Stitchable Neural Networks

Presented by Lujun LI

CVPR 2023

OUTLINE

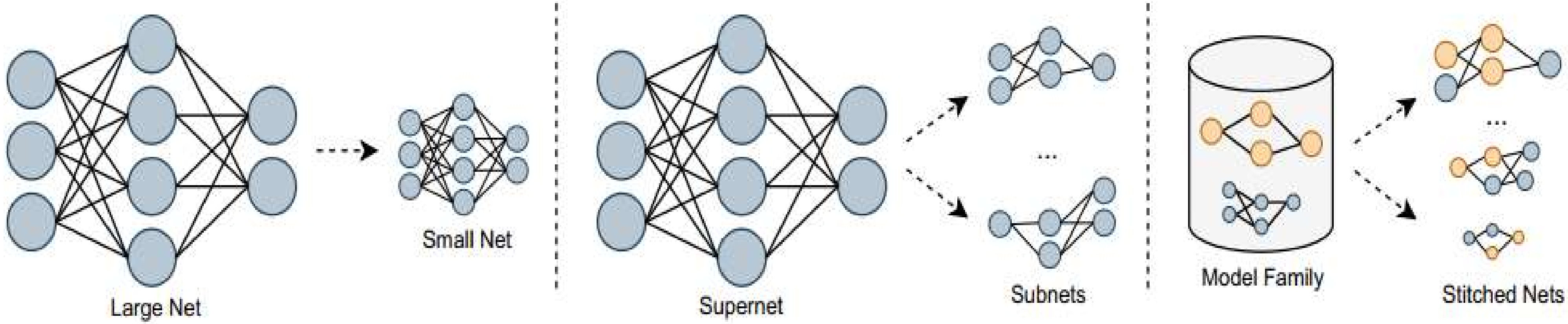
CONTENTS

- Background
- Method
- Experiment
- Conclusion

Background

Model deployment

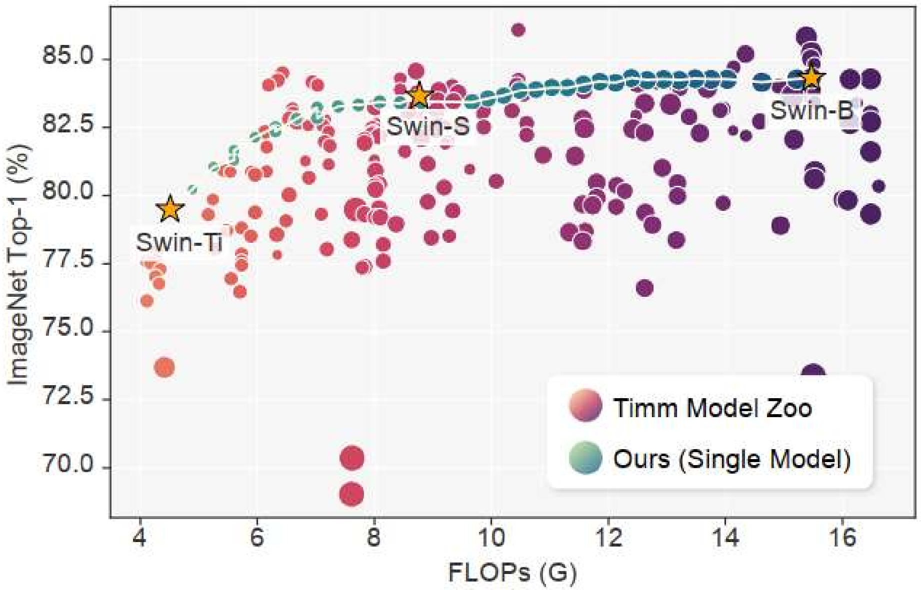
- Network Compression: pruning, quantization and knowledge distillation
- One-shot neural architecture search
- Stitchable Neural Network directly stitches the off-the-rack family of pretrained models

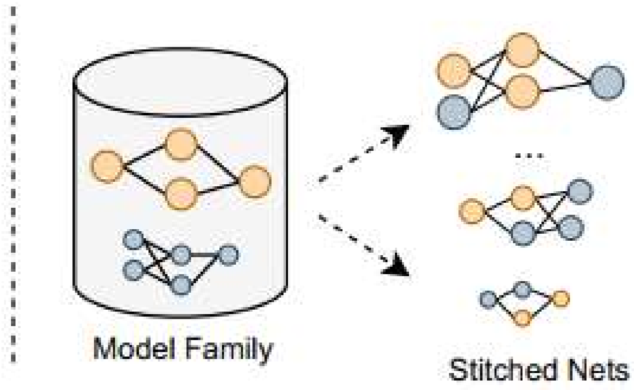


(a) Network Compression (One-to-One) (b) One-shot Neural Architecture Search (One-to-Many) (c) Stitchable Neural Networks (Many-to-Many)

Model deployment

- Network Compression: pruning, quantization and knowledge distillation
- One-shot neural architecture search
- Stitchable Neural Network directly stitches the off-the-rack family of pretrained models

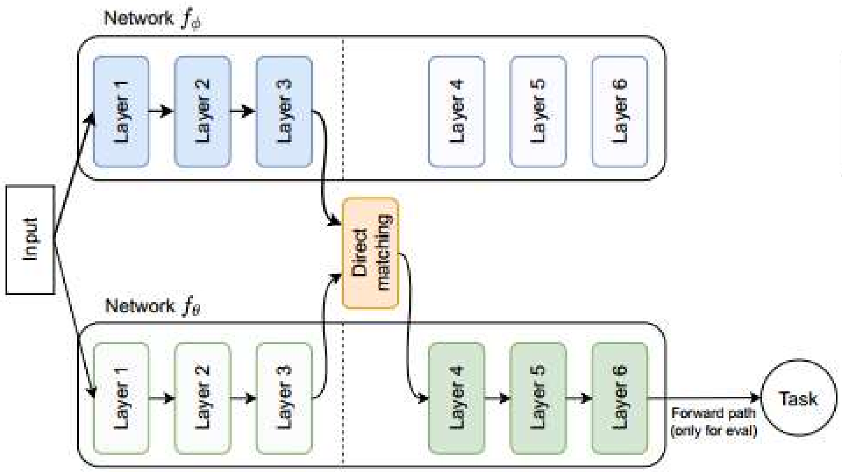




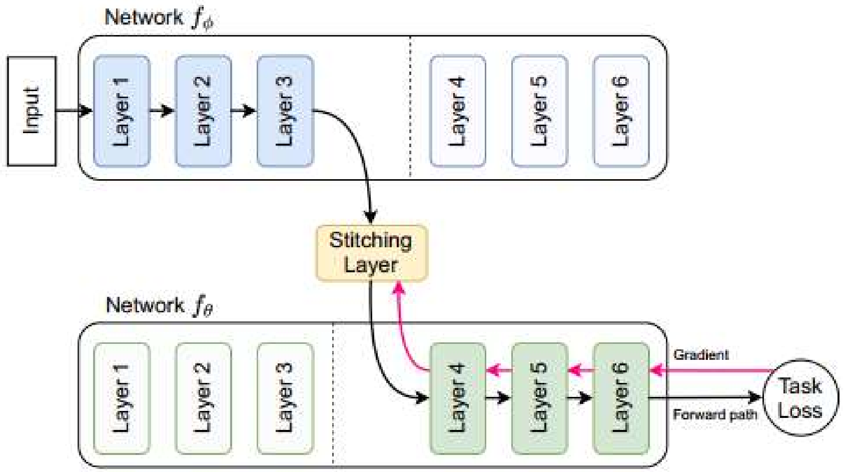
(c) Stitchable Neural Networks
(Many-to-Many)

Model stitching

- A trained network can be connected with another trained network by a 1×1 convolution stitching layer without a significant performance drop
- Representations similarity indices (e.g., CKA , CCA, SVCCA)
- DeRy: stitching pretrained model families in the large-scale model zoo



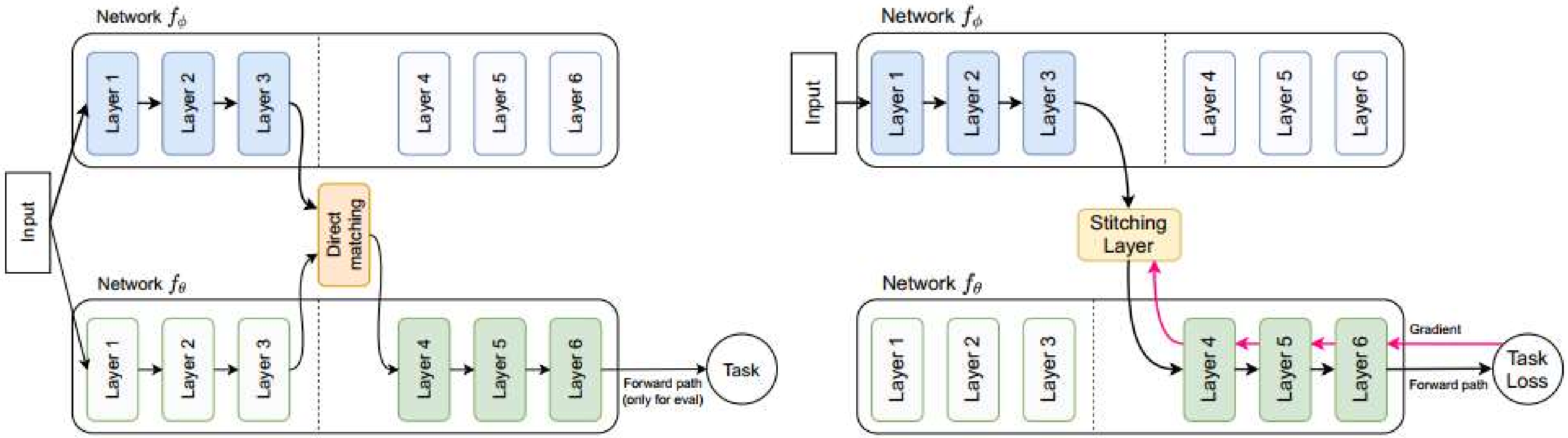
(a) Direct matching of representations.



(b) Task loss matching of representations.

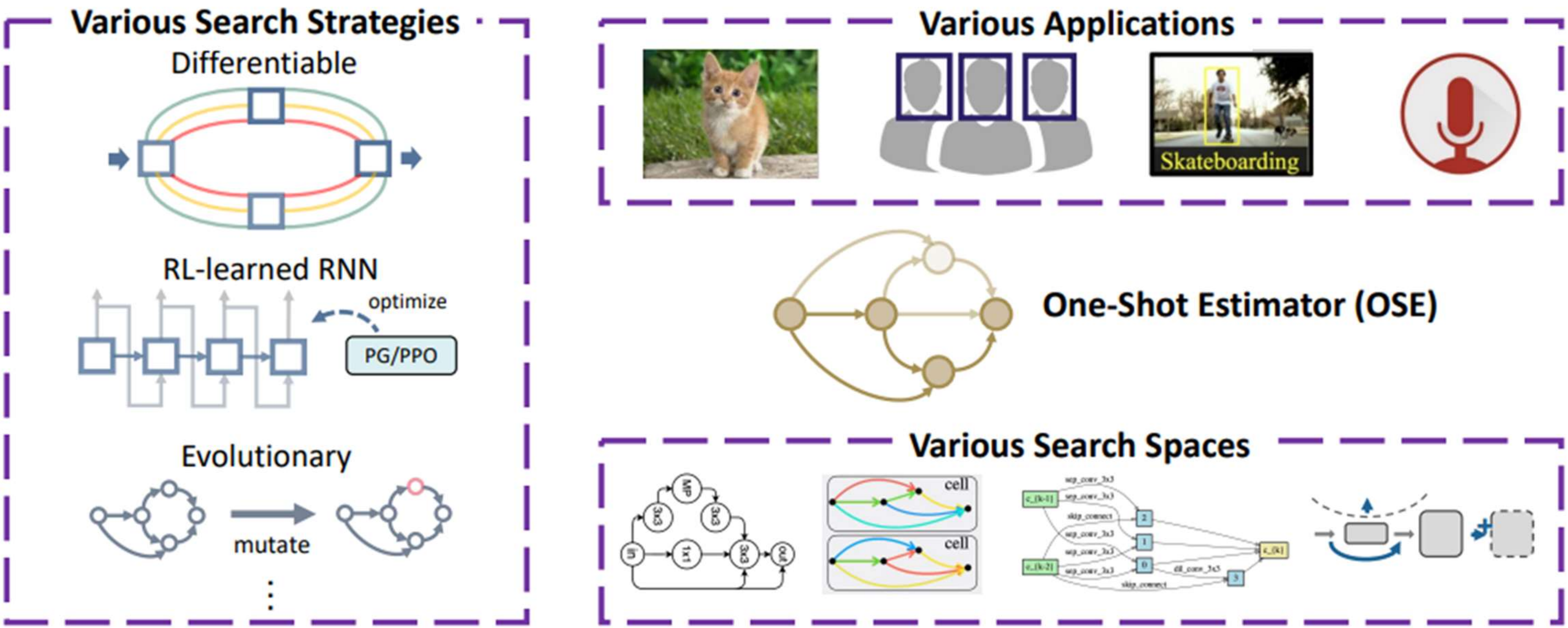
Model stitching

- A trained network can be connected with another trained network by a 1×1 convolution stitching layer without a significant performance drop
- Representations similarity indices (e.g., CKA , CCA, SVCCA)
- DeRy: stitching pretrained model families in the large-scale model zoo



(a) Direct matching of representations. (b) Task loss matching of representations.

Neural architecture search



Method

Stitchable Neural Networks

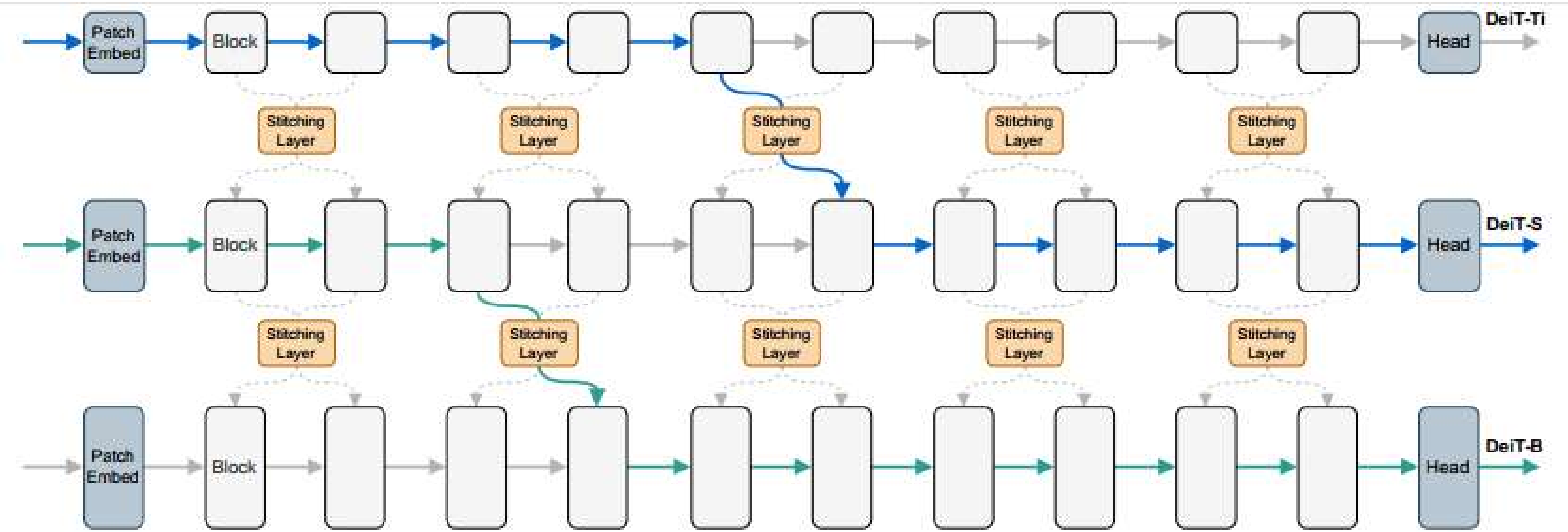


Figure 3. Illustration of the proposed **Stitchable Neural Network**, where three pretrained variants of DeiTs are connected with simple stitching layers (1×1 convolutions). We share the same stitching layer among neighboring blocks (*e.g.*, 2 blocks with a stride of 2 in this example) between two models. Apart from the basic anchor models, we obtain many sub-networks (stitches) by stitching the nearest pairs of anchors in complexity, *e.g.*, DeiT-Ti and DeiT-S (the blue line), DeiT-S and DeiT-B (the green line). Best viewed in color.

Stitchable Neural Networks

- Preliminaries of Model Stitching:
- What to stitch: the choice of anchors: ViTs and CNNs
- How to stitch: initialization
  - Kaiming initialization + SGD
  - LS Init + SGD
- Where to stitch: the stitching directions
- Way to stitch: stitching as sliding windows
- Stitching space
- **Training strategy**

Algorithm 1 Training Stitchable Neural Networks

Require: M pretrained anchors to be stitched. Configuration set $E = \{e_1, ..., e_Q\}$ with Q stitching positions.

Initialize all stitching layers by least-squares matching
for $i = 1, ..., n_{iters}$ **do**
  Get next mini-batch of data $\mathbf{X}$ and label $\mathbf{Y}$.
  Clear gradients, $optimizer.zero\_grad()$.
  Randomly sample a stitching e_q from set E.
  Execute the current stitch, $\hat{\mathbf{Y}} = F_{e_q}(\mathbf{X})$.
  Compute loss, $loss = criterion(\hat{\mathbf{Y}}, \mathbf{Y})$.
  Compute gradients, $loss.backward()$.
  Update weights, $optimizer.step()$.
end for

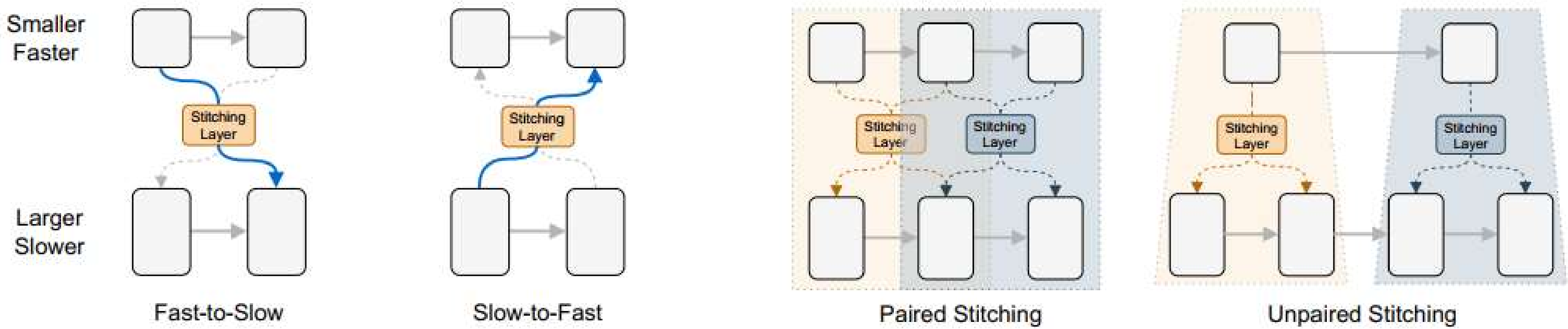
Stitchable Neural Networks

- Preliminaries of Model Stitching:
- What to stitch: the choice of anchors: ViTs and CNNs
- How to stitch: the stitching layer and its initialization
  - Kaiming initialization + SGD
  - LS Init + SGD

Definition 6.1. The coefficient of determination *of the best fit is given in terms of the optimal least squares matching $M_{LS} = A^{\dagger}B$ as:*

$$R^2_{LR}(A, B) = 1 - \frac{\|AM_{LS} - B\|_F^2}{\|B\|_F^2}.$$

Stitchable Neural Networks



- Where to stitch: the stitching directions： Fast-to-Slow and Slow-to-Fast
- Way to stitch: stitching as sliding windows
- Stitching space
- Training strategy

Stitchable Neural Networks

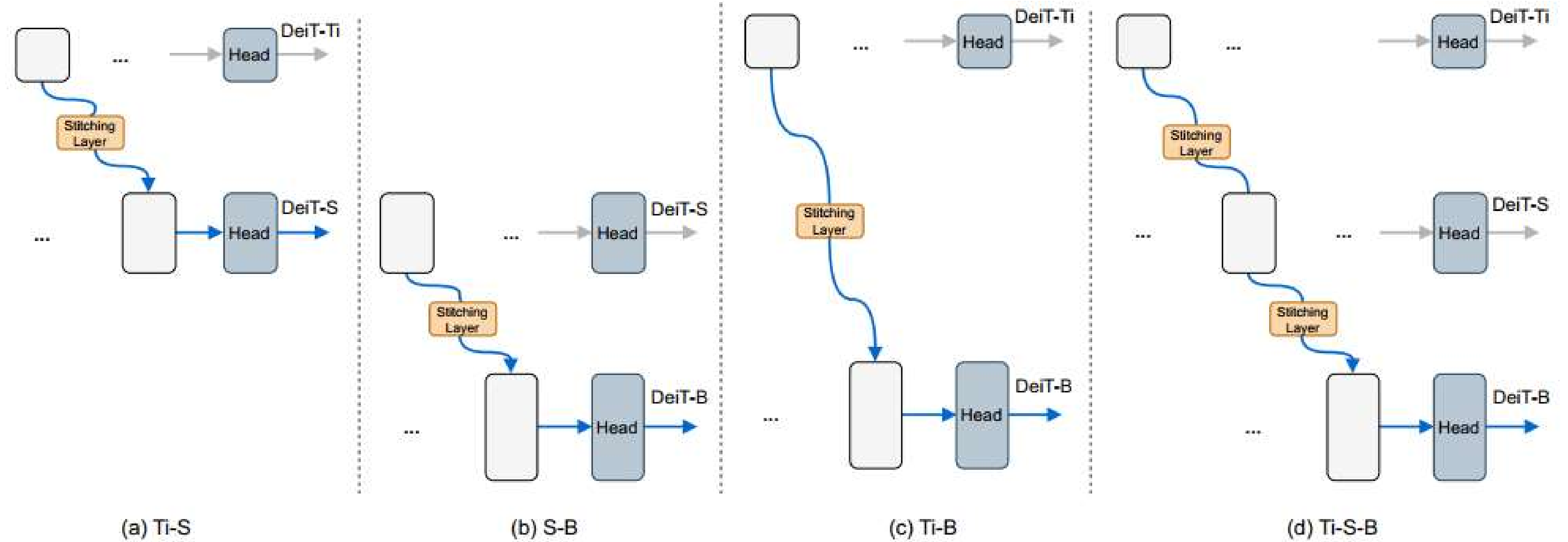
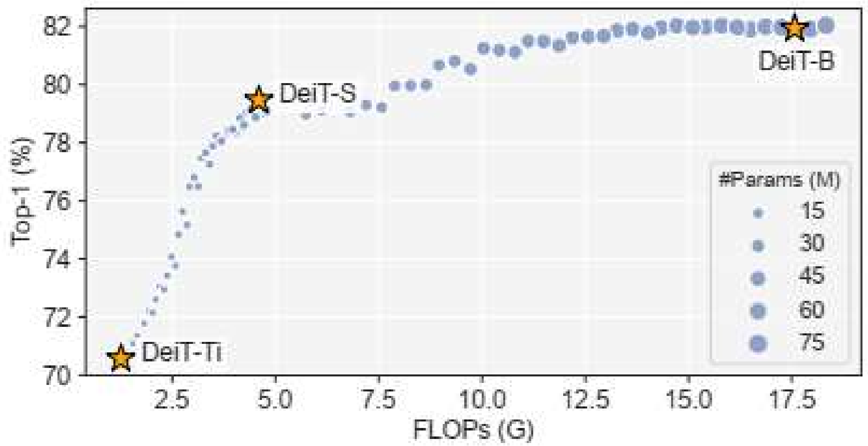


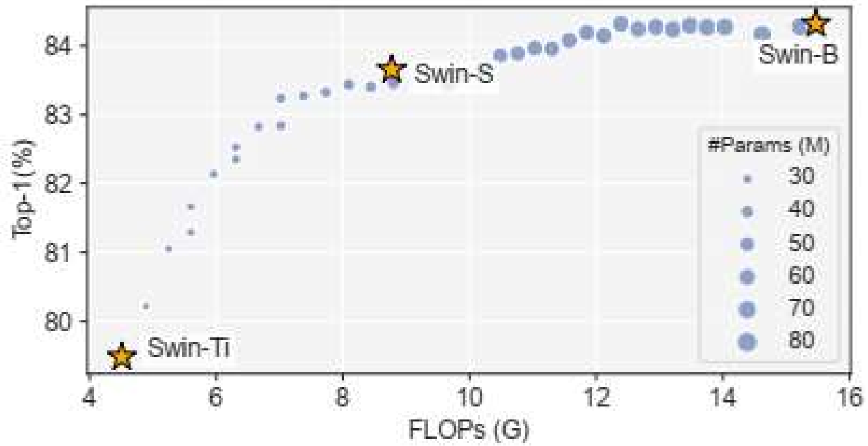
Figure 10. Four types of stitches based on DeiT-Ti/S/B. Under the proposed nearest stitching strategy, we limit the stitching between two anchors of the nearest model complexity/performance, *i.e.*, Figure (a) and (b), while excluding stitching anchors with a larger complexity/performance gap (Figure (c)) or sequentially stitching more than two anchors (Figure (d)).

Experiment

Stitching plain ViT

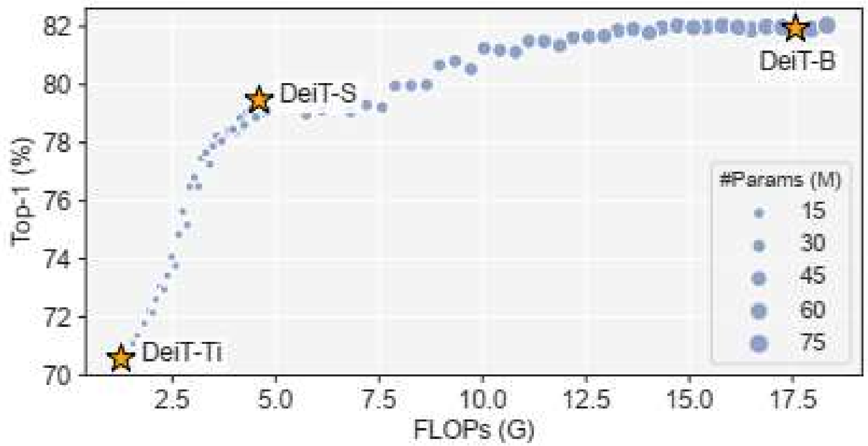
# Ti Blocks	# S Blocks	# B Blocks	FLOPs (G)	Throughput (images/s)	Individually Trained		SN-Net	
					Params (M)	Top-1 (%)	Params (M)	Top-1 (%)
12	0	0	1.3	2,839	5.7	72.1		70.6
9	3	0	2.1	2,352	10.0	75.9		72.6
6	6	0	2.9	1,963	14.0	78.2		76.5
3	9	0	3.8	1,673	18.0	79.4		78.2
0	12	0	4.6	1,458	22.1	79.8	118.4	79.5
0	9	3	7.9	1,060	38.7	79.4		80.0
0	6	6	11.2	828	54.6	failed		81.5
0	3	9	14.3	679	70.6	80.3		82.0
0	0	12	17.6	577	86.6	81.8		81.9

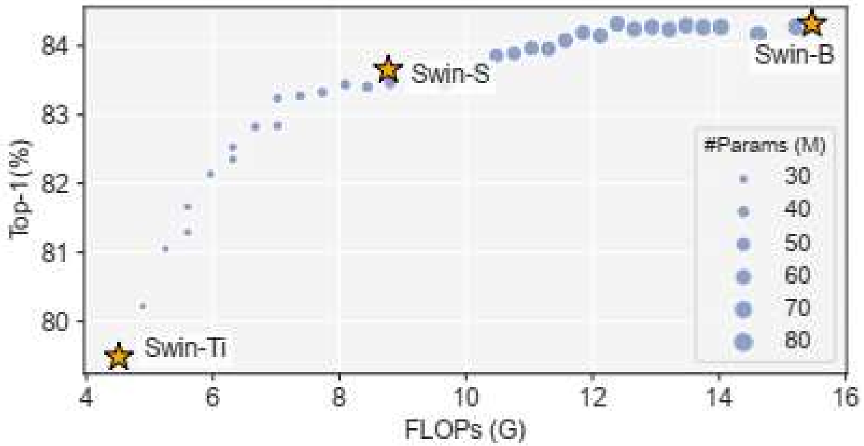




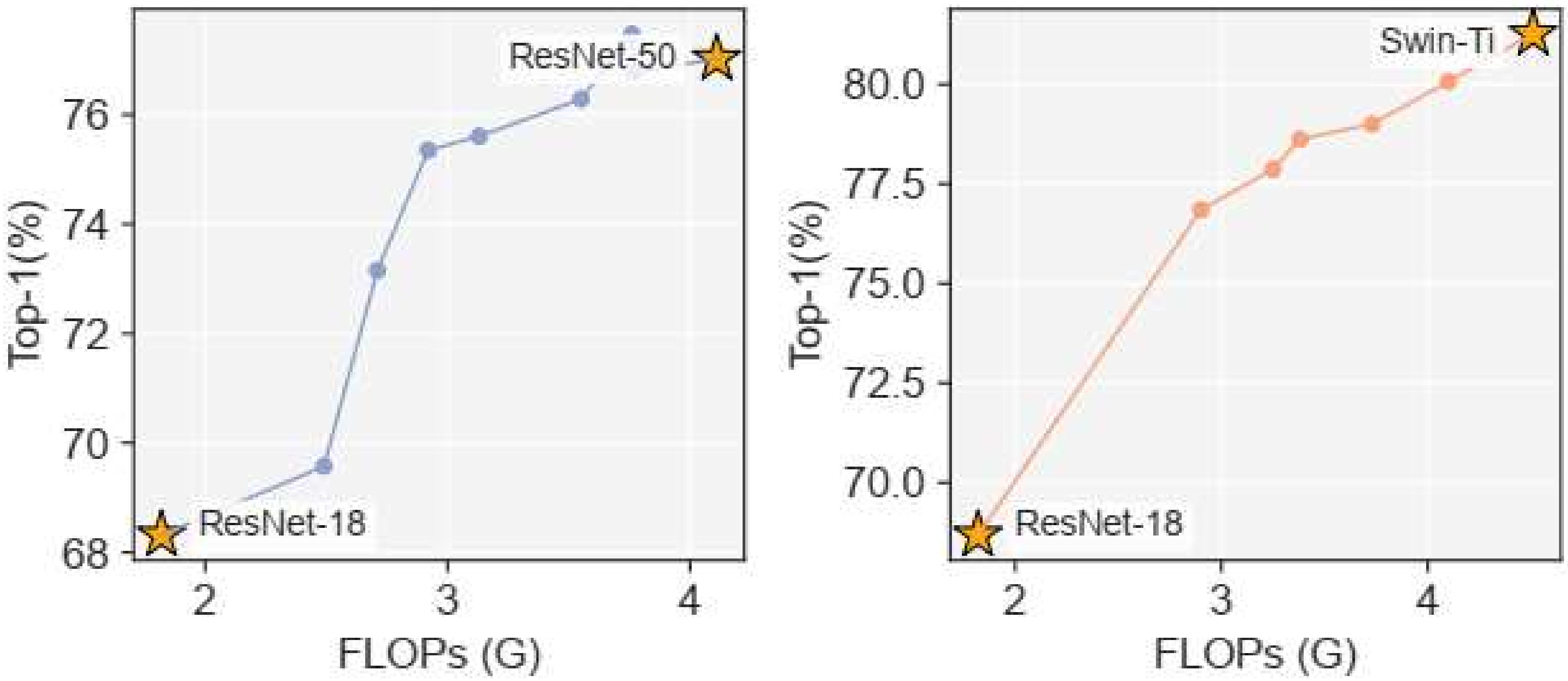
Stitching plain ViT & hierarchical ViTs

# Ti Blocks	# S Blocks	# B Blocks	FLOPs (G)	Throughput (images/s)	Individually Trained Params (M)	Individually Trained Top-1 (%)	SN-Net Params (M)	SN-Net Top-1 (%)
12	0	0	1.3	2,839	5.7	72.1	118.4	70.6
9	3	0	2.1	2,352	10.0	75.9		72.6
6	6	0	2.9	1,963	14.0	78.2		76.5
3	9	0	3.8	1,673	18.0	79.4		78.2
0	12	0	4.6	1,458	22.1	79.8		79.5
0	9	3	7.9	1,060	38.7	79.4		80.0
0	6	6	11.2	828	54.6	failed		81.5
0	3	9	14.3	679	70.6	80.3		82.0
0	0	12	17.6	577	86.6	81.8		81.9





Stitching CNNs and CNN-ViT



Ablation Study

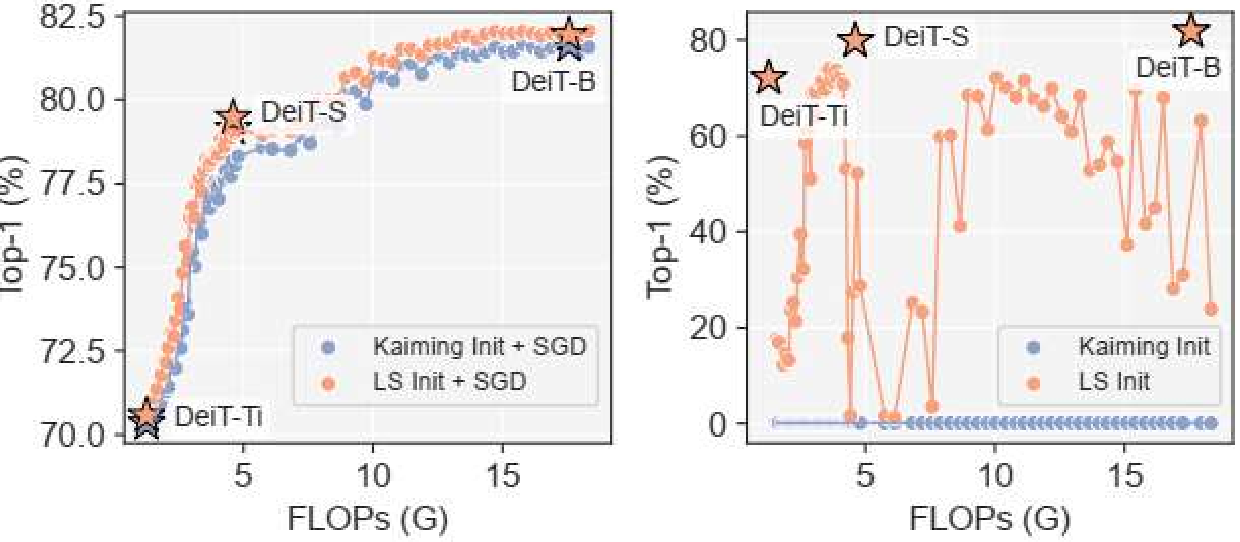


Figure 8. Different learning strategies for stitching layers.

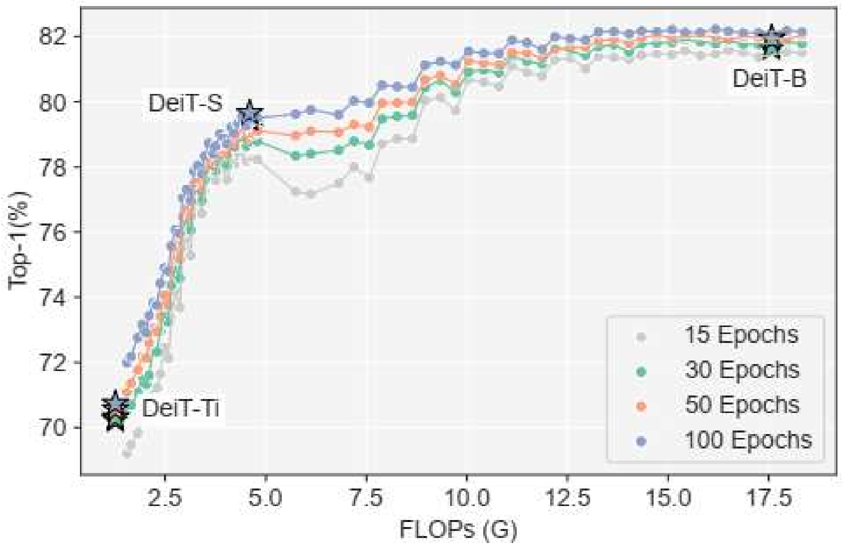


Figure 12. Effect of different training epochs.

Ablation Study

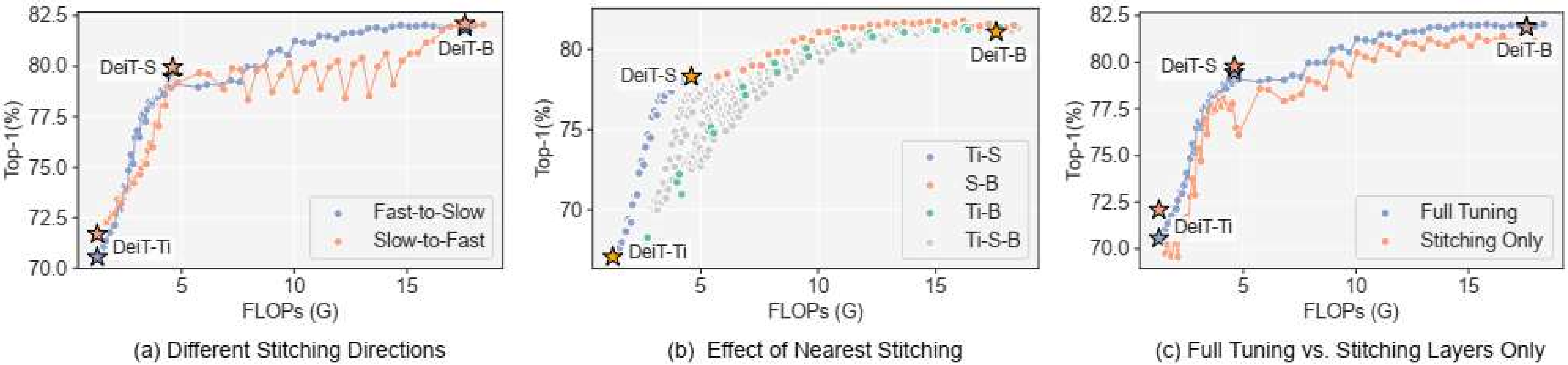


Figure 9. From left to right, Figure (a) shows the effect of different stitching directions. Figure (b) presents the effect of nearest stitching based on DeiT, where "Ti", "S", "B" denote the stitched anchors. For example, "Ti-S-B" refers to a stitch that defined by connecting the tiny, small and base variants of DeiT, sequentially. Figure (c) shows the comparison of full model tuning vs. tuning stitching layers only.

Ablation Study

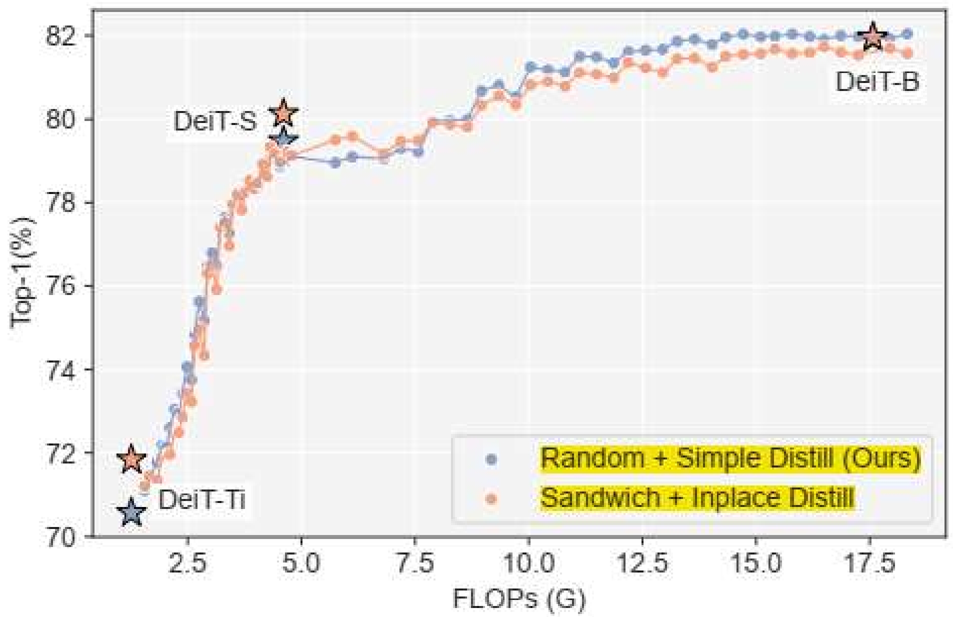


Figure 13. Comparison between our training strategy and common supernet training strategy in NAS (*i.e.*, sandwich sampling rule and inplace distillation [61]).

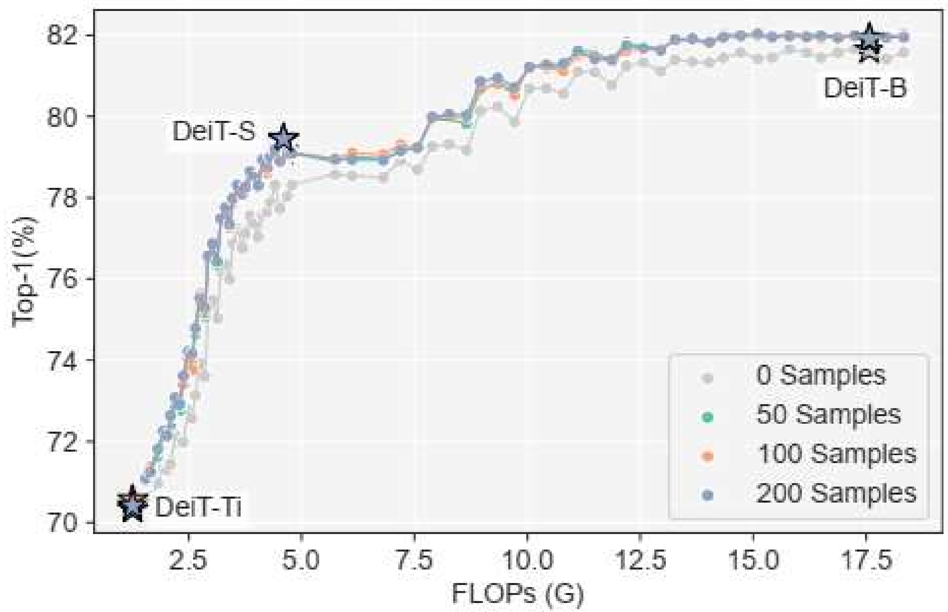


Figure 14. Effect of different number of samples for initializing stitching layers. With 0 samples, the initialization is equivalent to the default Kaiming initialization in PyTorch.

Conclusion

- New universal framework for directly utilising the pretrained model families in model zoo
- Practical principles to design and train SNNet, laying down the foundations

- Much larger stitching space?
- Stitches not be sufficiently trained?

Thanks for Listening