Less is More: Selective Layer Finetuning with SubTuning

Gal Kaplun*
Harvard University & Mobileye

Andrey Gurevich
Mobileye

Tal Swisa
Mobileye

Mazor David
Mobileye

Shai Shalev-Shwartz
Hebrew University & Mobileye

Eran Malach
Hebrew University & Mobileye

Agenda

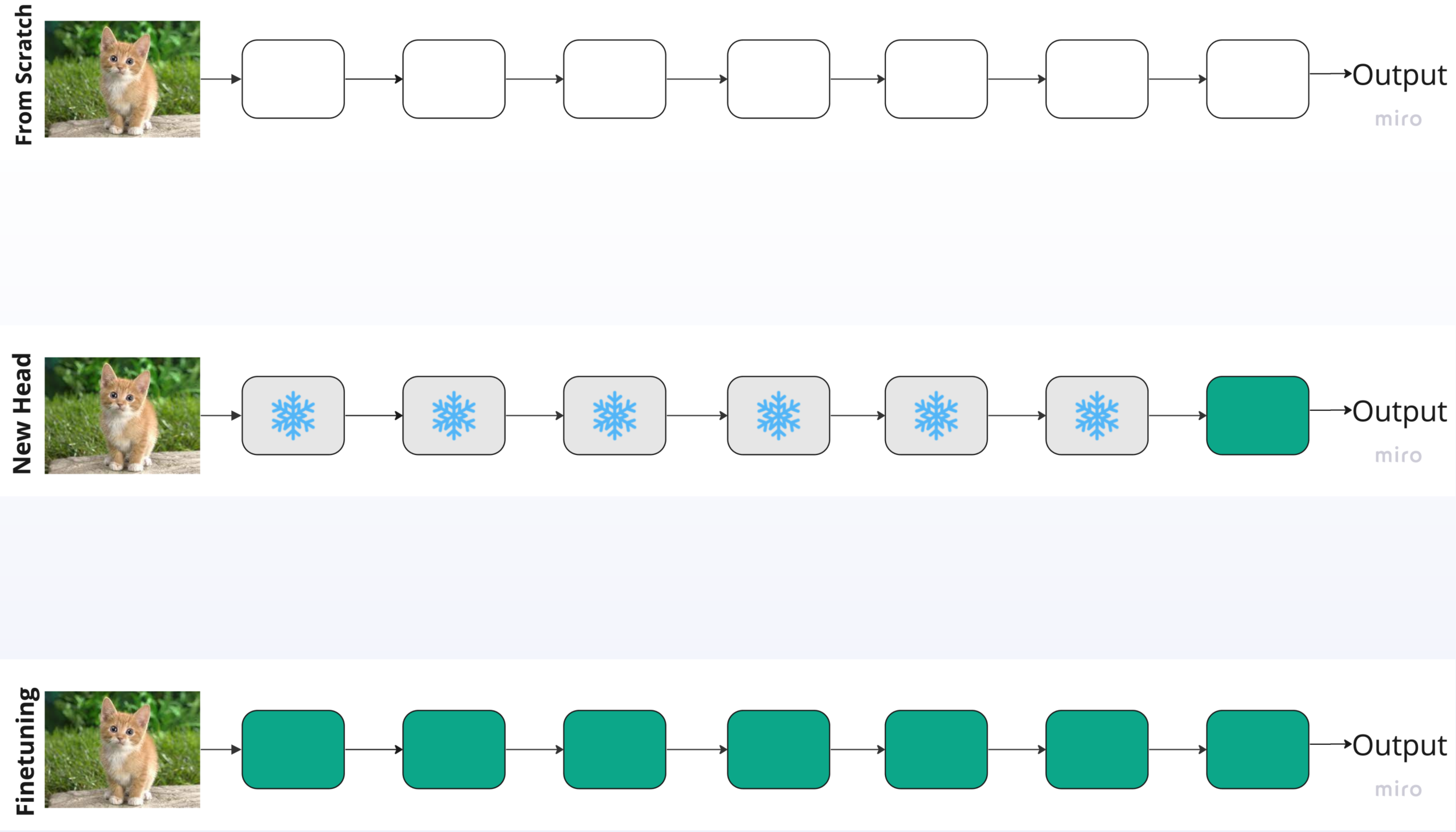
- Motivation
- Finetuning Profile
- SubTuning
  - The algorithm
  - Sparce & Corrupted data
  - Multitask

1

Motivation

Training DNNs – Basic Methods

- From scratch
  - Requires a lot of data & compute
  - Low performance
- Pretrained model
  - Allowing rapid convergence
  - Enhanced performance
- New head on pretrained model
  - Very fast and efficient
  - Low capacity
- Finetuning
  - Better performance
  - Costly in data & compute



Other Methods

- Head2Toe
  - Intemediate features may have useful information
  - Feature selection is computationally complex
- LORA – Low-Rank Adaptaion of LLMs
  - Reduces trainable params by x10,000
  - No additional cost at inference

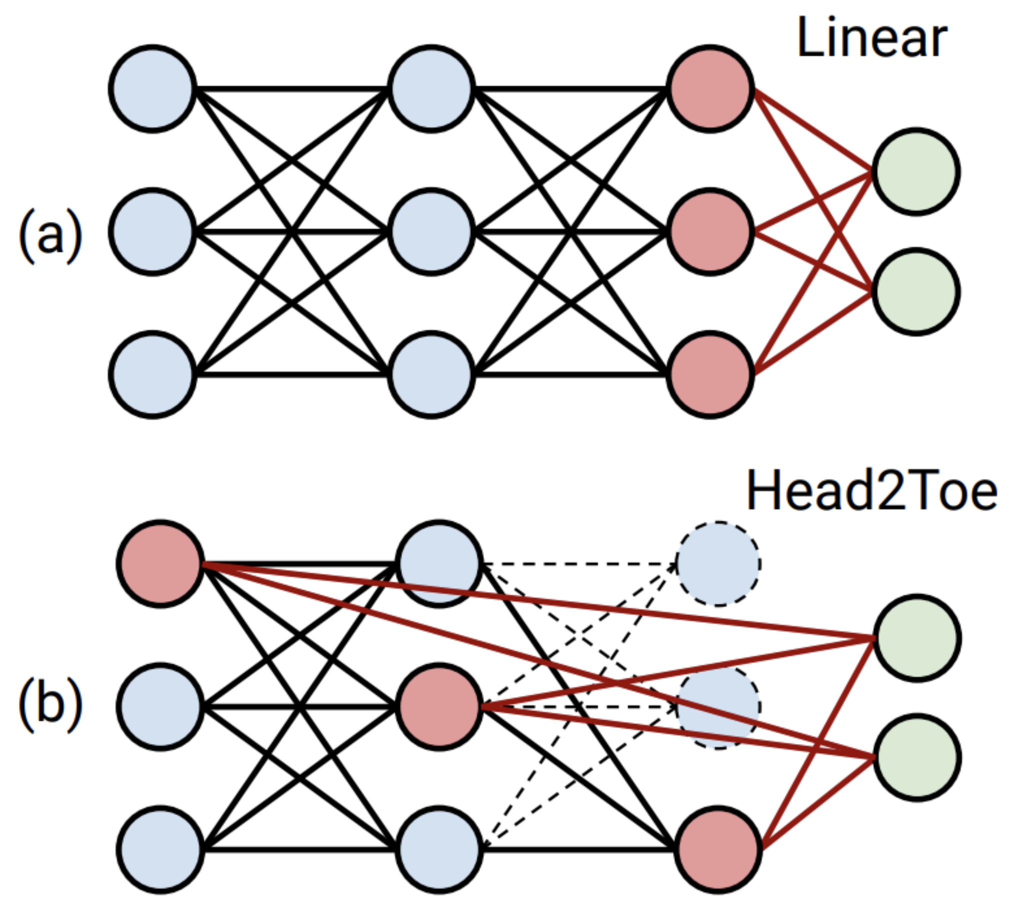


Figure 1. **(a)** Whereas LINEAR utilizes only the last layer for transfer learning, **(b)** HEAD2TOE selects the most useful features from the entire network and trains a linear head on top.

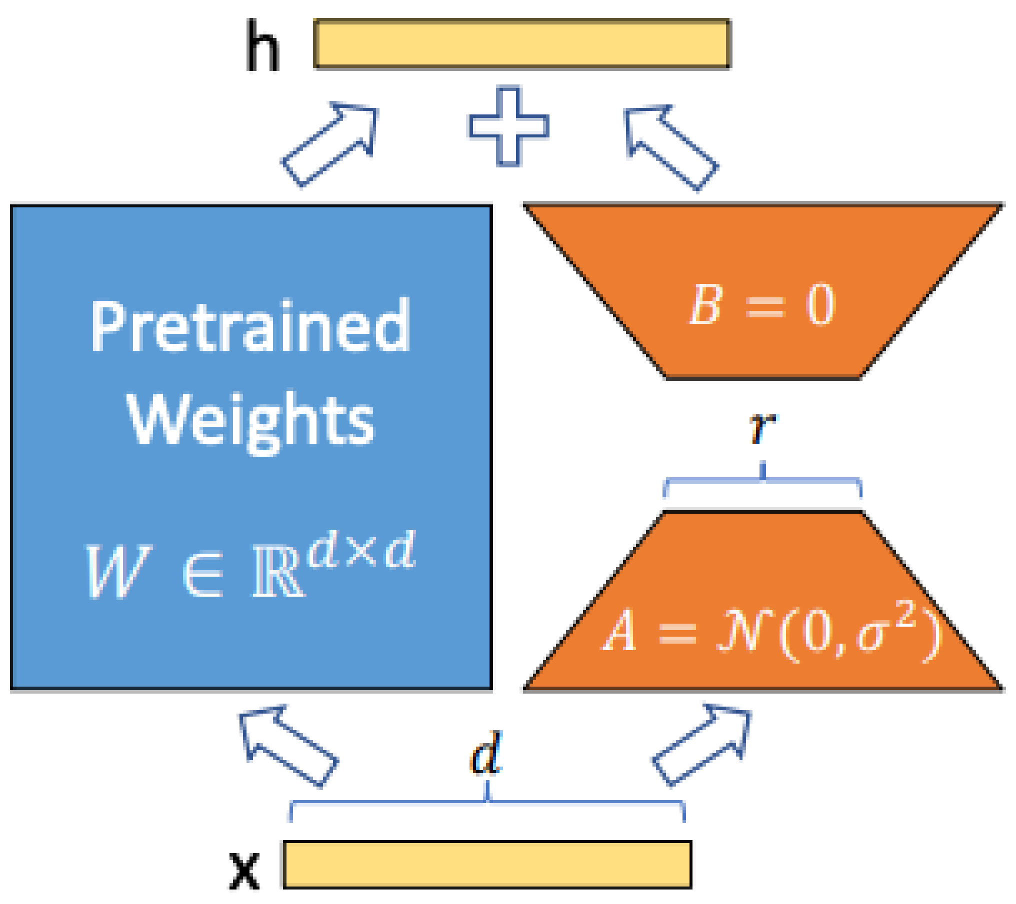


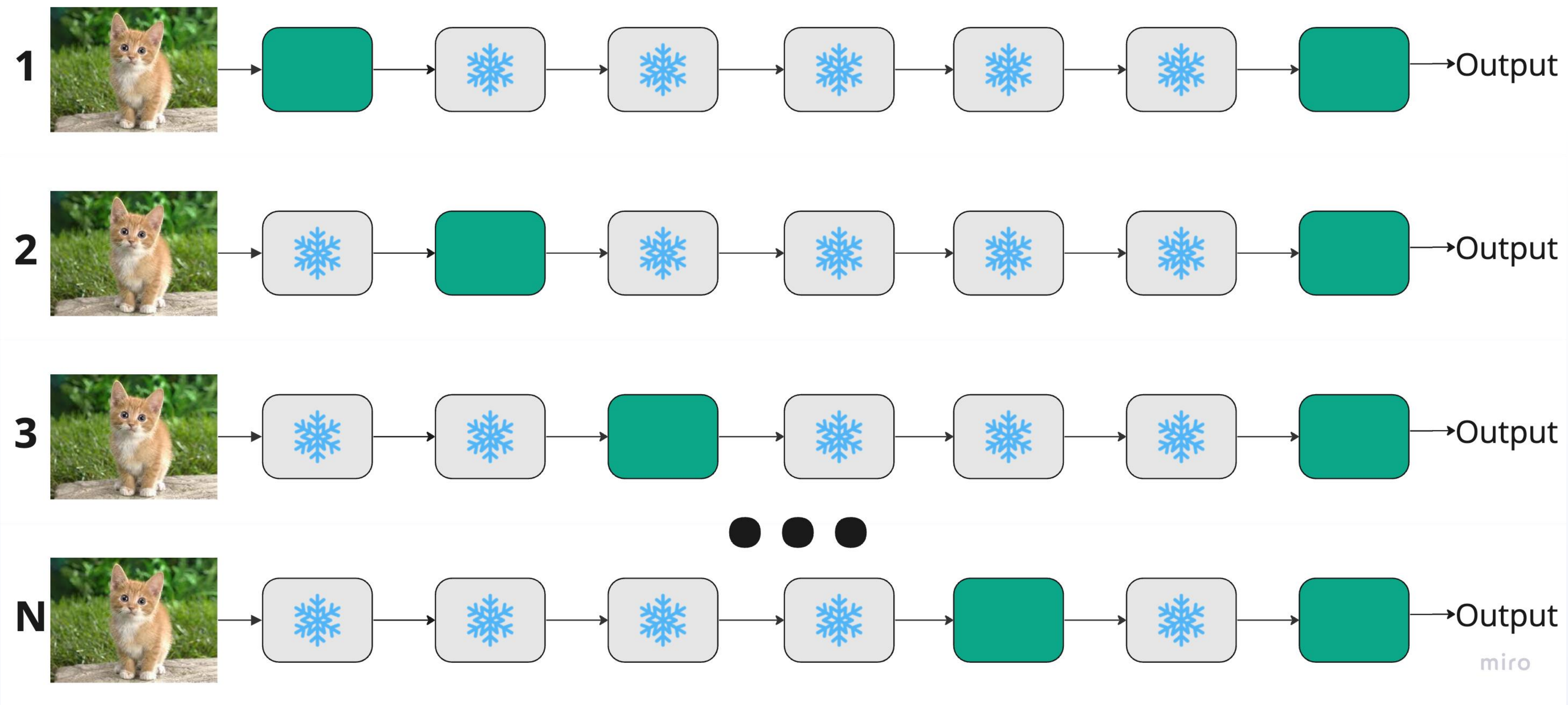
Figure 1: Our reparametrization. We only train A and B.

2

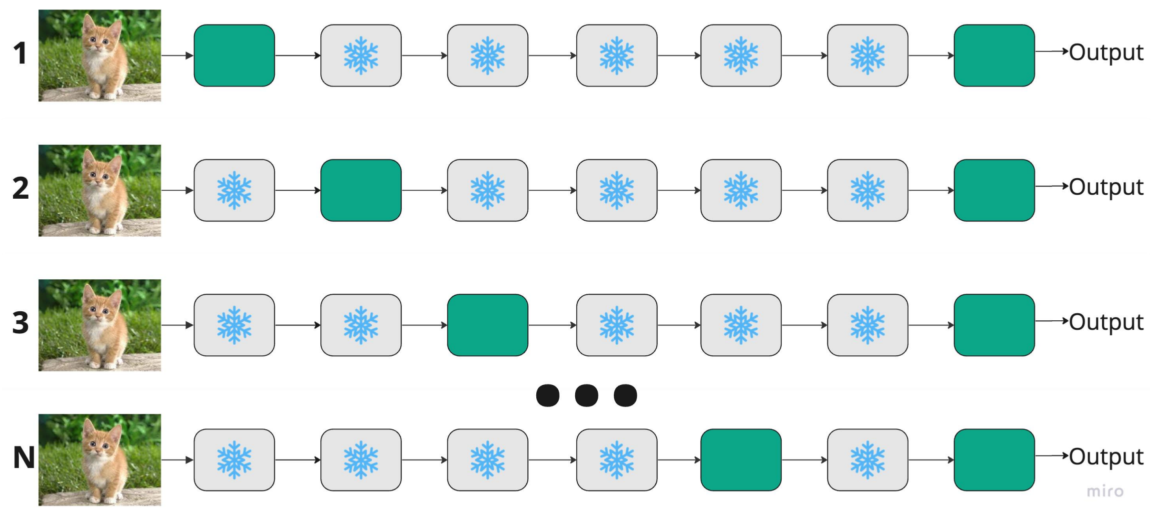
Finetuning Profile

Finetuning Profile

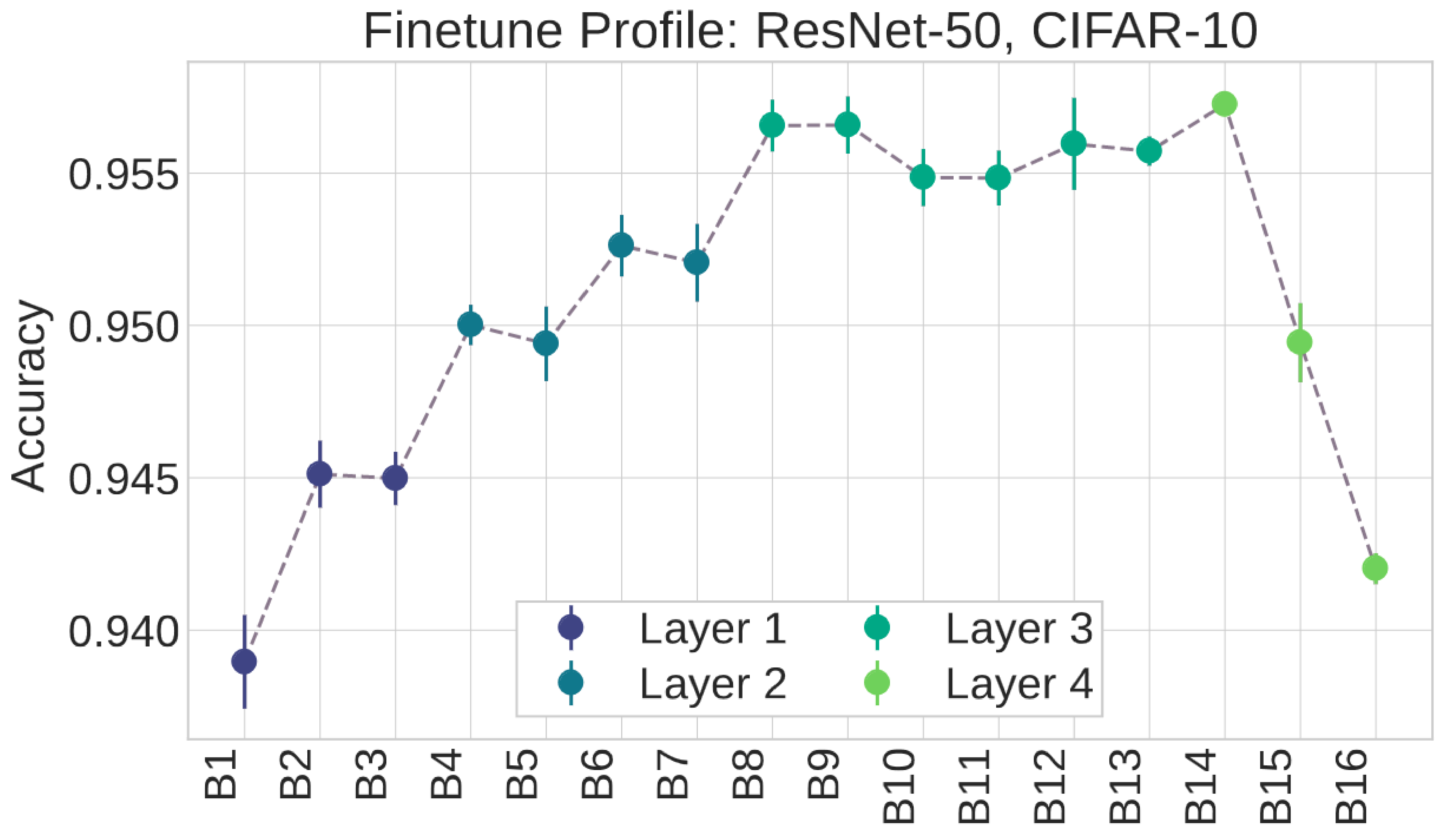
- What if we finetune only a subset of layers?
- Will we achieve the benefits of all worlds?



Finetuning Profile



- ResNet50 has
  - 16 ResBlocks
  - 4 resolutions
- Not all layers are created equal
- Different layers -> different contribution to performance



Finetuning Profile

Optimal choice of layers depends on

- Target task
- Architecture
- Pretraining

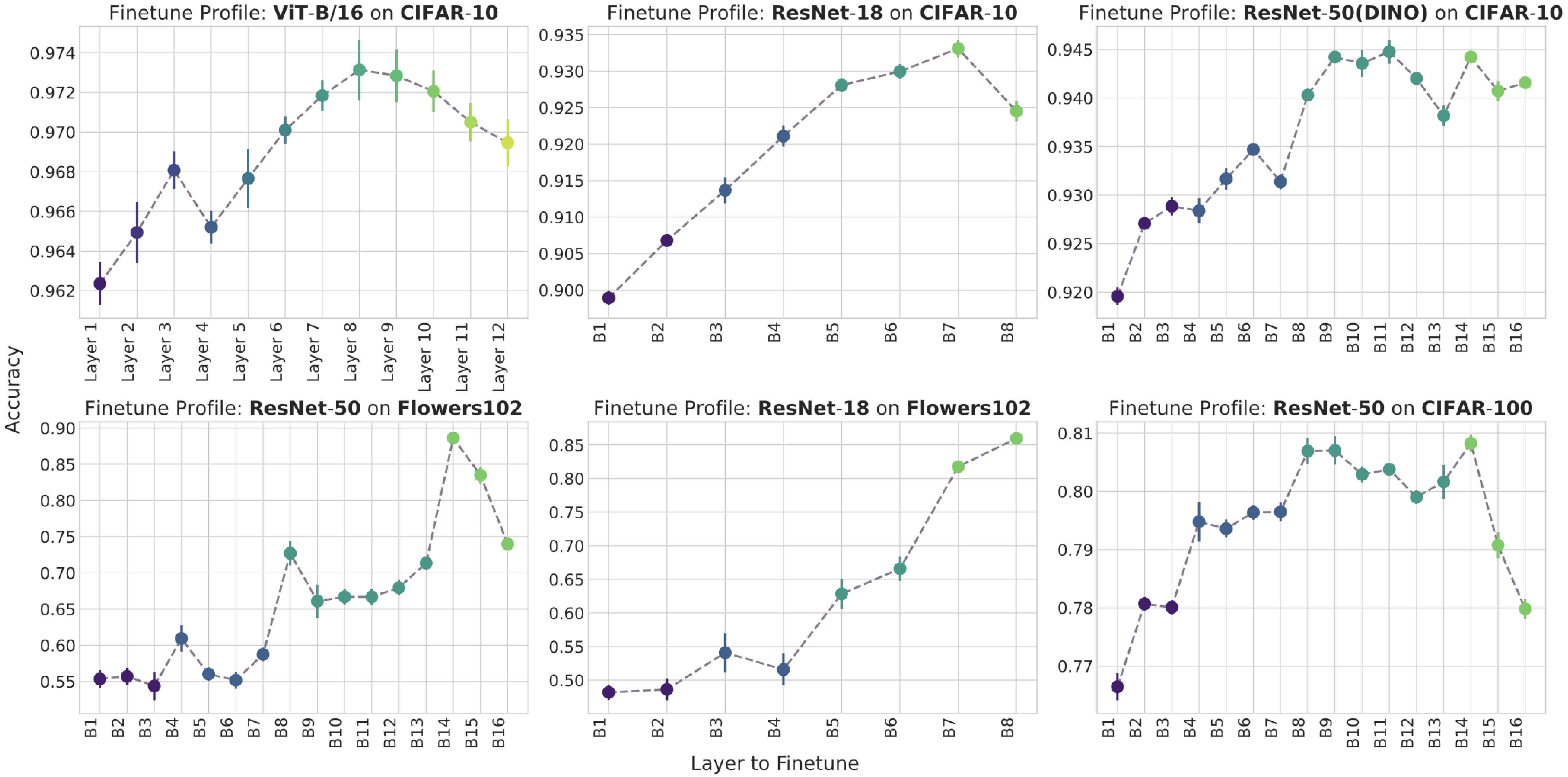


Figure 2: Finetuning profiles for different architectures, initializations and datasets.

3

SubTuning

SubTuning Algorithm

- We want to finetune a subset of layers
  - SubTuning
  - Fined best subset via Finetuning Profile
- This may be expensive -> Greedy Algorithm
  - Iteratively find the best layer to finetune
  - Stop when improvement < ε.

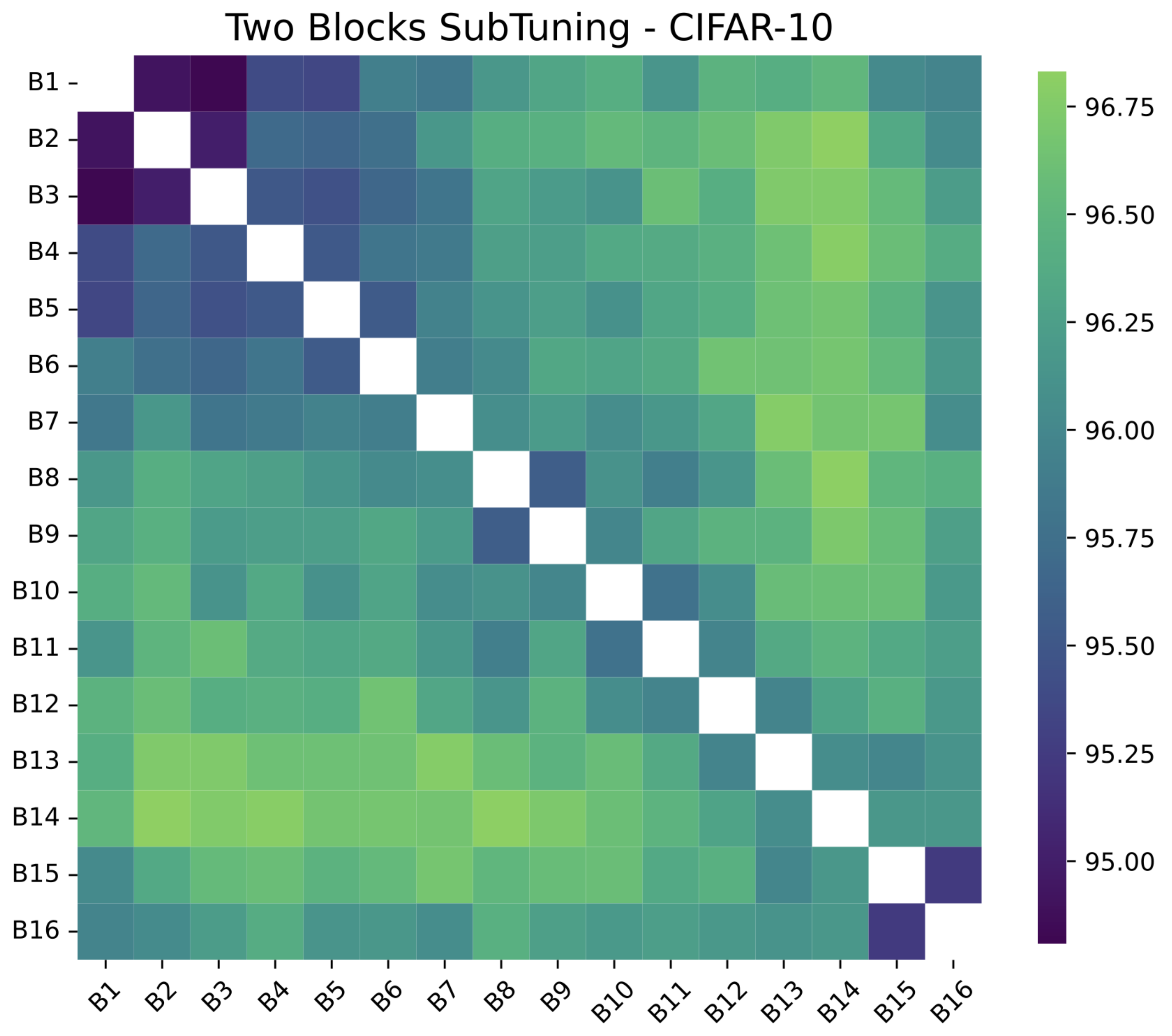
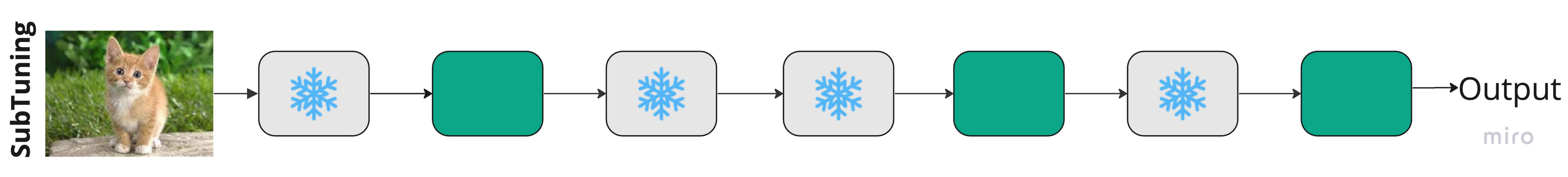


Figure 3: 2-block finetuning profile for ResNet-50 over CIFAR-10.



Results - Scarce Data

When only limited data is available

- Finetuning results in overfitting
- But SubTuning has great results

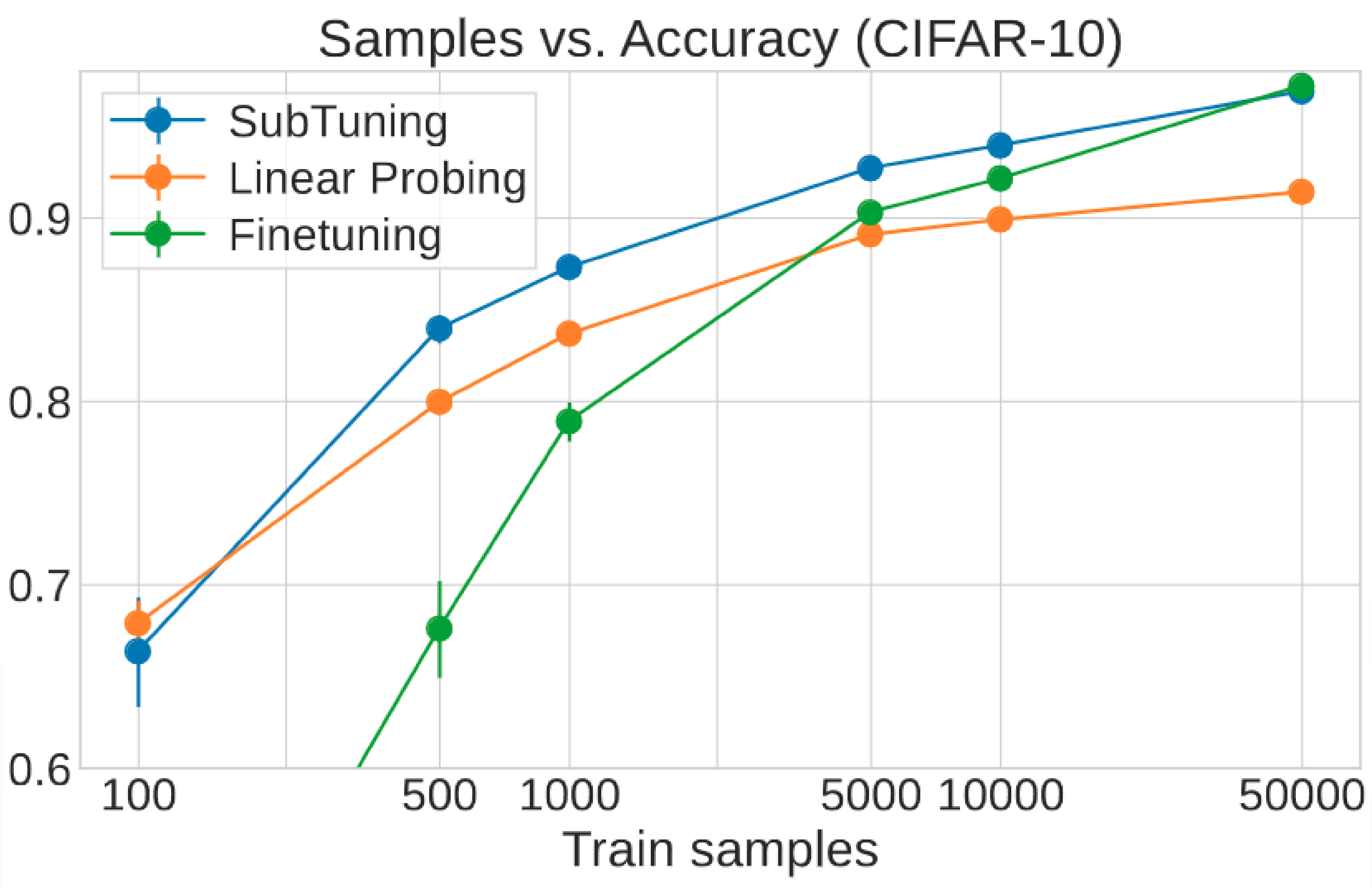
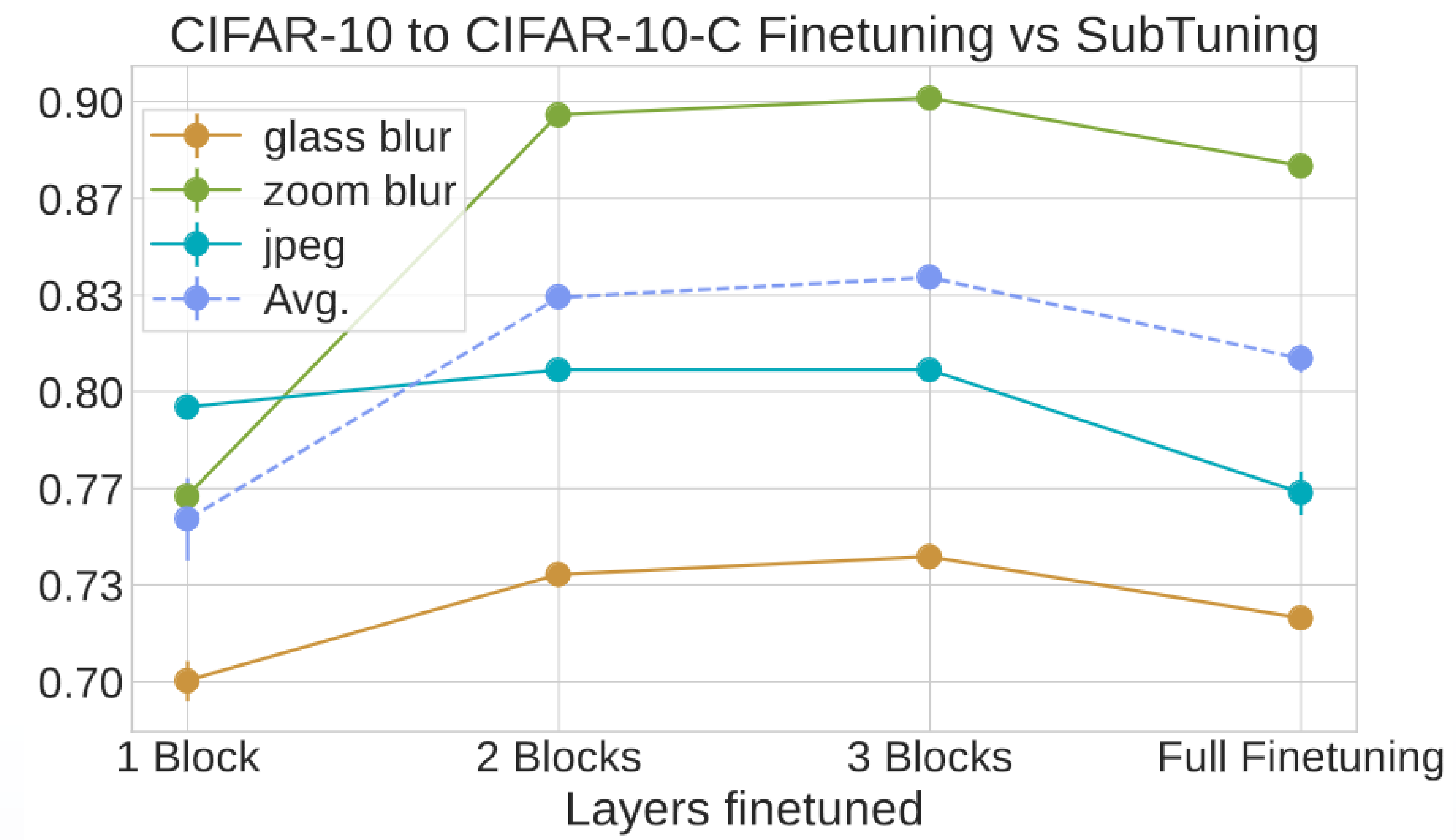


Table 1: Performance of ResNet-50 and ViT-b/16 pretrained on ImageNet and finetuned on datasets from VTAB-1k. FT denotes finetuning while LP stands for linear probing. Standard deviations reported in Table 5 in the appendix.

	ResNet50				ViT-b/16			
	CIFAR-100	Flowers102	Caltech101	DMLAB	CIFAR-100	Flowers102	Caltech101	DMLab
Ours	**54.6**	**90.5**	86.5	**51.2**	**68.0**	**97.7**	86.5	36.4
H2T [2] [13]	47.1	85.6	**88.8**	43.9	58.2	85.9	**87.3**	**41.6**
FT	33.7	87.3	78.7	48.2	47.8	91.2	80.7	34.3
LP	35.4	64.2	67.1	36.3	29.9	84.7	72.7	31.0
LoRA [22]	-	-	-	-	40.4	88.3	79.2	36.4

Results - Distribution Shift

- CIFAR-10 to CIFAR-10-C distribution shift
- Corrupted data



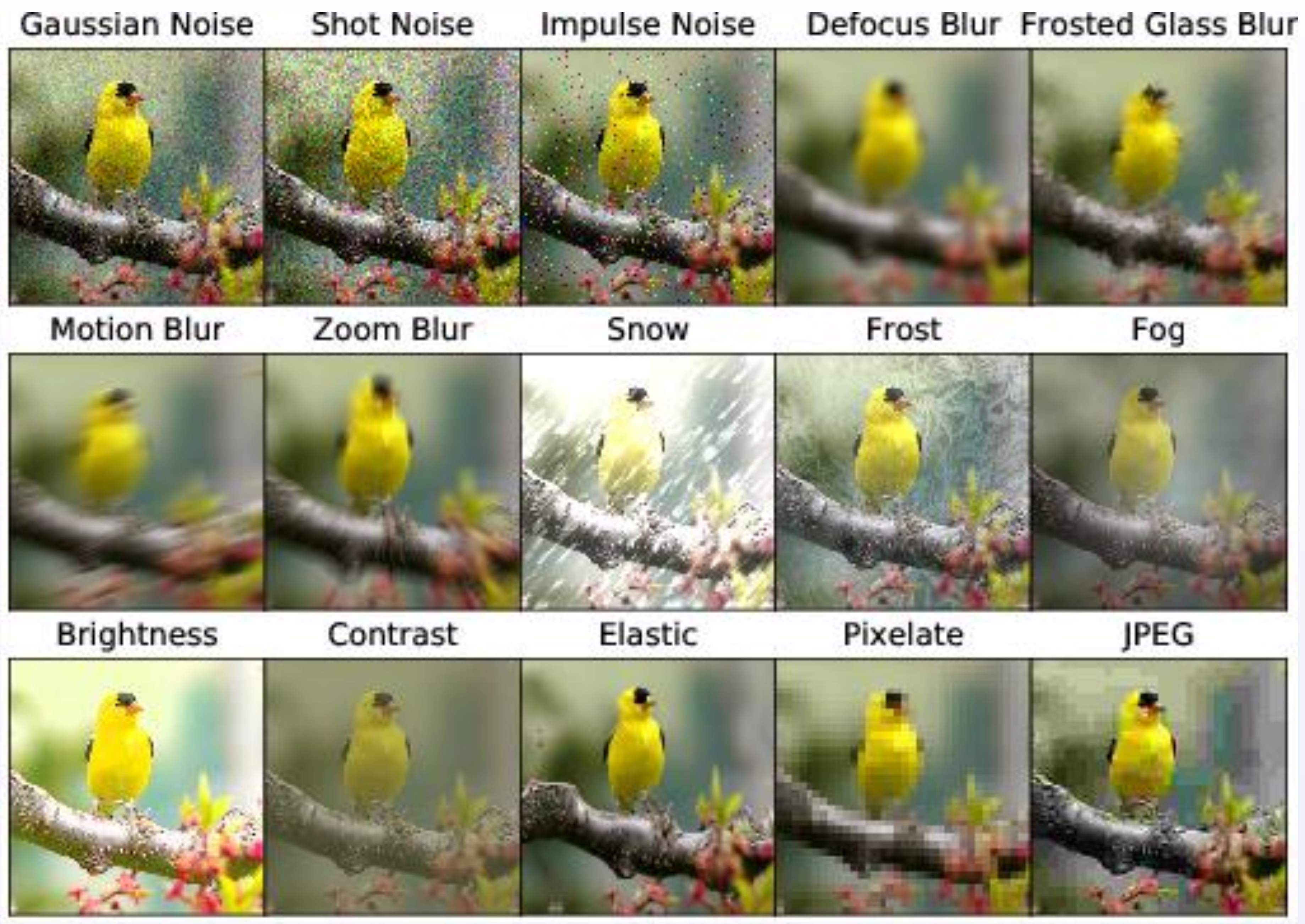


Table 2: CIFAR-10 to CIFAR-10-C distribution shift.

Distribution shift	**SubTuning**	**Finetuning**	**Surgical L1**	**Surgical L2**	**Surgical L3**	**Linear**
zoom blur	$\mathbf{90.0 \pm 0.1}$	87.8 ± 0.4	89.2 ± 0.1	89.1 ± 0.2	85.5 ± 0.3	68.7 ± 0.04
speckle noise	$\mathbf{81.5 \pm 0.2}$	77.8 ± 0.6	78.4 ± 0.1	74.8 ± 0.1	71.1 ± 0.1	51.5 ± 0.01
spatter	89.2 ± 0.2	86.8 ± 0.3	$\mathbf{89.4 \pm 0.1}$	87.4 ± 0.2	85.3 ± 0.0	80.4 ± 0.07
snow	$\mathbf{86.0 \pm 0.2}$	84.1 ± 0.2	84.8 ± 0.2	84.3 ± 0.1	82.2 ± 0.2	78.7 ± 0.07
shot noise	$\mathbf{82.0 \pm 0.3}$	77.6 ± 0.4	77.0 ± 0.9	74.2 ± 0.1	69.9 ± 0.1	46.4 ± 0.01
saturate	$\mathbf{92.0 \pm 0.1}$	89.5 ± 0.3	91.7 ± 0.0	91.2 ± 0.0	90.4 ± 0.0	89.8 ± 0.04
pixelate	$\mathbf{86.1 \pm 0.0}$	82.8 ± 0.5	85.8 ± 0.1	83.6 ± 0.2	78.5 ± 0.2	54.8 ± 0.02
motion blur	$\mathbf{87.3 \pm 0.1}$	85.5 ± 0.3	86.7 ± 0.1	86.9 ± 0.1	83.4 ± 0.1	72.9 ± 0.03
jpeg compression	$\mathbf{80.8 \pm 0.2}$	76.5 ± 0.7	80.1 ± 0.5	76.8 ± 0.1	74.9 ± 0.1	72.0 ± 0.04
impulse noise	$\mathbf{75.4 \pm 0.5}$	70.8 ± 0.7	69.6 ± 0.3	63.8 ± 0.1	56.7 ± 0.1	35.2 ± 0.01
glass blur	$\mathbf{74.3 \pm 0.3}$	72.2 ± 0.2	69.9 ± 0.4	71.5 ± 0.1	67.8 ± 0.1	55.2 ± 0.06
gaussian noise	$\mathbf{80.0 \pm 0.2}$	75.1 ± 1.2	72.7 ± 0.1	71.0 ± 0.1	66.6 ± 0.2	41.1 ± 0.01
gaussian blur	$\mathbf{89.5 \pm 0.2}$	86.4 ± 0.4	88.1 ± 0.0	87.3 ± 0.1	80.0 ± 0.0	41.7 ± 0.05
frost	$\mathbf{84.2 \pm 0.2}$	83.1 ± 0.4	$\mathbf{84.2 \pm 0.3}$	83.2 ± 0.1	80.4 ± 0.2	68.5 ± 0.03
Average	$\mathbf{84.2 \pm 0.2}$	81.1 ± 0.5	82.0 ± 0.2	80.4 ± 0.1	76.6 ± 0.1	61.2 ± 0.04

MultiTask

- Ideal for Multi-Task
  - High algorithmic performance
  - Allows to add new task to deployed models
    - No effect on existing task
    - Low inference cost - Concat intermediate outputs on batch

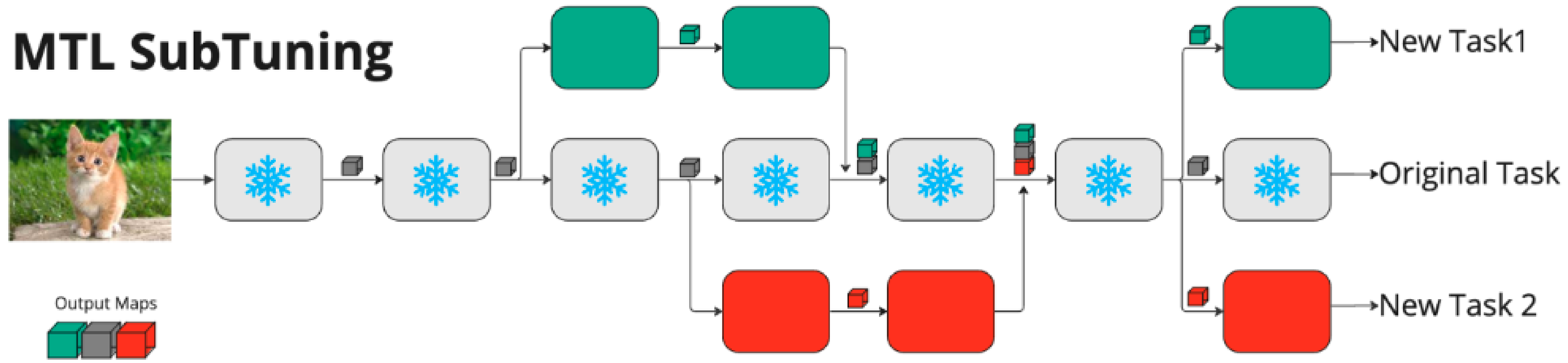
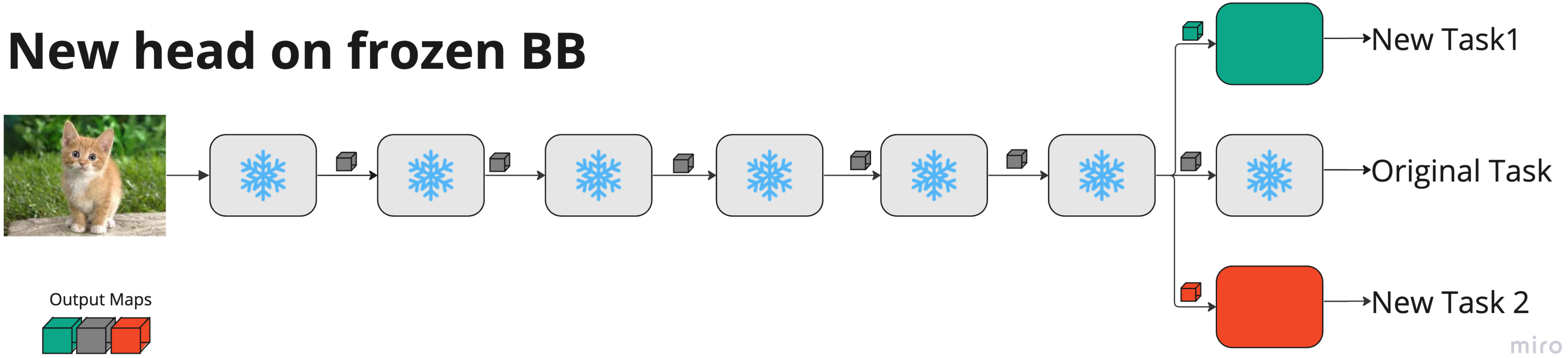


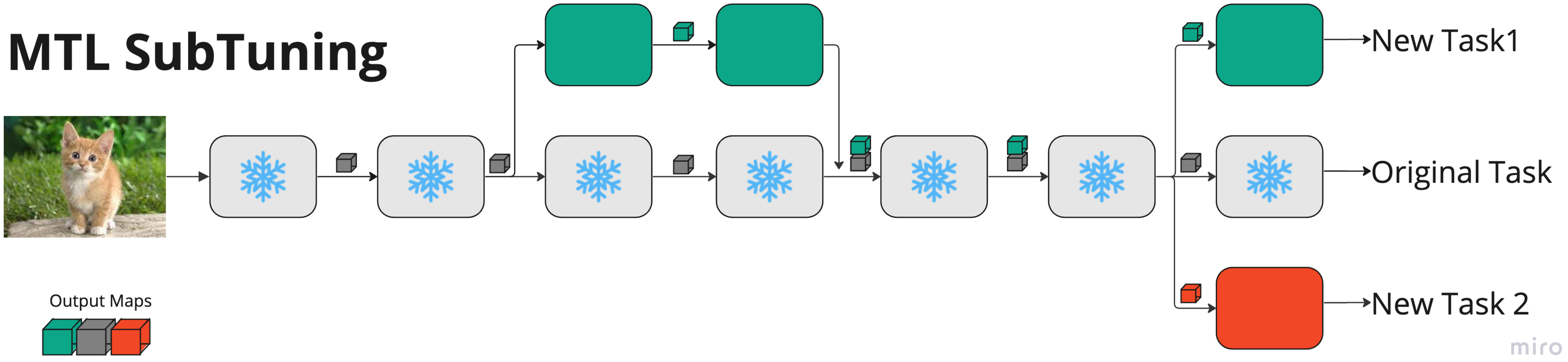
Figure 6: SubTuning for MTL. Each new task utilizes a consecutive subset of layers of a network and shares the others. At the end of the split, the outputs of different tasks are concatenated and parallelized along the batch axis for computational efficiency.

MultiTask – Efficient Runtime

New head on frozen BB



MTL SubTuning



Results - MultiTask

ImageNet to CIFAR-10 transfer learning

- Linear probing - 91.8%
  - only a small inference delay
- Finetuning - 97.1%
  - +100% inference cost

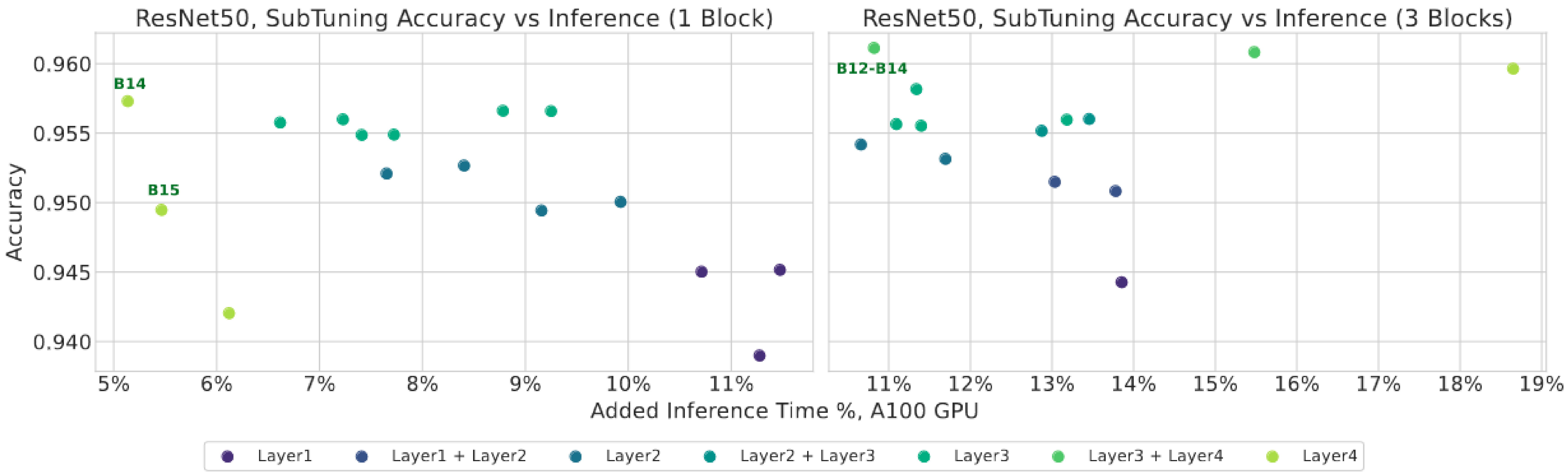
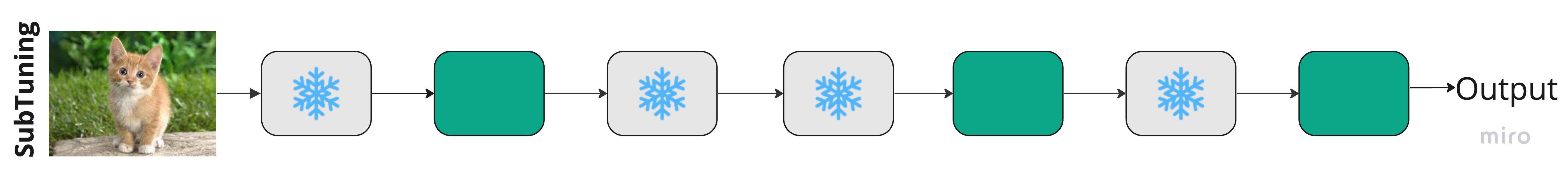


Figure 7: Accuracy on CIFAR-10 vs A100 latency with batch size of 1 and input resolution of 224.

Summary

- SubTuning is simple yet efficient
  - Selects a subset of layers to finetune
  - Greedy algorithm for fast perfomance
  - Achives SoTA performance
- Finetuning Profile
  - Not all layers are created equal
- Ideal for Multi-Task on a deployed model
  - Low inference cost
  - High performance
  - No effect on existing task



Thank you!

Contact me at
gurevichan@gmail.com