X-HRNET: TOWARDS LIGHTWEIGHT HUMAN POSE ESTIMATION WITH SPATIALLY UNIDIMENSIONAL SELF-ATTENTION

Yixuan Zhou, Xuanhan Wang, Xing Xu*, Lei Zhao and Jingkuan Song

Center for Future Media & School of Computer Science and Engineering
University of Electronic Science and Technology of China, China

ABSTRACT

High-resolution representation is necessary for human pose estimation to achieve high performance, and the ensuing problem is high computational complexity. In particular, predominant pose estimation methods estimate human joints by 2D single-peak heatmaps. Each 2D heatmap can be horizontally and vertically projected to and reconstructed by a pair of 1D heat vectors. Inspired by this observation, we introduce a lightweight and powerful alternative, **S**patially **U**nidimensional **S**elf-**A**ttention (**SUSA**), to the pointwise (1×1) convolution that is the main computational bottleneck in the depthwise separable 3×3 convolution. Our SUSA reduces the computational complexity of the pointwise (1×1) convolution by 96% without sacrificing accuracy. Furthermore, we use the SUSA as the main module to build our lightweight pose estimation backbone X-HRNet, where X represents the estimated cross-shape attention vectors. Extensive experiments on the COCO benchmark demonstrate the superiority of our X-HRNet, and comprehensive ablation studies show the effectiveness of the SUSA modules. The code is publicly available at https://github.com/cool-xuan/x-hrnet.

Index Terms— High-resolution representation learning, human pose estimation, lightweight backbone

1. INTRODUCTION

2D human pose estimation aims to localize all human joints from the whole image space. Recent works [1–6] demonstrate that the high-resolution representation is necessary for this pixel-level regression task to achieve high performance. But massive computational resources are required in these high-resolution models, and it is hard to deploy them on the increasingly popular mobile devices with limited resources. Therefore, building a lightweight and efficient pose estimator has become an urgent challenge.
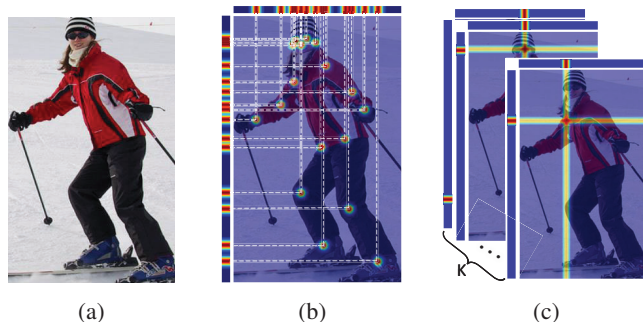
Fig. 1: (a) Input image. (b) 2D confidence heatmap can be horizontally and vertically projected to two 1D vectors. (c) K pairs of 1D heat vectors reconstruct K 2D heatmaps.

There are various lightweight schemes [8–10] for the classification task that can be borrowed from, where depthwise separable convolution [11, 12] (a depthwise 3×3 convolution followed by a pointwise convolution) is widely used to replace the standard 3×3 convolution. This substitution reduces the computational complexity by 89% without a performance drop. However, the pointwise (1×1) convolution is still the main computational bottleneck [13]. The straightforward optimization scheme reduces the channel number. For example, ShuffleNetV2 [9] alleviates the FLOPs of pointwise (1×1) convolution by only feeding half of the features to convolutions. Nevertheless, these lightweight methods tailored for classification focus on channel dimension, neglecting the computational overhead caused by spatial dimensions that are considerable for pixel-wise regression tasks.

Unlike conventional classification networks where the resolution of features shrinks with the increase of the depth, the networks for pixel-wise regression tasks maintain high-resolution features [4, 5, 14] or adopt the encoder-decoder structure [1,15,16]. For example, the HRNet [3] has four parallel branches with different resolutions, whose architecture is shown as follows:

$$
\begin{aligned}
\mathcal{N}_{11} &\longrightarrow \mathcal{N}_{21} \longrightarrow \mathcal{N}_{31} \longrightarrow \mathcal{N}_{41}\\
&\searrow \mathcal{N}_{22} \longrightarrow \mathcal{N}_{32} \longrightarrow \mathcal{N}_{42}\\
&\qquad\quad \searrow \mathcal{N}_{33} \longrightarrow \mathcal{N}_{43}\\
&\qquad\qquad\qquad \searrow \mathcal{N}_{44}
\end{aligned}
\tag{1}
$$

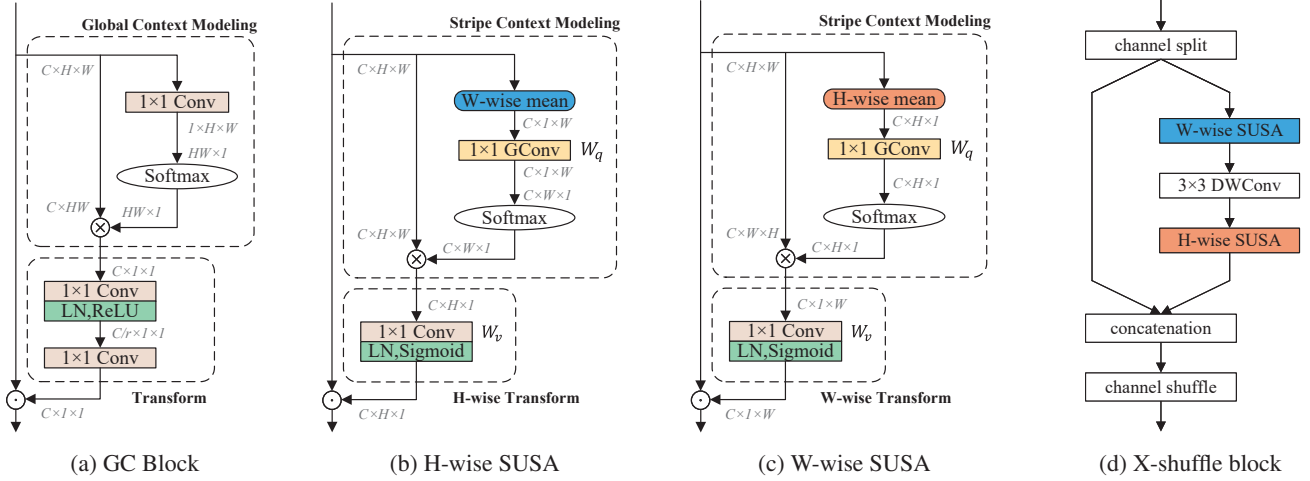where $\mathcal{N}_{sb}$ denotes the subnetwork of the s stage in the b

Fig. 2: **Architecture of the GC block [7], our SUSA and X-shuffle block.** For intuitive understanding, the features are abstracted as feature dimension, *e.g.*, $C \times H \times W$ denotes a feature map with channel number C, height H and Width W. $\otimes$ denotes matrix inner-product, $\odot$ denotes element-wise multiplication, and $\oplus$ denotes element-wise addition.

branch. The first branch maintains the highest resolution, whose size is a quarter of the one for input images. In HRNet, the width H and the height W of the feature map $x \in \mathbb{R}^{C \times H \times W}$ are on the same order of magnitude as the number of channel C. Therefore, we shift the focus from C to spatial dimensions: H and W for reducing the computational complexity in high-resolution pose estimation. Similar to our scheme, [13] presents the Lite-HRNet to reduce the computational complexity in spatial dimension as well. In the Lite-HRNet, all features over multiple resolutions are downsampled to the minimum resolution and concatenated together. The fused features are fed into the conditional channel weighting (CCW) to estimate the weights conditioned on the input features. However, high-resolution details are missing severely because of linear interpolation due to simply downsampling features.

To reduce the computational complexity without harming high-resolution details, we explore the essential characteristic of the pose estimation task: maximizing the confidence of all human joint regions in the whole 2D space, and the estimated 2D confidence map can be horizontally and vertically projected to two 1D confidence vectors as illustrated in Figure 1b. Particularly in the predominant top-down pipeline (i.e., detect and estimate) [3, 4, 16, 17], K independent 2D heatmaps are separately estimated to represent the confidence for K human joints. And the 2D heatmap for each joint can be reconstructed by a pair of 1D heat vectors, as shown in Figure 1c.

Inspired by this, we introduce a lightweight unit called **S**patially **U**nidimensional **S**elf-**A**ttention (**SUSA**) as an alternative to the computationally costly pointwise (1×1) convolution in high-resolution pose estimation. It firstly groups the input features along one spatial dimension (H or W) by **S**tripe **C**ontext **M**odeling (**SCM**). And then, the grouped features are

fed into **S**patially **U**nidimensional **T**ransform (**SUT**) to learn vertical and horizontal attention vectors. It is noteworthy that the SUT only needs to apply convolution on spatially unidimensional features whose computational complexity is linear to the spatial dimension, while the pointwise (1×1) convolution is on quadratic complexity. We substitute both pointwise (1×1) convolutions in the shuffle block [9] and employ the resulting block to build our lightweight high-resolution pose estimation network X-HRNet, where X represents the estimated cross-shape attention vectors. Without harming the high-resolution details, SUSA is equipped with comparable representation capacity but only requires 2.6% of the FLOPs of the pointwise (1×1) convolution.

Our main contributions can be summarized as follows:
- We introduce a lightweight and efficient module, SUSA, for high-resolution pose estimation. Our SUSA performs calculations over the spatially unidimensional feature instead of the 2D feature, significantly reducing computational complexity.
- With the proposed SUSA, we introduce an efficient network, X-HRNet. And X-HRNet achieves the state-of-the-art in terms of complexity and accuracy trade-off on the COCO benchmark.

2. METHODOLOGY

In this section, the SUSA module is first introduced and comprehensively compared with the GC block [7]. Then we use the SUSA module as the main module to build our X-HRNet.

2.1. Spatially Unidimensional Self-Attention

The SUSA module follows the design pattern of the global context block (GC block) [7] whose detailed structure as illustrated in Figure 2a, and it is formed of **S**tripe **C**ontext

Modeling (**SCM**) and **S**patially **U**nidimensional **T**ransform (**SUT**). In the following subsections, we introduce how SCM groups feature along one spatial dimension and how SUT estimates the attention vector for the orthogonal dimension in the wake of an overview of SUSA. Additionally, we expose the relationship between the GC block and our method.

Overview of SUSA. With the input feature map $x \in \mathbb{R}^{C \times H \times W}$, there are two spatial dimensions: H and W. Accordingly, we propose two corresponding SUSA: H-wise and W-wise SUSA. As illustrated in Figure 2b and Figure 2c, they are exactly identical except for processing different spatial dimensions. SUSA can be divided into three procedures: 1) **Stripe context modeling (SCM).** The SCM only groups features along one spatial dimension (H or W) with the grouping matrix x_q and outputs stripe context features, unlike global context modeling in GC block that groups the features of all positions together. 2) **Spatially unidimensional transform (SUT)**. The SUT transforms the grouped features via a pointwise (1×1) convolution which learns the attention vector over the remaining spatial dimension. 3) **Feature Aggregation.** Element-wise multiplication is employed to aggregate the learned attention vector with the input feature map. Our SUSA is formulated as

$$SUSA(x) = x * SUT(\overbrace{x_v \times x_q}^{SCM(x)}), \qquad (2)$$

where $*$ is element-wise multiplication, $\times$ refers to matrix inner-product, and x_v is reshaped from the input feature x. We instantiate the H-wise SUSA for introducing details of SCM and SUT, and the W-wise one works on the W dimension in the same way.

Stripe Context Modeling (SCM). To achieve the trade-off of presentation capacity and efficiency, we adopt a 1×1 group convolution W_q (group=C) on $x_w \in \mathbb{R}^{C \times 1 \times W}$ to compute a grouping matrix x_q, and the x_w is calculated from x via weighted averaging along H dimension. The x_q is subsequently activated by Softmax normalization to increase the dynamic range of attention. The calculation formula of x_q is shown as

$$x_q = \text{Softmax}(W_q(\Phi(x))), \qquad (3)$$

where Φ is H-wise weighted averaging. With the grouping matrix x_q and x_v reshaped from x, SCM computes the stripe context features f_h by aggregating x_v and x_q together through matrix inner-product.

Spatially Unidimensional Transform (SUT). Both CCW [13] and GC block use two cascaded 1×1 convolutions with bottleneck structure to learn the conditional weights. This trick reduces the FLOPs but introduces extra convolutions, actually slowing down the inference speed. For simplification, our H-wise SUT encodes f_h by a single 1×1 convolution and outputs the final horizontal attention vector a_h. In particular, a_h is normalized by a LayerNorm (LN) on C dimension like GC block and activated by Sigmoid function. The estimated

(a) Addition   (b) Multiplication

Fig. 3: Multiplication fusion produces a sharper peak value and smaller focus region than addition fusion. The output values are normalized after fusion.

a_h is broadcast multiplied back to x as the horizontal attention. Correspondingly, the W-wise SUT learns the vertical attention and combines it to x by element-wise multiplication. The formula of SUT is written as

$$a_h = \text{Sigmoid}(\text{LN}(W_v(f_h))). \qquad (4)$$

Relationship to global context block. Our SUSA module borrows the design scheme from the GC block [7]. The GC block is an efficient variant of Non-Local Network [18], and it aims to capture long-range dependencies in the whole 2D space. We exploit the capacity of capturing long-range dependency into grouping features along one spatial dimension and estimating the stripe context features instead of the global context features. Notably, we aggregate the stripe context features to the input features as horizontal or vertical attention vector by multiplication, while GC Block aggregates global context by addition. A toy example is shown in Figure 3, multiplication fusion produces a shaper peak value and smaller activation region than addition fusion. The GC block is designed to model the long-range dependency, and addition fusion learns a large receptive field. However, our SUSA module targets pixel-level peak value maximization, for which multiplication fusion is more suitable.

2.2. X-HRNet

To verify the effectiveness of our SUSA modules, we employ them to replace two costly pointwise (1×1) convolutions in the shuffle block [9] and name it **X-shuffle block**, where X represents the estimated cross-shape attention vectors. In the X-shuffle block shown in Figure 2d, both pointwise (1×1) convolutions are replaced by our H-wise and W-wise SUSA in order. And the order of substitution affects the performance surprisingly.

Based on the X-shuffle blocks, we stack them following the structure of HRNet to maintain the high-resolution representation and build our lightweight backbone **X-HRNet**. Particularly, there are three stages in our X-HRNet, and each stage has 2, 3, 4 branches, respectively. The channel dimensions of each resolution branch are C, $2C$, $4C$, $8C$, where C is set as 40, following the setting in [13]. Besides, all standard 1×1 convolutions in fusion modules of HRNet are replaced by depthwise separable 3×3 convolutions like [13]. We also

Table 1: **Architectures of X-HRNet backbones.** N in X-HRNet-N means the number of blocks. #channels indicates the current stage containing multi-branches with a corresponding number of channels.

layer	block	#channels	#blocks	#modules	
				X-HRNet-18	X-HRNet-30
image		3			
stem	conv2d	32	1	1	1
	shuffle	32	1		
stage1	X-shuffle	40,80	2	2	3
	fusion	40,80	1		
stage2	X-shuffle	40,80,160	2	4	8
	fusion	40,80,160	1		
stage3	X-shuffle	40,80,160,320	2	2	3
	fusion	40,80,160,320	1		
#Params				1.3M	2.1M
FLOPs				194.5M	300.4M

instantiate X-HRNet-18 and X-HRNet-30 for a fair comparison with [13]. The exhaustive architectures of X-HRNet are illustrated in Table 1.

3. EXPERIMENTS

We evaluate our approach on the COCO dataset [19] and report the comparisons with start-of-the-arts. We also perform comprehensive ablation studies to investigate the effects of each proposed component in our approach.

3.1. Settings

Datasets. The MS-COCO dataset [19] includes over 60k images and 250k person instances annotated with 17 human joint categories, *e.g.,* left-/right-ear, left-/right-elbow. We train our X-HRNet on the COCO tran2017 dataset, including $57K$ images and $150K$ person instances. The val2017 set and test-dev2017 set are utilized to evaluate our approaches, containing $5K$ and $20K$ images, respectively.

Training. We extend the human detection boxes in height or width to a fixed aspect ratio: height : width = 4 : 3, and crop the resized box from the image, which is further rescaled to a fixed size, 256×192 or 384×288. Following the data augmentation settings in [13], we randomly augment each person sample with a random scale factor ($[0.75, 1.25]$), random rotation($[-30°, 30°]$), horizontal flipping and additional half body augmentation. The network is trained from scratch on 8 NVIDIA V100 with a mini-batch size per GPU. For optimization, we adopt Adam optimizer with an initial learning rate $2e^{-3}$. The training process is terminated within 210 epochs, and the learning rate is dropped to $2e^{-4}$ and $2e^{-5}$ at the 170-th and the 200-th epoch, respectively.

Testing. We adopt the top-down paradigm (detect and estimate) with the detection results provided by previous work [16] for a fair comparison. Following the common practice [3, 4, 6], we compute the final heatmap by averaging the
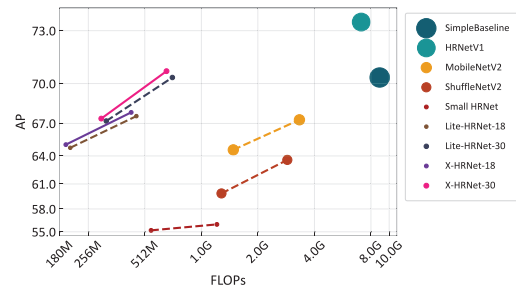

Fig. 4: **Comparisons on the COCO val2017 set.** The size of points denotes the #Params for the corresponding method. Points with the same color represent the same method trained with different input sizes: 256×192 and 384×288. The single points represent those large networks trained with 384×288. To highlight our method, we draw solid lines for our method and dashed lines for other methods. Best viewed in color.

heatmaps of the original and flipped images. The coordinates of keypoints are decoded from the heatmaps by adjusting the highest response location.

Evaluation metric. The OKS-based mAP metric is adopted to evaluate our approach. The OKS (Object Keypoint Similarity) measures the similarity between ground truth poses and predicted poses. We report standard mean average precision and recall scores: AP (the mean of AP scores with OKS threshold = [0.5:0.95:0.05]), AP^{50}, AP^{75}, AP^M (AP score for instances with medium size), AP^L (AP score for instances with large size), AR for a fair comparison with start-of-the-arts.

3.2. Comparison with the State-of-the-Arts

COCO val2017. As illustrated in Figure 4, our X-HRNet requires much less FLOPs and achieves competitive performance when compared with other start-of-the-art methods. When the input size is 256×192, our X-HRNet-30 achieves a 67.5 AP score, outperforming existing lightweight methods. Compared to the dominant lightweight networks designed for classification task: MobileNetV2 [10] and ShuffleNetV2 [9], our X-HRNet achieves superior performance far less computational cost and parameters. When compared with ShuffleNetV2 [9] that is initialized by the weights pretrained on the ImageNet dataset [20], the X-HRNet-30 significantly achieves 7.6 AP gain and only requires 15% parameters and 28% FLOPs.

As for existing pose estimation approaches, our X-HRNet-18 outperforms the small HRNet by 9.9 AP points only requiring 35% FLOPs. The performance gap is further enlarged to 11.8 when the input size is expanded to 384×288, which demonstrates the superiority of our SUSA module in capturing high-resolution details in our approach. Regardless of different input sizes or model depths, X-HRNet comprehensively surpasses Lite-HRNet in performance with less FLOPs. Compared with the large pose estimation network,

Table 2: **Comparisons on the COCO val2017 dataset.** #Params and FLOPs are only calculated for the estimation network, not including those for human detection and keypoint grouping. 'pretrain' refers to pretraining the backbone for ImageNet classification [20].

model	backbone	pretrain	input size	#Params	FLOPs	AP	AP^{50}	AP^{75}	AP^M	AP^L	AR
Large Networks											
SimpleBaseline [16]	ResNet-50	Y	256×192	34.0M	8.90G	70.4	88.6	78.3	67.1	77.2	76.3
HRNetV1 [4]	HRNetV1-W32	N	256×192	28.5M	7.10G	73.4	89.5	80.7	70.2	80.1	78.9
Lightweight Classification Networks											
MobileNetV2 [10]	MobileNetV2	Y	256×192	9.6M	1.48G	64.6	87.4	72.3	61.1	71.2	70.7
ShuffleNetV2 [9]	ShuffleNetV2	Y	256×192	7.6M	1.28G	59.9	85.4	66.3	56.6	66.2	66.4
DY-ReLU [21]	MobileNetV2	Y	256×192	9.0M	1.03G	68.1	88.5	76.2	64.8	74.3	-
Lightweight Pose Estimation Networks											
Small HRNet	HRNet-W16	N	256×192	1.3M	0.54G	55.2	83.7	62.4	52.3	61.0	62.1
Small HRNet	HRNet-W16	N	384×288	1.3M	1.21G	56.0	83.8	63.0	52.4	62.6	62.6
Lite-HRNet [13]	Lite-HRNet-18	N	256×192	**1.1M**	205.2M	64.8	86.7	**73.0**	62.1	70.5	71.2
X-HRNet (**Ours**)	X-HRNet-18	N	256×192	1.3M	**194.4M**	**65.1**	86.7	72.7	**62.3**	**70.9**	**71.2**
Lite-HRNet	Lite-HRNet-18	N	384×288	**1.1M**	461.6M	67.6	**87.8**	75.0	64.5	73.7	**73.7**
X-HRNet (**Ours**)	X-HRNet-18	N	384×288	1.3M	**433.2M**	**67.9**	87.6	**75.5**	**64.7**	**73.9**	73.6
Lite-HRNet	Lite-HRNet-30	N	256×192	**1.8M**	319.2M	67.2	**88.0**	75.0	64.3	73.1	73.3
X-HRNet (**Ours**)	X-HRNet-30	N	256×192	2.1M	**300.2M**	**67.4**	87.5	**75.4**	**64.5**	**73.3**	**73.5**
Lite-HRNet	Lite-HRNet-30	N	384×288	**1.8M**	717.8M	70.4	88.7	77.7	67.5	76.3	**76.2**
X-HRNet (**Ours**)	X-HRNet-30	N	384×288	2.1M	**668.0M**	**70.6**	**88.9**	77.7	**67.6**	**76.5**	76.1

Table 3: **Comparisons on the COCO test-dev2017 dataset.** The input size is set as 384×288. #Params and FLOPs are only calculated for the estimation network, not including those for human detection and keypoint grouping.

model	backbone	#Params	FLOPs	AP	AP^{50}	AP^{75}	AR
Large Networks							
SimpleBaseline [16]	ResNet-152	68.6M	35.6G	73.7	91.9	81.1	79.0
HRNetV1 [4]	HRNetV1-w48	63.6M	32.9G	75.5	92.5	83.3	80.5
Lightweight networks							
MobileNetV2 [10]	MobileNetV2	9.6M	3.33G	66.8	90.0	74.0	72.3
ShuffleNetV2 [9]	ShuffleNetV2	7.6M	2.87G	62.9	88.5	69.4	68.9
Small HRNet	HRNet-16	1.3M	1.21G	55.2	85.8	61.4	61.5
Lite-HRNet [13]	Lite-HRNet-18	**1.1M**	461.6M	66.9	89.4	74.4	72.6
X-HRNet (**Ours**)	X-HRNet-18	1.3M	**433.2M**	67.3	89.8	74.8	73.0
Lite-HRNet	Lite-HRNet-30	**1.8M**	717.8M	69.7	**90.7**	77.5	75.4
X-HRNet (**Ours**)	X-HRNet-30	2.1M	**668.0M**	70.0	90.6	77.7	75.5

SimpleBaseline [16], the X-HRNet-30 significantly reduces the complexity by 97% but merely a 4% performance drop.

COCO test-dev2017. With the input size 384×288, our X-HRNet-18, X-HRNet-30 respectively achieve 67.3 and 70.0 AP scores on COCO test-dev2017 dataset, as reported in Table 3. When compared with the Lite-HRNet, our X-HRNet maintains the superiority, which demonstrates the robustness of our X-HRNet. Compared to large networks, our X-HRNet-30 also achieves comparable performance but requires far less FLOPs and parameters.

3.3. Ablation Study

We perform ablation studies and report the results on the COCO val2017 set. And all results are obtained with the input size of 256×192.

The effect of SUSA modules. To compare our SUSA modules with the costly 1×1 convolution, we make the comparison between our X-HRNet and wider naive Lite-HRNet (WNL-HRNet) [13] whose main block is the standard shuffle block. And we simply remove two 1×1 convolutions in all shuffle blocks and set the resulting network as the baseline. As reported in Table 4, the WNL-HRNet achieves a 65.7 AP score, which is decreased by 4.4 points when removing two 1×1 convolutions in shuffle blocks. Meanwhile, the removing also reduces the FLOPs by 122.2M, and 61.1 MFLOPs for each 1×1 convolution in shuffle block. By inserting one H-wise or W-wise SUSA after the depthwise 3×3 convolution in each incomplete shuffle block, the baseline is respectively improved by 2.0, 1.6. And with both of them (X-HRNet), the performance is consistently increased by 3.8, which is larger than the sum of AP improvements (3.6) of adding either one. It is worth noting that our SUSA modules only cost 2.7 MFLOPs that is only 4% of the 1×1 convolution but achieves a comparable AP score, 65.1. The 1×1 convolution plays a significant role in information exchange across channels, but it is computationally costly. Our SUSA module plays a similar role as 1×1 convolution by learning cross-shape attention vectors and only needs to perform convolution calculation on a single spatial dimension.

The effect of different fusion types. The different choices on fusion are influential for the SUSA module to achieve high performance. And we respectively apply addition and multiplication fusion on the GC block and our SUSA module to verify our analysis in 2.1. Ablation results on different fusion are shown in Table 5. The experimental results are consistent with our aforementioned theoretical analysis. The multiplication fusion is more suitable for the human pose estimation task than addition fusion, which is inverse for object detection or segmentation as illustrated in [7].

Table 4: **Ablation study about SUSA modules.** All results are evaluated on COCO val2017 set and the input size is 256×192. WNL-HRNet-18 = wider naive Lite-HRNet [13], baseline refers to dropping two 1×1 convolutions from shuffle blocks in WNL-HRNet-18, + SUSA = inserting the SUSA module into the incomplete shuffle block.

model	#Params	FLOPs	AP	AP^{50}	AP^{75}	AR
WNL-HRNet-18	1.3M	311.1M	65.7	87.0	73.3	71.8
Baseline	0.9M	188.9M	61.3	85.3	68.7	67.7
+ W-wise SUSA	1.1M	191.2M	62.9	85.7	70.5	69.1
+ H-wise SUSA	1.1M	191.9M	63.3	85.9	70.7	69.6
X-HRNet-18	1.3M	194.4M	**65.1**	**86.7**	**72.7**	**71.2**

Table 5: **Ablation study about different fusion types.** All results are evaluated on the COCO val2017 set, and the input size is 256×192. The 'add' denotes addition fusion, and the 'mul' refers to multiplication fusion.

model	#Params	FLOPs	AP	AP^{50}	AP^{75}	AR
X-HRNet (add)	1.3M	194.4M	63.3	85.9	70.7	69.6
X-HRNet (mul)	1.3M	194.4M	**65.1**	**86.7**	**72.7**	**71.2**

The effect of the order of H-wise and W-wise SUSA. As mentioned in Sec. 2.2, the order of H-wise and W-wise SUSA in the X-shuffle block has a surprising influence on performance. The X-HRNet outperforms the reversed one by 0.7 AP points, and they have the same #Params and FLOPs, as shown in Table 6. This peculiar discrepancy is caused by the depthwise 3×3 convolution between two SUSA modules, which enlarges the receptive field for the second SUSA. And for the COCO benchmark dataset, the prescriptive input size is 256×192 or 384×288, where height is larger than width. Consequently, the X-shuffle block whose H-wise SUSA is equipped with the larger receptive field outperforms the reversed one.

4. CONCLUSION

This paper presents a lightweight human pose estimation approach named X-HRNet. We explore the essential characteristic of the pose estimation task: estimating human joints by 2D single-peak heatmaps, where each 2D heatmap can be horizontally and vertically projected to and reconstructed by a pair of 1D heat vectors. Inspired by this characteristic, we propose a lightweight module named Spatially Unidimensional Self-Attention (SUSA). The resulting lightweight pose estimation network X-HRNet achieves the state-of-the-art in terms of complexity and accuracy trade-off on the COCO benchmark.

Table 6: **Ablation study about the arrangement order of the H-wise and W-wise SUSA.** ⊤ indicates the arrangement order of the H-wise and W-wise SUSA in the X-shuffle block is reversed.

model	#Params	FLOPs	AP	AP^{50}	AP^{75}	AR
X-HRNet-18⊤	1.3M	194.4M	64.5	86.4	72.3	70.8
X-HRNet-18	1.3M	194.4M	**65.1**	**86.7**	**72.7**	**71.2**

5. REFERENCES

[1] Vijay Badrinarayanan et al., "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *Trans.PAMI*, vol. 39, no. 12, pp. 2481–2495, 2017.

[2] Liang-Chieh Chen et al., "Semantic image segmentation with deep convolutional nets and fully connected crfs," *ICLR*, 2014.

[3] Ke Sun et al., "Deep high-resolution representation learning for human pose estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 5693–5703.

[4] Jingdong Wang et al., "Deep high-resolution representation learning for visual recognition," *Trans.PAMI*, 2020.

[5] Bowen Cheng et al., "Higherhrnet: Scale-aware representation learning for bottom-up human pose estimation," in *CVPR*, 2020, pp. 5386–5395.

[6] Jia Li et al., "Simple pose: Rethinking and improving a bottom-up approach for multi-person pose estimation," in *AAAI*, 2020, pp. 11354–11361.

[7] Yue Cao et al., "Gcnet: Non-local networks meet squeeze-excitation networks and beyond," in *CVPR*, 2019.

[8] Xiangyu Zhang et al., "Shufflenet: An extremely efficient convolutional neural network for mobile devices," in *CVPR*, 2018, pp. 6848–6856.

[9] Ningning Ma et al., "Shufflenet v2: Practical guidelines for efficient cnn architecture design," in *ECCV*, 2018, pp. 116–131.

[10] Andrew Howard et al., "Inverted residuals and linear bottlenecks: Mobile networks for classification, detection and segmentation," in *CVPR*, 2018.

[11] François Chollet, "Xception: Deep learning with depthwise separable convolutions," in *CVPR*, 2017, pp. 1251–1258.

[12] Andrew G Howard et al., "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.

[13] Changqian Yu et al., "Lite-hrnet: A lightweight high-resolution network," in *CVPR*, 2021, pp. 10440–10450.

[14] Changqian Yu et al., "Bisenet: Bilateral segmentation network for real-time semantic segmentation," in *ECCV*, 2018, pp. 325–341.

[15] Tsung-Yi Lin et al., "Feature pyramid networks for object detection," in *CVPR*, 2017, pp. 2117–2125.

[16] Bin Xiao et al., "Simple baselines for human pose estimation and tracking," in *ECCV*, 2018, pp. 466–481.

[17] Yilun Chen et al., "Cascaded pyramid network for multi-person pose estimation," in *CVPR*, 2018, pp. 7103–7112.

[18] Xiaolong Wang et al., "Non-local neural networks," in *CVPR*, 2018, pp. 7794–7803.

[19] Tsung-Yi Lin et al., "Microsoft coco: Common objects in context," in *ECCV*. Springer, 2014, pp. 740–755.

[20] Alex Krizhevsky et al., "Imagenet classification with deep convolutional neural networks," *NeurIPS*, vol. 25, pp. 1097–1105, 2012.

[21] Yinpeng Chen et al., "Dynamic relu," in *ECCV*. Springer, 2020, pp. 351–367.