Supplementary Material: Aligned Structured Sparsity Learning for Efficient Image Super-Resolution

Yulun Zhang[1,†] **Huan Wang**[1,†,*] **Can Qin**[1] **Yun Fu**[1,2]
[1]Department of ECE, Northeastern University
[2]Khoury College of Computer Science, Northeastern University

1 Method

1.1 Algorithm

Our proposed aligned structured sparsity learning (ASSL) algorithm is summarized in Algorithm 1.

2 Experimental Results

2.1 Regularization Visualization

More plots of the pruning process of our ASSL approach on the EDSR_baseline network are shown in Fig. 1. There are in total 16 residual blocks in EDSR_baseline. We evenly pick 5 blocks to plot here (layer name is texted as title in each sub-plot). The free Conv layers are plotted in the left and constrained Conv layers plotted in the right. As seen, similar to layer `model.body.8.body.0` depicted in the main text, all the other plots pose the same trend that *pruned WN scales are compressed towards zero and the kept WN scales arise accordingly to compensate the signal energy loss*. This spontaneous reaction of the kept WN scale parameters prevents the network from catastrophic expressivity damage, hence the performance superiority of our method against the counterparts.

2.2 Visual Comparisons

We provide more visual comparisons in Fig. 2. For example, in the right part of img_005, we can observe that most of the compared methods cannot recover structural details with proper directions. In contrast, our ASSLN can better recover more structural details. In img_033, most compared methods suffer from blurring artifacts. While, our ASSLN can better alleviate the blurring artifacts. We can find similar observations in other images. These visual comparisons are consistent with the quantitative results shown in the main paper, demonstrating the superiority of our method. Our ASSLN learns the aligned structured sparsity from a large network and prunes it to a much smaller one, but still maintains most representation ability.

3 Explanations for Checklist

3.1 Limitations

In this work, we mainly focus on the most commonly used network module: residual block (RB). For other basic modules, like dense connection [4], we have not investigated yet.

[†] Equal Contribution
[*] Corresponding author: Huan Wang (wang.huan@northeastern.edu)

35th Conference on Neural Information Processing Systems (NeurIPS 2021), Sydney, Australia.

Algorithm 1: Aligned Structural Sparsity Learning (ASSL)

```
Input: Pretrained SR network $\Theta$, regularization increment $\Delta$, interval $T$, penalty ceiling limit $\tau$, iterations
  of SSA $N_{SSA}$.
Output: Small model $\Theta'$.
Init: $\alpha = 0$ for all filters; iteration $i = 0$; constrained Conv layer set $\mathcal{C}$.
Init: Insert a weight normalization layer after each convolutional layer.
Init: for $l$ = $1 \sim L$ do
    Set $S^{(l)}$ by $L_1$-norm sorting of the filter norms, if $l$ not in $\mathcal{C}$.
end
Pruning:
while True do
    for $l$ = $1 \sim L$ do
        if $l$ not in $\mathcal{C}$ then
            if $i\%T = 0$ then
                $\alpha_j^{(l)} = \min(\alpha_j^{(l)} + \Delta, \tau)$ for $j \in S^{(l)}$.
            end
            Add $\mathcal{L}_{SI}$ (Eq. (4), $\mathcal{L}_{SI} = \alpha \sum_{l=1}^{L} \sum_{i \in S^{(l)}} \gamma_i^2$) to the total loss.
        end
        else
            if $i$ < $N_{SSA}$ then
                Add $\mathcal{L}_{SSA}$ (Eq. (5), $\mathcal{L}_{SSA} = -\frac{1}{K} \sum_{k=1}^{K} (MM^T)_k$) to the total loss.
            end
            if $i$ = $N_{SSA}$ then
                Set $S^{(l)}$ by $L_1$-norm sorting of the filter norms.
            end
            if $i$ > $N_{SSA}$ then
                if $i\%T = 0$ then
                    $\alpha_j^{(l)} = \min(\alpha_j^{(l)} + \Delta, \tau)$ for $j \in S^{(l)}$.
                end
                Add $\mathcal{L}_{SI}$ (Eq. (4), $\mathcal{L}_{SI} = \alpha \sum_{l=1}^{L} \sum_{i \in S^{(l)}} \gamma_i^2$) to the total loss.
            end
        end
    end
    Loss backward and parameter update by SGD.
    Iteration adds up: $i$ += 1.
    if $\alpha$'s in all layers reach $\tau$ then
        break. // all layers finished pruning
    end
end
Remove the filters in $S^{(l)}$ for each $l$-th layer. Remove all weight normalization layers (scales merged with
filter weights). Rebuild to obtain the pruned model.
Finetune the pruned model and output the final model as $\Theta'$.
```

3.2 Potential Negative Societal Impacts

We believe that our efficient image super-resolution (SR) technique: ASSL would benefit to both academic and industry. We think there has few potential negative societal impacts.

References

[1] Namhyuk Ahn, Byungkon Kang, and Kyung-Ah Sohn. Fast, accurate, and lightweight super-resolution with cascading residual network. In *ECCV*, 2018. 5

[2] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Learning a deep convolutional network for image super-resolution. In *ECCV*, 2014. 5

[3] Chao Dong, Chen Change Loy, and Xiaoou Tang. Accelerating the super-resolution convolutional neural network. In *ECCV*, 2016. 5

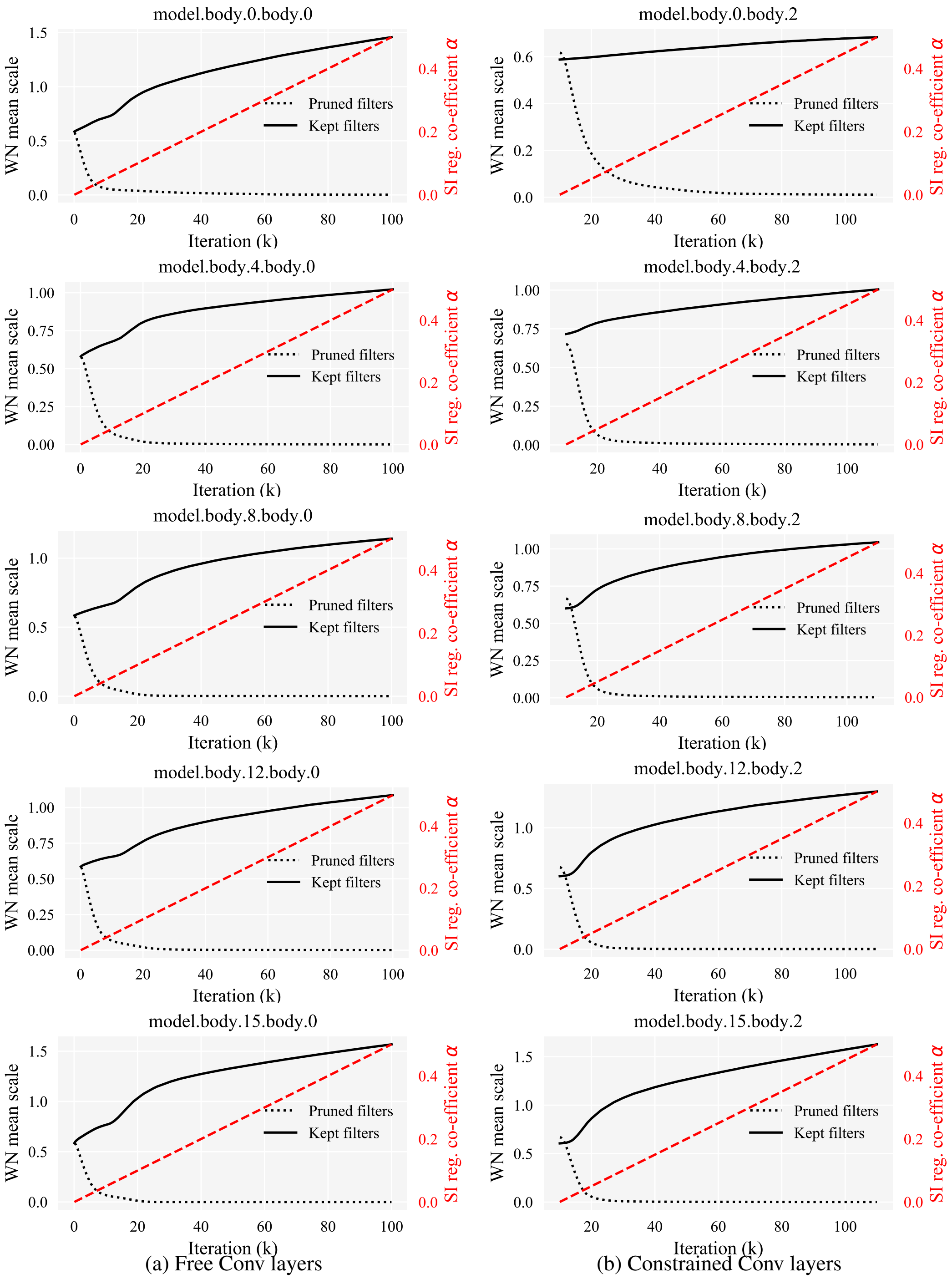
[4] Gao Huang, Zhuang Liu, Kilian Q Weinberger, and Laurens van der Maaten. Densely connected convolutional networks. In *CVPR*, 2017. 1

[5] Zheng Hui, Xinbo Gao, Yunchu Yang, and Xiumei Wang. Lightweight image super-resolution with information multi-distillation network. In *ACM MM*, 2019. 5

[6] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee. Accurate image super-resolution using very deep convolutional networks. In *CVPR*, 2016. 5

[7] Wei-Sheng Lai, Jia-Bin Huang, Narendra Ahuja, and Ming-Hsuan Yang. Deep laplacian pyramid networks for fast and accurate super-resolution. In *CVPR*, 2017. 5

[8] Ying Tai, Jian Yang, Xiaoming Liu, and Chunyan Xu. Memnet: A persistent memory network for image restoration. In *ICCV*, 2017. 5



(a) Free Conv layers

(b) Constrained Conv layers

Figure 1: Illustration of the pruning process of five free Conv layers (a) and constrained Conv layers (b) in EDSR_baseline. The weight normalization (WN) mean scale of pruned or kept filters are plotted to the left y-axis (in black); Sparsity-inducing (SI) regularization co-efficient is plotted to the right y-axis (in red). In the main paper, layer "model.body.8.body.0" is presented.

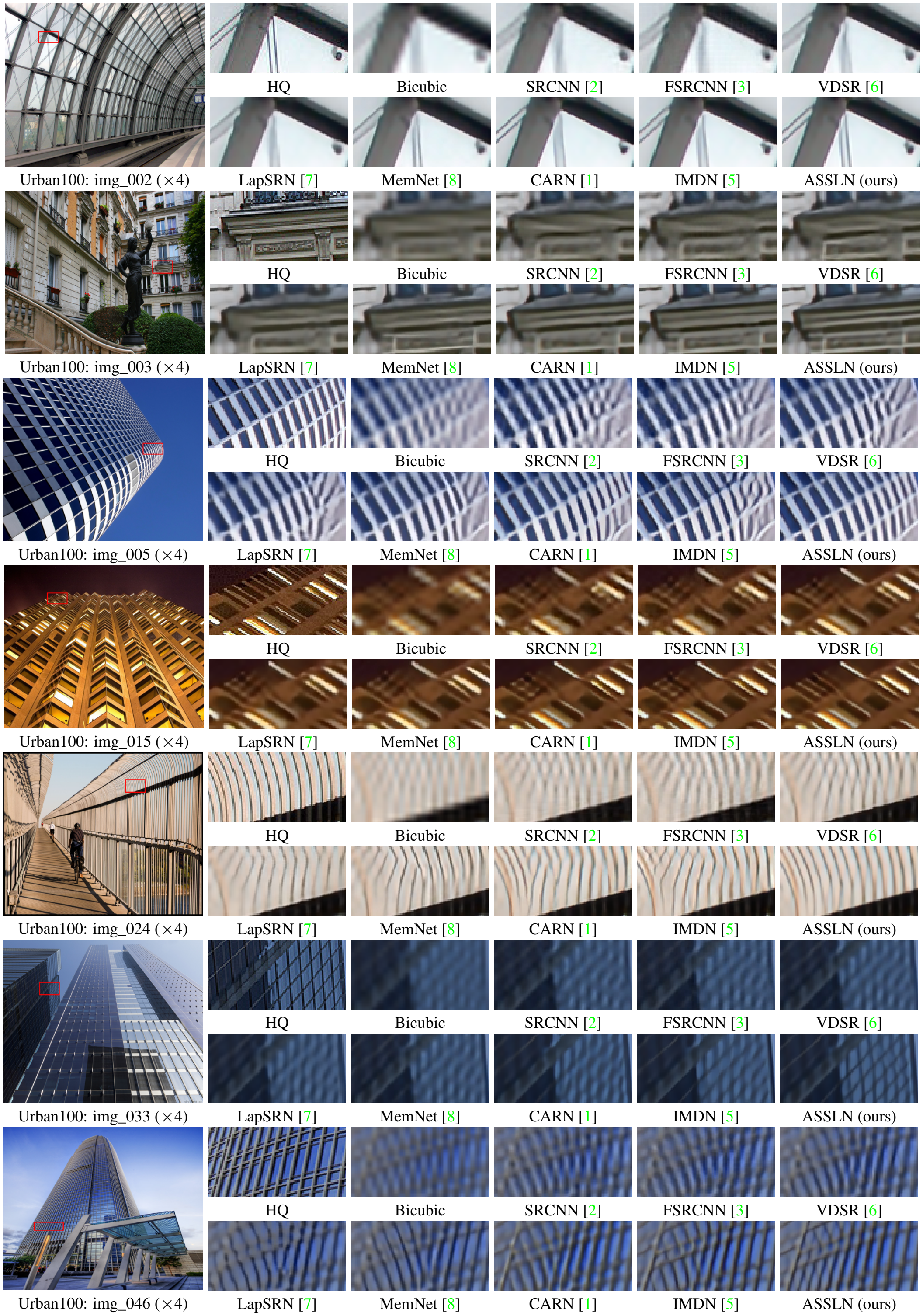


Figure 2: Visual comparison (×4) with lightweight SR networks on Urban100 dataset.