Supplementary Material for Sparsifiner: Learning Sparse Instance-Dependent Attention for Efficient Vision Transformers

Cong Wei[1,3*] Brendan Duke[1,3*] Ruowei Jiang[3] Parham Aarabi[1,3] Graham W. Taylor[2,4] Florian Shkurti[1,4]

[1]University of Toronto [2]University of Guelph [3]Modiface, Inc. [4]Vector Institute

1. Visualization

We present more visualization results (Fig. 1) and (Fig. 2) to show the connectivity mask predicted by Sparsifiner. We show the original input image and the connectivity mask of the query patch, where the dark regions represent tokens that are not attended to by the query patch token. For each attention head, Sparsifiner generates a corresponding connectivity mask. We also visualize the sparse attention map efficiently generated using the connectivity mask and compare it with the full attention map. The visualization 1 shows that Sparsifiner is able to generate different connectivity masks for different tokens in the same image. In particular, the connectivity mask for the query patch on the person focuses on the region surrounding the person. In contrast, the connectivity mask for the query patch on the horse focuses on the region surrounding the horse. This further demonstrates the effectiveness of Sparsifiner's mask predictor in producing semantically meaningful connectivity masks. In the second group of visualization 2, we show the connectivity mask generated by Sparsifiner with a very low attention budget of 20 which is 1% of total number of tokens. In particular, the connectivity mask for the query patch on one rooster focuses on the region surrounding other roosters. This further demonstrates that Sparsifiner is able to utilize limited attention connectivities effectively.

2. Training Details

We conduct our experiments on the ImageNet dataset [1]. We train our models on the training set, which consists of 1.28M images. We report the top-1 accuracy on the 50k validation images. The image resolution in training and testing is 224×224 unless otherwise specified. By default, the connectivity mask predictor module is incorporated into every layer of DeiT-S [3] and LV-ViT-S [2]. In all of our experiments, we set the reduced dimension n_{down} to 32, since this setting leads to a decent trade-off between complexity and performance. The attention budget B is in the range $(0, \text{number of tokens}]$. Budget B is directly determined by the attention keep rate in $(0, 1]$ as the ceiling of the keep rate multiplied by the total number of tokens. In the default Sparsifiner-S, we set the attention keep rate to 0.25, thus the attention budget is $B = 49$.

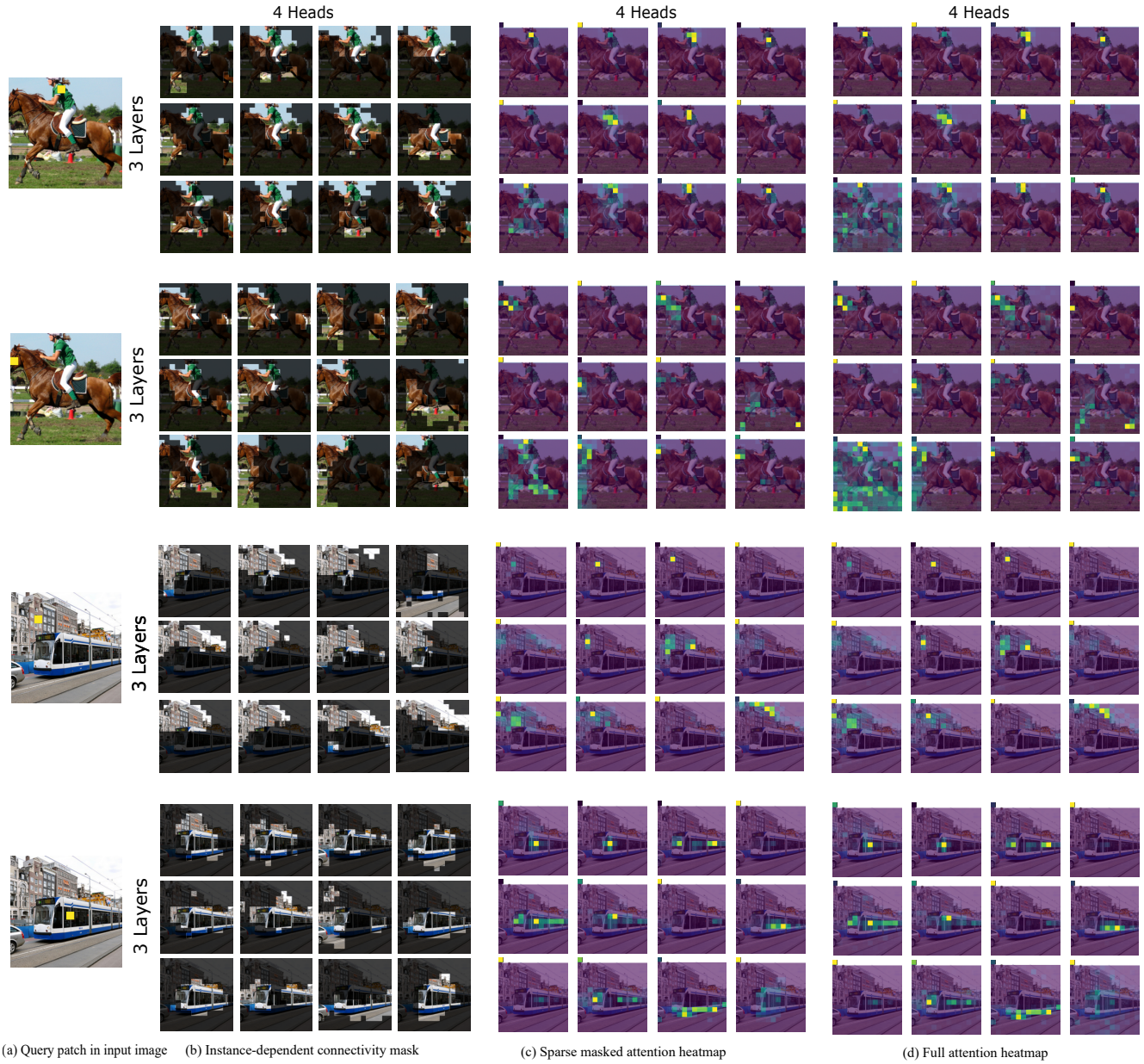
We follow most of the training techniques used in DeiT-S and LV-ViT-S. We use pre-trained vision transformer models to initialize the backbone models. The default Sparsifiner-S uses DeiT-S as the backbone model. To improve speed of convergence, we propose a two-phase training strategy. In the first phase, we freeze the backbone model and train the connectivity mask predictor module with attention distillation loss and L2 regularization only for 5 epochs. Specifically, we set $\lambda^{\text{token}}_{\text{distill}} = 0.0$, $\lambda^{\text{cls}}_{\text{distill}} = 0.0$, $\lambda^{\text{attn}}_{\text{distill}} = 1.0$ and we set the weight decay as 0.05 in the optimizer. There is no threshold applied in the first phase. We found that the first phase training helps the connectivity mask predictor to learn W^{up} quickly and the loss converges within 5 epochs. After the first stage of training, we apply a threshold on the learned basis W^{up} to prune noisy values in the basis. By default we set the threshold to be 0.01, which prunes over 90% of the basis value. In the second phase, we freeze the connectivity mask predictor module and fine-tune the backbone for another 40 epochs. We set $\lambda^{\text{token}}_{\text{distill}} = 0.5$, $\lambda^{\text{cls}}_{\text{distill}} = 0.5$, and $\lambda^{\text{attn}}_{\text{distill}} = 0.0$. We set the threshold τ to 0.05 on the basis coefficients A^{down} during the second phase training. After pruning, the basis coefficient $\tilde{A}^{\text{down}}$ has a ratio 87% of sparsity. So the basis and basis coefficients are both sparse. Then the connectivity score map can be computed by sparse-sparse matrix multiplication.

The batch size is adjusted adaptively for different models according to the GPU memory. All of our models are trained on a single machine with 8 GPUs.

3. Comparisons of Different Sparsifying Regularizers

We compare the performance of Sparsifiner-S trained under L1 regularization and L2 regularization in Table 1. We can

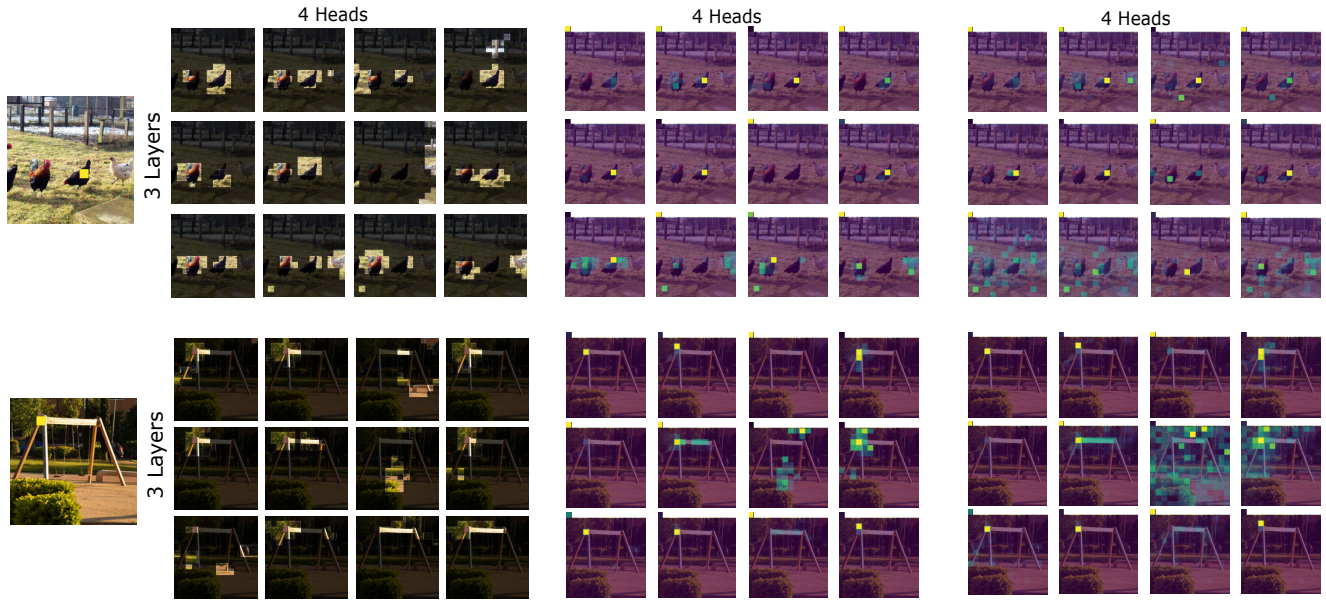
*Equal contribution.



(a) Query patch in input image (b) Instance-dependent connectivity mask (c) Sparse masked attention heatmap (d) Full attention heatmap

Figure 1. Visualization of connectivity mask with budget size of 49 (b) with sparse (c) and full (d) attention maps for a given query patch (a). In the heatmaps, the blue darker color indicates lower, and yellow brighter color indicates higher attention value. Here we visualize the attention maps for only 3 layers and 4 heads of the ViT. For both images we visualize layers 3–5. The query patch on the person (first) produces distinctly different sparse attention maps and connectivity masks compared with the query patch on the horse (second). In particular, the connectivity mask for the person focuses on the region surrounding the person, while the connectivity mask for the horse focuses on the region surrounding the horse. The query patch on the building (third) produces distinctly different sparse attention maps and connectivity masks compared with the query patch on the street car (fourth). In particular, the connectivity mask for the building focuses on the background region, while the connectivity mask for the street car focuses on the region surrounding the street car.

see that the basis trained with L1 regularization has similar sparsity ratio compared with L2 regularization, but the model degraded by almost 1% absolute percentage point in top-1 accuracy. This is because L1 regularization encourages pruning basis, which limits the expressiveness of the mask predictor. We visualize the sparse basis of the first six layers of Sparsifiner learned under L1 regularization and L2 regularization in Figure 3. We can observe the basis pruning and collapse issue under L1 regularization. Especially in the first and last layers, many bases have zero values af-



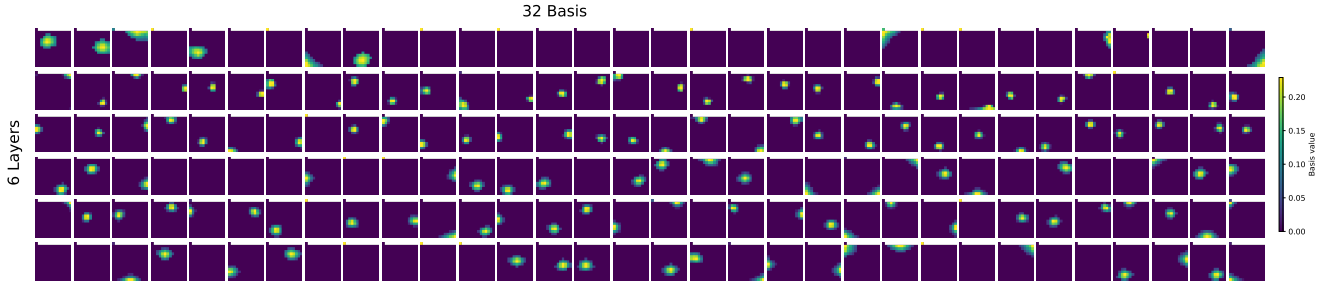
(a) Query patch in input image (b) Instance-dependent connectivity mask (c) Sparse masked attention heatmap (d) Full attention heatmap

Figure 2. Visualization of connectivity mask with budget size of 20 (b) with sparse (c) and full (d) attention maps for a given query patch (a). In the heatmaps, the blue darker color indicates lower, and yellow brighter color indicates higher attention value. Here we visualize the attention maps for only 3 layers and 4 heads of the ViT. For both images we visualize layers 3–5. Under an extremely low attention budget, the Sparsifiner can still produce meaningful connectivity masks for the rooster (top) and swing set (bottom). In particular, the connectivity mask for the rooster focuses on the region surrounding other roosters, while the connectivity mask for the swing set focuses on different parts of the swing set.

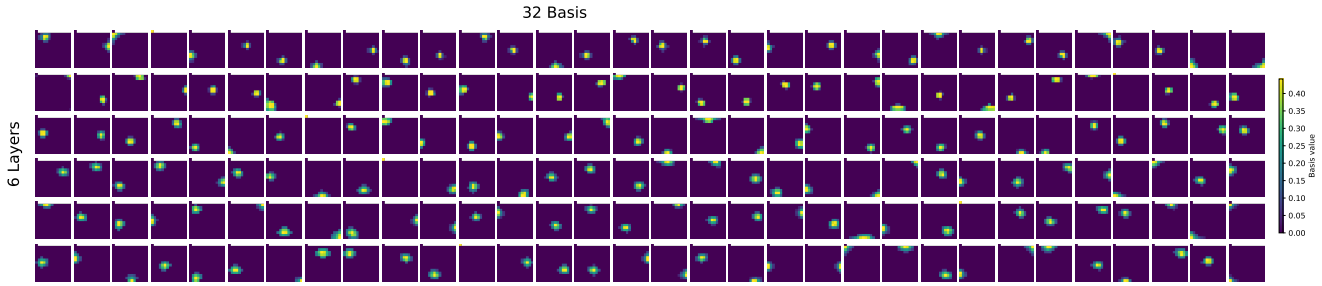
ter applying the threshold, and some of the remaining bases collapse into large dense bases. In contrast, L2 regularization discourages pruning entire bases and none of the bases get pruned. Note that some bases only have high intensity value in the top-left corner are bases corresponding to the [CLS] token. After pruning low values, the basis trained under L2 regularization still has a decent level of sparsity and is more expressive.

Model	Regularization strategy	Basis threshold	Basis sparsity ratio	Top-1 Acc (%)
Sparsifiner-S	L1	1e-2	91.3%	78.81
Sparsifiner-S	L2	1e-2	90.8%	79.79

Table 1. Comparison of Sparsifiner-S trained under L1 regularization and L2 regularization. After applying the same threshold on basis values, basis trained with L2 regularization has similar sparsity ratio compared with L1 regularization, while the model achieves almost 1% absolute percentage point improvement in top-1 accuracy.



(a) Visualization of sparse basis learned under L1 regularization.



(b) Visualization of sparse basis learned under L2 regularization.

Figure 3. Visualization of the up-projection matrix W^{up} of the first 6 layers of Sparsifiner-S, which we refer to here as a sparse basis, under different regularization strategy. We visualize 32 dimensions of the sparse basis. Dark blue weights indicate low values, which are pruned after training so that only the bright yellow weights are left over. Since L1 regularization encourages sparsity over W^{up}, some bases have almost zero value and are completely pruned. The remaining bases collapse, which limits the expressiveness of the mask predictor. While L2 regularization discourages pruning entire bases. After pruning low values in W^{up}, the basis has a decent level of sparsity and is more expressive.

References

[1] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A Large-scale Hierarchical Image Database. In *CVPR*, 2009.

[2] Zi-Hang Jiang, Qibin Hou, Li Yuan, Daquan Zhou, Yujun Shi, Xiaojie Jin, Anran Wang, and Jiashi Feng. All tokens matter: Token labeling for training better vision transformers. *Advances in Neural Information Processing Systems*, 34:18590–18602, 2021.

[3] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Herve Jegou. Training Data-efficient Image Transformers & Distillation through Attention. In *ICML*, 2021.