	32.22	0.9294	38.87	0.9774
FMEN [15]	DF2K	×2	172.0G	748K	38.10	0.9609	33.75	0.9192	32.26	0.9007	32.41	0.9311	38.95	0.9778
NGswin (ours)	**D2K**	×2	**140.4G**	**998K**	**38.05**	**0.9610**	**33.79**	**0.9199**	**32.27**	**0.9008**	**32.53**	**0.9324**	**38.97**	**0.9777**
EDSR-baseline [28]	D2K	×3	160.2G	1,555K	34.37	0.9270	30.28	0.8417	29.09	0.8052	28.15	0.8527	33.45	0.9439
MemNet [50]	219	×3	2,662.4G	677K	34.09	0.9248	30.00	0.8350	28.96	0.8001	27.56	0.8376	-	-
CARN [2]	D2K+291	×3	118.8G	1,592K	34.29	0.9255	30.29	0.8407	29.06	0.8034	28.06	0.8493	33.50	0.9440
IMDN [23]	D2K	×3	71.5G	703K	34.36	0.9270	30.32	0.8417	29.09	0.8046	28.17	0.8519	33.61	0.9445
LatticeNet [37]	D2K	×3	76.3G	765K	34.40	0.9272	30.32	0.8416	29.10	0.8049	28.19	0.8513	33.63	0.9442
RFDN-L [30]	D2K	×3	65.6G	633K	34.47	0.9280	30.35	0.8421	29.11	0.8053	28.32	0.8547	33.78	0.9458
SRPN-Lite [65]	DF2K	×3	62.7G	615K	34.47	0.9276	30.38	0.8425	29.16	0.8061	28.22	0.8534	-	-
HNCT [16]	D2K	×3	37.8G	363K	34.47	0.9275	30.44	0.8439	29.15	0.8067	28.28	0.8557	33.81	0.9459
FMEN [15]	DF2K	×3	77.2G	757K	34.45	0.9275	30.40	0.8435	29.17	0.8063	28.33	0.8562	33.86	0.9462
NGswin (ours)	**D2K**	×3	**66.6G**	**1,007K**	**34.52**	**0.9282**	**30.53**	**0.8456**	**29.19**	**0.8078**	**28.52**	**0.8603**	**33.89**	**0.9470**
EDSR-baseline [28]	D2K	×4	114.0G	1,518K	32.09	0.8938	28.58	0.7813	27.57	0.7357	26.04	0.7849	30.35	0.9067
MemNet [50]	291	×4	2,662.4G	677K	31.74	0.8893	28.26	0.7723	27.40	0.7281	25.50	0.7630	-	-
CARN [2]	D2K+291	×4	90.9G	1,592K	32.13	0.8937	28.60	0.7806	27.58	0.7349	26.07	0.7837	30.47	0.9084
IMDN [23]	D2K	×4	40.9G	715K	32.21	0.8948	28.58	0.7811	27.56	0.7353	26.04	0.7838	30.45	0.9075
LatticeNet [37]	D2K	×4	43.6G	777K	32.18	0.8943	28.61	0.7812	27.57	0.7355	26.14	0.7844	30.54	0.9075
RFDN-L [30]	D2K	×4	37.4G	643K	32.28	0.8957	28.61	0.7818	27.58	0.7363	26.20	0.7883	30.61	0.9096
SRPN-Lite [65]	DF2K	×4	35.8G	623K	32.24	0.8958	28.69	0.7836	27.63	0.7373	26.16	0.7875	-	-
HNCT [16]	D2K	×4	22.0G	373K	32.31	0.8957	28.71	0.7834	27.63	0.7381	26.20	0.7896	30.70	0.9112
FMEN [15]	DF2K	×4	44.2G	769K	32.24	0.8955	28.70	0.7839	27.63	0.7379	26.28	0.7908	30.70	0.9107
NGswin (ours)	**D2K**	×4	**36.4G**	**1,019K**	**32.33**	**0.8963**	**28.78**	**0.7859**	**27.66**	**0.7396**	**26.45**	**0.7963**	**30.80**	**0.9128**

Table 3. Comparison of state-of-the-art lightweight super-resolution results. SwinIR-NG is SwinIR-light improved with the N-Gram. The mark ↓ and § indicates reduced-channel and DF2K, respectively. 'Year' indicates the publication year of each paper. The best and second best results are in red and blue.

Method	Year	Scale	Mult-Adds	#Params	Set5		Set14		BSD100		Urban100		Manga109	
					PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
SwinIR-light [27]	2021	×2	243.7G	910K	38.14	0.9611	33.86	0.9206	32.31	0.9012	32.76	0.9340	39.12	0.9783
ESRT [35]	2022	×2	191.4G	677K	38.03	0.9600	33.75	0.9184	32.25	0.9001	32.58	0.9318	39.12	0.9774
ELAN-light [63]	2022	×2	168.4G	582K	38.17	0.9611	33.94	0.9207	32.30	0.9012	32.76	0.9340	39.12	0.9783
DiVANet [6]	2023	×2	189.0G	902K	38.16	0.9612	33.80	0.9195	32.29	0.9012	32.60	0.9325	39.08	0.9775
SwinIR-NG (ours)	**2023**	×2	**274.1G**	**1,181K**	**38.17**	**0.9612**	**33.94**	**0.9205**	**32.31**	**0.9013**	**32.78**	**0.9340**	**39.20**	**0.9781**
SwinIR-light [27]	2021	×3	109.5G	918K	34.62	0.9289	30.54	0.8463	29.20	0.8082	28.66	0.8624	33.98	0.9478
ESRT [35]	2022	×3	96.4G	770K	34.42	0.9268	30.43	0.8433	29.15	0.8063	28.46	0.8574	33.95	0.9455
ELAN-light [63]	2022	×3	75.7G	590K	34.61	0.9288	30.55	0.8463	29.21	0.8081	28.69	0.8624	34.00	0.9478
DiVANet [6]	2023	×3	89.0G	949K	34.60	0.9285	30.47	0.8447	29.19	0.8073	28.58	0.8603	33.94	0.9468
SwinIR-NG (ours)	**2023**	×3	**114.1G**	**1,190K**	**34.64**	**0.9293**	**30.58**	**0.8471**	**29.24**	**0.8090**	**28.75**	**0.8639**	**34.22**	**0.9488**
SwinIR-light [27]	2021	×4	61.7G	930K	32.44	0.8976	28.77	0.7858	27.69	0.7406	26.47	0.7980	30.92	0.9151
ESRT [35]	2022	×4	67.7G	751K	32.19	0.8947	28.69	0.7833	27.69	0.7379	26.39	0.7962	30.75	0.9100
ELAN-light [63]	2022	×4	43.2G	601K	32.43	0.8975	28.78	0.7858	27.69	0.7406	26.54	0.7982	30.92	0.9150
DiVANet [6]	2023	×4	57.0G	939K	32.41	0.8973	28.70	0.7844	27.65	0.7391	26.42	0.7958	30.73	0.9119
SwinIR-NG↓ (ours)	**2023**	×4	**42.5G**	**770K**	**32.44**	**0.8978**	**28.80**	**0.7863**	**27.70**	**0.7407**	**26.47**	**0.7977**	**30.97**	**0.9147**
SwinIR-NG↓§ (ours)			**42.5G**	**770K**	**32.48**	**0.8979**	**28.83**	**0.7868**	**27.71**	**0.7411**	**26.54**	**0.7998**	**31.12**	**0.9158**
SwinIR-NG (ours)			**63.0G**	**1,201K**	**32.44**	**0.8980**	**28.83**	**0.7870**	**27.73**	**0.7418**	**26.61**	**0.8010**	**31.09**	**0.9161**

trained by the same strategies except warm-start [29]. While we trained NGswin and SwinIR-NG from scratch on ×2 and by warm-start (using pre-trained ×2 weights) on ×3 and ×4, HNCT-NG was trained from scratch on all tasks. Other details are in the supplementary Sec. B.

Evaluation. We evaluated the performances of the different models on the five benchmark datasets, composed of Set5 [7], Set14 [60], BSD100 [40], Urban100 [22], and Manga109 [41]. We used PSNR (dB) and SSIM [54] scores on the Y channel of the YCbCr space as the metrics. LR images were acquired by the MATLAB bicubic kernel from corresponding HR images matching each SR task.

4.2. Comparisons of Super-Resolution Results

In Tab. 2, we compare NGswin with other efficient SR models, including EDSR-baseline (CVPRW17) [28], MemNet (ICCV17) [50], CARN (ECCV18) [2], IMDN (ACMMM19) [23], LatticeNet (ECCV20) [37], RFDN-L (ECCV20) [30], SRPN-Lite (ICLR22) [65], HNCT

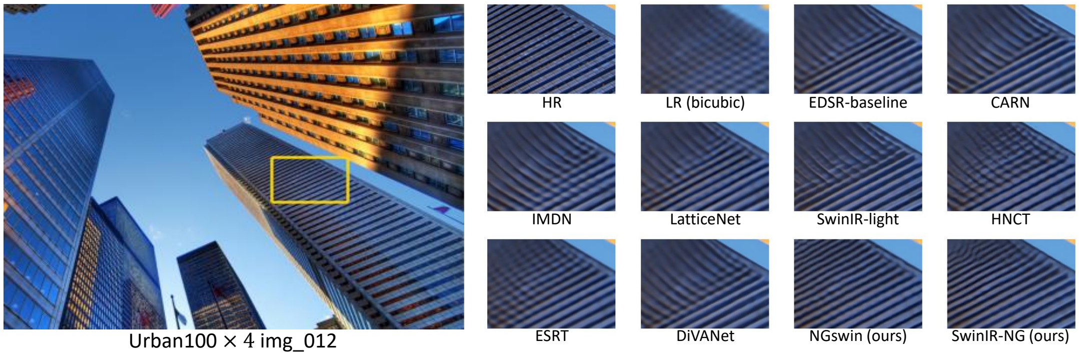


Figure 7. Visual comparisons with other models. 'LR' is an input image. More results are illustrated in the supplementary Sec. D.

Table 4. Ablation study on the N-Gram context. The top and bottom tables are PSNR / SSIM of NGswin and HNCT, respectively.

NGswin without *vs.* with N-Gram

N-Gram	Scale	Mult-Adds	#Params	Set14	Urban100	Manga109
w/o	×2	138.20G	750K	33.70 / 0.9194	32.39 / 0.9304	38.86 / 0.9775
w/		**140.41G**	**998K**	**33.79 / 0.9199**	**32.53 / 0.9324**	**38.97 / 0.9777**
w/o	×3	65.53G	759K	30.48 / 0.8451	28.37 / 0.8573	33.81 / 0.9464
w/		**66.56G**	**1,007K**	**30.53 / 0.8456**	**28.52 / 0.8603**	**33.89 / 0.9470**
w/o	×4	35.89G	771K	28.70 / 0.7844	26.25 / 0.7918	30.70 / 0.9123
w/o (channel up)		53.71G	1,189K	28.75 / 0.7854	26.28 / 0.7927	30.73 / 0.9129
w/o (depth up)		47.88G	1,061K	28.75 / 0.7853	26.37 / 0.7946	30.78 / **0.9133**
w/		**36.44G**	**1,019K**	**28.78 / 0.7859**	**26.45 / 0.7963**	**30.80** / 0.9128

HNCT [16] *vs.* HNCT-NG (ours)

N-Gram	Scale	Mult-Adds	#Params	Set14	Urban100	Manga109
w/o	×2	82.39G	357K	**33.65** / 0.9182	32.22 / 0.9294	38.87 / **0.9774**
w/		**83.19G**	**424K**	33.64 / **0.9195**	**32.35 / 0.9306**	**38.94 / 0.9774**
w/o	×3	37.78G	363K	30.44 / 0.8439	28.28 / 0.8557	**33.81** / 0.9459
w/		**38.14G**	**431K**	**30.48 / 0.8450**	**28.38 / 0.8573**	**33.81 / 0.9464**
w/o	×4	22.01G	373K	28.71 / 0.7834	26.20 / 0.7896	30.70 / 0.9112
w/		**22.21G**	**440K**	**28.72 / 0.7846**	**26.23 / 0.7912**	**30.71 / 0.9114**

(CVPRW22) [16], and FMEN (CVPRW22) [15]. PSNR and SSIM were evaluated on the three SR tasks. We reported the training dataset, Mult-Adds, and the number of parameters for comparing efficiency. The result shows that NGswin outperformed previous leading models on all benchmarks with a relatively efficient structure. Compared to SRPN-Lite and FMEN, NGswin was data-efficient with PSNR margins up to 0.3dB and 0.19dB, respectively. Note that the results of LatticeNet were referred from [36].

As shown in Tab. 3, we improved SwinIR-light [27] with the N-Gram context (named as SwinIR-NG). It was compared with the current best lightweight SR methods, including SwinIR-light (ICCVW21), ESRT (CVPRW22) [35], ELAN-light (ECCV22) [63], and DiVANet (PR23) [6], all of which were trained on DIV2K. SwinIR-NG outperformed them and established state-of-the-art lightweight SR on all benchmarks. Since the restoration of highly distorted regions needs much neighbor information, the impact of the N-Gram context was especially strong for ×3 or ×4 tasks and Urban100 or Manga109 datasets by a PSNR margin up to 0.24dB, compared to *w/o* N-Gram. For fair comparison with respect to #parameters, we also reduced the channels of SwinIR-NG from 60 to 48, which is denoted as ↓. In addition, SwinIR-NG↓ were also trained on DF2K (denoted as §) for further comparison. The performance of the reduced model on ×4 were reported and still better than the others with the fewest computations. We corrected the parameters of [27] that omitted the relative position bias tables.

The visual comparisons of each model are in Fig. 7.

Table 5. Ablation study on N-Gram interaction. "Direction": how many directions the network see N-Gram neighbors from. "Type": the method for N-Gram interaction. The bottom row is a default setting. PSNR / SSIM are evaluated on ×2 task with NGswin.

Direction	Type	Mult-Adds	#Params	Urban100	Manga109
1	WSA	152.41G	1,238,056	**32.54** / 0.9322	38.90 / **0.9777**
4	WSA	139.56G	935,272	32.52 / 0.9317	38.92 / 0.9776
1	CNN	139.80G	1,327,528	32.45 / 0.9316	38.86 / 0.9775
2	CNN	139.38G	998,568	**32.54** / 0.9321	38.90 / 0.9776
4	CNN	139.17G	936,488	32.52 / 0.9320	38.93 / **0.9777**
2	**WSA**	**140.41G**	**998,384**	32.53 / **0.9324**	**38.97 / 0.9777**

4.3. Ablation Studies

Tab. 4 demonstrates that the N-Gram context enhanced Swin Transformer-based SR models with a reasonable level of sacrificed efficiency. We denote HNCT [16] improved with the N-Gram context as HNCT-NG. The results of SwinIR-light [27] and a corresponding SwinIR-NG can be referred to in Tab. 3. Interestingly, the N-Gram increased SSIM in general. That is, our method tended to produce perceptually more similar images to the ground-truth. Moreover, our method was robust to Urban100 and Manga109 datasets, which are hard to recover with DIV2K training dataset [38]. Fig. 8 visualizes the examples. One can doubt the marginal gain of HNCT-NG. This was because HNCT contains 8 Swin Transformer layers (Swins), while NGswin had 20. The application of the N-Gram method on fewer Swins of HNCT-NG yielded less effective results. Afterwards, we increased the depth or channel of NGswin without N-Gram for more compelling comparisons regarding to the number of parameters. The variations were applied to ×4 task, which still fell behind our proposed NGswin de-

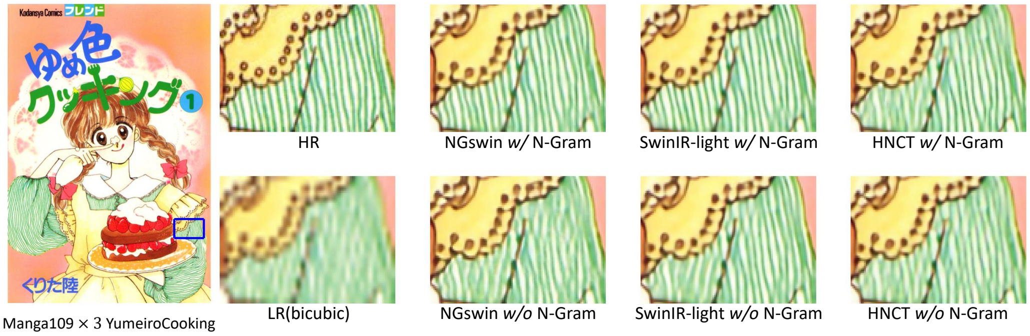


Figure 8. Visual comparisons of *w/ vs. w/o* N-Gram context for NGswin, SwinIR-light, and HNCT. At the top and bottom row, the 2nd to 4th columns show the models with and without N-Gram, respectively. More visual comparisons are in the supplementary Sec. D.

Table 6. Ablation study on extra stages and SCDP bottleneck.

(a) The specifications of models with different stages. dep.: # of NSTBs / res.: training input resolution. The total number of NSTBs is kept as 20.

Stages	encoder1	encoder2	encoder3	encoder4	decoder1	decoder2
	dep. / res.	dep. / res.	dep. / res.	dep. / res.	dep. / res.	dep. / res.
extra	4 / 64×64	4 / 32×32	4 / 16×16	4 / 8×8	2 / 32×32	2 / 64×64
default	**6 / 64×64**	**4 / 32×32**	**4 / 16×16**	**- / -**	**6 / 64×64**	**- / -**

(b) Impacts of extra stages and SCDP bottleneck. PSNR / SSIM.

Stages	SCDP	Scale	Mult-Adds	#Params	Urban100	Manga109
extra	*w/o*	×2	87.98G	997K	32.28 / 0.9298	38.72 / 0.9773
default	*w/o*		138.88G	992K	32.48 / 0.9321	38.92 / 0.9776
default	***w/***		**140.41G**	**998K**	**32.53 / 0.9324**	**38.97 / 0.9777**
extra	*w/o*	×3	42.10G	1,006K	28.33 / 0.8562	33.67 / 0.9453
default	*w/o*		65.85G	1,001K	28.47 / 0.8596	33.81 / 0.9464
default	***w/***		**66.56G**	**1,007K**	**28.52 / 0.8603**	**33.89 / 0.9470**
extra	*w/o*	×4	23.33G	1,018K	26.22 / 0.7900	30.46 / 0.9090
default	*w/o*		36.06G	1,013K	26.38 / 0.7954	30.71 / 0.9121
default	***w/***		**36.44G**	**1,019K**	**26.45 / 0.7963**	**30.80 / 0.9128**

spite similar parameters but much more computations.

In Tab. 5, we examined the N-Gram interaction. “Direction” counts how many directions the network sees N-Gram neighbors from. “Type” indicates the method for N-Gram interaction. The two directions and WSA type were the same as explained in Sec. 3.4. At one direction, the network saw the neighbors from a lower-right only. The uni-Gram embedding never reduced channels, which was harmful for the efficiency. In the case of four directions, neighbors from lower-right, lower-left, upper-right, and upper-left were viewed as N-Gram. In implementation, uni-Gram embedding layer produced the four N-Gram features that have $\frac{D}{4}$ dimension. They were merged into D dimension of the N-Gram context. For CNN interaction, we operated a conventional convolution after uni-Gram embedding rather than *sliding-WSA*. As a result, the bi-directional N-Gram context with *sliding-WSA* model was the best optimized model in terms of trade-off between performance and efficiency. In addition, *sliding-WSA* had an advantage over sliding-winodw-convolution in that SA could compute the correlations within each N-Gram.

As shown in Tab. 6, while both default and extra stages had the same number of NSTBs, the extra stages handled lower resolutions. But the extra stages appended to the encoder and decoder dropped the performance. This is because reconstruction of high-frequency information from the space preserving HR information richly is easier than from the space preserving HR information insufficiently. Since our SCDP bottleneck took all outputs of the encoder stages (*i.e.*, various resolutions), we prevented the performance drop despite hierarchical encoders. In implementation, the bottleneck without SCDP had the depth-wise and point-wise convolutions only, and took the output of the last NSTB in the third encoder stage.

Each ablation study on the other benchmarks not reported due to page limit are in the appendix Sec. C.

5. Conclusion

This paper successfully introduced the N-Gram context from the text to the vision domain for the first time in history. Our N-Gram interaction by *sliding-WSA* made NGswin, SwinIR, and HNCT overcome the limitation of Swin Transformer, where the broad regions are ignored. SCDP bottleneck prevented the hierarchical encoder from dropping performance. The hierarchical encoder, small decoder, and uni-Gram embedding decreased the operations significantly. With the components above, NGswin showed competitive results compared with the previous leading SR methods. Moreover, SwinIR-NG established state-of-the-art results. For future works, we hope our N-Gram context can succeed on other low-level vision tasks, such as denoising and deblurring. In closing, if the N-Gram context is extended to the universal Transformer architectures, more developments for computer vision could be expected.

Acknowledgements. This paper was supported by Institute of Information & Communications Technology Planning & Evaluation (IITP) grant (No.2022-0-00956) and Korea Health Industry Development Institute (KHIDI) grant (No. H122C1983) funded by the Korea government (MSIT). Special thanks to Rescale Inc. for HPC cloud computing resources and Somang Kim for valuable comments.

References

[1] Eirikur Agustsson and Radu Timofte. Ntire 2017 challenge on single image super-resolution: Dataset and study. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 126–135, 2017. 5

[2] Namhyuk Ahn, Byungkon Kang, and Kyung-Ah Sohn. Fast, accurate, and lightweight super-resolution with cascading residual network. In *Proceedings of the European conference on computer vision (ECCV)*, pages 252–268, 2018. 2, 4, 6

[3] Namhyuk Ahn, Byungkon Kang, and Kyung-Ah Sohn. Efficient deep neural network for photo-realistic image super-resolution. *Pattern Recognition*, 127:108649, 2022. 4

[4] Pablo Arbelaez, Michael Maire, Charless Fowlkes, and Jitendra Malik. Contour detection and hierarchical image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 33(5):898–916, 2010. 6

[5] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016. 4

[6] Parichehr Behjati, Pau Rodriguez, Carles Fernández, Isabelle Hupont, Armin Mehri, and Jordi Gonzàlez. Single image super-resolution based on directional variance attention network. *Pattern Recognition*, 133:108997, 2023. 1, 4, 5, 6, 7

[7] Marco Bevilacqua, Aline Roumy, Christine Guillemot, and Marie Line Alberi-Morel. Low-complexity single-image super-resolution based on nonnegative neighbor embedding. 2012. 6

[8] Peter F Brown, Vincent J Della Pietra, Peter V Desouza, Jennifer C Lai, and Robert L Mercer. Class-based n-gram models of natural language. *Computational linguistics*, 18(4):467–480, 1992. 1

[9] Hanting Chen, Yunhe Wang, Tianyu Guo, Chang Xu, Yiping Deng, Zhenhua Liu, Siwei Ma, Chunjing Xu, Chao Xu, and Wen Gao. Pre-trained image processing transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12299–12310, 2021. 1

[10] Taco Cohen and Max Welling. Group equivariant convolutional networks. In *International conference on machine learning*, pages 2990–2999. PMLR, 2016. 1, 4

[11] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 2

[12] Shizhe Diao, Jiaxin Bai, Yan Song, Tong Zhang, and Yonggang Wang. Zen: Pre-training chinese text encoder enhanced by n-gram representations. *arXiv preprint arXiv:1911.00720*, 2019. 2, 3

[13] Mingyu Ding, Bin Xiao, Noel Codella, Ping Luo, Jingdong Wang, and Lu Yuan. Davit: Dual attention vision transformers. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXIV*, pages 74–92. Springer, 2022. 2

[14] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 1

[15] Zongcai Du, Ding Liu, Jie Liu, Jie Tang, Gangshan Wu, and Lean Fu. Fast and memory-efficient network towards efficient image super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 853–862, 2022. 1, 6, 7

[16] Jinsheng Fang, Hanjiang Lin, Xinyu Chen, and Kun Zeng. A hybrid network of cnn and transformer for lightweight image super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1103–1112, 2022. 1, 2, 4, 5, 6, 7

[17] Runze Han, Yachen Xiang, Peng Huang, Yihao Shan, Xiaoyan Liu, and Jinfeng Kang. Flash memory array for efficient implementation of deep neural networks. *Advanced Intelligent Systems*, 3(5):2000161, 2021. 1

[18] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16000–16009, 2022. 1, 3, 4

[19] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 4, 5

[20] Xinwei He, Tengteng Huang, Song Bai, and Xiang Bai. View n-gram network for 3d object retrieval. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7515–7524, 2019. 2

[21] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017. 4, 5

[22] Jia-Bin Huang, Abhishek Singh, and Narendra Ahuja. Single image super-resolution from transformed self-exemplars. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5197–5206, 2015. 6

[23] Zheng Hui, Xinbo Gao, Yunchu Yang, and Xiumei Wang. Lightweight image super-resolution with information multi-distillation network. In *Proceedings of the 27th acm international conference on multimedia*, pages 2024–2032, 2019. 2, 6

[24] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee. Accurate image super-resolution using very deep convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1646–1654, 2016. 4

[25] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5

[26] Pradnya Kulkami, Andrew Stranieri, and Julien Ugon. Texture image classification using pixel n-grams. In *Proceedings of 2016 IEEE International Conference on Signal and Image Processing (ICSIP)*, pages 137–141. IEEE, 2016. 2

[27] Jingyun Liang, Jiezhang Cao, Guolei Sun, Kai Zhang, Luc Van Gool, and Radu Timofte. Swinir: Image restoration us-

ing swin transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1833–1844, 2021. 1, 2, 4, 5, 6, 7

[28] Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee. Enhanced deep residual networks for single image super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 136–144, 2017. 2, 4, 6

[29] Zudi Lin, Prateek Garg, Atmadeep Banerjee, Salma Abdel Magid, Deqing Sun, Yulun Zhang, Luc Van Gool, Donglai Wei, and Hanspeter Pfister. Revisiting rcan: Improved training for image super-resolution. *arXiv preprint arXiv:2201.11279*, 2022. 6

[30] Jie Liu, Jie Tang, and Gangshan Wu. Residual feature distillation network for lightweight image super-resolution. In *European Conference on Computer Vision*, pages 41–55. Springer, 2020. 6

[31] Ze Liu, Han Hu, Yutong Lin, Zhuliang Yao, Zhenda Xie, Yixuan Wei, Jia Ning, Yue Cao, Zheng Zhang, Li Dong, et al. Swin transformer v2: Scaling up capacity and resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12009–12019, 2022. 2, 4

[32] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10012–10022, 2021. 1, 2, 4

[33] Inigo Lopez-Gazpio, Montse Maritxalar, Mirella Lapata, and Eneko Agirre. Word n-gram attention models for sentence similarity and inference. *Expert Systems with Applications*, 132:1–11, 2019. 2, 3

[34] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016. 5

[35] Zhisheng Lu, Hong Liu, Juncheng Li, and Linlin Zhang. Efficient transformer for single image super-resolution. *arXiv preprint arXiv:2108.11084*, 2021. 1, 2, 4, 6, 7

[36] Xiaotong Luo, Yanyun Qu, Yuan Xie, Yulun Zhang, Cuihua Li, and Yun Fu. Lattice network for lightweight image restoration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022. 7

[37] Xiaotong Luo, Yuan Xie, Yulun Zhang, Yanyun Qu, Cuihua Li, and Yun Fu. Latticenet: Towards lightweight image super-resolution with lattice block. In *European Conference on Computer Vision*, pages 272–289. Springer, 2020. 2, 6

[38] Salma Abdel Magid, Zudi Lin, Donglai Wei, Yulun Zhang, Jinjin Gu, and Hanspeter Pfister. Texture-based error analysis for image super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2118–2127, 2022. 7

[39] P Majumder, M Mitra, and BB Chaudhuri. N-gram: a language independent approach to ir and nlp. In *International conference on universal knowledge and language*, 2002. 2

[40] David Martin, Charless Fowlkes, Doron Tal, and Jitendra Malik. 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*, volume 2, pages 416–423. IEEE, 2001. 6

[41] Yusuke Matsui, Kota Ito, Yuji Aramaki, Azuma Fujimoto, Toru Ogawa, Toshihiko Yamasaki, and Kiyoharu Aizawa. Sketch-based manga retrieval using manga109 dataset. *Multimedia Tools and Applications*, 76(20):21811–21838, 2017. 6

[42] Ben Niu, Weilei Wen, Wenqi Ren, Xiangde Zhang, Lianping Yang, Shuzhen Wang, Kaihao Zhang, Xiaochun Cao, and Haifeng Shen. Single image super-resolution via a holistic attention network. In *European conference on computer vision*, pages 191–207. Springer, 2020. 5

[43] Matteo Pagliardini, Prakhar Gupta, and Martin Jaggi. Unsupervised learning of sentence embeddings using compositional n-gram features. *arXiv preprint arXiv:1703.02507*, 2017. 2, 3

[44] Yatian Pang, Wenxiao Wang, Francis EH Tay, Wei Liu, Yonghong Tian, and Li Yuan. Masked autoencoders for point cloud self-supervised learning. *arXiv preprint arXiv:2203.06604*, 2022. 3, 4

[45] Russell Reed. Pruning algorithms-a survey. *IEEE transactions on Neural Networks*, 4(5):740–747, 1993. 2

[46] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015. 2, 3, 4, 5

[47] Hans-Christian Ruiz-Euler, Unai Alegre-Ibarra, Bram van de Ven, Hajo Broersma, Peter A Bobbert, and Wilfred G van der Wiel. Dopant network processing units: towards efficient neural network emulators with high-capacity nanoelectronic nodes. *Neuromorphic Computing and Engineering*, 1(2):024002, 2021. 1

[48] Wenzhe Shi, Jose Caballero, Ferenc Huszár, Johannes Totz, Andrew P Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1874–1883, 2016. 4, 5

[49] Yan Song, Tong Zhang, Yonggang Wang, and Kai-Fu Lee. Zen 2.0: Continue training and adaption for n-gram enhanced text encoders. *arXiv preprint arXiv:2105.01279*, 2021. 2

[50] Ying Tai, Jian Yang, Xiaoming Liu, and Chunyan Xu. Memnet: A persistent memory network for image restoration. In *Proceedings of the IEEE international conference on computer vision*, pages 4539–4547, 2017. 6

[51] Radu Timofte, Eirikur Agustsson, Luc Van Gool, Ming-Hsuan Yang, and Lei Zhang. Ntire 2017 challenge on single image super-resolution: Methods and results. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 114–125, 2017. 6

[52] Zhengzhong Tu, Hossein Talebi, Han Zhang, Feng Yang, Peyman Milanfar, Alan Bovik, and Yinxiao Li. Maxvit: Multi-axis vision transformer. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, Octo-*

ber 23–27, 2022, Proceedings, Part XXIV, pages 459–479. Springer, 2022. 1, 2

[53] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 2

[54] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004. 6

[55] Zhendong Wang, Xiaodong Cun, Jianmin Bao, Wengang Zhou, Jianzhuang Liu, and Houqiang Li. Uformer: A general u-shaped transformer for image restoration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17683–17693, 2022. 1, 5

[56] Jianwei Yang, Chunyuan Li, Pengchuan Zhang, Xiyang Dai, Bin Xiao, Lu Yuan, and Jianfeng Gao. Focal self-attention for local-global interactions in vision transformers. *arXiv preprint arXiv:2107.00641*, 2021. 1, 2

[57] Jianchao Yang, John Wright, Thomas S Huang, and Yi Ma. Image super-resolution via sparse representation. *IEEE transactions on image processing*, 19(11):2861–2873, 2010. 6

[58] Qihang Yu, Yingda Xia, Yutong Bai, Yongyi Lu, Alan L Yuille, and Wei Shen. Glance-and-gaze vision transformer. *Advances in Neural Information Processing Systems*, 34:12992–13003, 2021. 1, 2

[59] Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, and Ming-Hsuan Yang. Restormer: Efficient transformer for high-resolution image restoration. In *CVPR*, 2022. 5

[60] Roman Zeyde, Michael Elad, and Matan Protter. On single image scale-up using sparse-representations. In *International conference on curves and surfaces*, pages 711–730. Springer, 2010. 6

[61] Jiale Zhang, Yulun Zhang, Jinjin Gu, Yongbing Zhang, Linghe Kong, and Xin Yuan. Accurate image restoration with attention retractable transformer. *arXiv preprint arXiv:2210.01427*, 2022. 1, 2

[62] Qiming Zhang, Yufei Xu, Jing Zhang, and Dacheng Tao. Vsa: learning varied-size window attention in vision transformers. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXV*, pages 466–483. Springer, 2022. 2

[63] Xindong Zhang, Hui Zeng, Shi Guo, and Lei Zhang. Efficient long-range attention network for image super-resolution. *arXiv preprint arXiv:2203.06697*, 2022. 1, 2, 4, 5, 6, 7

[64] Yulun Zhang, Kunpeng Li, Kai Li, Lichen Wang, Bineng Zhong, and Yun Fu. Image super-resolution using very deep residual channel attention networks. In *Proceedings of the European conference on computer vision (ECCV)*, pages 286–301, 2018. 5

[65] Yulun Zhang, Huan Wang, Can Qin, and Yun Fu. Learning efficient image super-resolution networks via structure-regularized pruning. In *International Conference on Learning Representations*, 2021. 2, 6

[66] Chen Zheng, Yulun Zhang, Jinjin Gu, Yongbing Zhang, Linghe Kong, and Xin Yuan. Cross aggregation transformer for image restoration. *arXiv preprint arXiv:2211.13654*, 2022. 1, 2