

EfficientML.ai Lecture 9

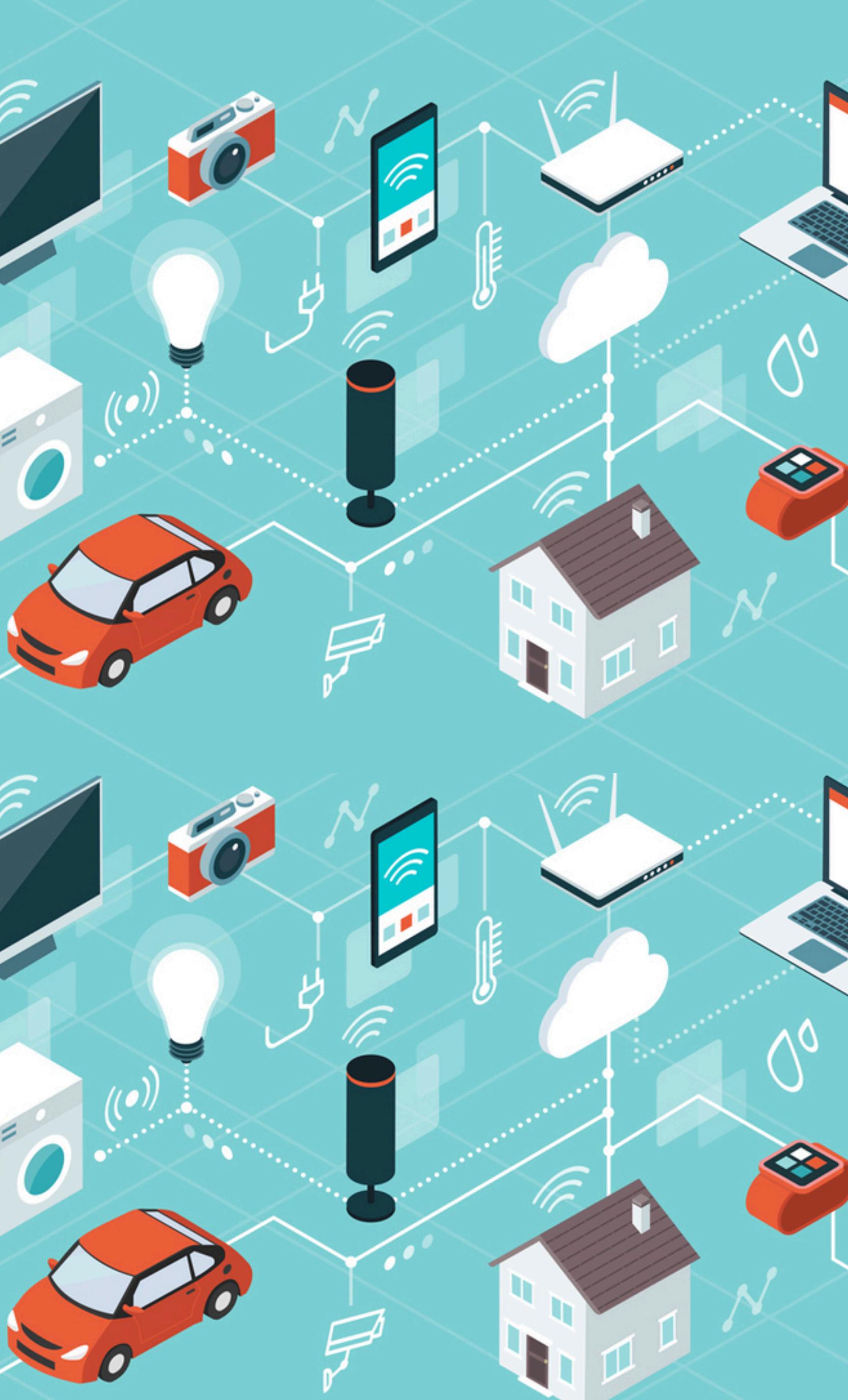
Knowledge Distillation



Song Han

Associate Professor, MIT
Distinguished Scientist, NVIDIA

 @SongHan/MIT



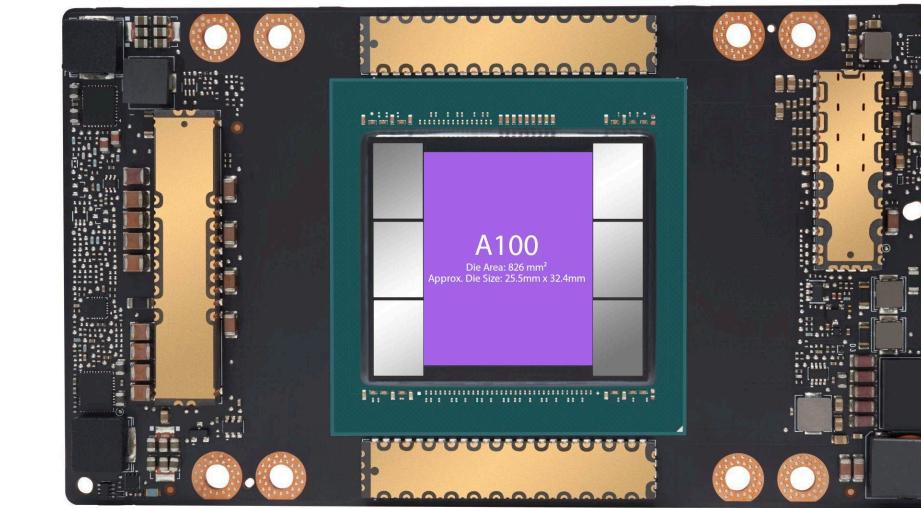
Knowledge Distillation

Today we will:

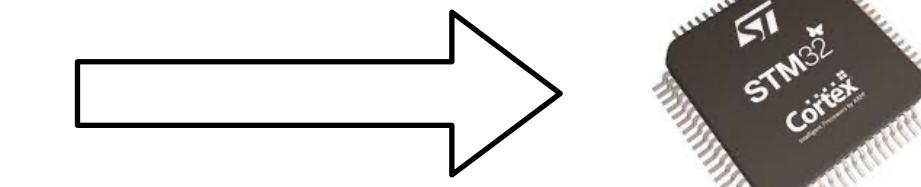
1. What is knowledge distillation;
2. What to match;
3. Self and online distillation;
4. Distillation for different tasks;
5. Network augmentation.



Challenge: limited hardware resources



Cloud AI



Tiny AI

Computation (fp32)

19.5 TFLOPS

MFLOPs

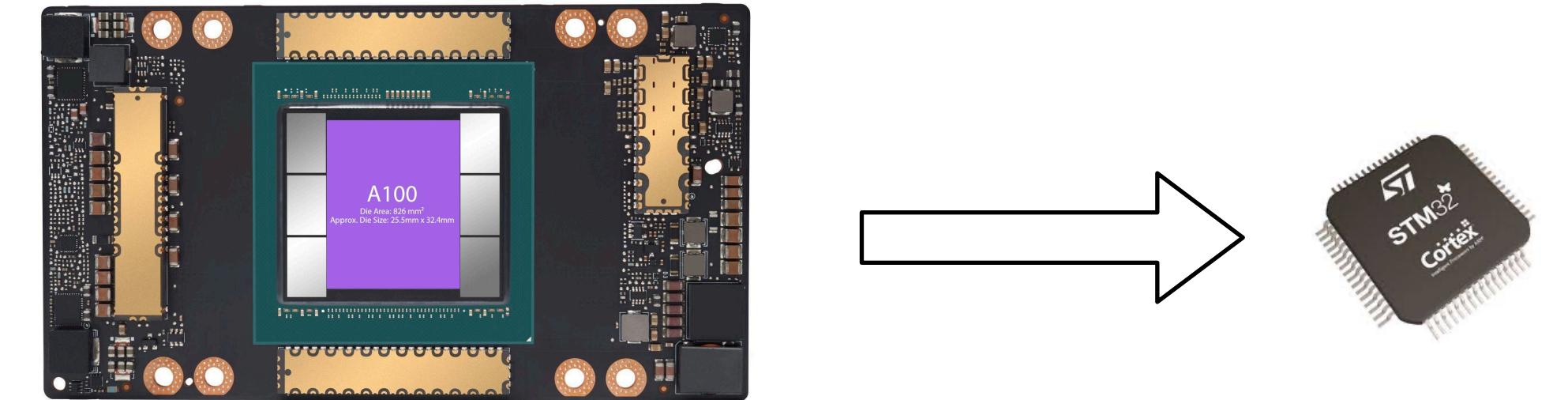
Memory

80GB

256kB

Network Augmentation for Tiny Deep Learning [Cai et al., ICLR 2022]

Challenge: limited hardware resources



Cloud AI

Tiny AI

Computation (fp32)

19.5 TFLOPS

MFLOPs

Memory

80GB

256kB

Neural Network

ResNet
ViT-Large
...

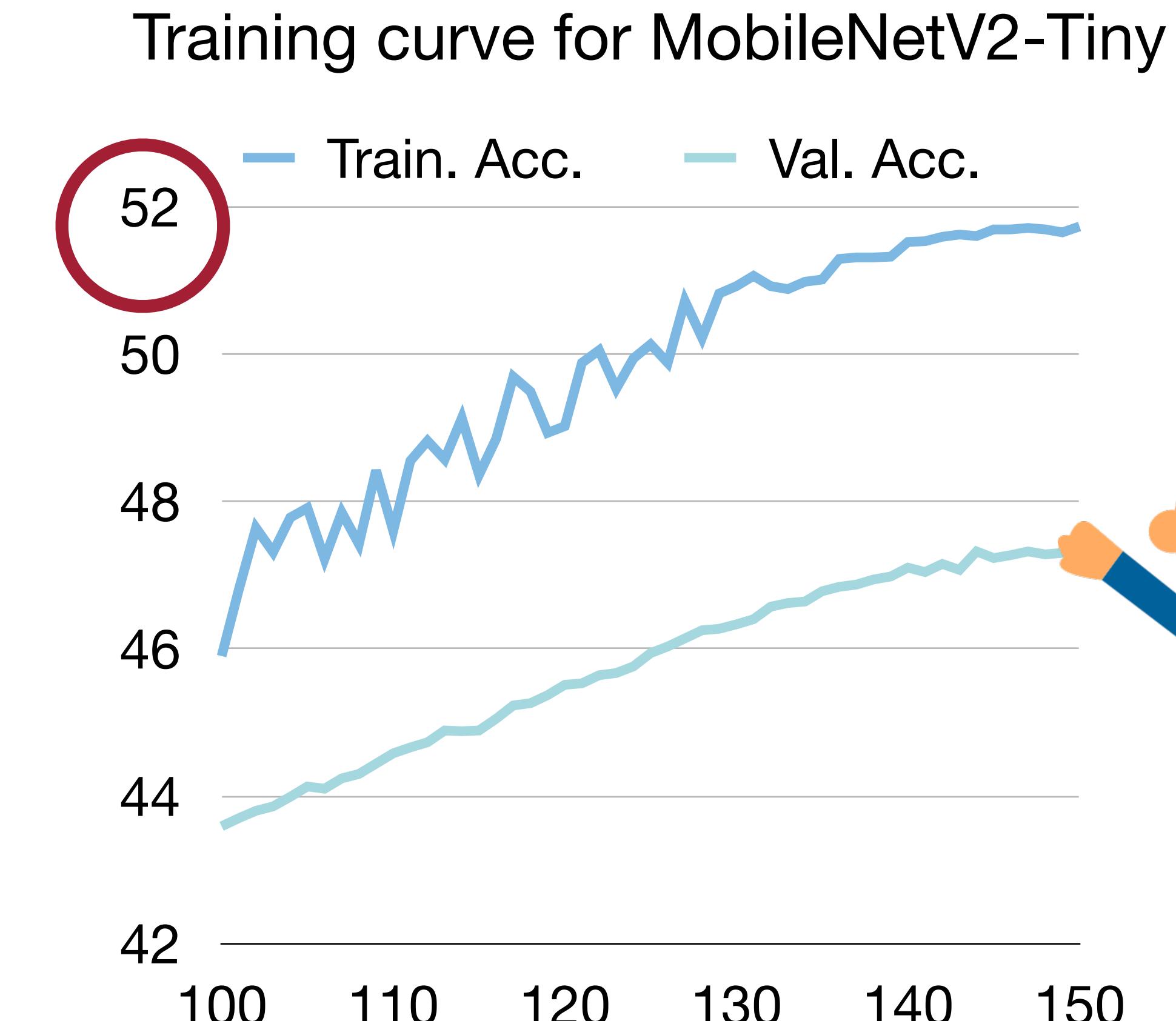
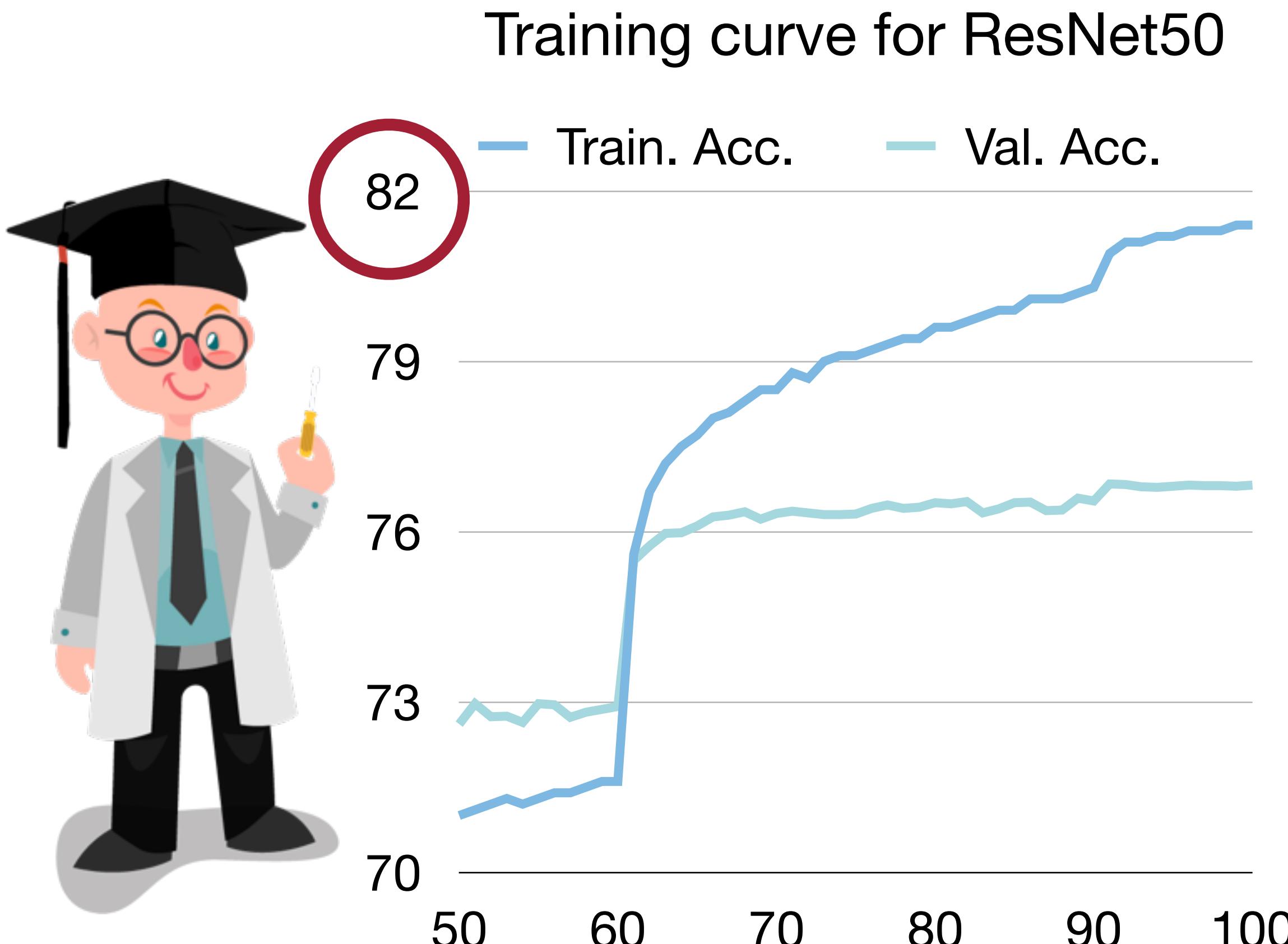
MCUNet
MobileNetV2-Tiny
...

- Neural network must be **tiny** to run efficiently on tiny edge devices. How to train tiny model with the help of large model?

Network Augmentation for Tiny Deep Learning [Cai et al., ICLR 2022]

Tiny models are hard to train

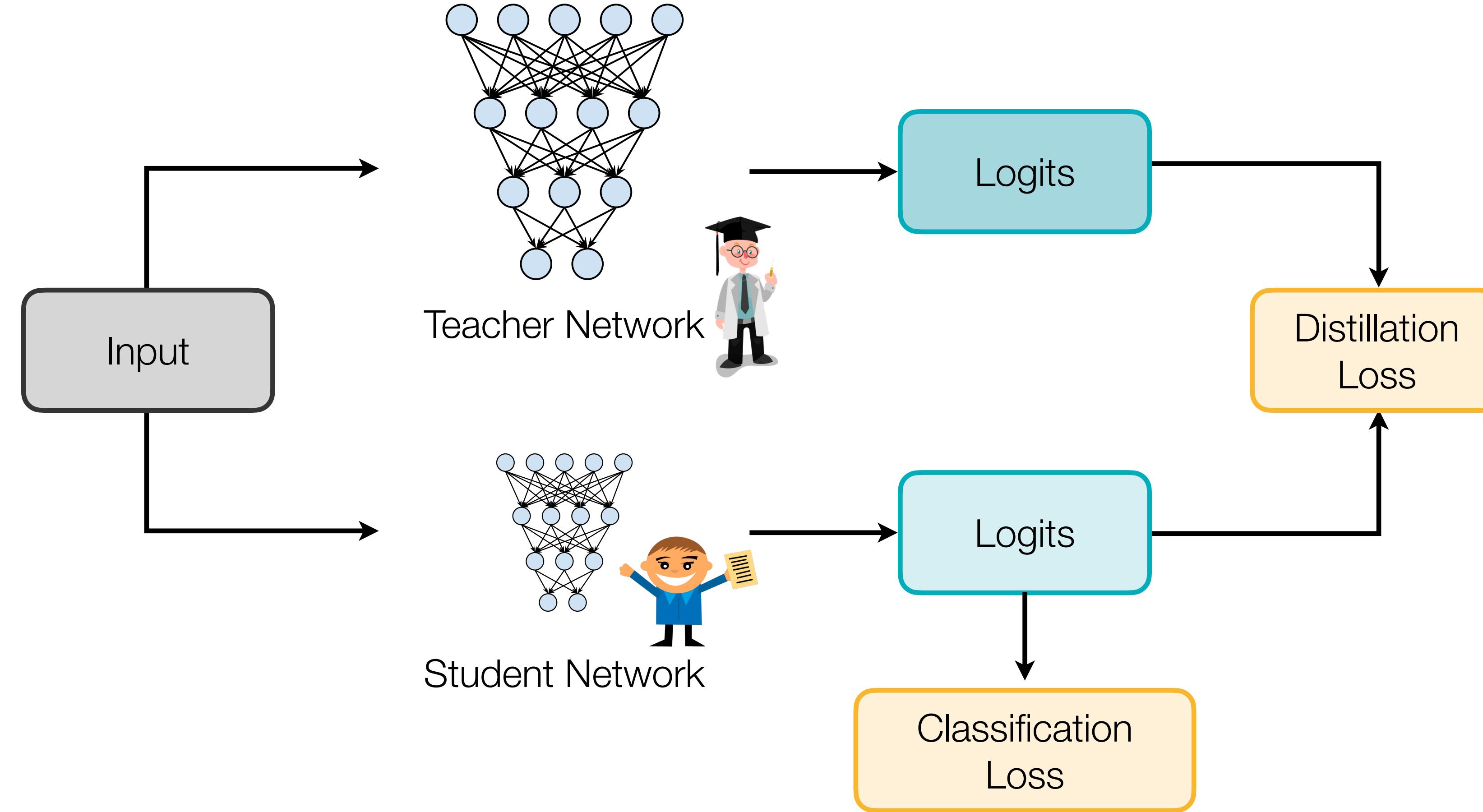
Tiny models underfit large datasets



Question: Can we help the training of tiny models with large models?

Network Augmentation for Tiny Deep Learning [Cai et al., ICLR 2022]

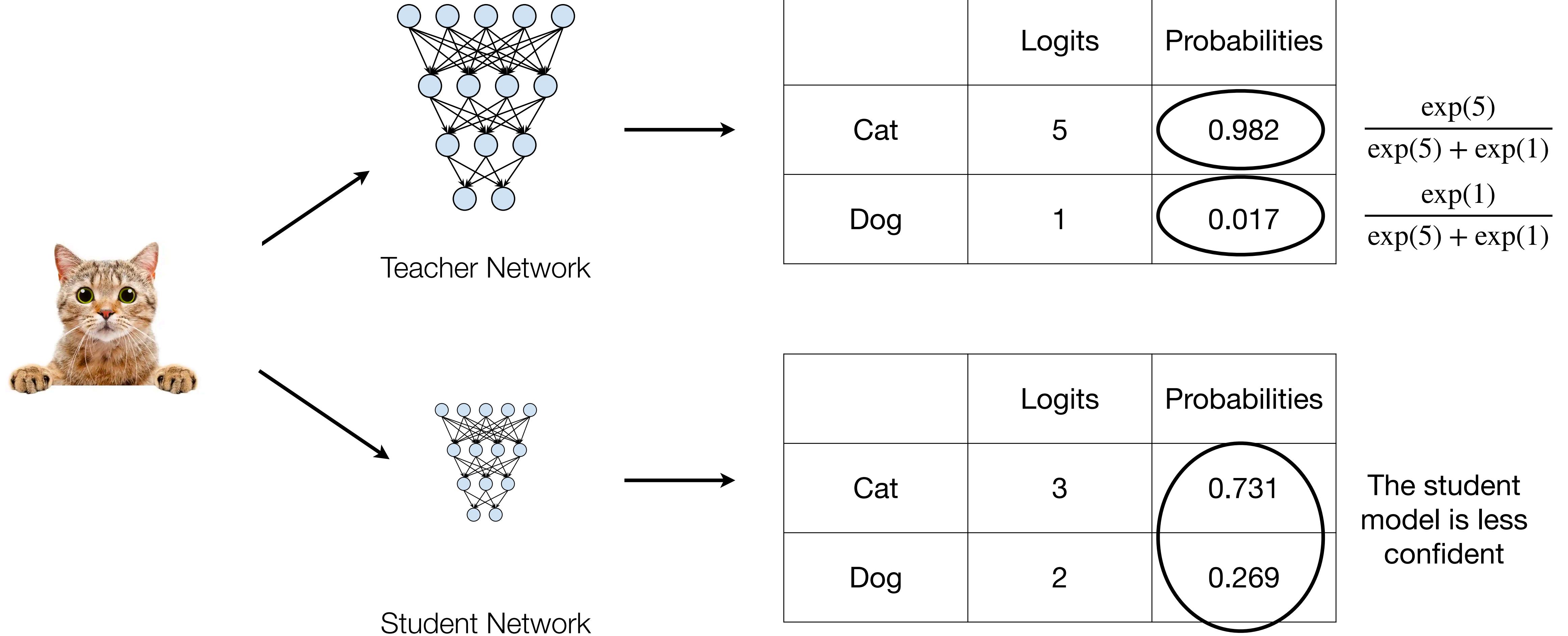
Illustration of knowledge distillation



Distilling the Knowledge in a Neural Network [Hinton et al., NeurIPS Workshops 2014]

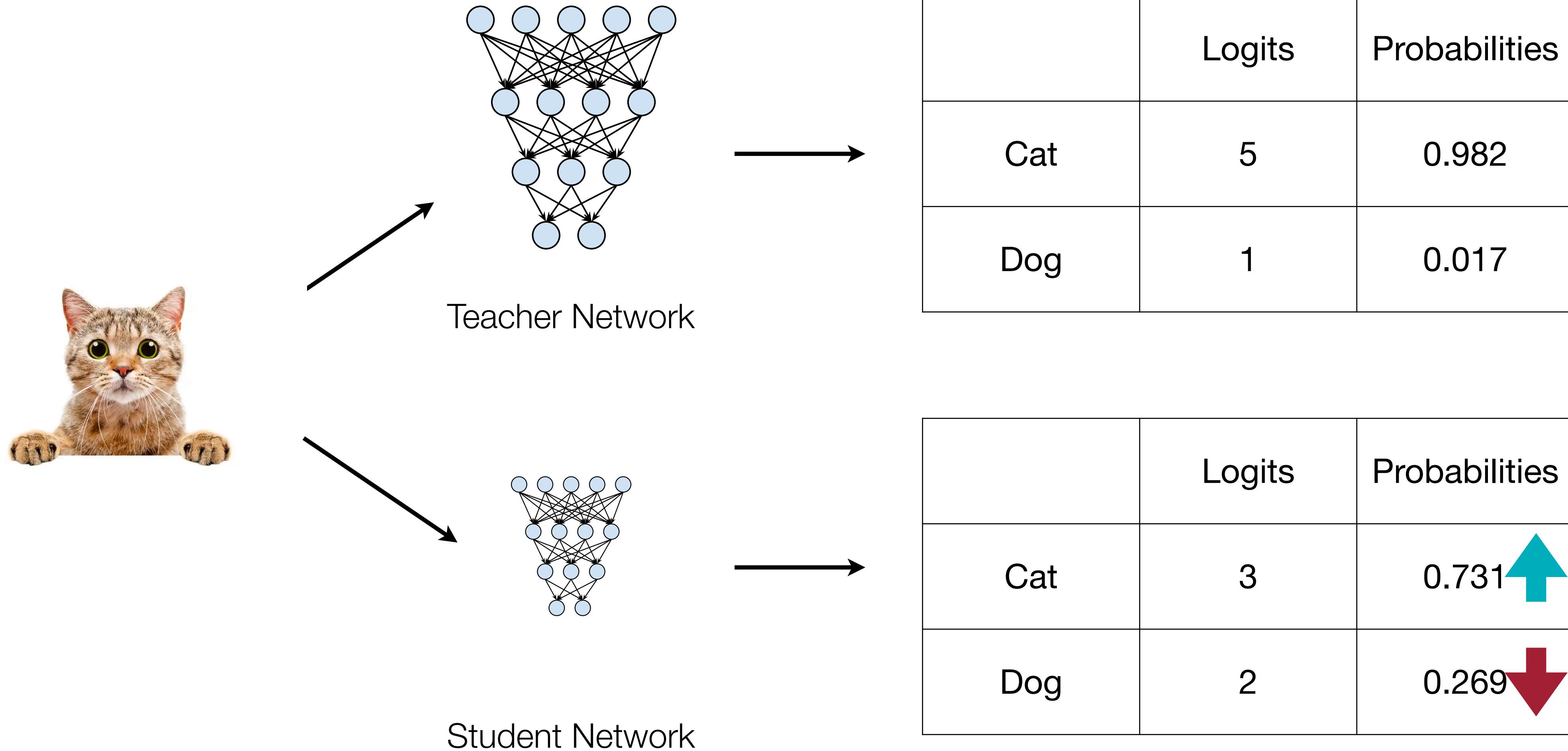
Intuition of knowledge distillation

Matching prediction probabilities between teacher and student



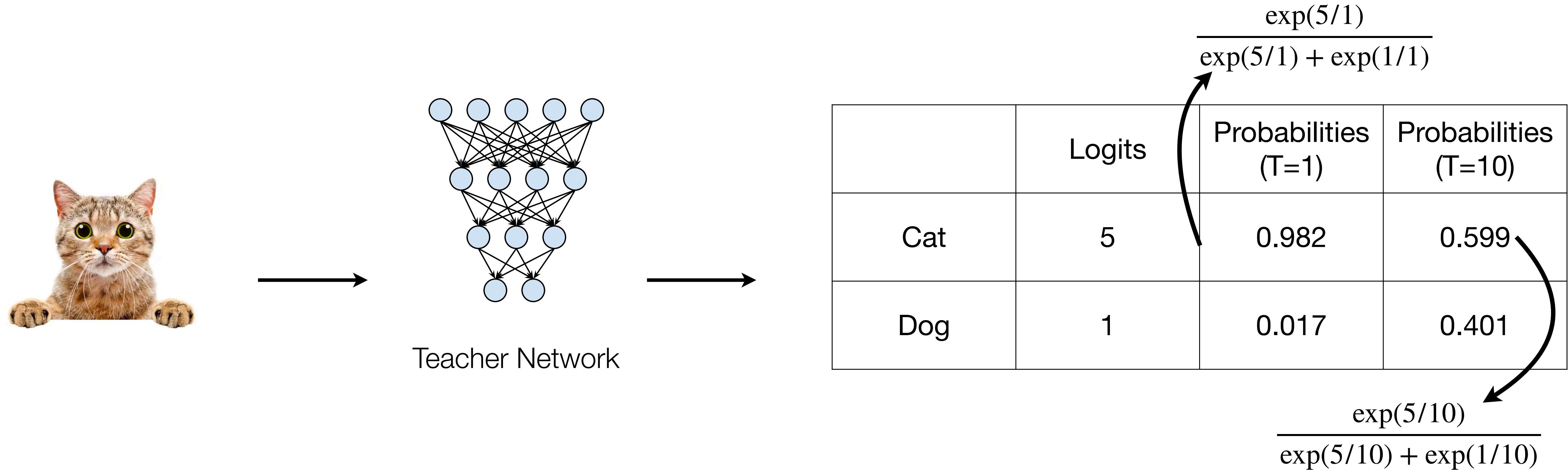
Intuition of knowledge distillation

Matching prediction probabilities between teacher and student



Intuition of knowledge distillation

Concept of temperature



A larger temperature smooths the output probability distribution.

Formal Definition of KD

- Neural networks typically use a softmax function to generate the **logits** z_i to class **probabilities** $p(z_i, T) = \frac{\exp(z_i/T)}{\sum_j \exp(z_j/T)}$. Here, $i, j = 0, 1, 2, \dots, C - 1$, where C is the number of classes. T is the temperature, which is normally set to 1.
- The goal of knowledge distillation is to **align the class probability distributions from teacher and student networks**.

Distilling the Knowledge in a Neural Network [Hinton *et al.*, NeurIPS Workshops 2014]

Knowledge Distillation

Today we will:

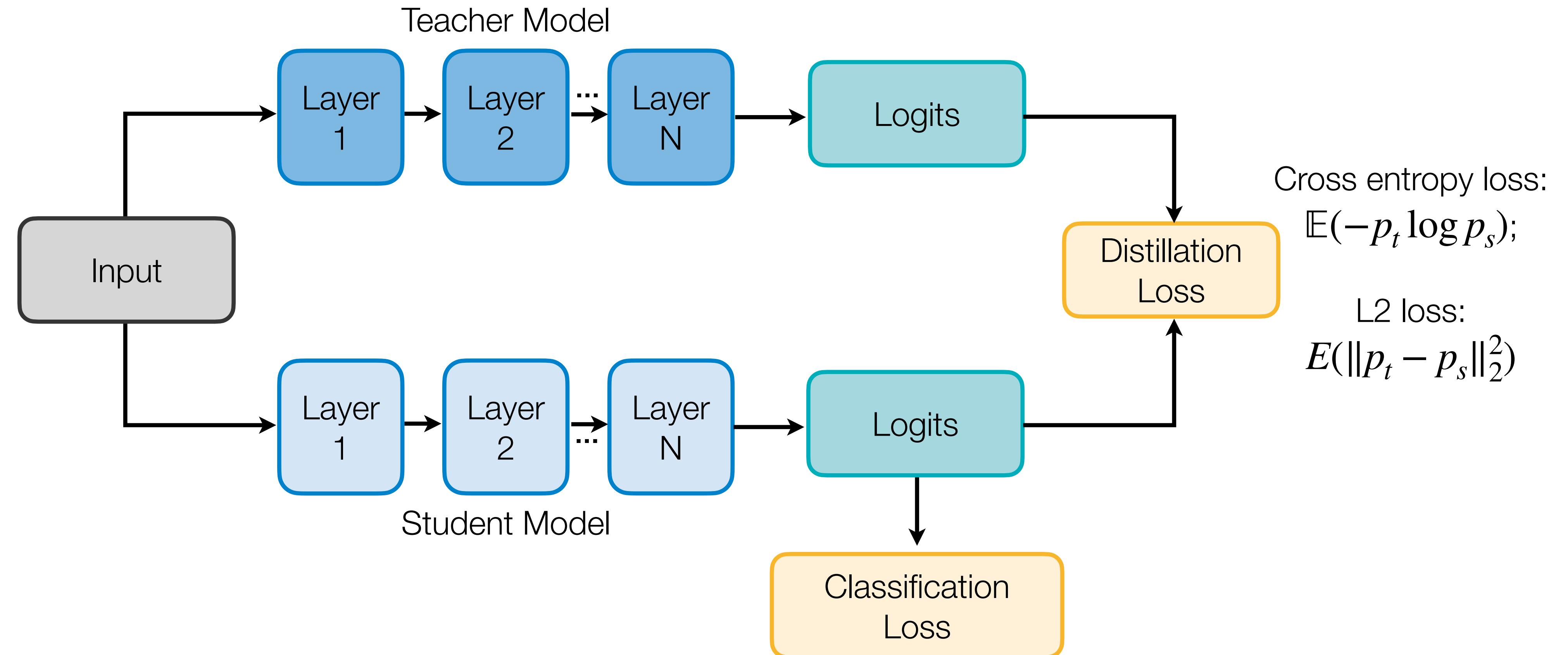
1. What is knowledge distillation;
2. What to match;
3. Self and online distillation;
4. Distillation for different tasks;
5. Network augmentation.



What to match?

- 1. Output logits**
2. Intermediate weights
3. Intermediate features
4. Gradients
5. Sparsity patterns
6. Relational information

Matching output logits



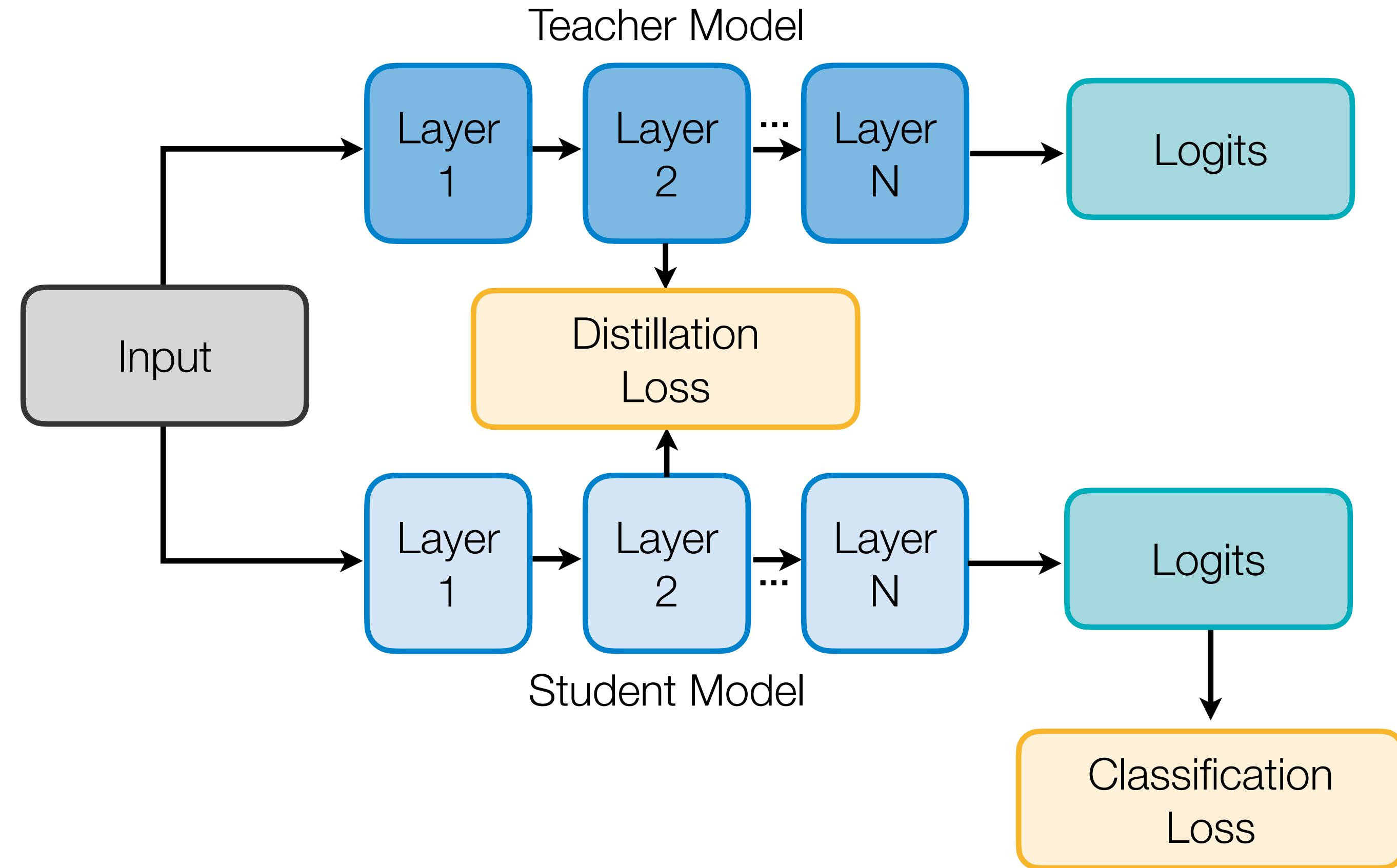
Distilling the Knowledge in a Neural Network [Hinton et al., NeurIPS Workshops 2014]
Do Deep Nets Really Need to be Deep? [Ba and Caruana, NeurIPS 2014]

What to match?

1. Output logits
2. **Intermediate weights**
3. Intermediate features
4. Gradients
5. Sparsity patterns
6. Relational information

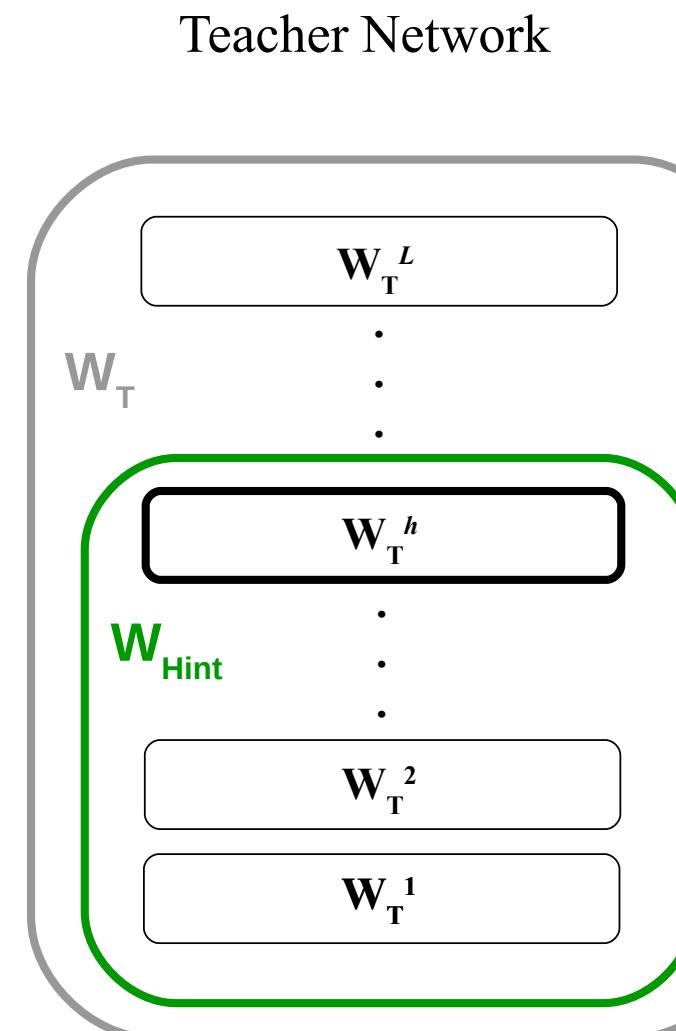
What else to match other than output logits?

Matching intermediate weights

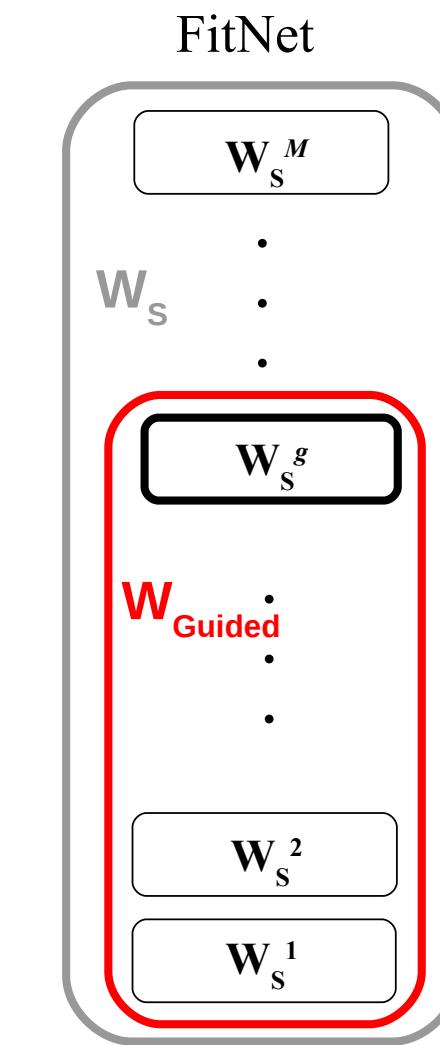


Knowledge Distillation: A Survey [Gou et al., IJCV 2020]

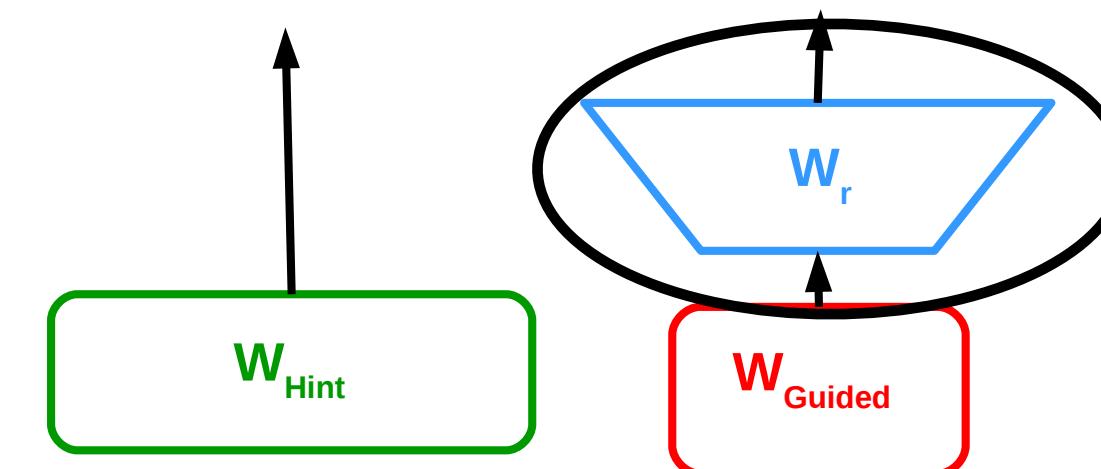
Matching intermediate weights



(a) Teacher and Student Networks



$$W_{\text{Guided}}^* = \underset{W_{\text{Guided}}}{\operatorname{argmin}} \mathcal{L}_{HT}(W_{\text{Guided}}, W_r)$$



(b) Hints Training

An FC layer used to align the shapes of teacher and student weights

- Other than the cross-entropy distillation loss, also add a L2 loss between teacher weights and student weights (linear transformation is applied to match the dimensionalities).

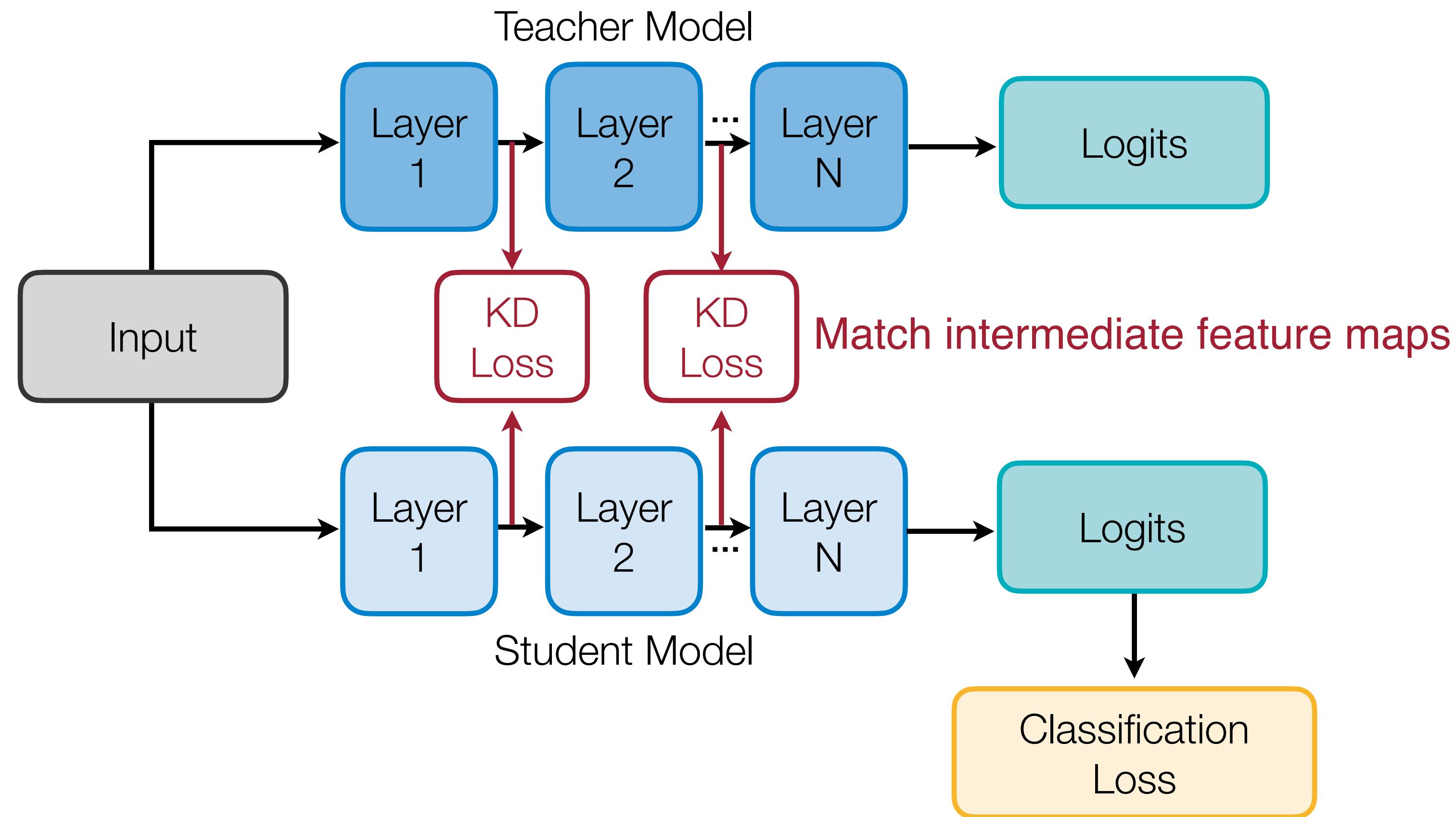
What to match?

1. Output logits
2. Intermediate weights
- 3. Intermediate features**
4. Gradients
5. Sparsity patterns
6. Relational information

Matching intermediate features

Minimizing maximum mean discrepancy between feature maps

- Intuition: teacher and student networks should have similar **feature** distributions, not just output probability distributions.



Like What You Like: Knowledge Distill via Neuron Selectivity Transfer [Huang and Wang, arXiv 2017]

What to match?

1. Output logits
2. Intermediate weights
3. Intermediate features
- 4. Gradients**
5. Sparsity patterns
6. Relational information

Intermediate attention maps

Gradients of feature maps are used to characterize “attention” of DNNs

- The attention of a CNN feature map x is defined as $\frac{\partial L}{\partial x}$, where L is the learning objective.
- Intuition: If $\frac{\partial L}{\partial x_{i,j}}$ is large, a small perturbation at i, j will significantly impact the final output. As a result, the network is putting more attention on position i, j .

input image



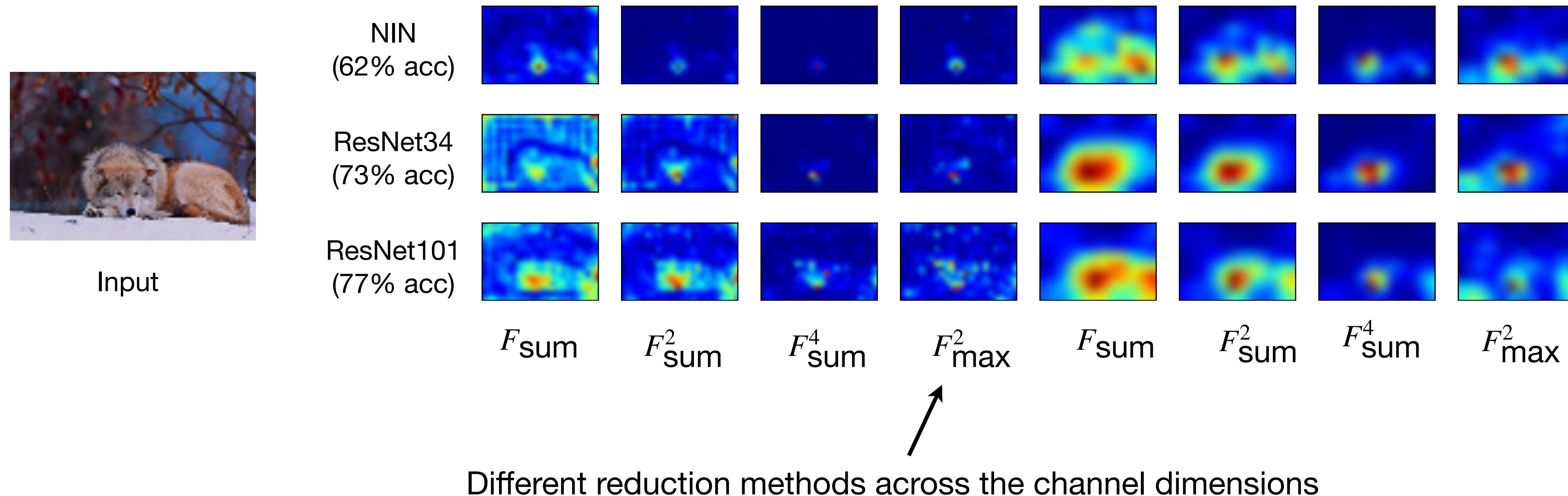
attention map



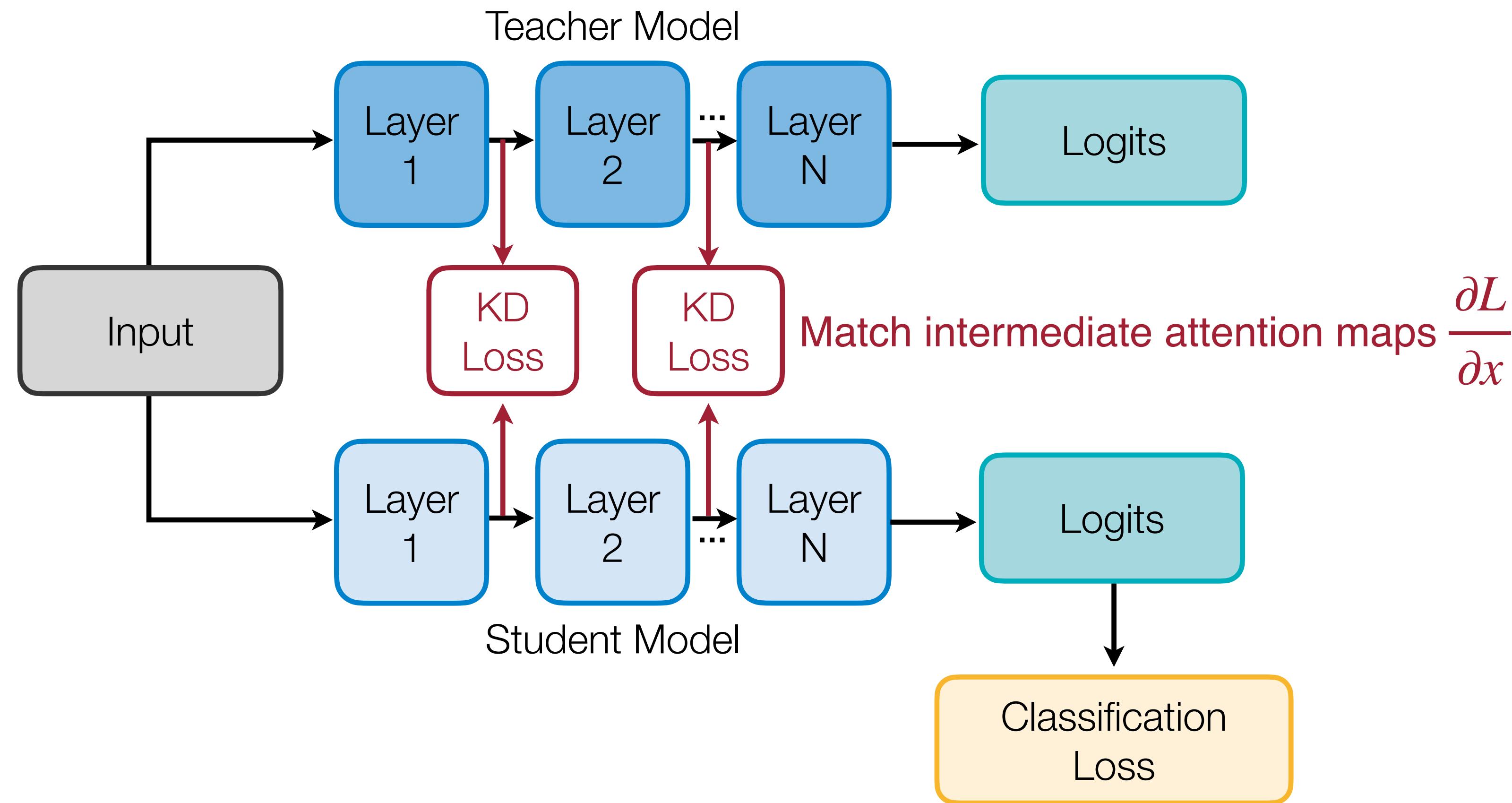
Intermediate attention maps

Performant models have similar attention maps

- Attention maps of performant ImageNet models (ResNets) are indeed similar to each other, but the less performant model (NIN) has quite different attention maps.



Matching intermediate attention maps



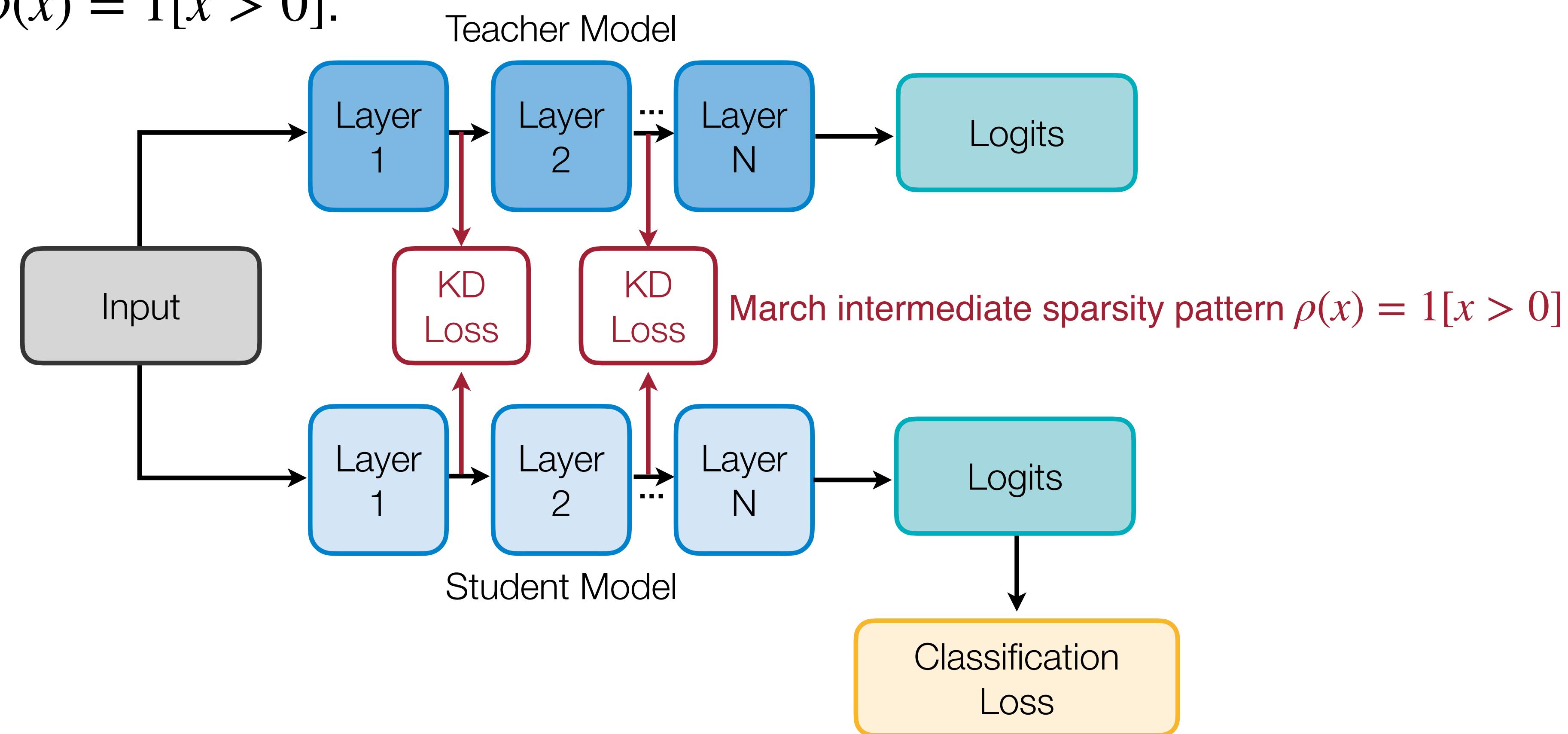
Paying More Attention to Attention: Improving the Performance of Convolutional Neural Networks via Attention Transfer [Zagoruyko and Komodakis, ICLR 2017]

What to match?

1. Output logits
2. Intermediate weights
3. Intermediate features
4. Gradients
- 5. Sparsity patterns**
6. Relational information

Matching sparsity patterns

- Intuition: the teacher and student networks should have similar sparsity patterns after the ReLU activation. A neuron is **activated** after ReLU if its value is larger than 0, denoted by the indicator function $\rho(x) = 1[x > 0]$.



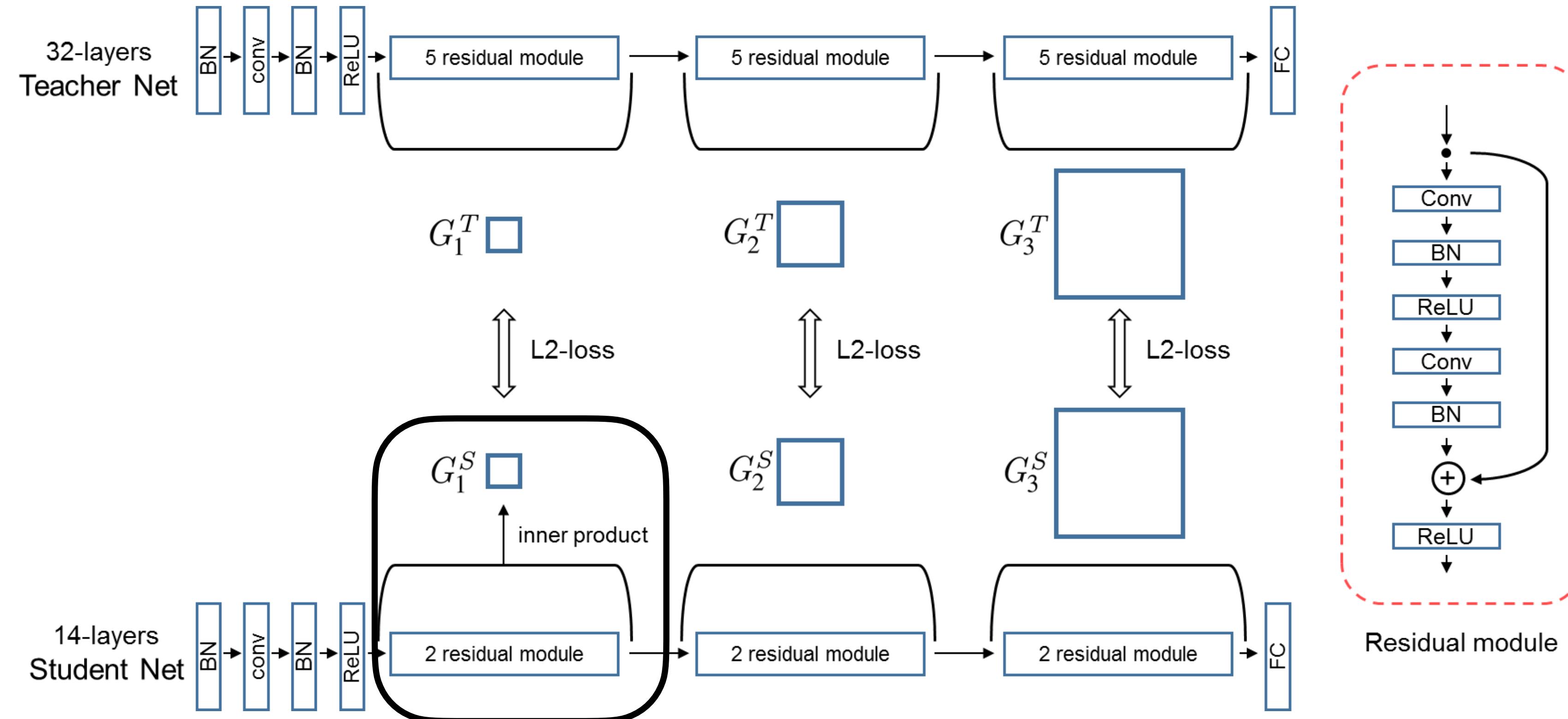
Knowledge Transfer via Distillation of Activation Boundaries Formed by Hidden Neurons [Heo et al., AAAI 2019]

What to match?

1. Output logits
2. Intermediate weights
3. Intermediate features
4. Gradients
5. Sparsity patterns
- 6. Relational information**

Matching relational information

Relations between different layers



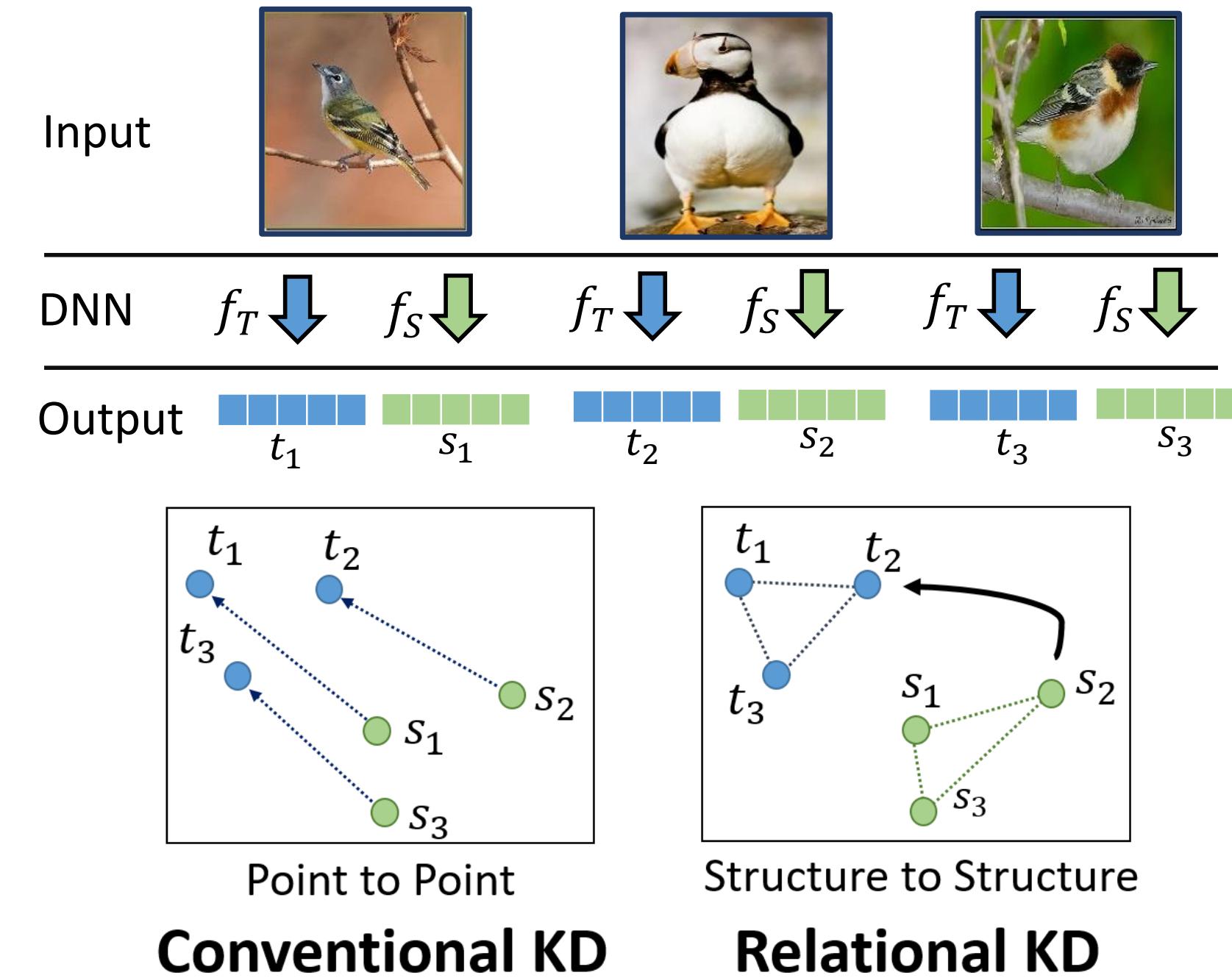
Use inner product to extract relational information (a matrix of shape $C_{\text{in}} \times C_{\text{out}}$, reduction on the spatial dimensions) for both student and teacher networks.

Note: the student and teacher networks only differ in number of layers, not number of channels.
Then match the resulting dot products between teacher and student networks (G_1^T, G_1^S).

A Gift from Knowledge Distillation: Fast Optimization, Network Minimization and Transfer Learning [Yim et al., CVPR 2017]

Matching relational information

Relations between different samples



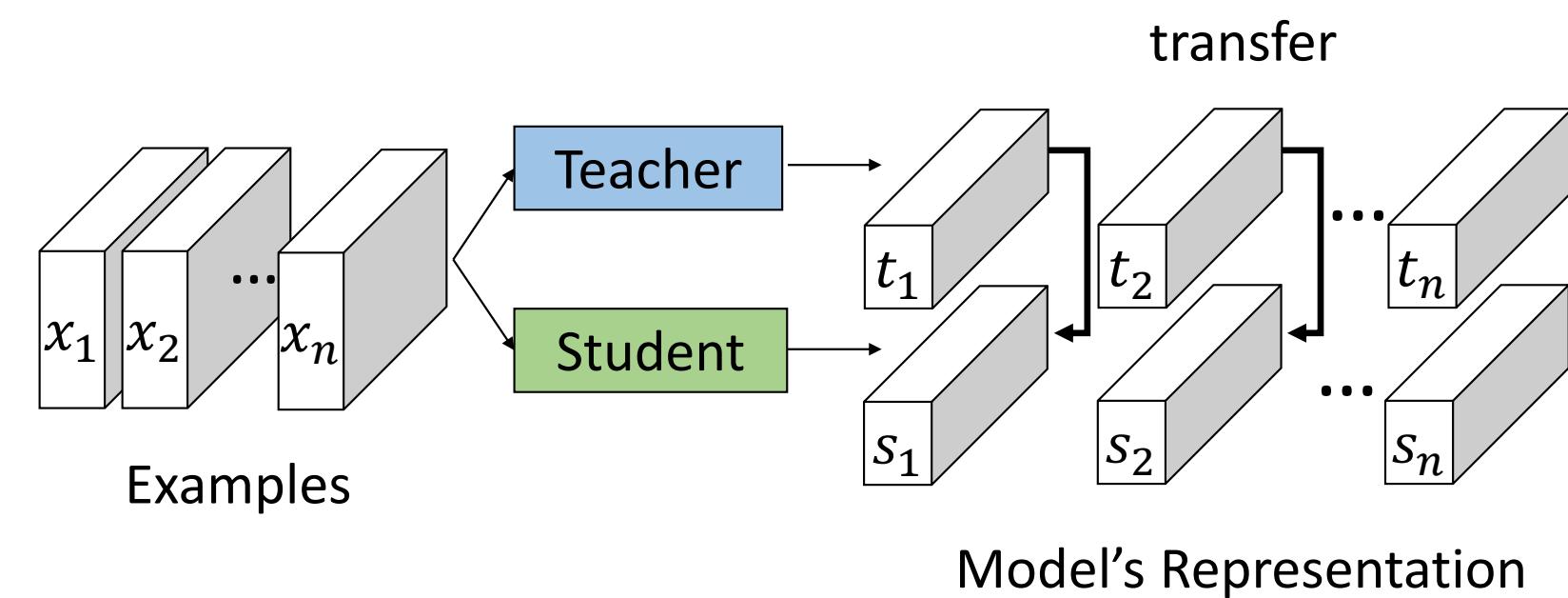
Conventional KD focuses on matching features / logits for **one input**.
Relational KD looks at the **relations** between intermediate features from **multiple inputs**.

Relational Knowledge Distillation [Park et al., CVPR 2019]

Matching relational information

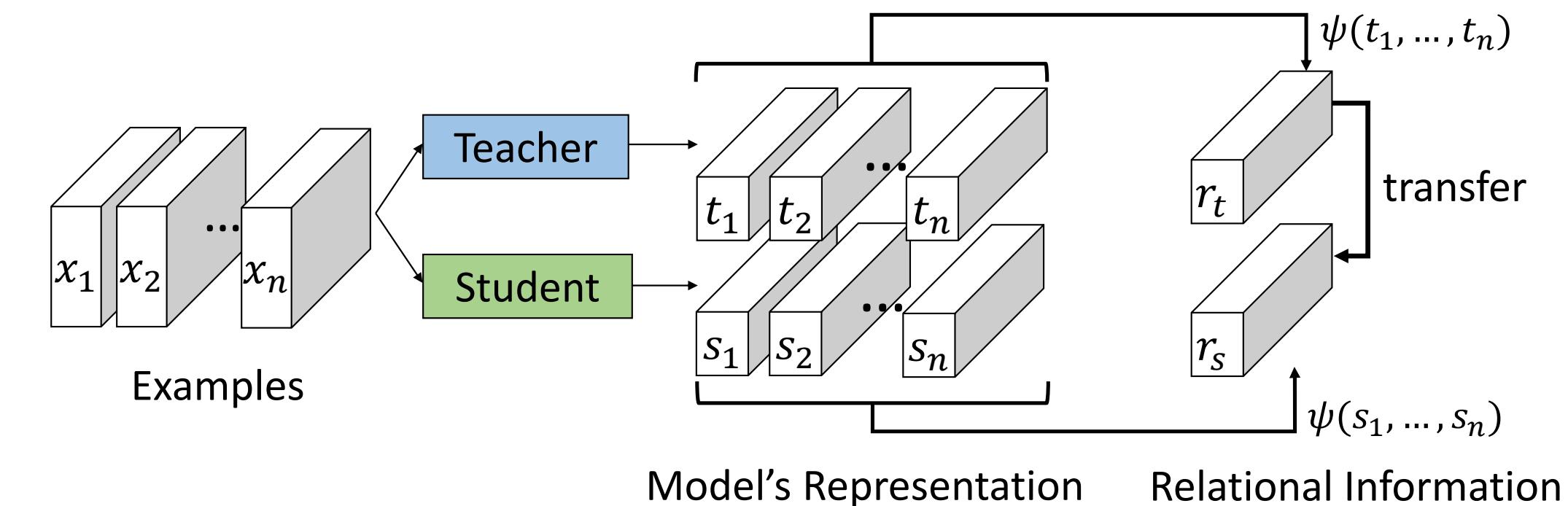
Relations between different samples

$\psi(s_1, s_2, \dots, s_n) = (\|s_1 - s_2\|_2^2, \|s_1 - s_3\|_2^2, \dots, \|s_1 - s_n\|_2^2, \dots, \|s_{n-1} - s_n\|_2^2)$ is a vector of length $n(n - 1)/2$ representing pairwise distances of feature vectors.



Individual Knowledge Distillation

Representative method: FSP (previous slide)
Relation **within** the model, calculated **separately** for each input



Relational Knowledge Distillation

Representative method: RKD
Relation **across** different samples

Relational Knowledge Distillation [Park et al., CVPR 2019]

Knowledge Distillation

Today we will:

1. What is knowledge distillation;
2. What to match;
3. Self and online distillation;
4. Distillation for different tasks;
5. Network Augmentation, a training technique for tiny machine learning models.

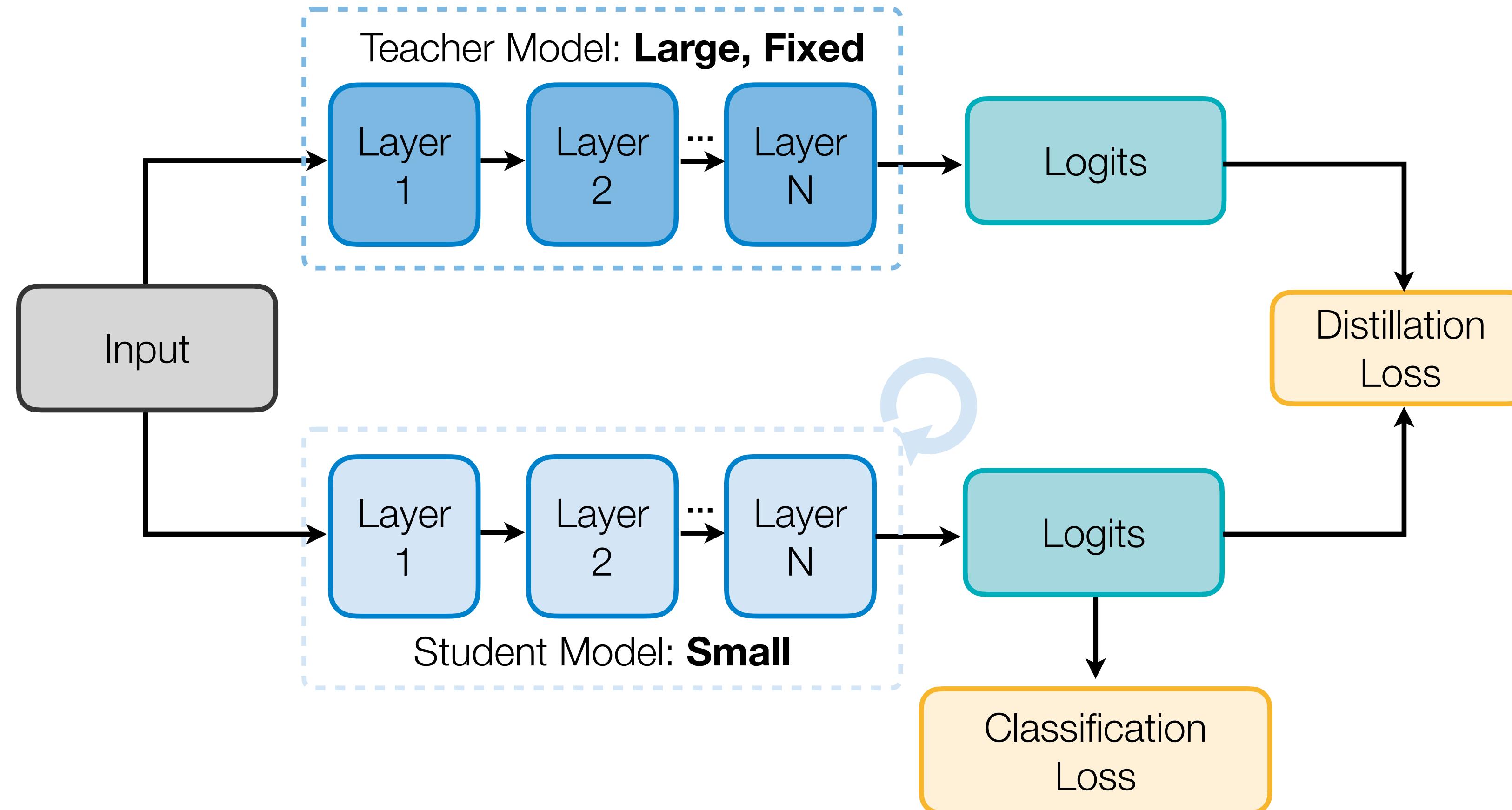


Self and Online Distillation

1. Self Distillation
2. Online Distillation
3. Combined

Overview of knowledge distillation

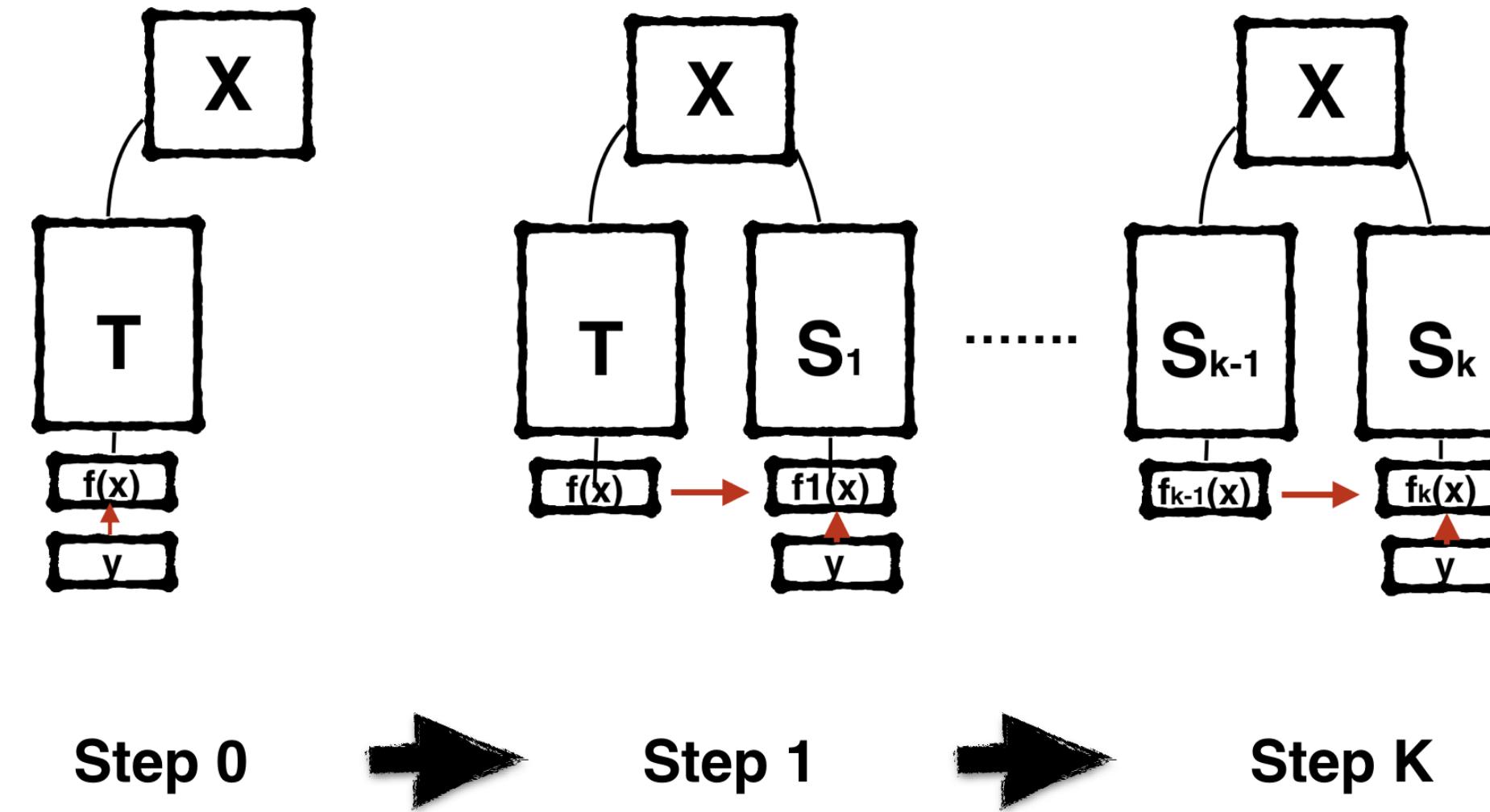
Teacher model is usually larger than the student model and is fixed



Discussion: What is the disadvantage of fixed large teachers? Does it have to be the case that we need a fixed large teacher in KD?

Knowledge Distillation: A Survey [Gou et al., IJCV 2020]

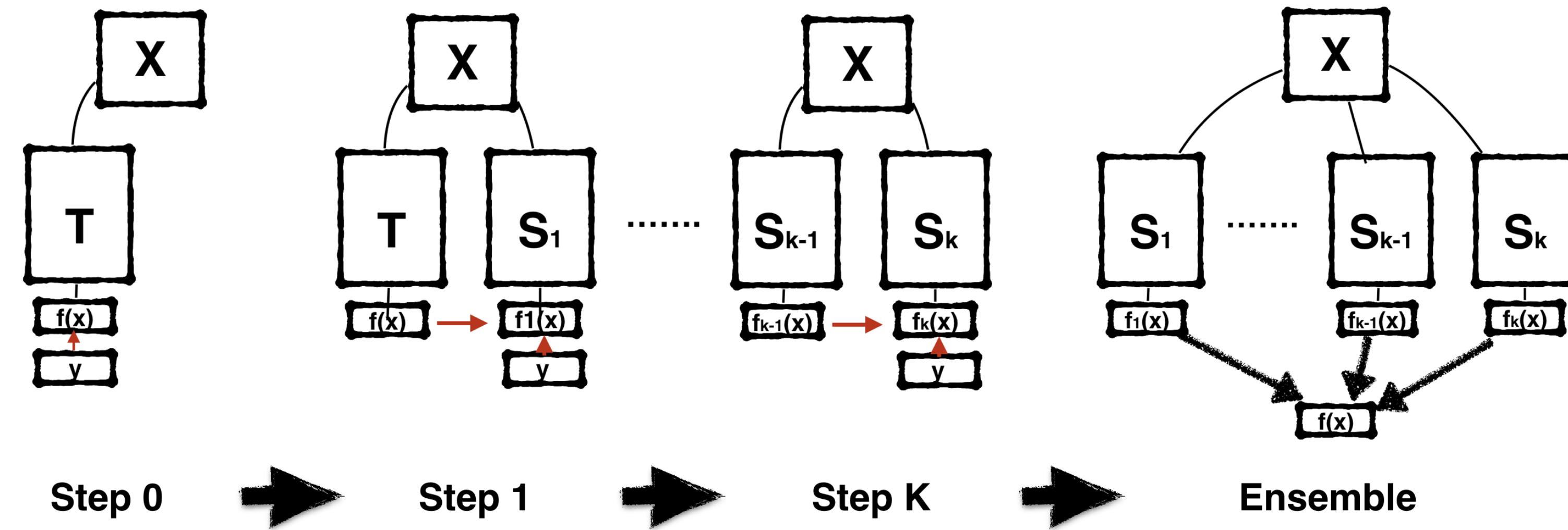
Self-Distillation with Born-Again NNs



- Born-Again Networks **adds iterative training stages** and **using both classification objective and distillation objective** in subsequent stages.
- Network architecture $T = S_1 = S_2 = \dots = S_k$.
- Network accuracy $T < S_1 < S_2 < \dots < S_k$.
- Can alternatively ensemble T, S_1, S_2, \dots, S_k to get even better performance.

Born-Again Neural Networks [Furlanello et al., ICML 2018]

Self-Distillation with Born-Again NNs



- Born-Again Networks adds **iterative training stages** and using both **classification objective** and **distillation objective** in subsequent stages.
- Network architecture $T = S_1 = S_2 = \dots = S_k$.
- Network accuracy $T < S_1 < S_2 < \dots < S_k$.
- Can alternatively ensemble T, S_1, S_2, \dots, S_k to get even better performance.

Born-Again Neural Networks [Furlanello et al., ICML 2018]

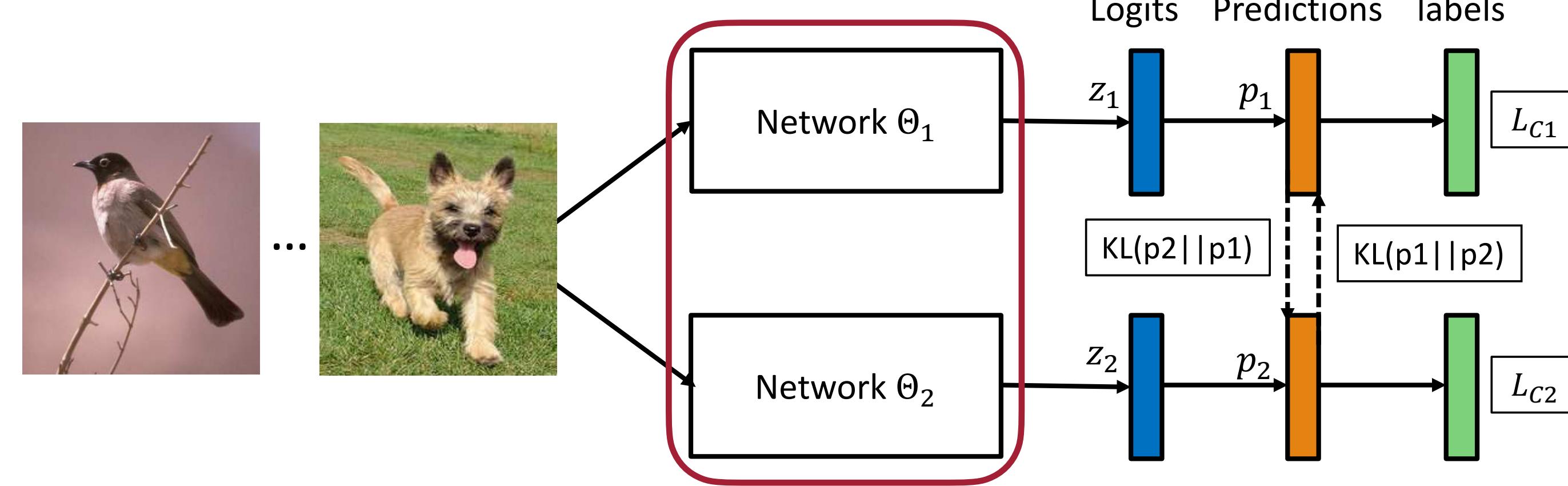
Self and Online Distillation

1. Self Distillation
2. **Online Distillation**
3. Combined

Online Distillation

Deep Mutual Learning

Θ_1, Θ_2 can be the same or different, and they are trained from scratch.



- Idea of deep mutual learning: for both teacher and student networks, we want to add a distillation objective that minimizes the output distribution of the other party.
- $\mathcal{L}(S) = \text{CrossEntropy}(S(I), y) + \text{KL}(S(I), T(I));$
- $\mathcal{L}(T) = \text{CrossEntropy}(T(I), y) + \text{KL}(T(I), S(I)).$
- Note: it is not necessary to pretrain T and $S = T$ is allowed.

Online Distillation

Deep Mutual Learning

Network Types		CIFAR-10						CIFAR-100					
		Independent		DML		DML-Ind		Independent		DML		DML-Ind	
Net 1	Net 2	Net 1	Net 2	Net 1	Net 2	Net 1	Net 2	Net 1	Net 2	Net 1	Net 2	Net 1	Net 2
Resnet-32	Resnet-32	92.47	92.47	92.68	92.80	0.21	0.33	68.99	68.99	71.19	70.75	2.20	1.76
WRN-28-10	Resnet-32	95.01	92.47	95.75	93.18	0.74	0.71	78.69	68.99	78.96	70.73	0.27	1.74
MobileNet	Resnet-32	93.59	92.47	94.24	93.32	0.65	0.85	73.65	68.99	76.13	71.10	2.48	2.11
MobileNet	MobileNet	93.59	93.59	94.10	94.30	0.51	0.71	73.65	73.65	76.21	76.10	2.56	2.45
WRN-28-10	MobileNet	95.01	93.59	95.73	94.37	0.72	0.78	78.69	73.65	80.28	77.39	1.59	3.74
WRN-28-10	WRN-28-10	95.01	95.01	95.66	95.63	0.65	0.62	78.69	78.69	80.28	80.08	1.59	1.39

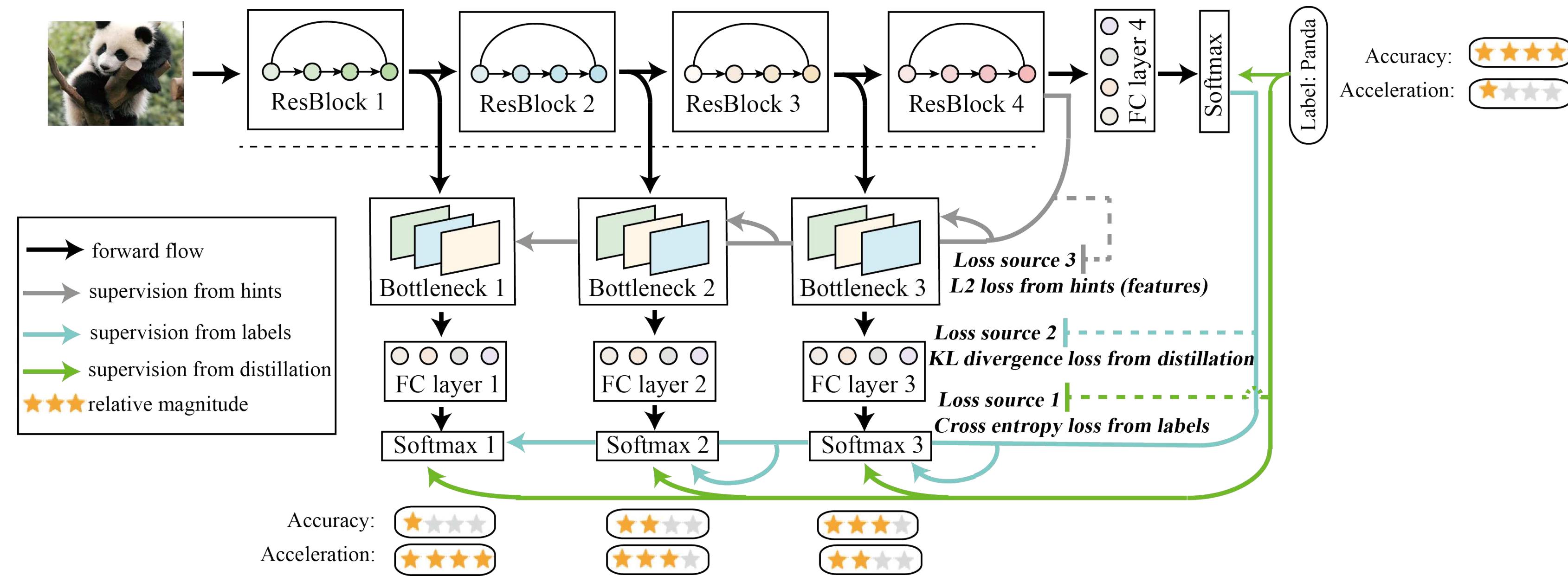
Deep mutual learning can improve both student (net 2) and teacher (net 1) models.

Self and Online Distillation

1. Self Distillation
2. Online Distillation
- 3. Combined**

Combining Online and Self-Distillation

Be Your Own Teacher: deep supervision + distillation



- Use deeper layers to distill shallower layers.
- **Intuition:** Labels at later stages are more reliable, so the authors use them to supervise the predictions from the previous stages.

Be Your Own Teacher: Improve the Performance of Convolutional Neural Networks via Self Distillation [Zhang et al., ICCV 2019]

Combining Online and Self-Distillation

Be Your Own Teacher: deep supervision + distillation

Neural Networks	Baseline	Classifier 1/4	Classifier 2/4	Classifier3/4	Classifier 4/4	Ensemble
VGG19(BN)	64.47	63.59	67.04	68.03	67.73	68.54
ResNet18	77.09	67.85	74.57	78.23	78.64	79.67
ResNet50	77.68	68.23	74.21	75.23	80.56	81.04
ResNet101	77.98	69.45	77.29	81.17	81.23	82.03
ResNet152	79.21	68.84	78.72	81.43	81.61	82.29
ResNeXt29-8	81.29	71.15	79.00	81.48	81.51	81.90
WideResNet20-8	79.76	68.85	78.15	80.98	80.92	81.38
WideResNet44-8	79.93	72.54	81.15	81.96	82.09	82.61
WideResNet28-12	80.07	71.21	80.86	81.58	81.59	82.09
PyramidNet101-240	81.12	69.23	78.15	80.98	82.30	83.51

- Results on CIFAR100 shows consistent performance improvements over the baseline.
- In addition, predictions from intermediate classifiers (1/4, 2/4, 3/4) can sometimes outperform the baseline. As such, inference efficiency can be improved.

Knowledge Distillation

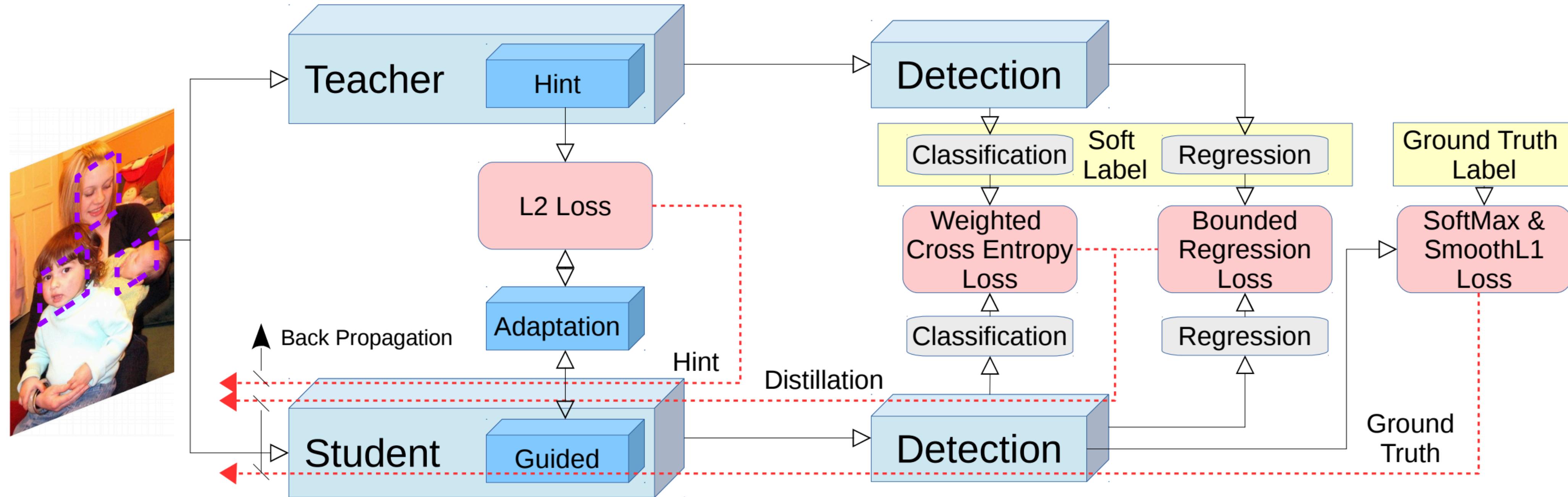
Today we will:

1. What is knowledge distillation;
2. What to match;
3. Self and online distillation;
4. Distillation for different tasks;
5. Network augmentation.



KD for Object Detection

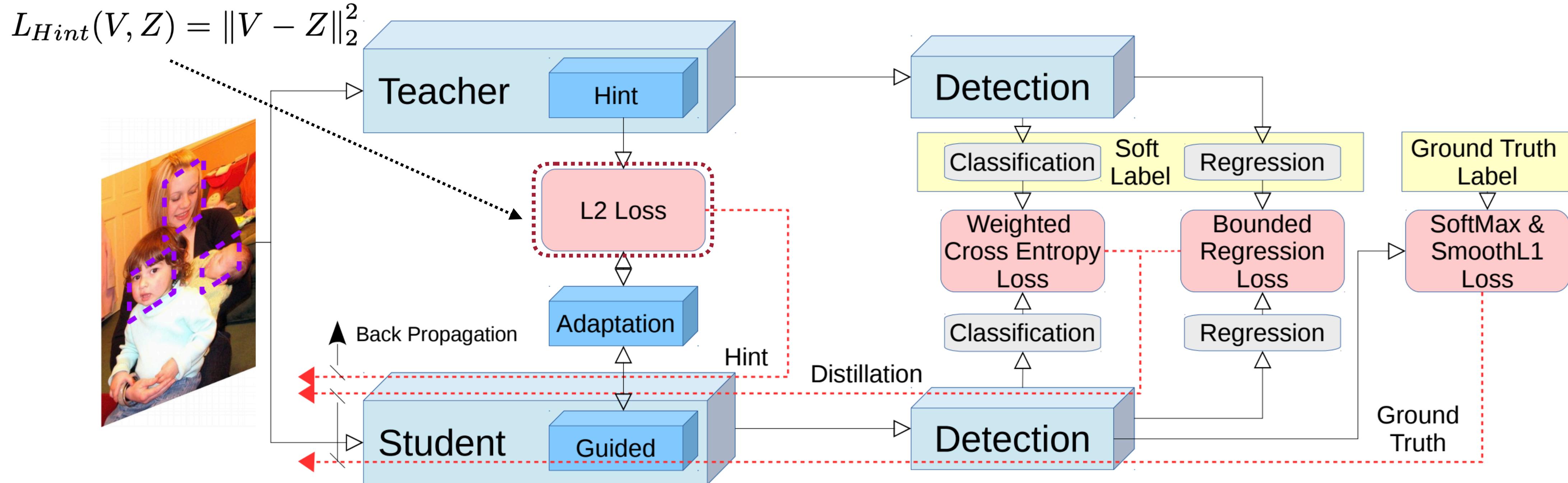
Feature Imitation



Learning Efficient Object Detection Models with Knowledge Distillation [Chen et al., NeurIPS 2017]

KD for Object Detection

Feature Imitation

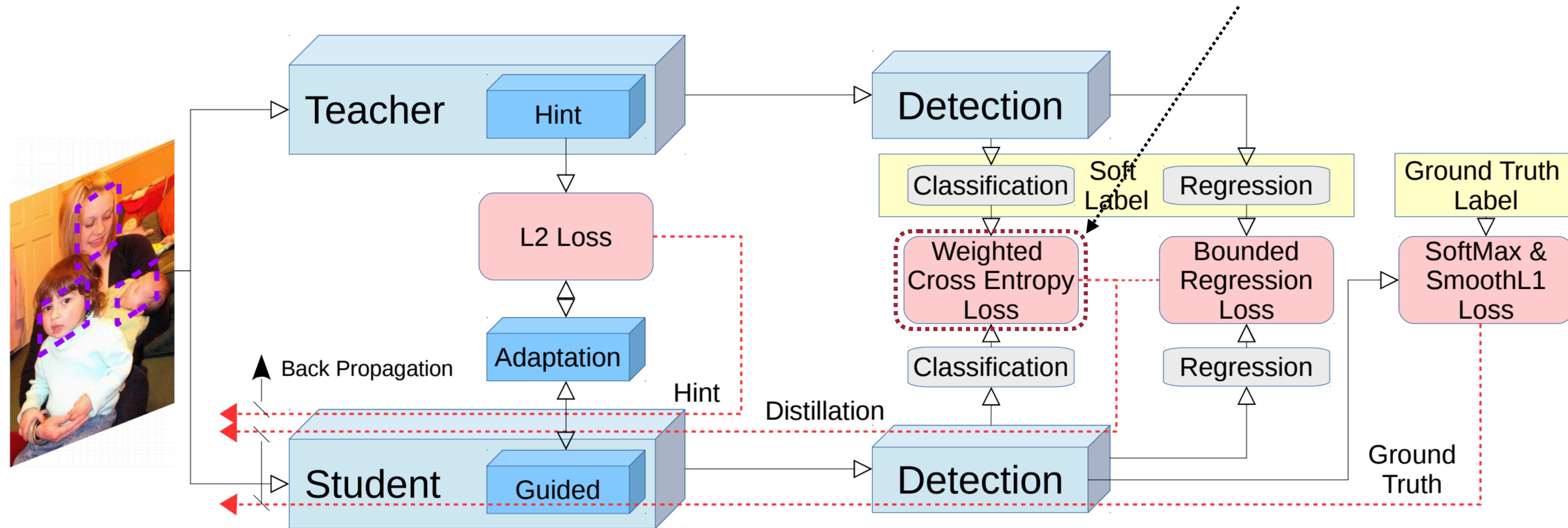


- Add a 1×1 convolution layer to match the shape.

KD for Object Detection

Feature Imitation

$$L_{soft}(P_s, P_t) = - \sum w_c P_t \log P_s$$

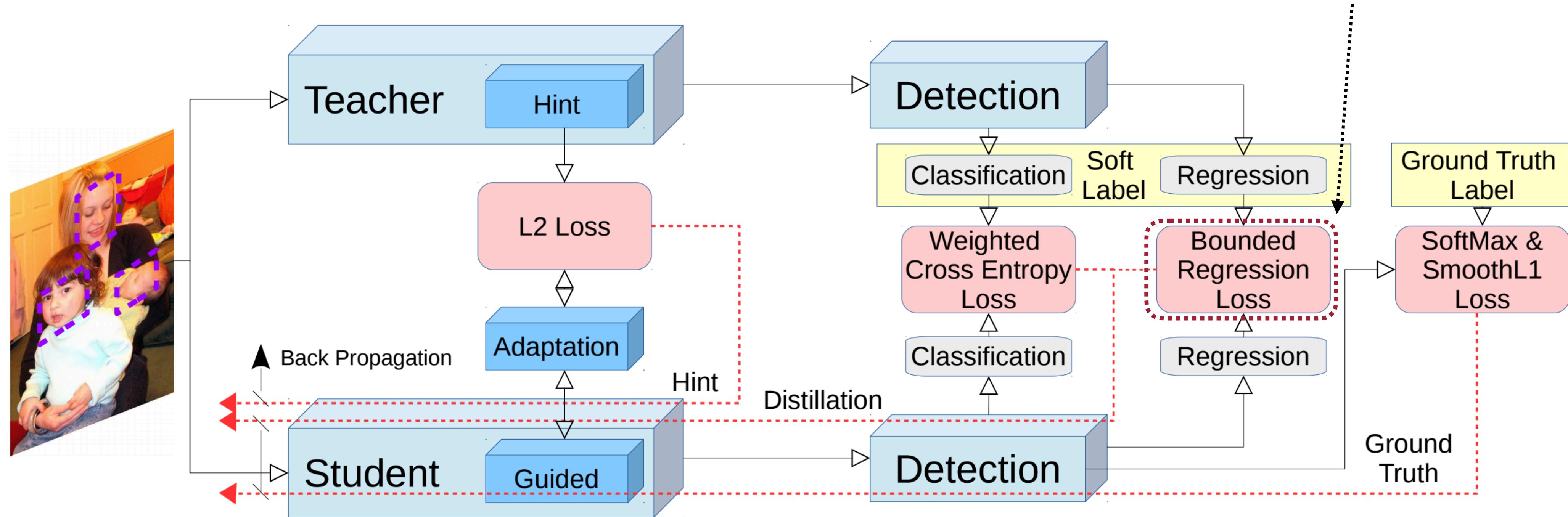


- Use different weights for foreground and background classes to handle the class imbalance problem.

KD for Object Detection

Feature Imitation

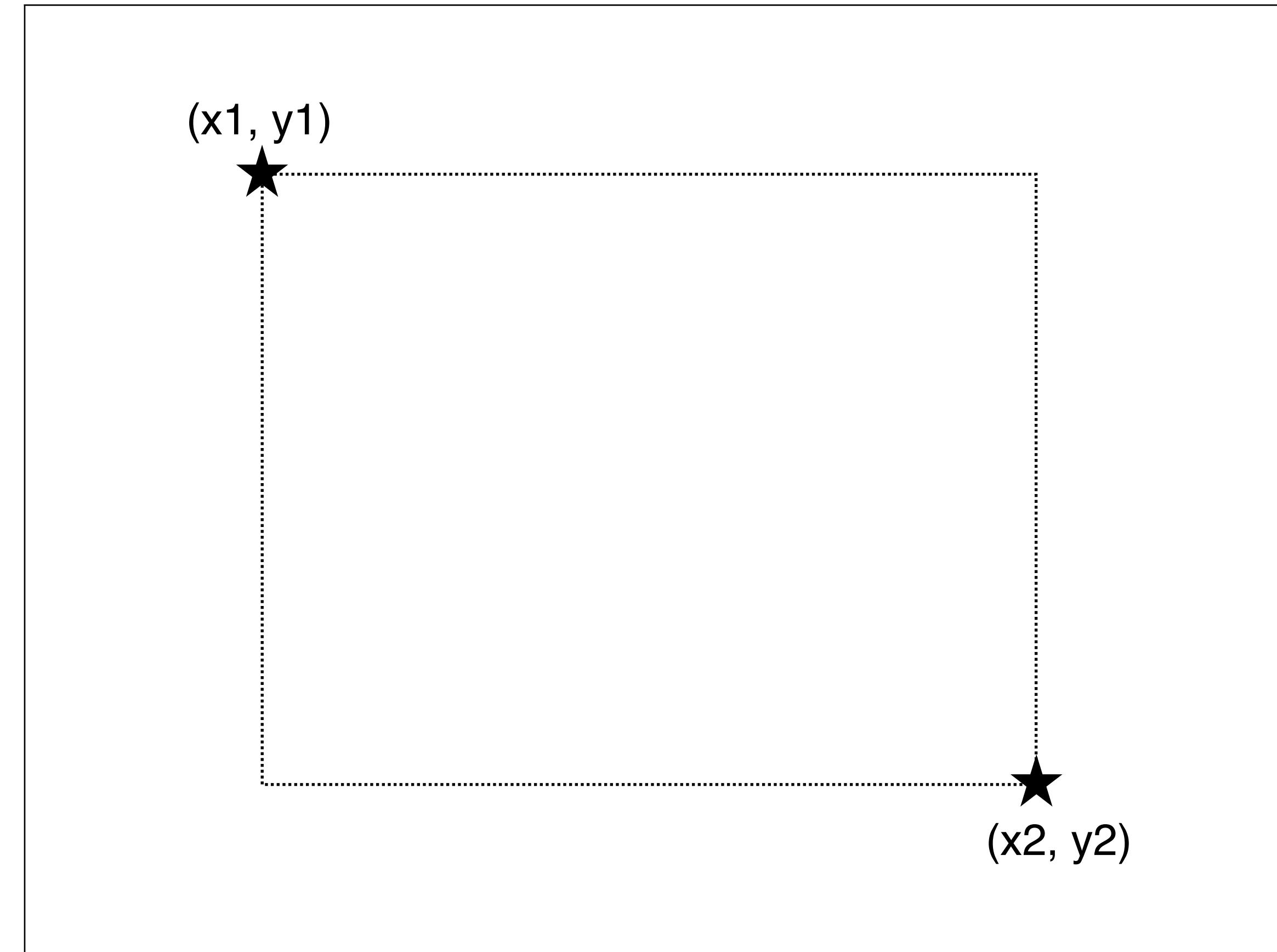
$$L_b(R_s, R_t, y) = \begin{cases} \|R_s - y\|_2^2, & \text{if } \|R_s - y\|_2^2 + m > \|R_t - y\|_2^2 \\ 0, & \text{otherwise} \end{cases}$$



- Exploit teacher's prediction as an upper bound for the student to achieve. Once the quality of the student surpasses that of the teacher with a certain margin, the loss becomes zero.

KD for Object Detection

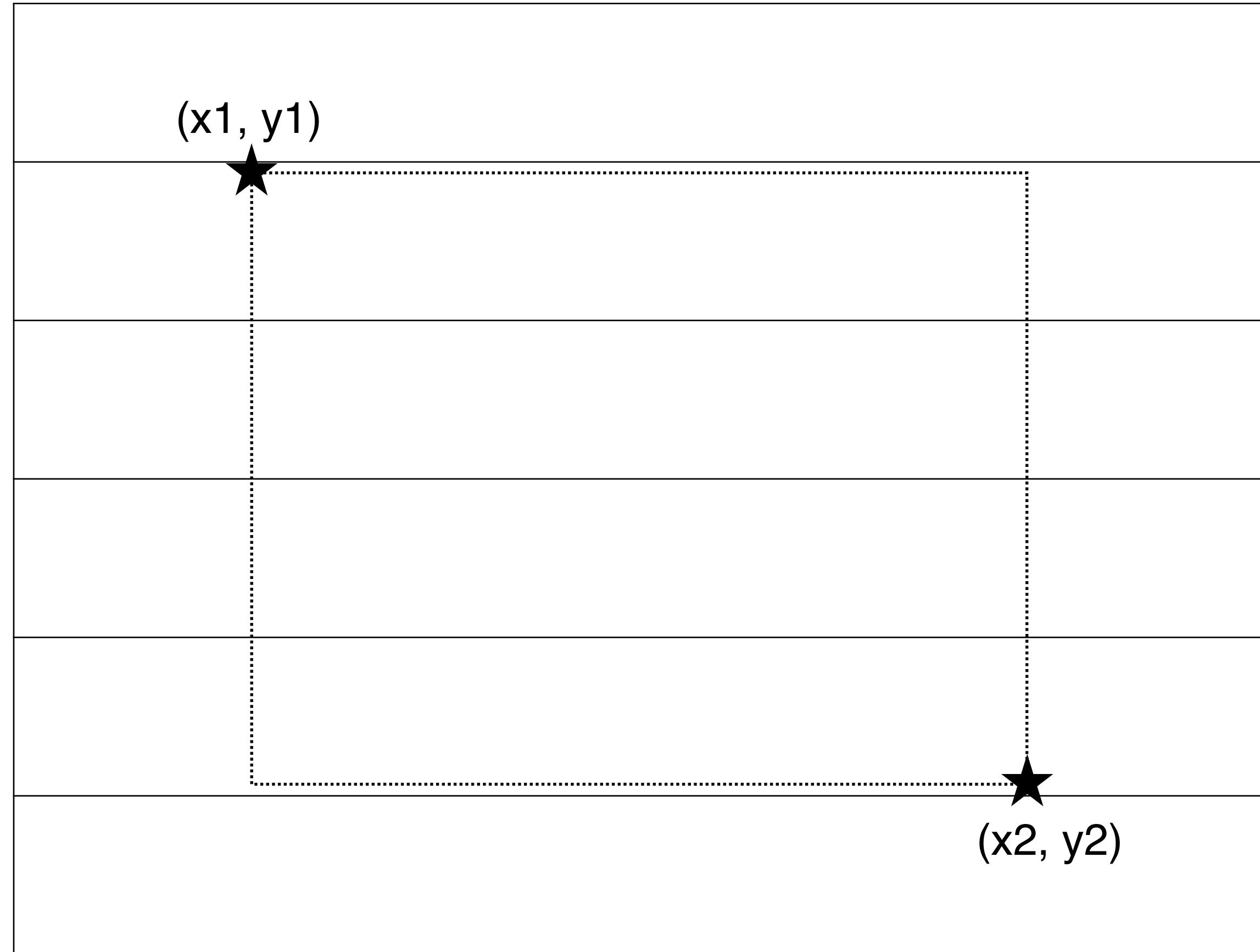
Convert bounding box regression to classification problem



KD for Object Detection

Convert bounding box regression to classification problem

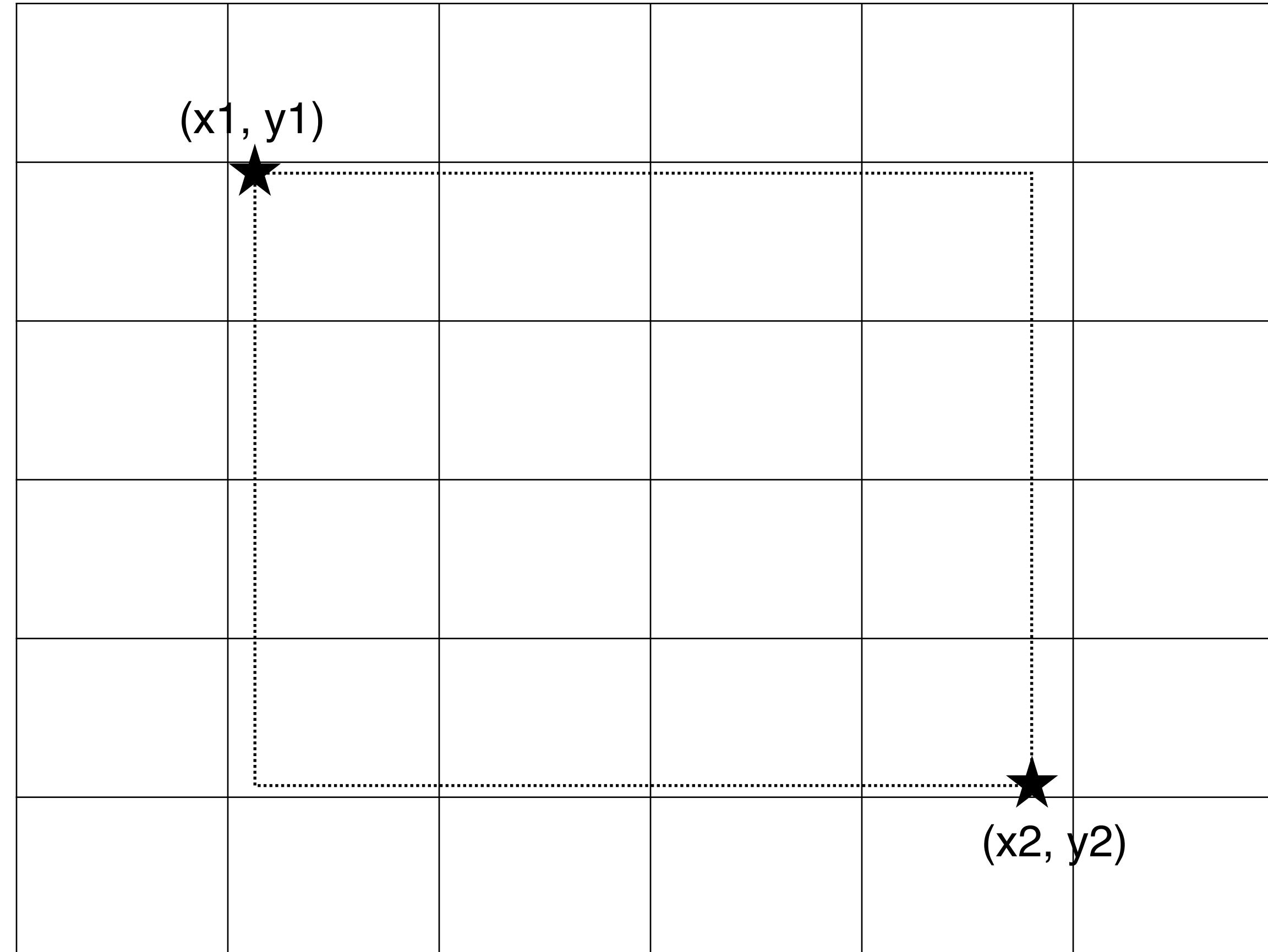
Divide the y-axis into 6 bins



KD for Object Detection

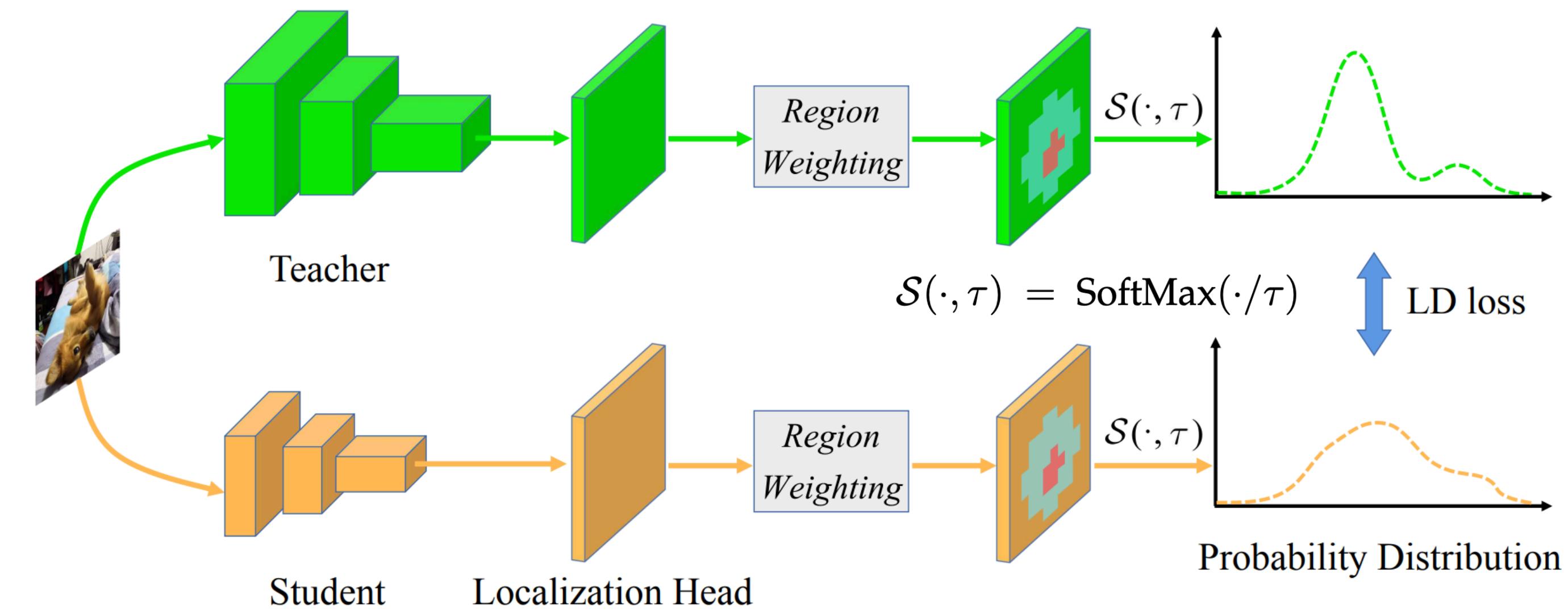
Convert bounding box regression to classification problem

Divide the y-axis into 6 bins & divide the x-axis into 6 bins



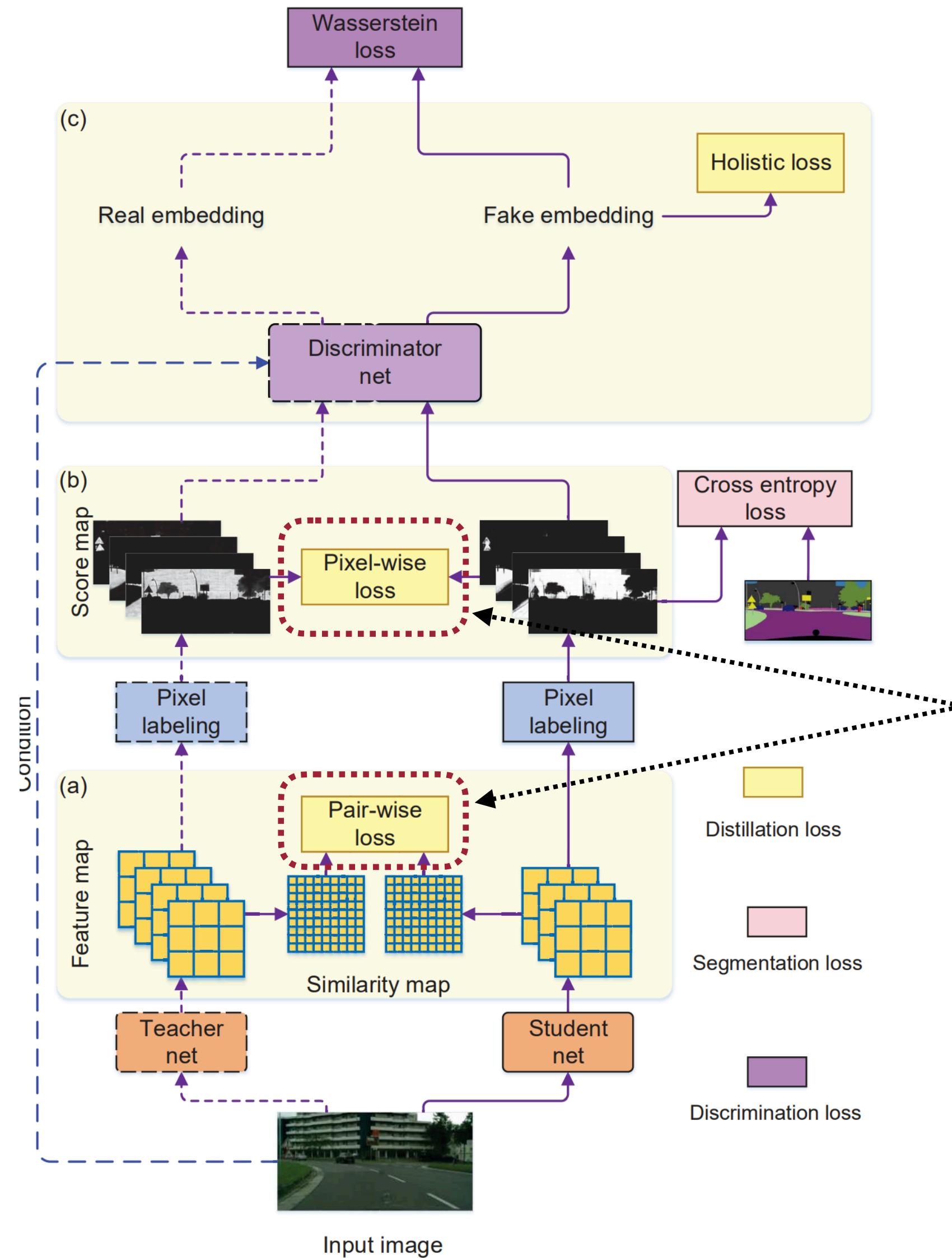
KD for Object Detection

Localization Distillation



- Calculate the distillation loss between two probability distributions predicted by the teacher and the student.

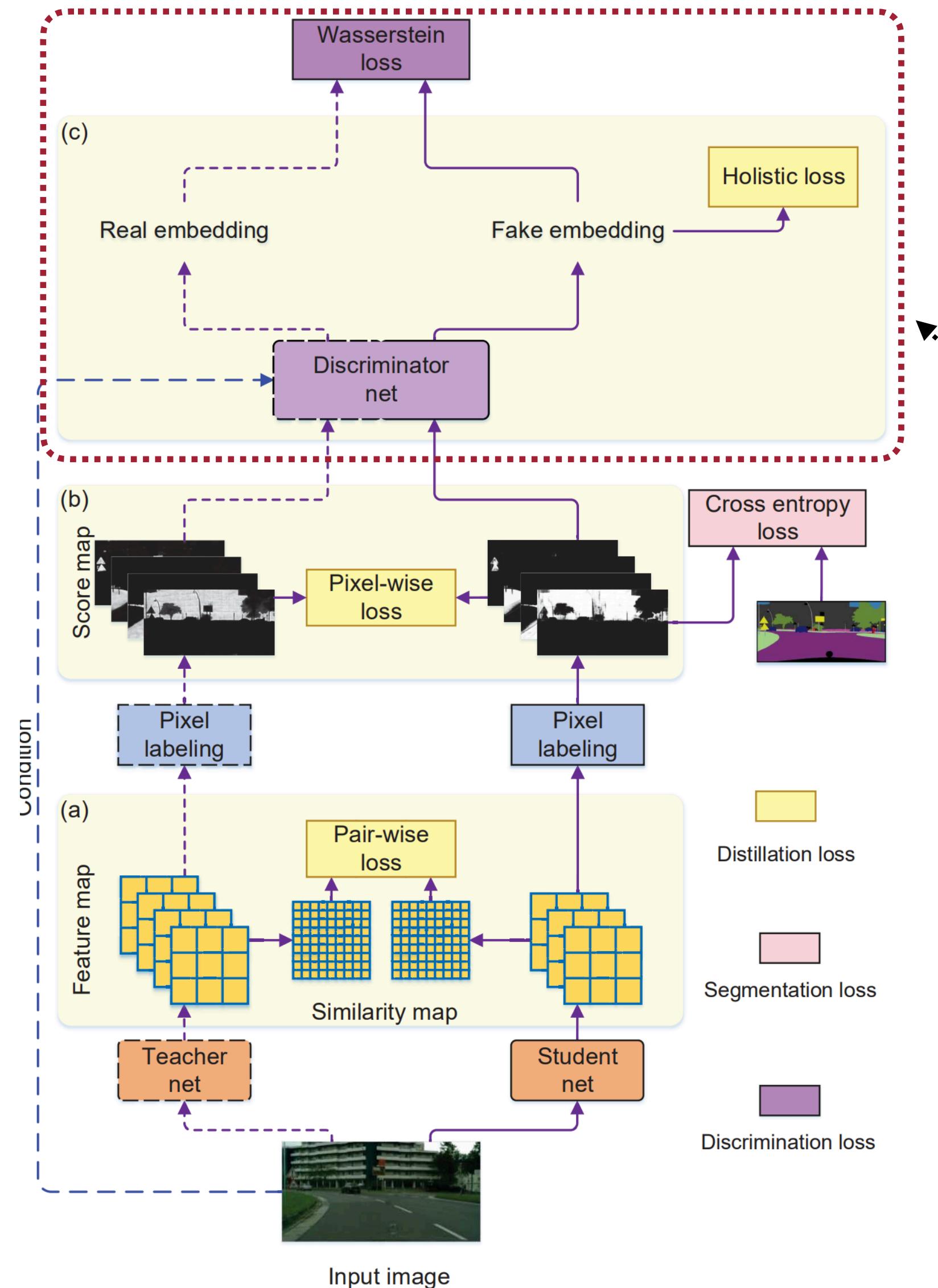
KD for Semantic Segmentation



Feature imitation similar to classification and detection

Structured Knowledge Distillation for Semantic Segmentation [Liu et al., CVPR 2019]

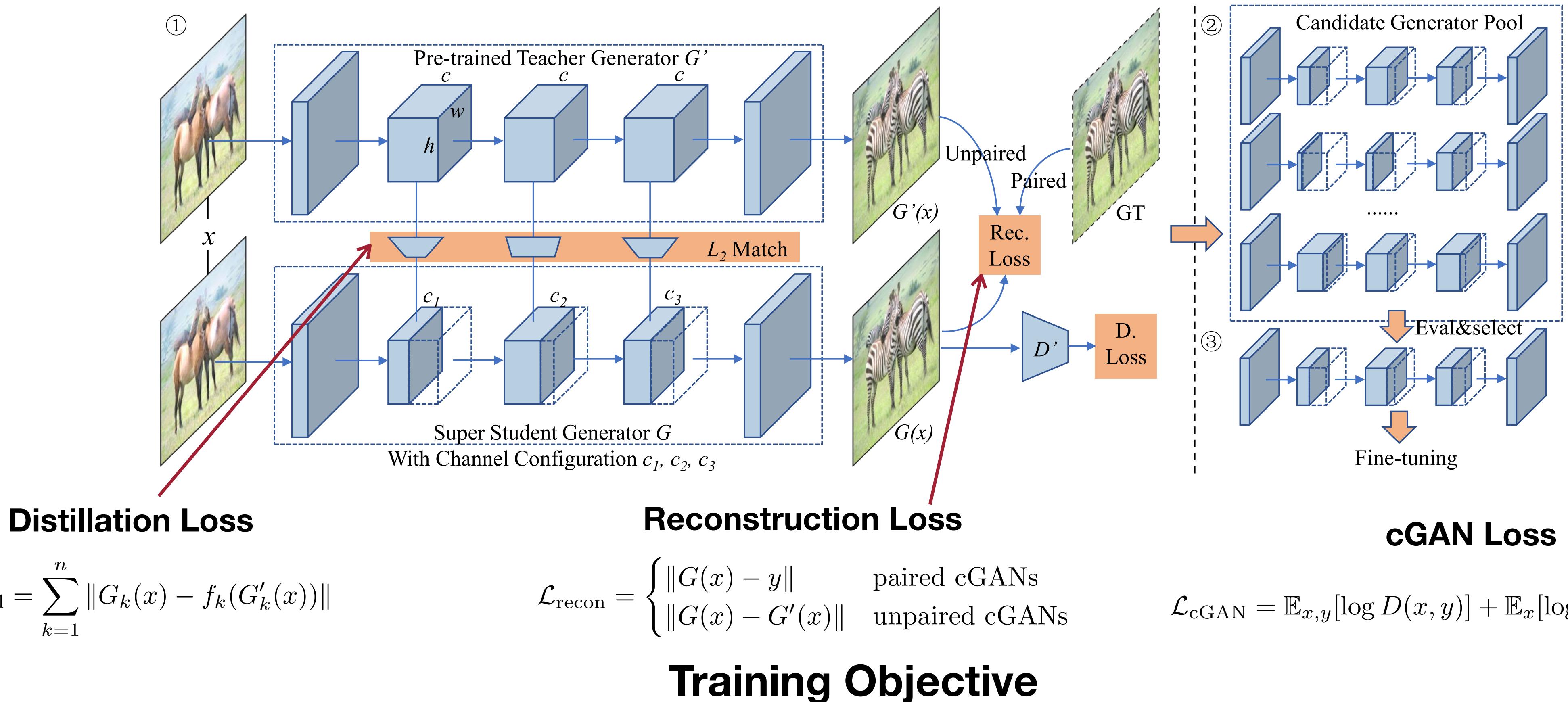
KD for Semantic Segmentation



Add a discriminator network to provide adversarial loss: the student is trained to fool the discriminator network

Structured Knowledge Distillation for Semantic Segmentation [Liu et al., CVPR 2019]

KD for GAN



GAN Compression: Efficient Architectures for Interactive Conditional GANs [Li et al., CVPR 2020]

Demos on Horse2zebra Dataset



Original CycleGAN; FLOPs: 56.8G; **FPS: 12.1**; FID: 61.5



GAN Compression; FLOPs: 3.50G (16.2x); **FPS: 40.0 (3.3x)**; FID: 53.6

Measured on NVIDIA **Jetson Xavier GPU**
Lower FID indicates better Performance.



Interactive image editing demo



Original CycleGAN; MACs: 56.8G; FPS: 1.6; FID: 24.2

Measured on NVIDIA **Jetson Nano GPU**
Lower FID indicates better Performance.



GAN Compression: Efficient Architectures for Interactive Conditional GANs [Li et al., CVPR 2020]

Interactive image editing demo



Original CycleGAN; MACs: 56.8G; **FPS: 1.6**; FID: 24.2



GAN Compression; MACs: 4.81G (11.8x); **FPS: 3.9 (2.5x)**; FID: 26.6

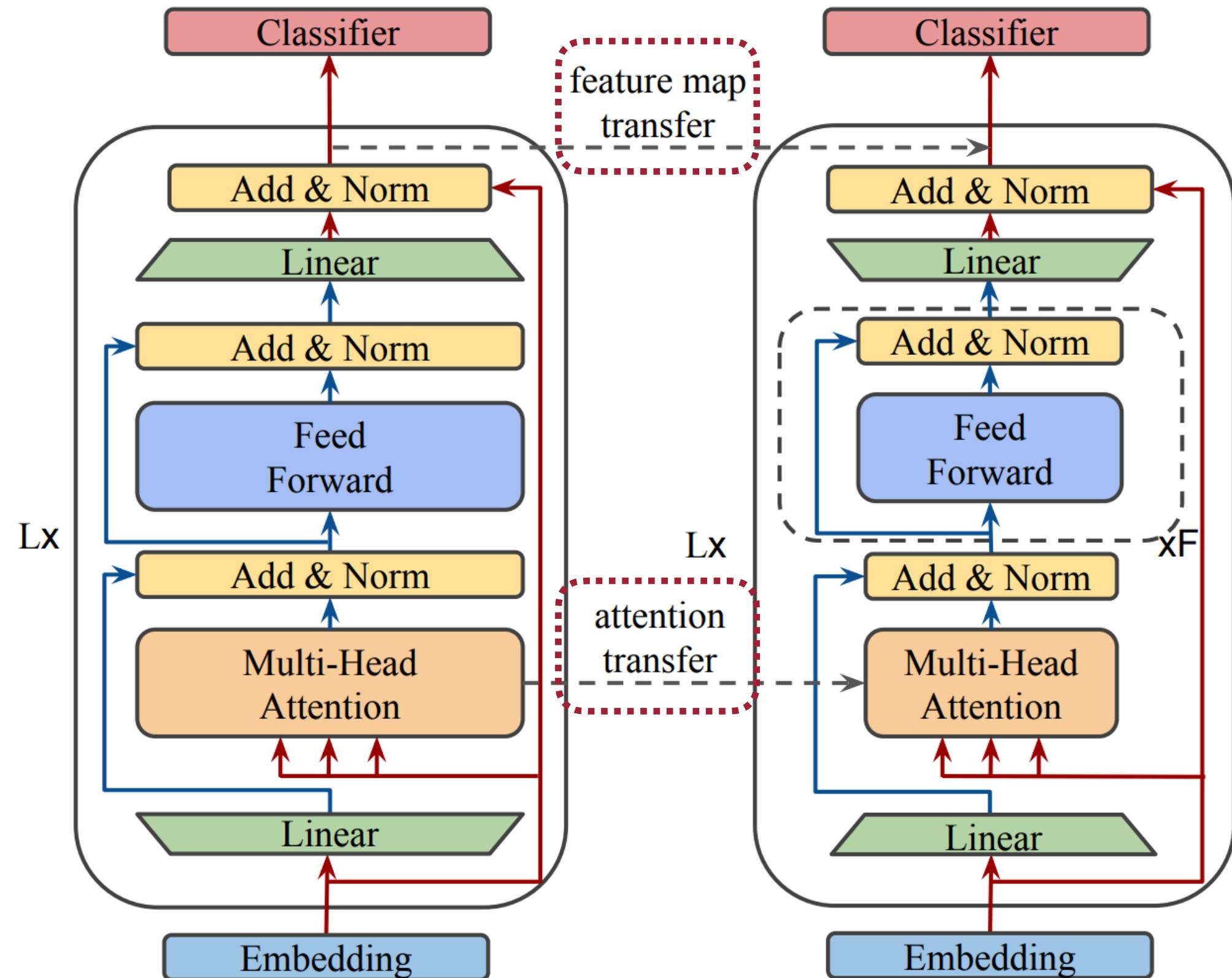
Measured on NVIDIA **Jetson Nano GPU**
Lower FID indicates better Performance.



GAN Compression: Efficient Architectures for Interactive Conditional GANs [Li et al., CVPR 2020]

KD for NLP

Attention Transfer

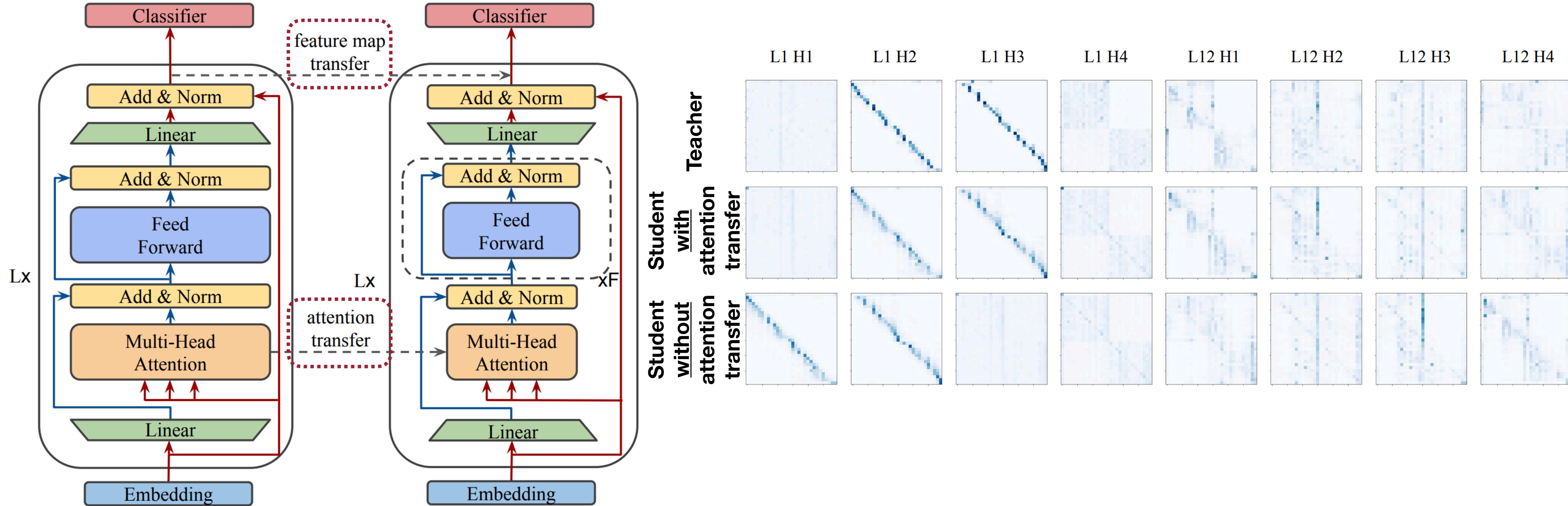


- In addition to feature imitation, the student model is trained to mimic teacher model's attention maps.

MobileBERT: a Compact Task-Agnostic BERT for Resource-Limited Devices [Sun et al., ACL 2020]

KD for NLP

Attention Transfer



- In addition to feature imitation, the student model is trained to mimic teacher model's attention maps.

MobileBERT: a Compact Task-Agnostic BERT for Resource-Limited Devices [Sun et al., ACL 2020]

Knowledge Distillation

Today we will:

1. What is knowledge distillation;
2. What to match;
3. Self and online distillation;
4. Distillation for different tasks;
5. Network augmentation.



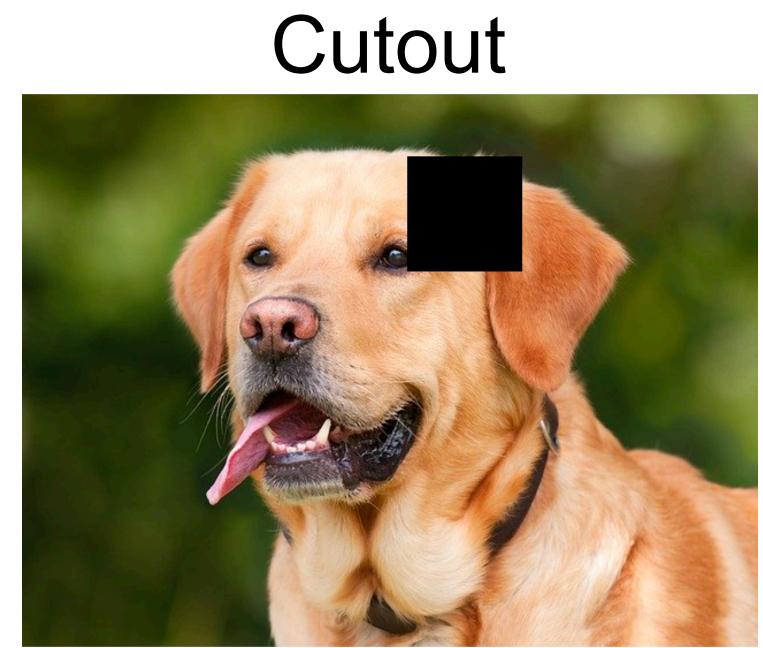
Conventional Approach

Data augmentation during training to avoid overfitting



Conventional Approach

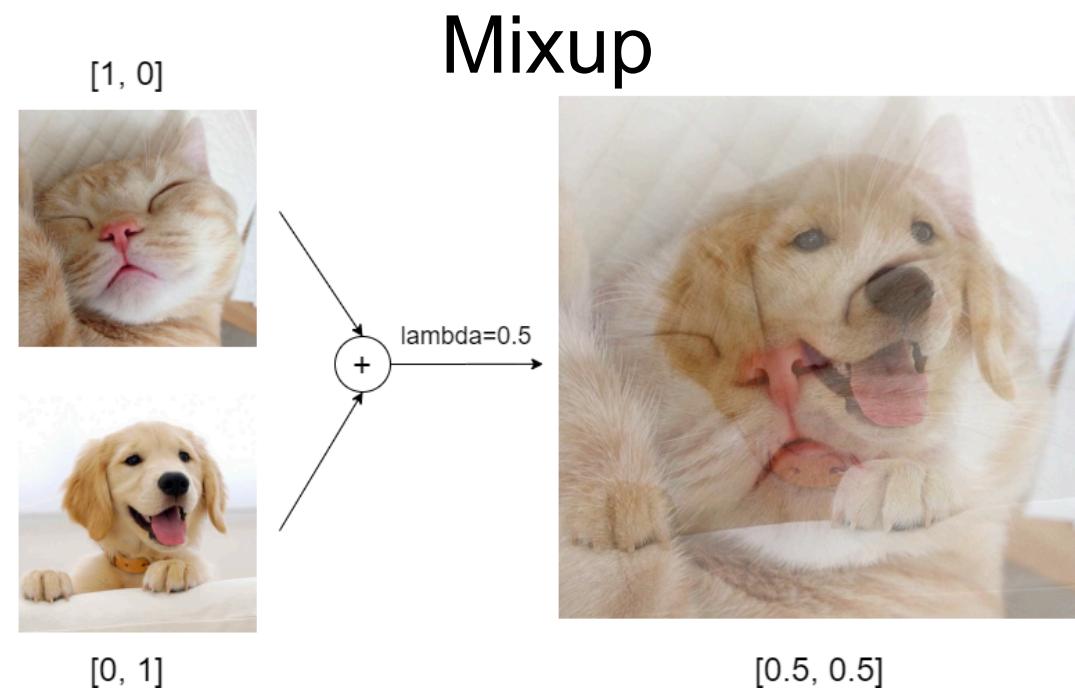
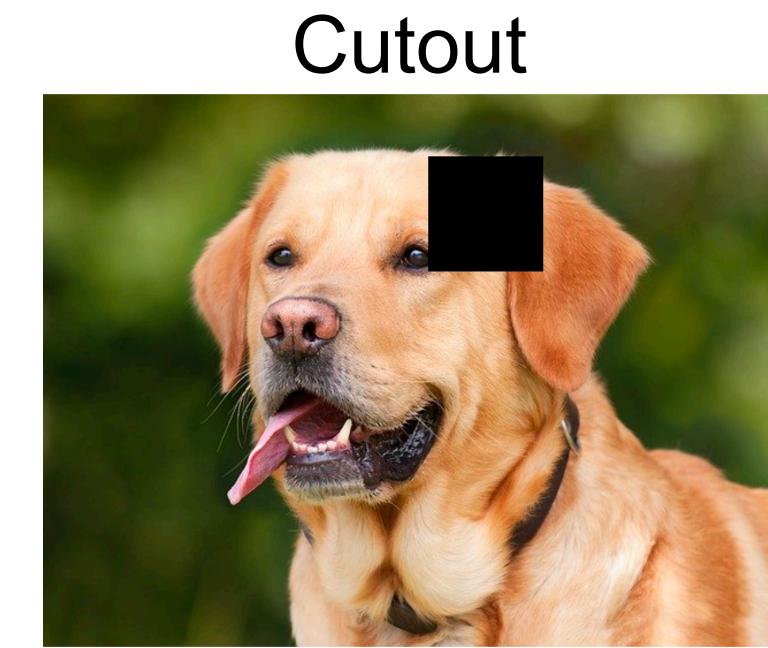
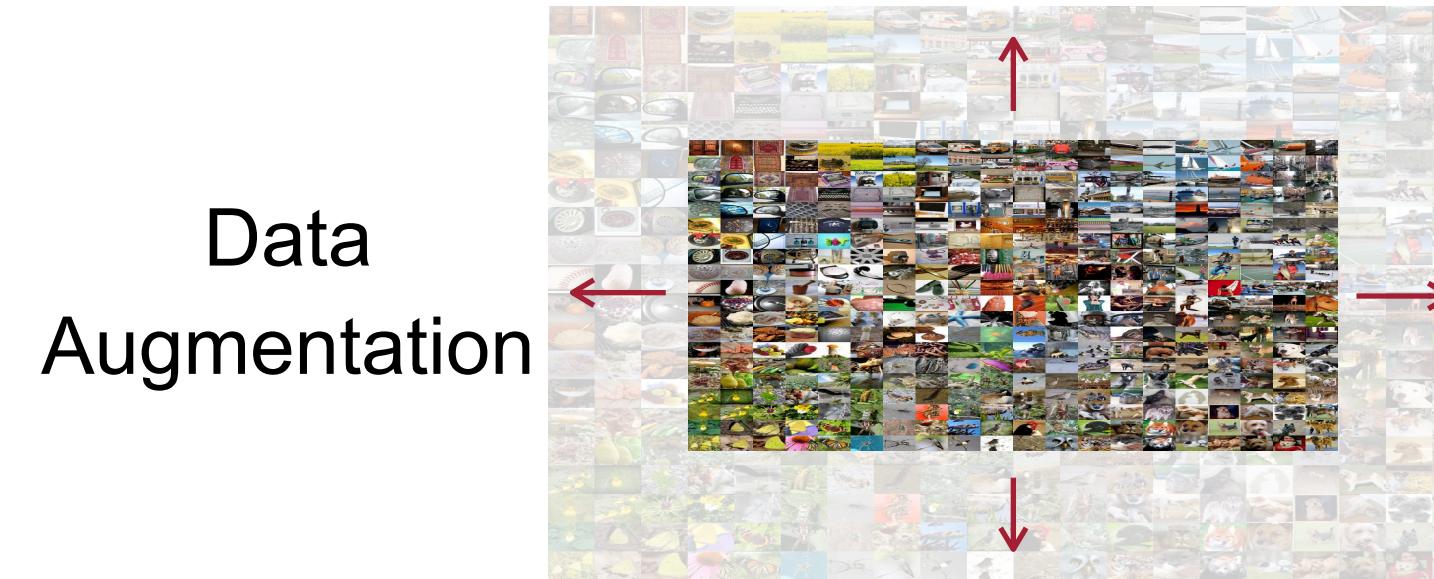
Data augmentation during training to avoid overfitting



Improved Regularization of Convolutional Neural Networks with Cutout [DeVries et al., arXiv 2017]

Conventional Approach

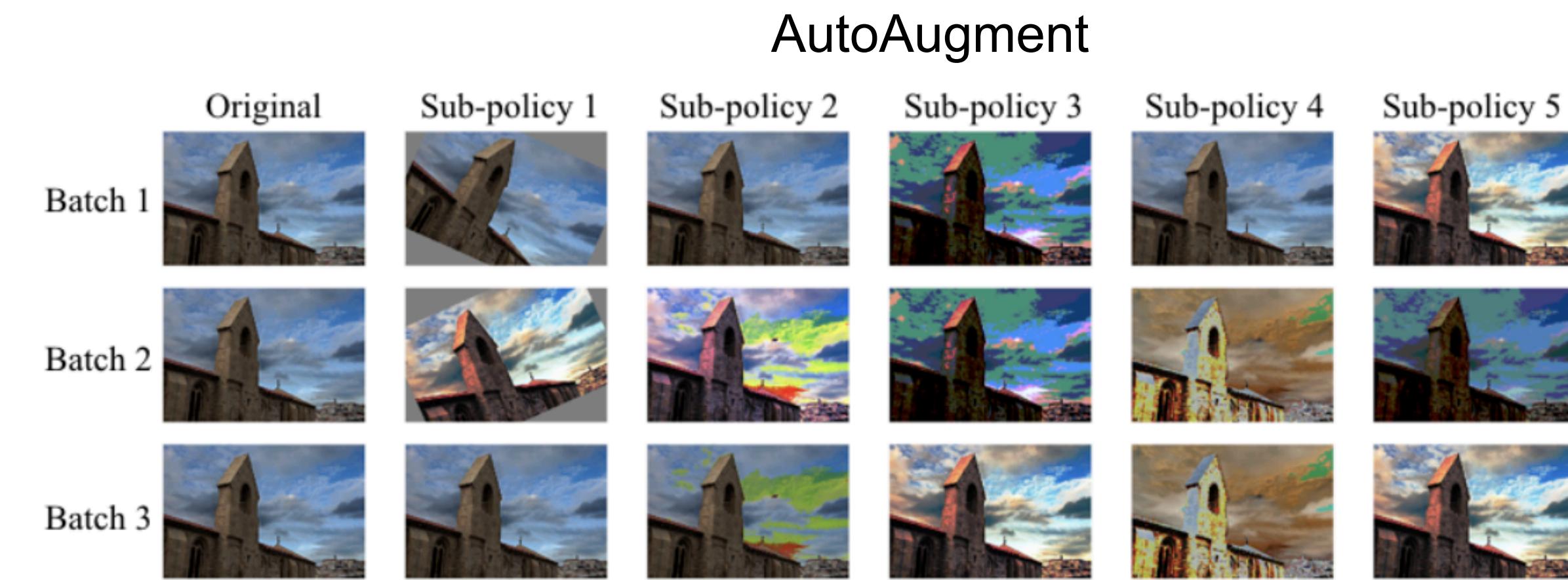
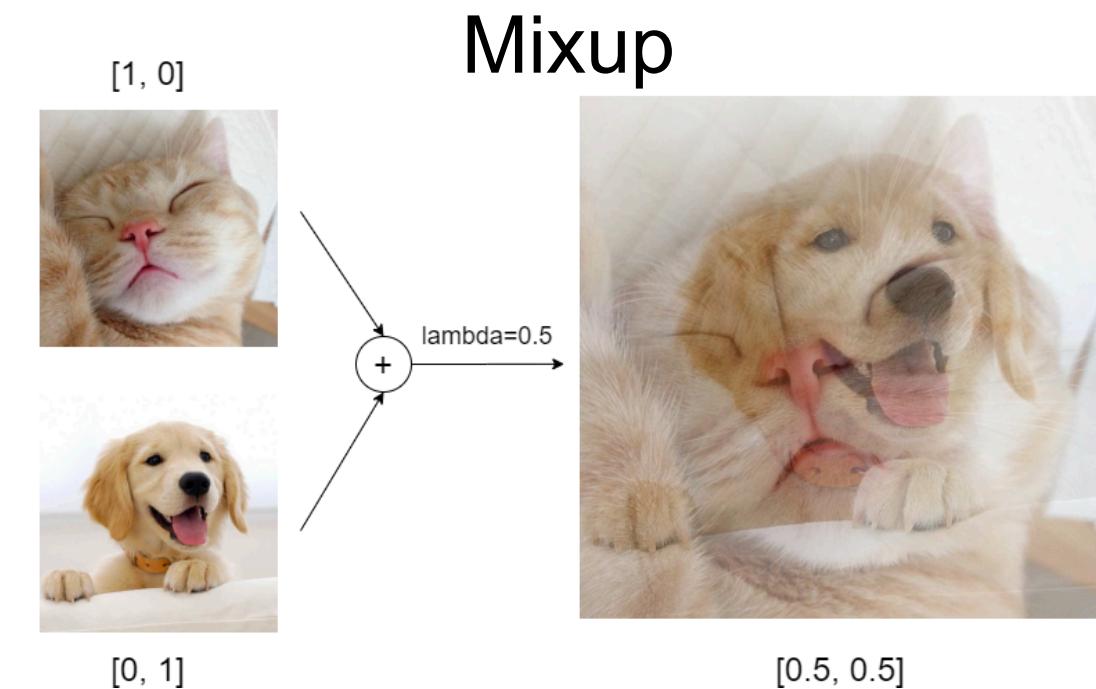
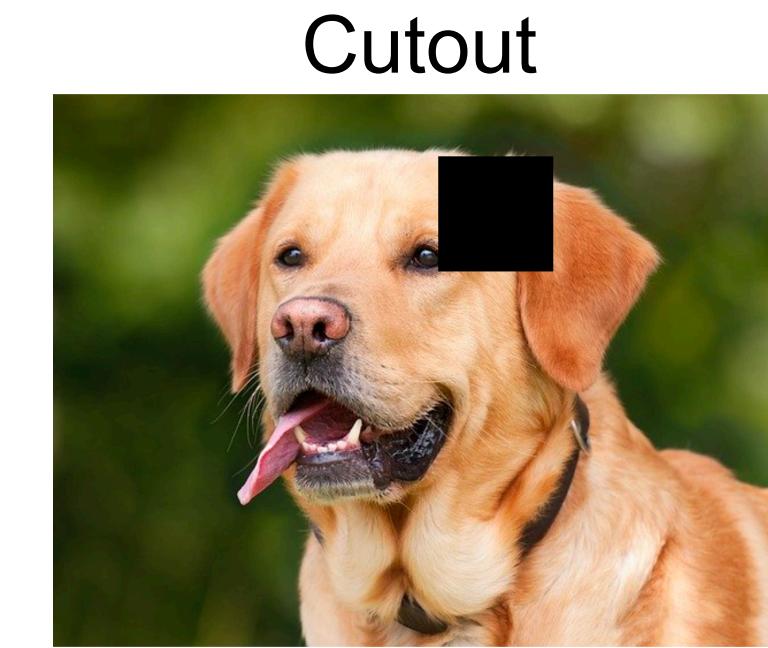
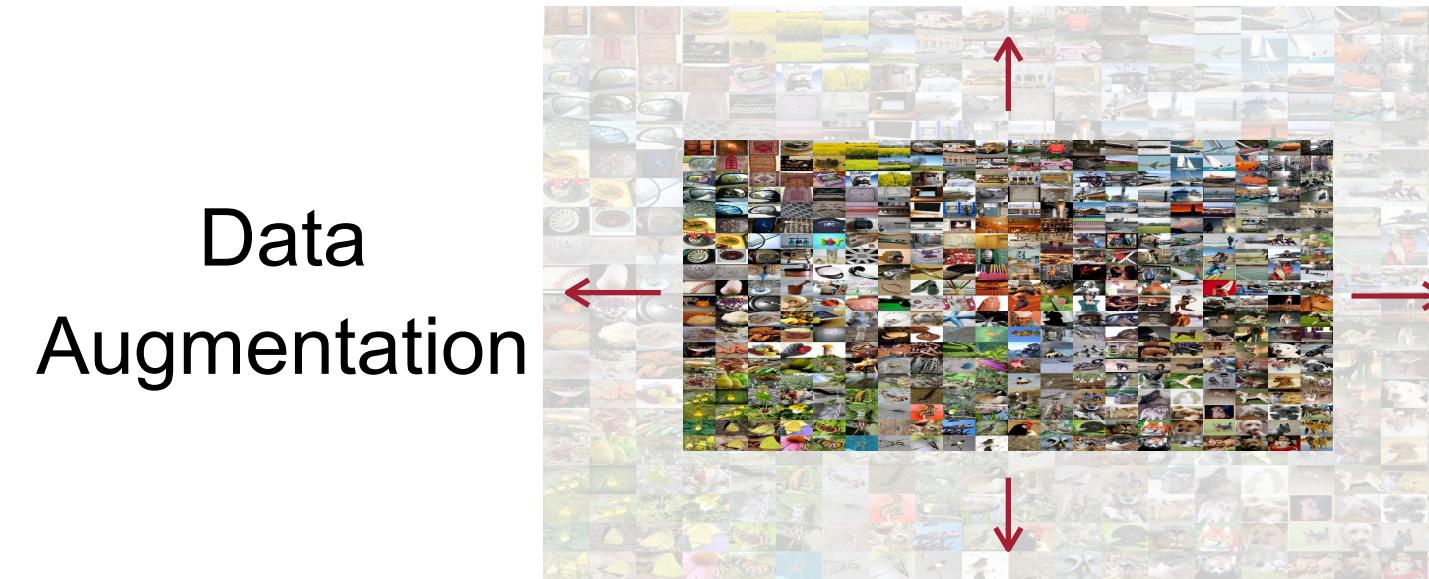
Data augmentation during training to avoid overfitting



mixup: Beyond Empirical Risk Minimization [Zhang et al., ICLR 2018]

Conventional Approach

Data augmentation during training to avoid overfitting



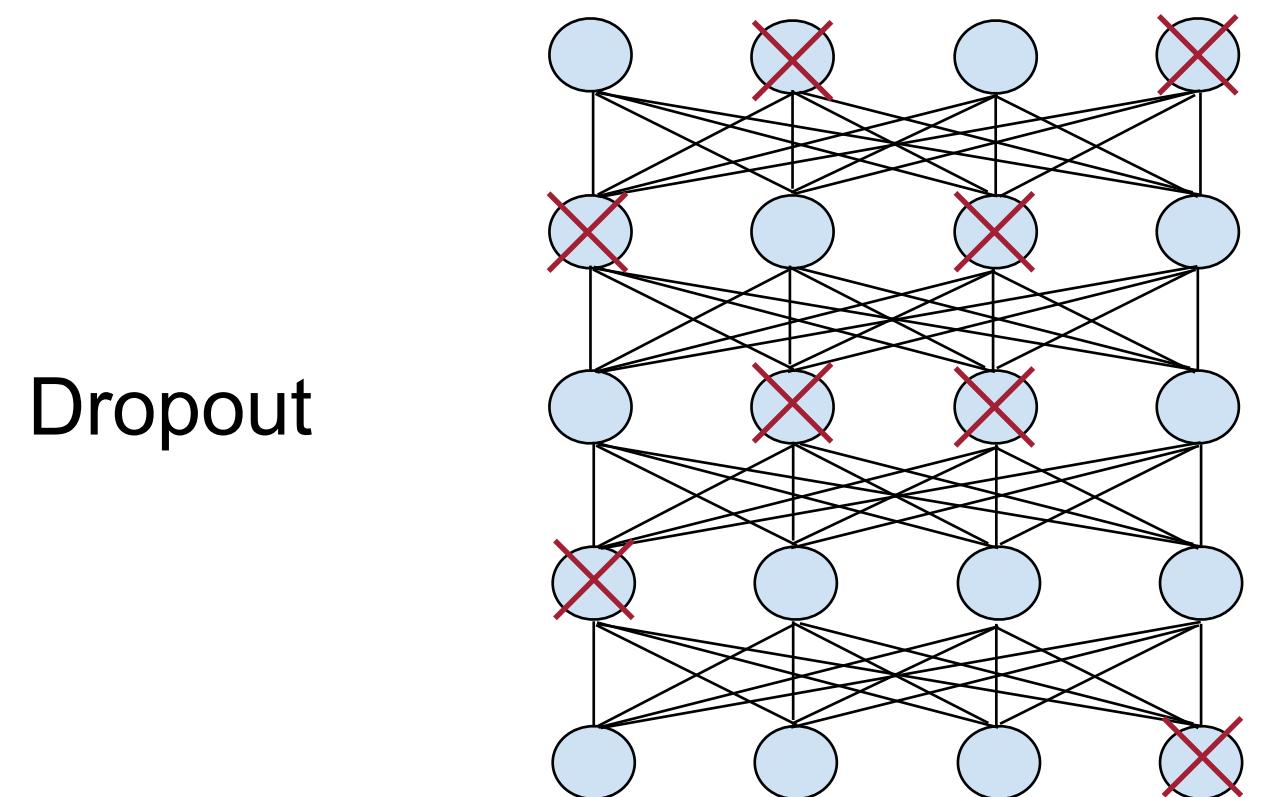
AutoAugment: Learning Augmentation Policies from Data [Cubuk et al., CVPR 2019]

Conventional Approach

Dropout during training to avoid overfitting



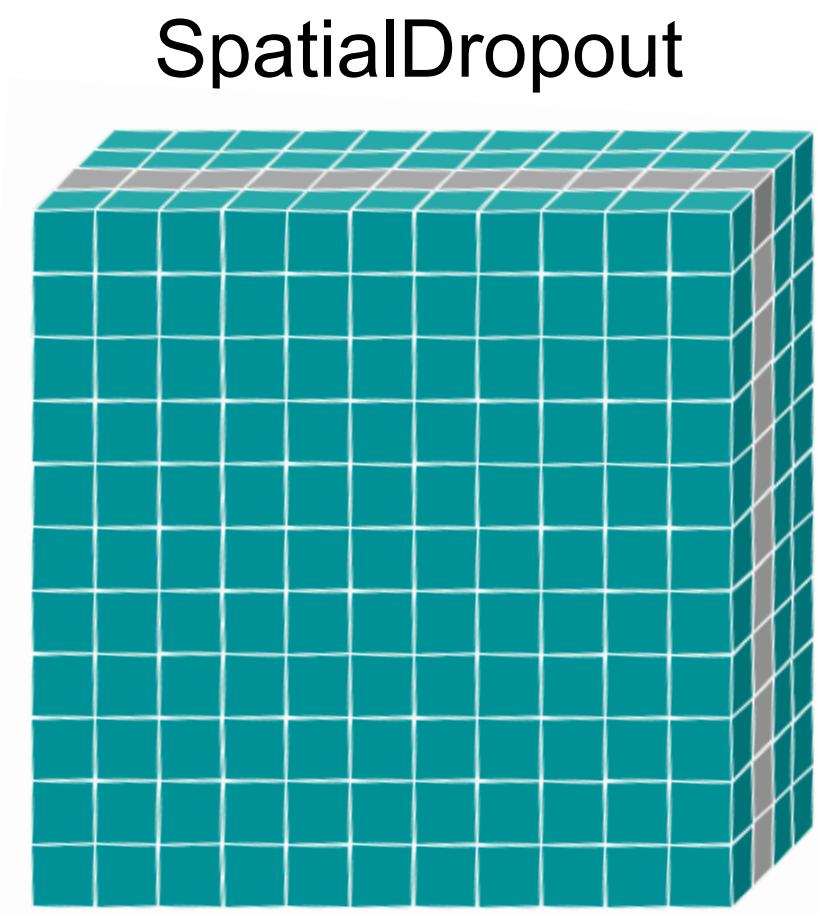
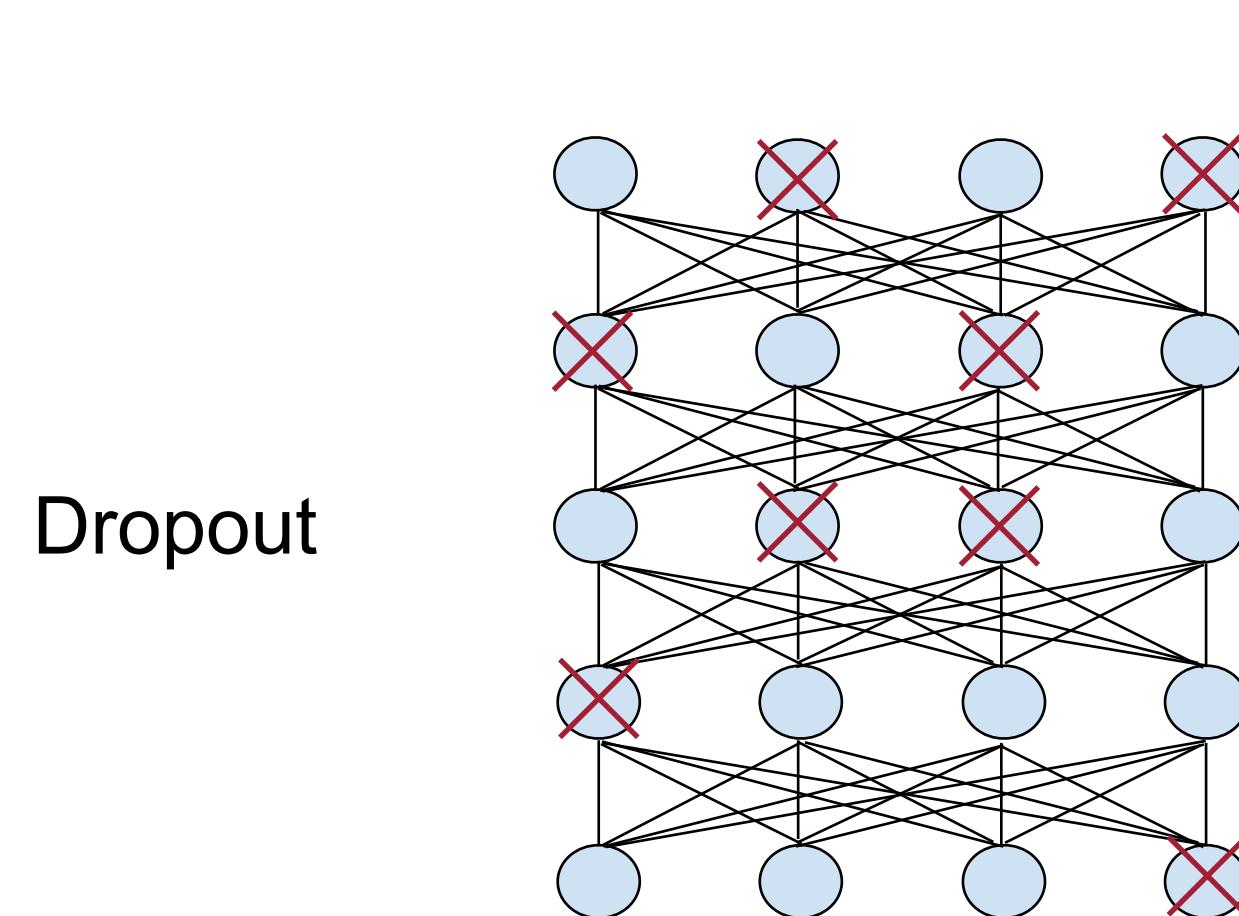
Data
Augmentation



Dropout

Conventional Approach

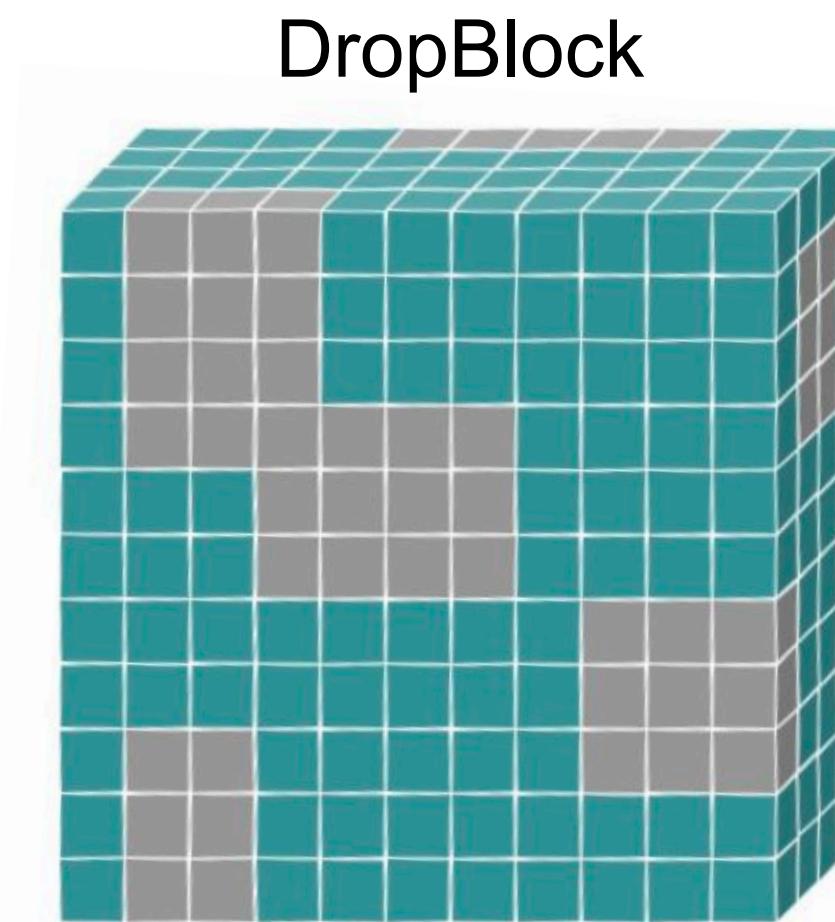
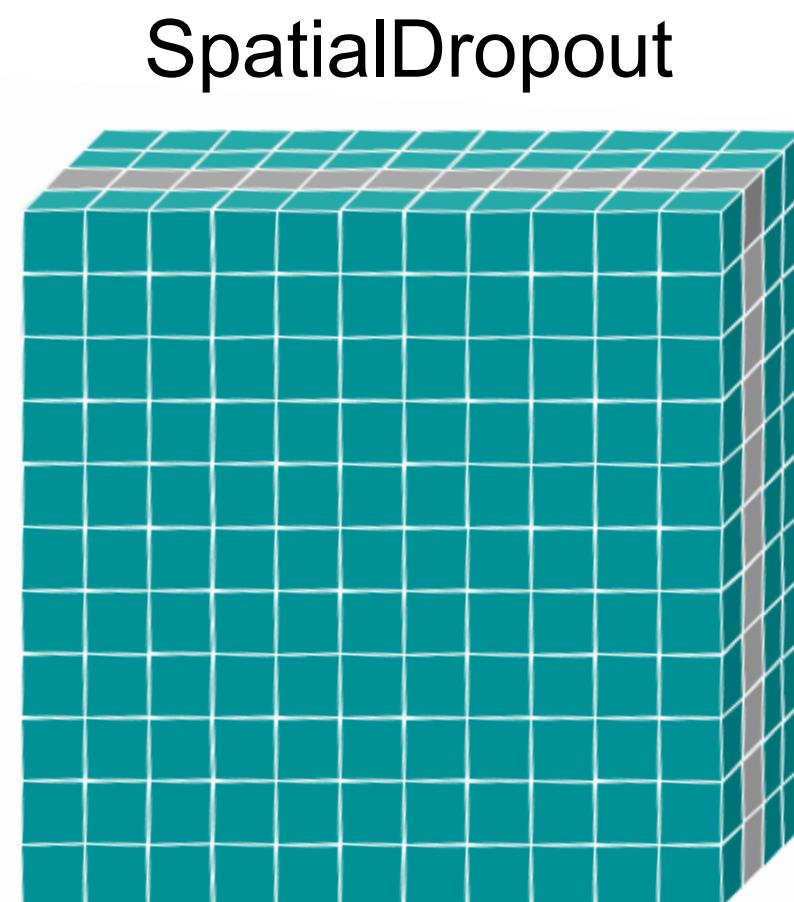
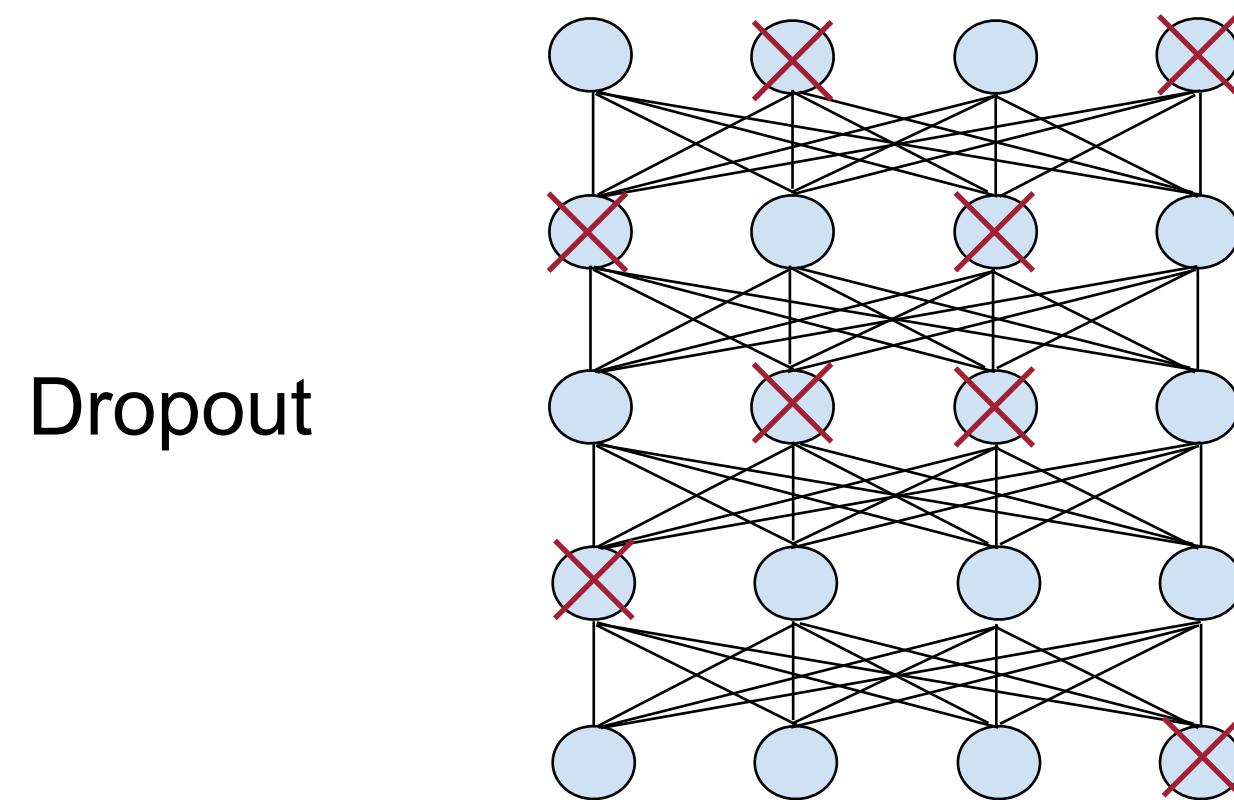
Dropout during training to avoid overfitting



Efficient Object Localization Using Convolutional Networks [Tompson et al., CVPR 2015]

Conventional Approach

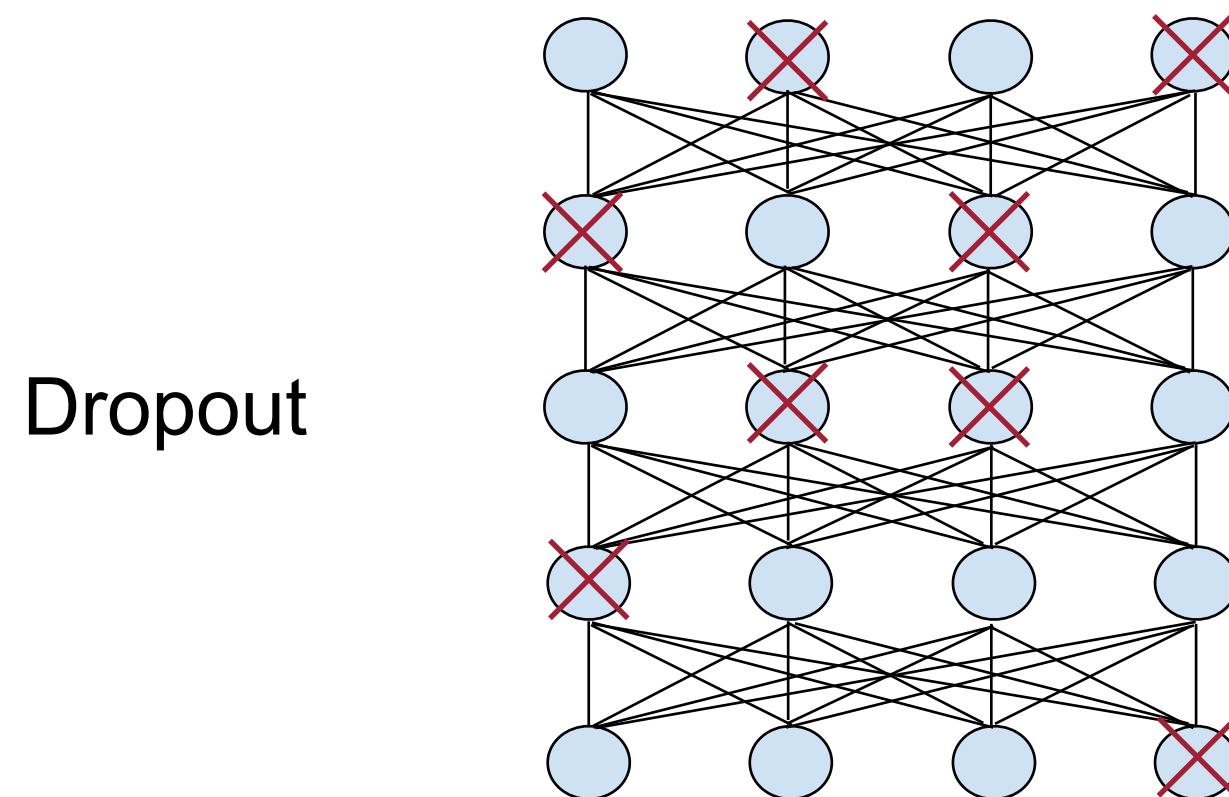
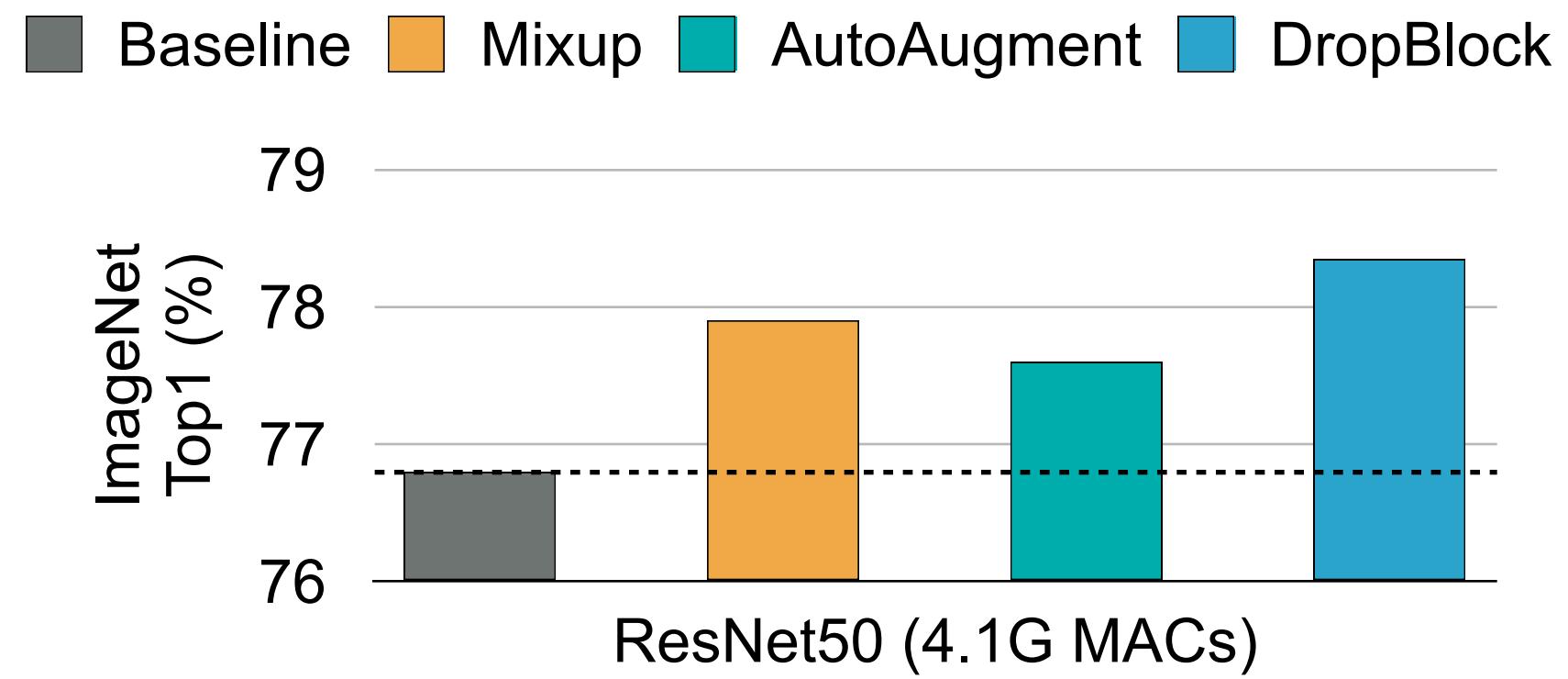
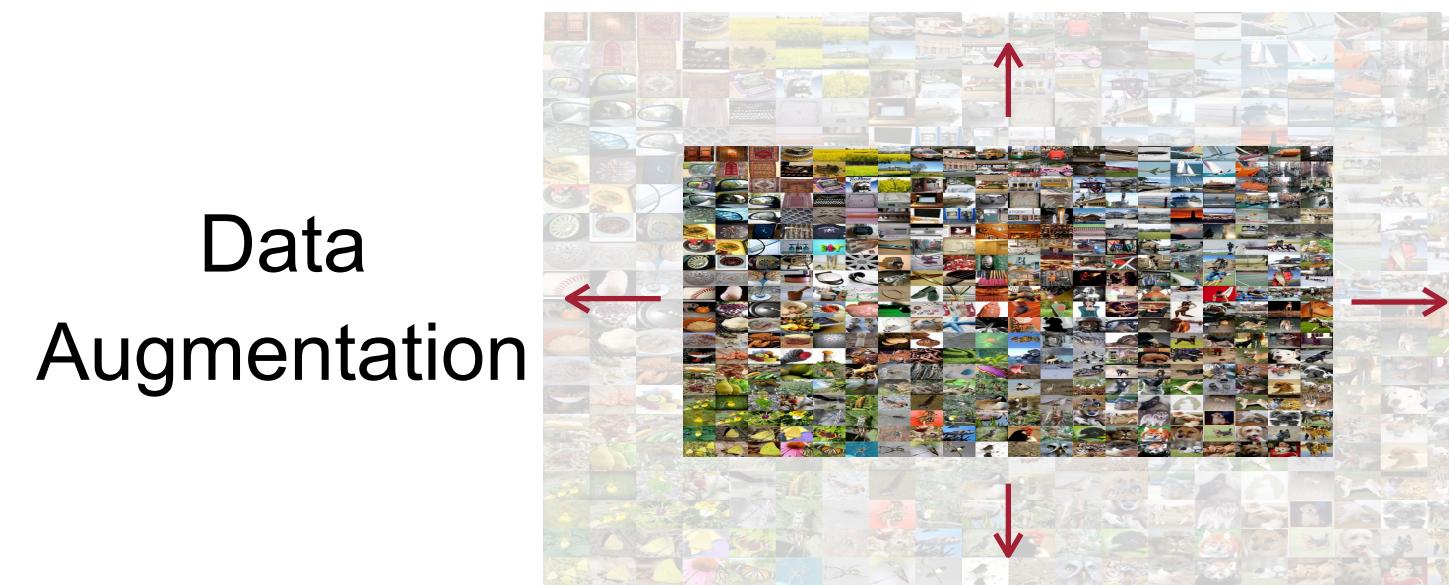
Dropout during training to avoid overfitting



DropBlock: A regularization method for convolutional networks [Ghiasi et al., NeurIPS 2018]

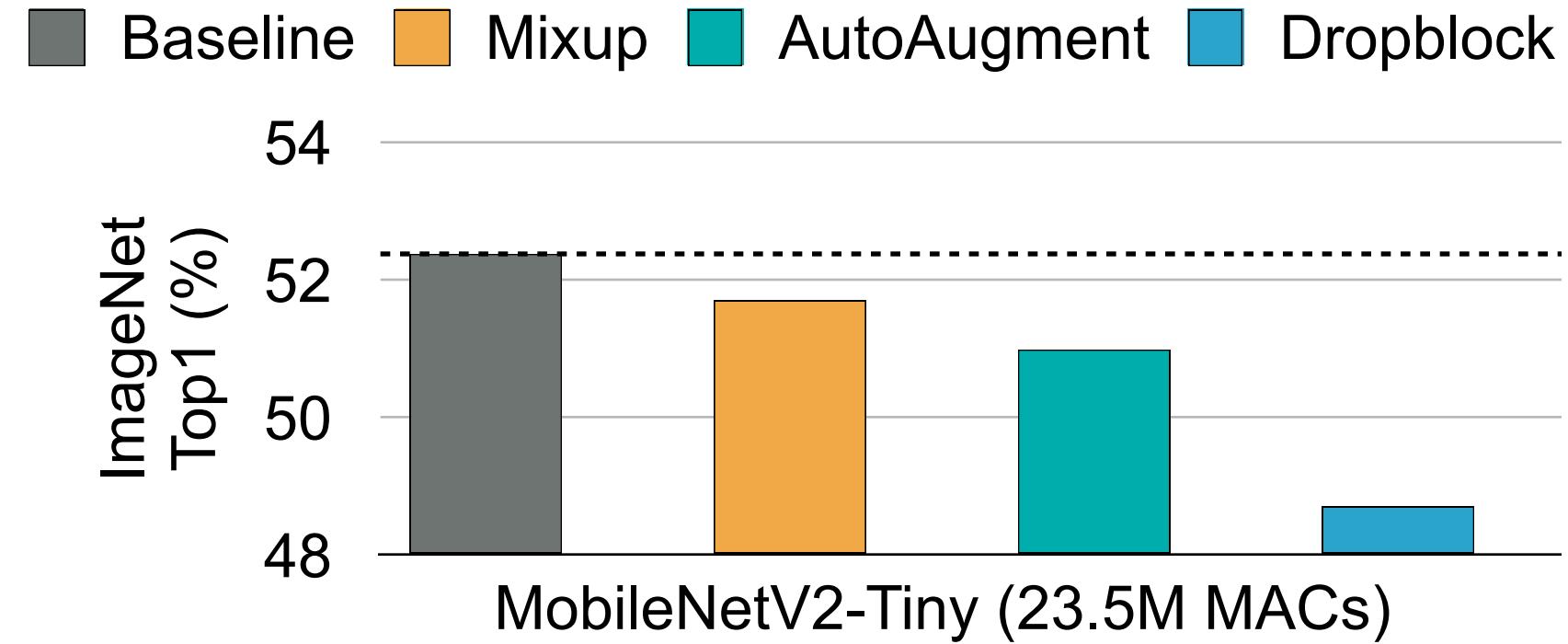
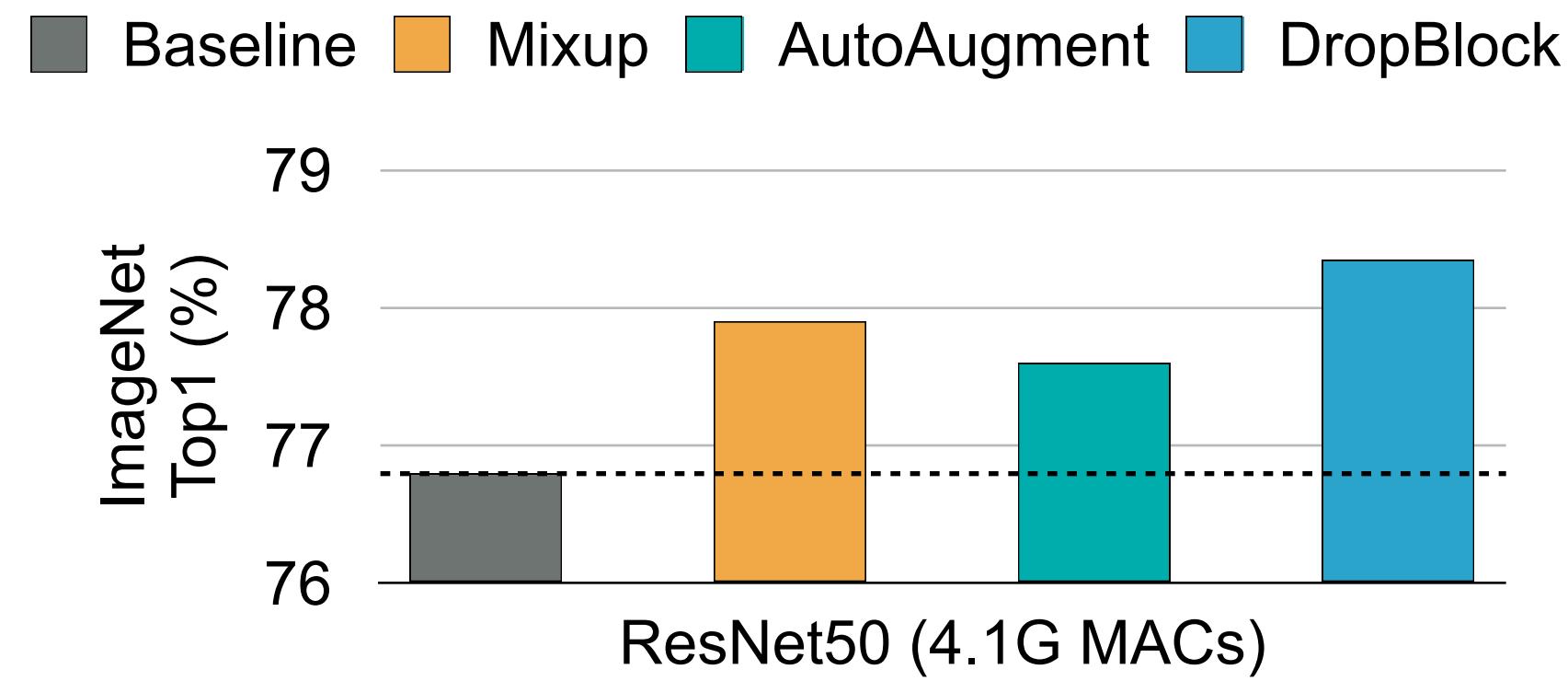
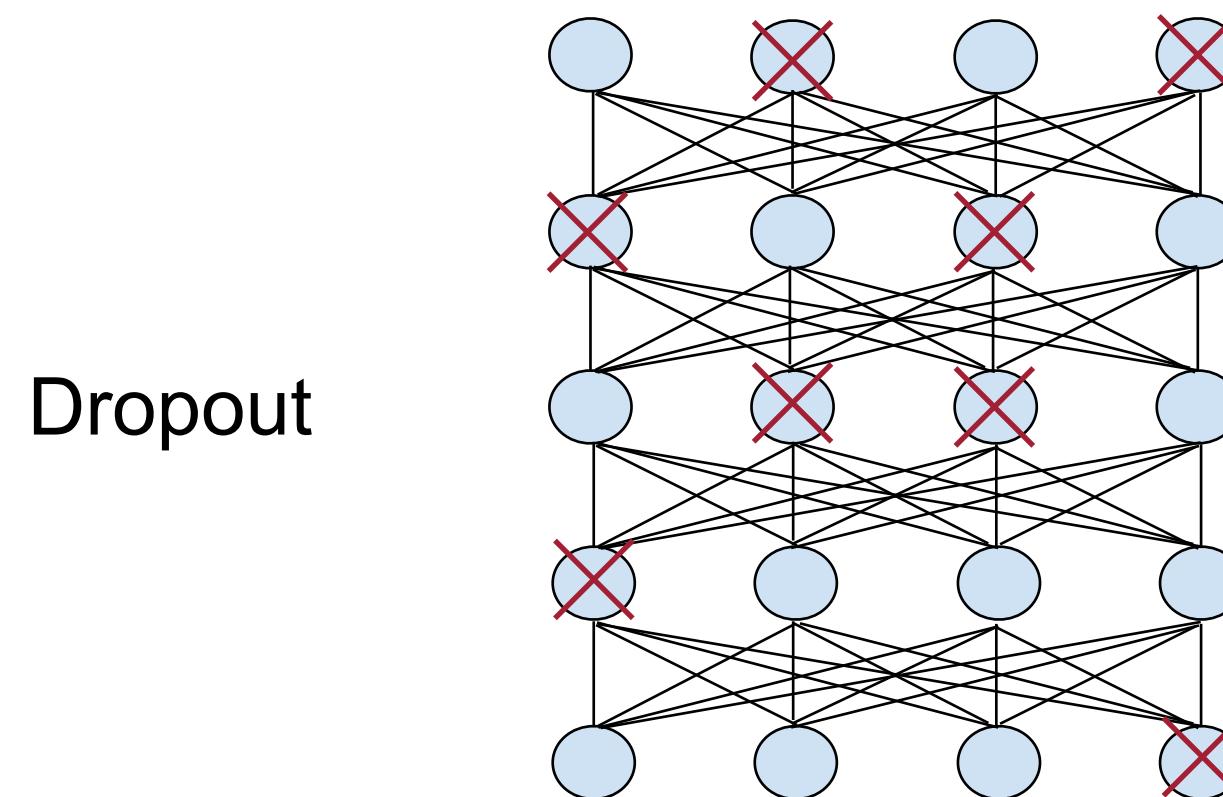
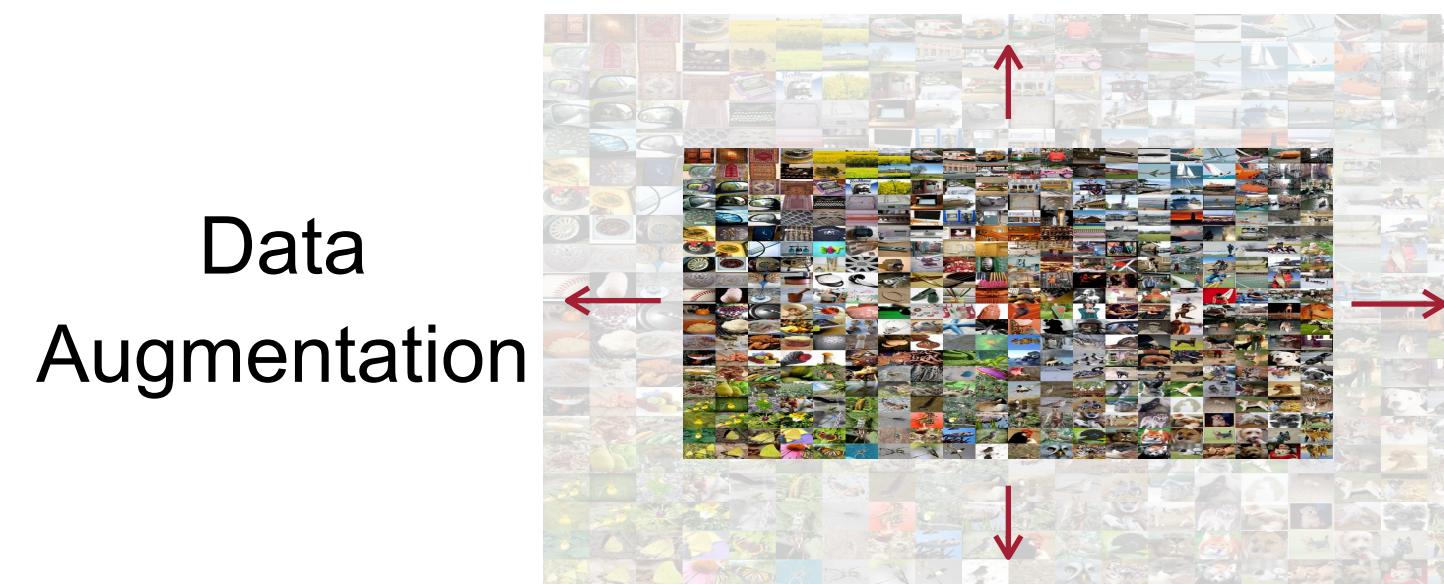
Conventional Approach

Dropout/Data augmentation improves large neural networks' performance



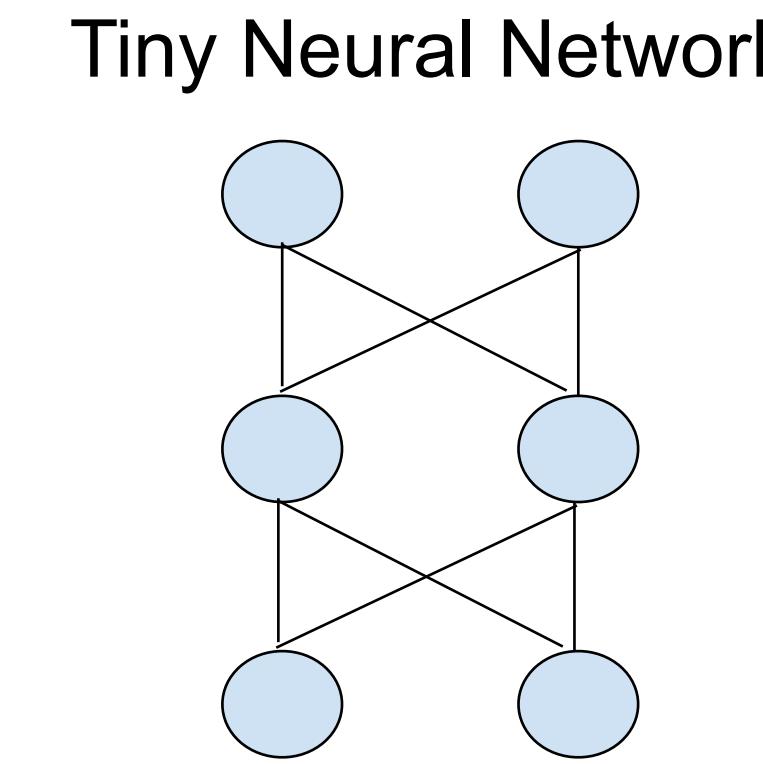
Conventional Approach

Dropout/Data augmentation hurts tiny neural networks' performance

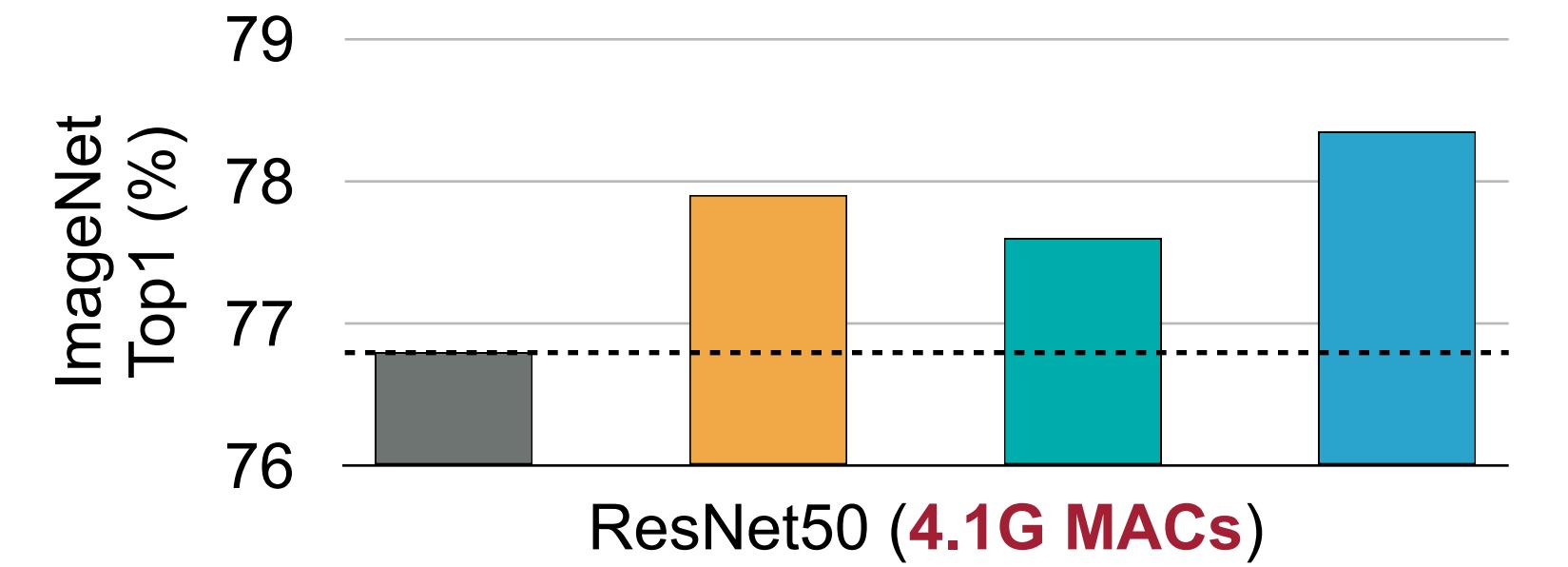


Conventional Approach

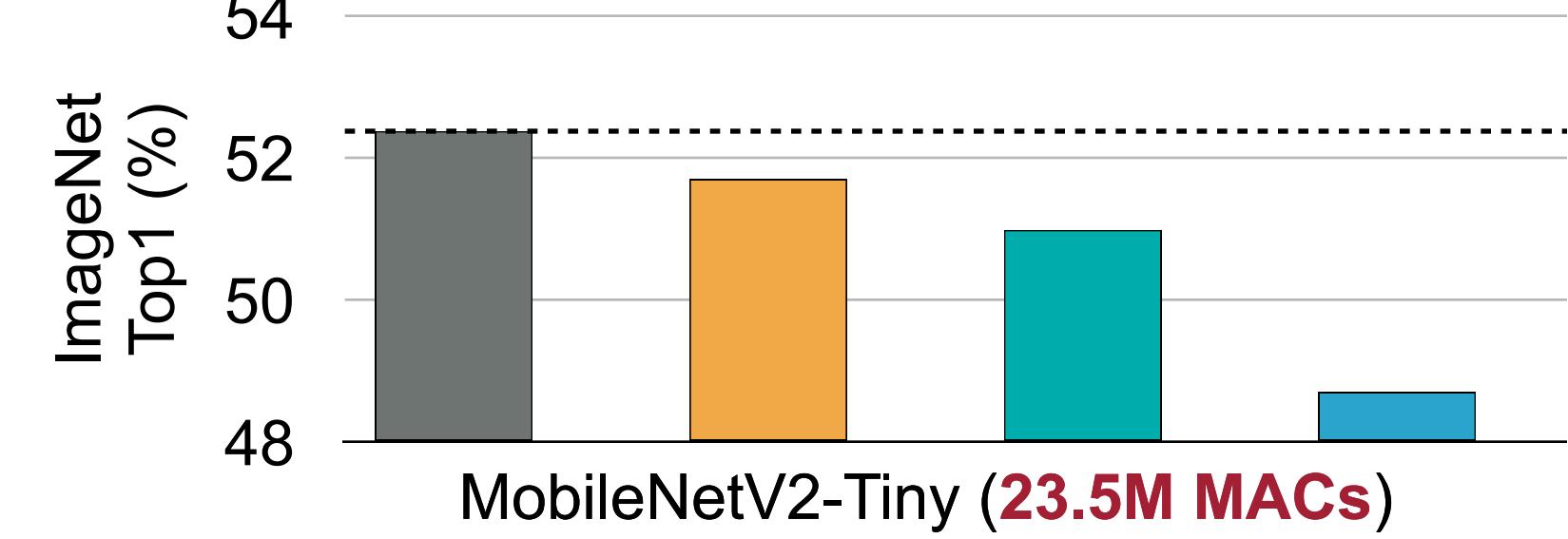
Tiny neural network lacks capacity



■ Baseline ■ Mixup ■ AutoAugment ■ DropBlock

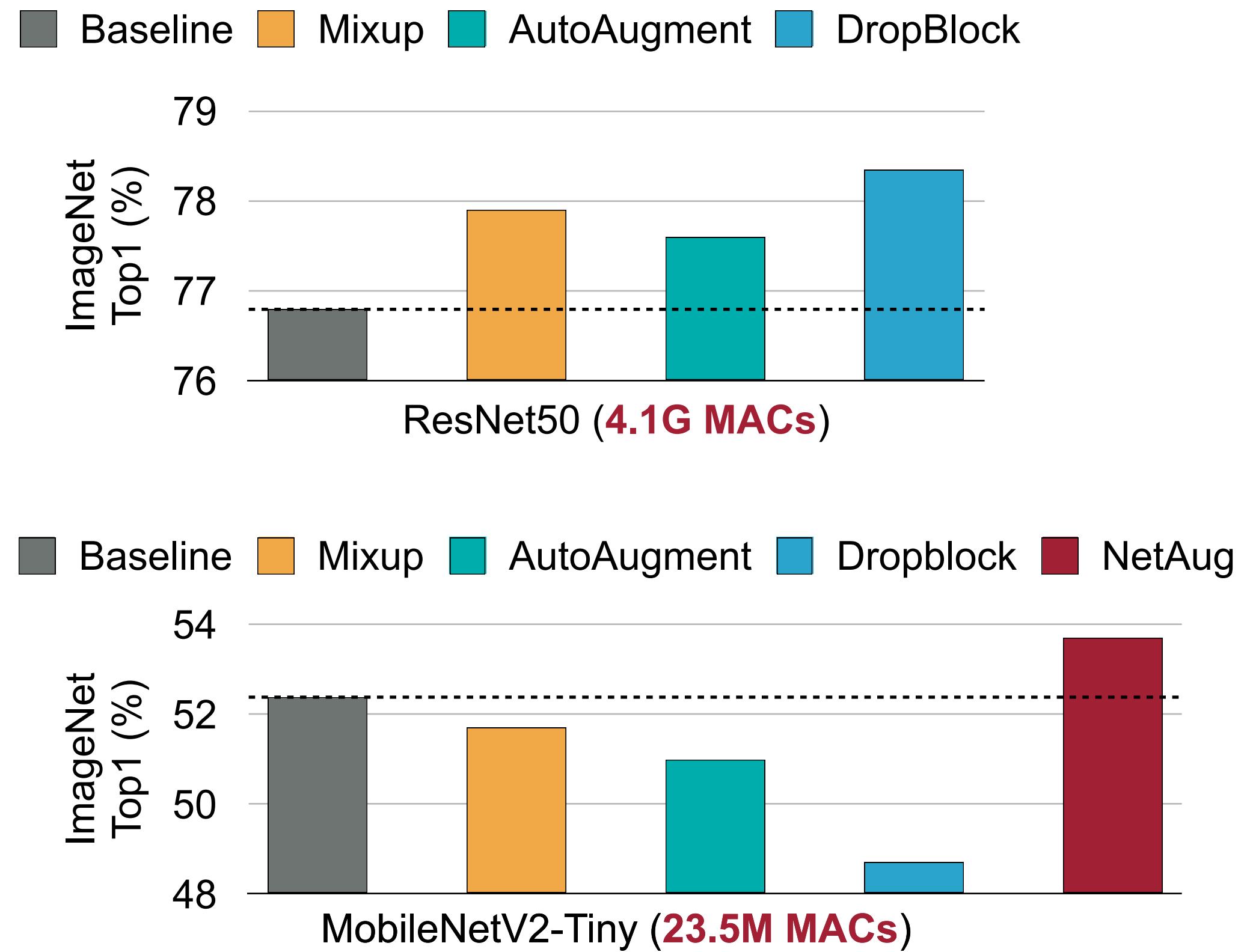
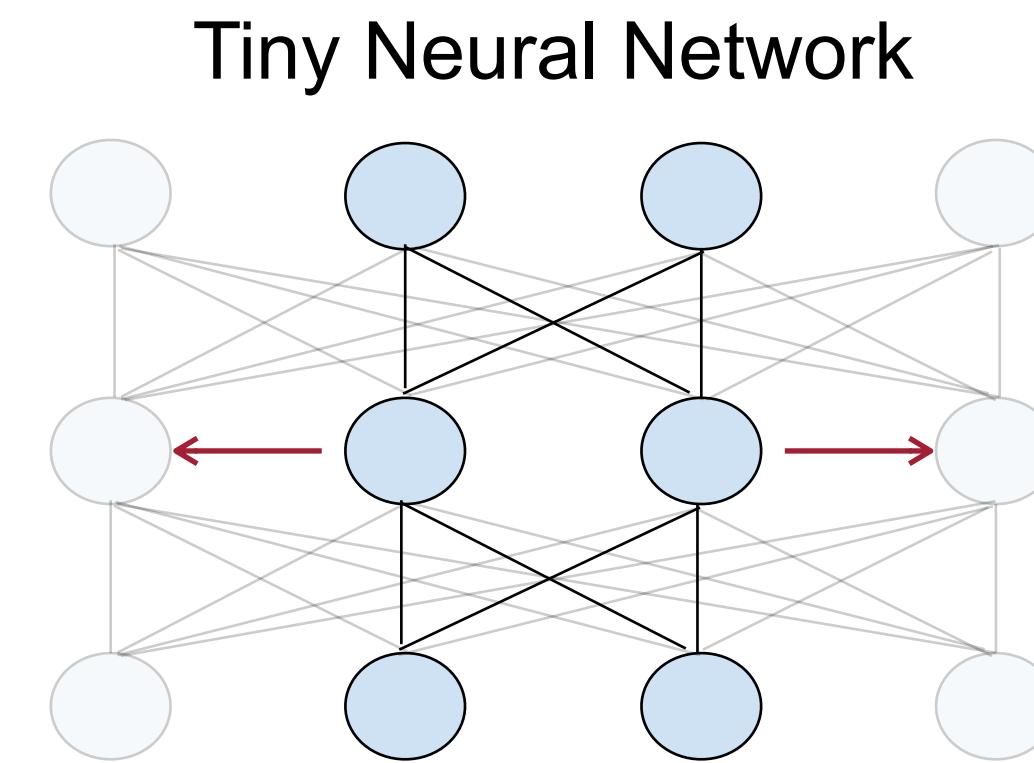


■ Baseline ■ Mixup ■ AutoAugment ■ Dropblock



Network Augmentation

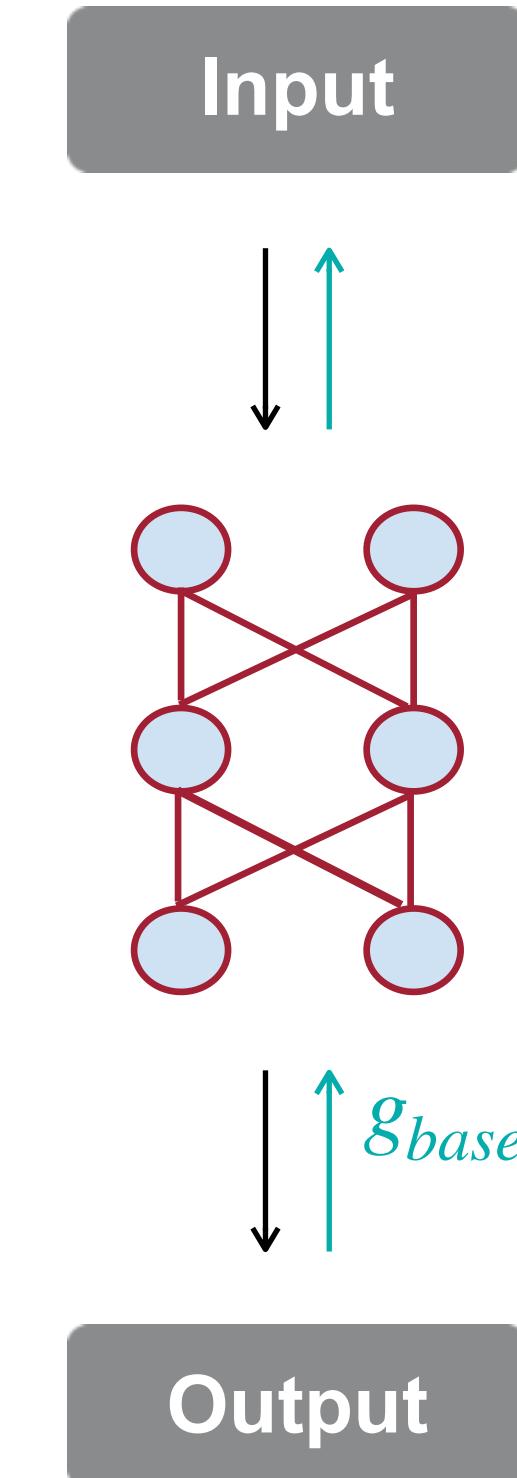
Augment the model to get extra supervision during training for tiny models



Network Augmentation for Tiny Deep Learning [Cai et al., ICLR 2022]

NetAug: Training Process

Augment the model to get extra supervision during training for tiny models

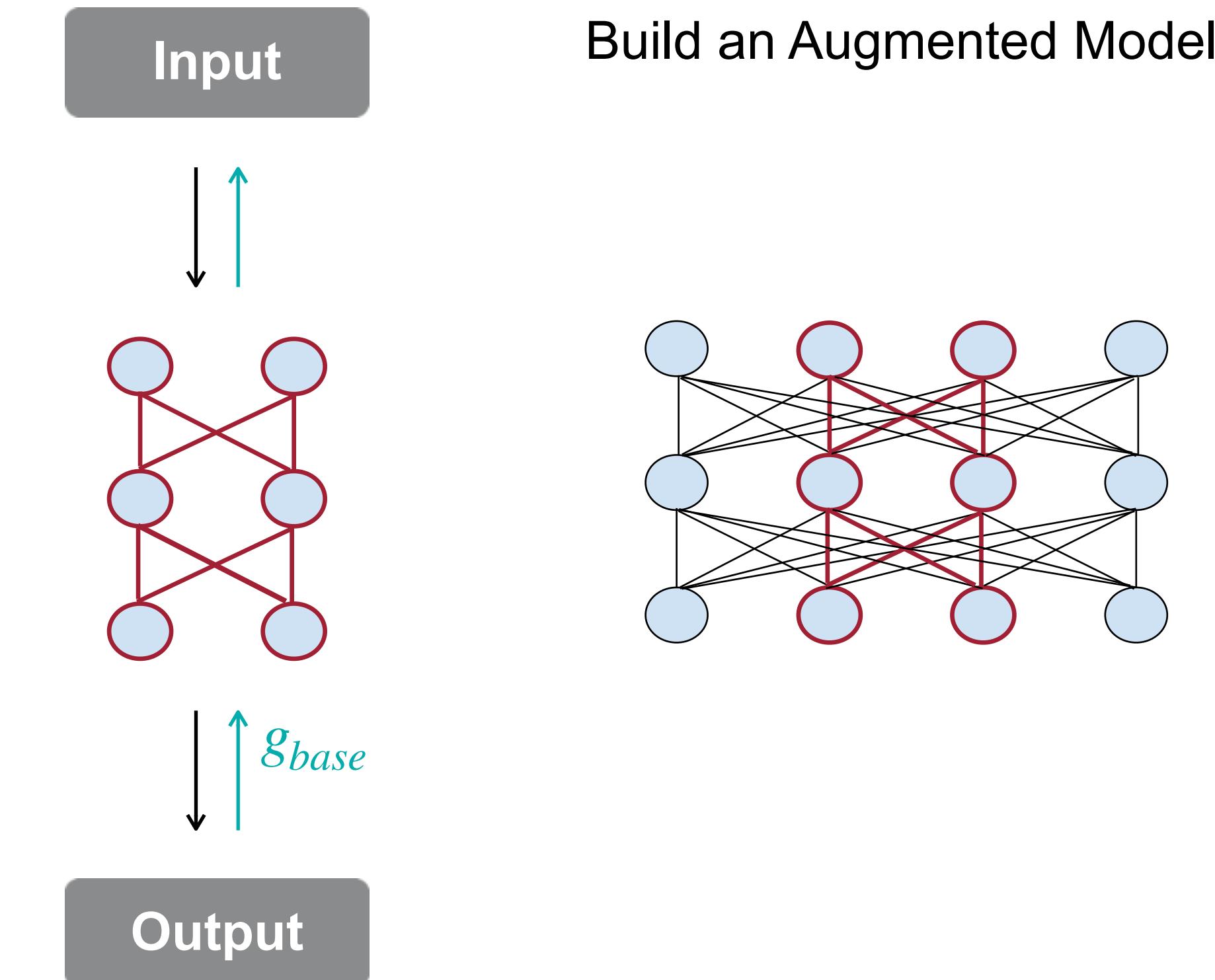


$$\mathcal{L}_{\text{aug}} = \underbrace{\mathcal{L}(W_{\text{base}})}_{\text{base supervision}}$$

Network Augmentation for Tiny Deep Learning [Cai et al., ICLR 2022]

NetAug: Training Process

Augment the model to get extra supervision during training for tiny models

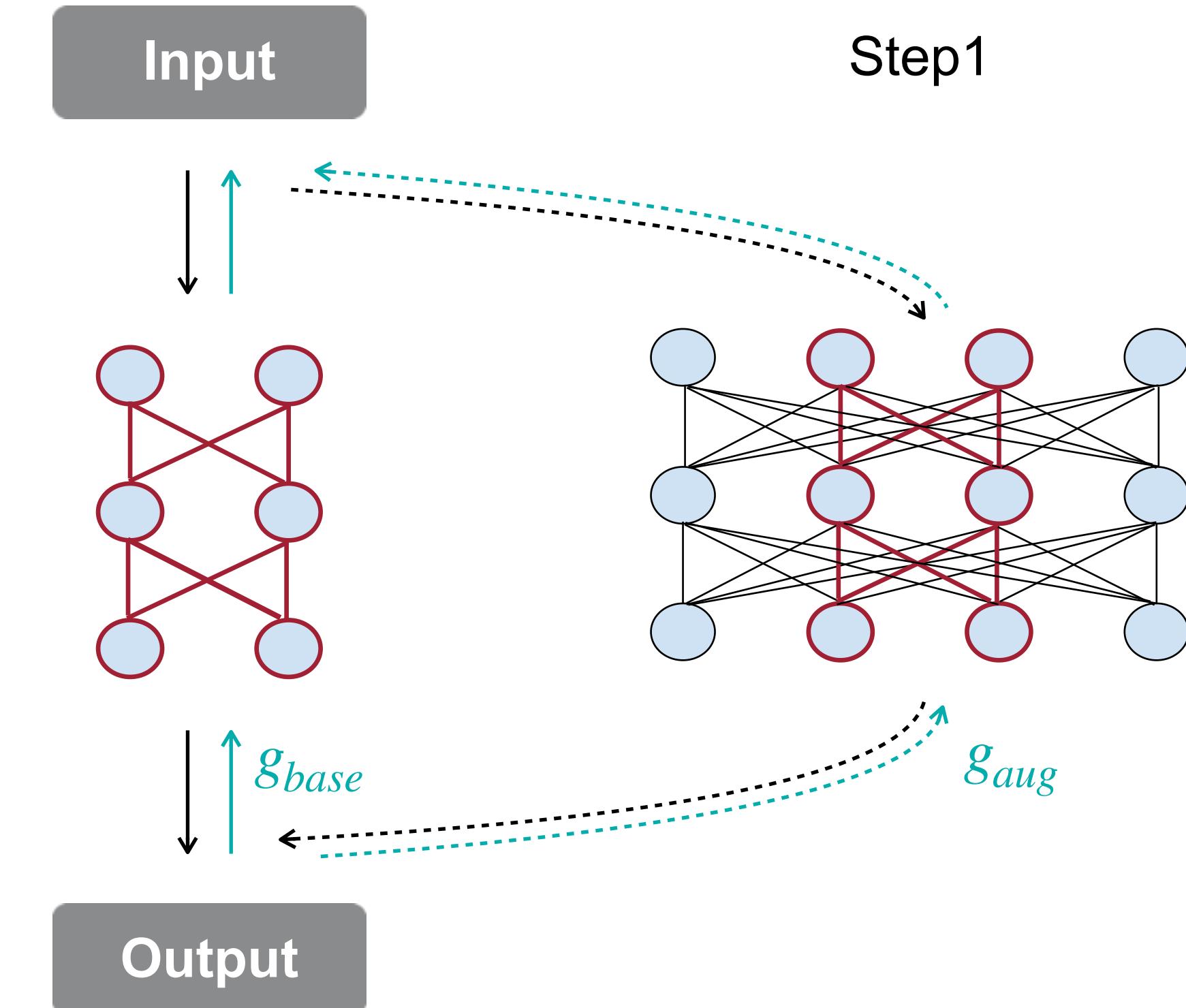


$$\mathcal{L}_{\text{aug}} = \underbrace{\mathcal{L}(W_{\text{base}})}_{\text{base supervision}}$$

Network Augmentation for Tiny Deep Learning [Cai et al., ICLR 2022]

NetAug: Training Process

Augment the model to get extra supervision during training for tiny models

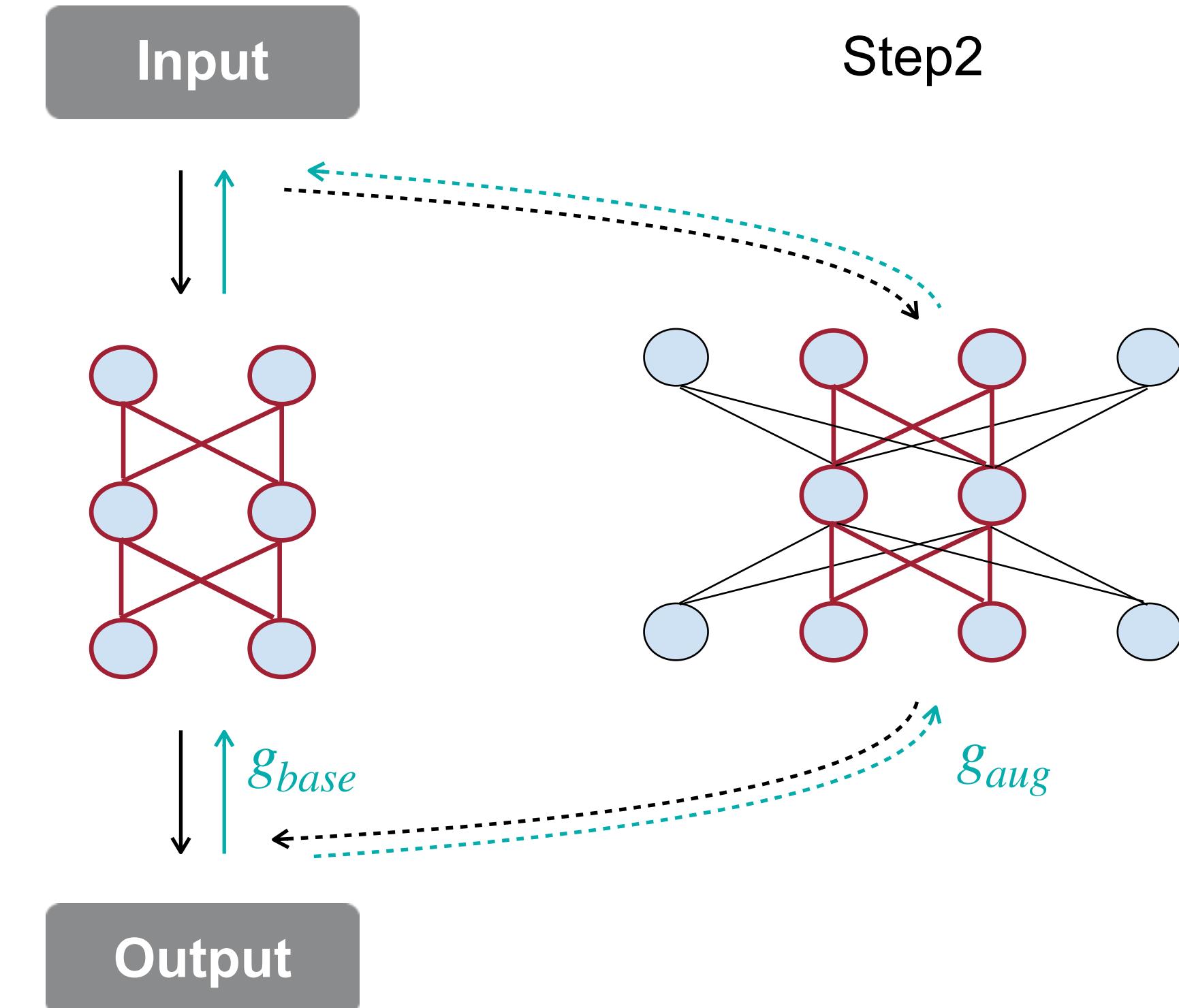


$$\mathcal{L}_{\text{aug}} = \underbrace{\mathcal{L}(W_{\text{base}})}_{\text{base supervision}} + \alpha \underbrace{\mathcal{L}([W_{\text{base}}, W_{\text{aug}}])}_{\text{auxiliary supervision}}$$

Network Augmentation for Tiny Deep Learning [Cai et al., ICLR 2022]

NetAug: Training Process

Augment the model to get extra supervision during training for tiny models

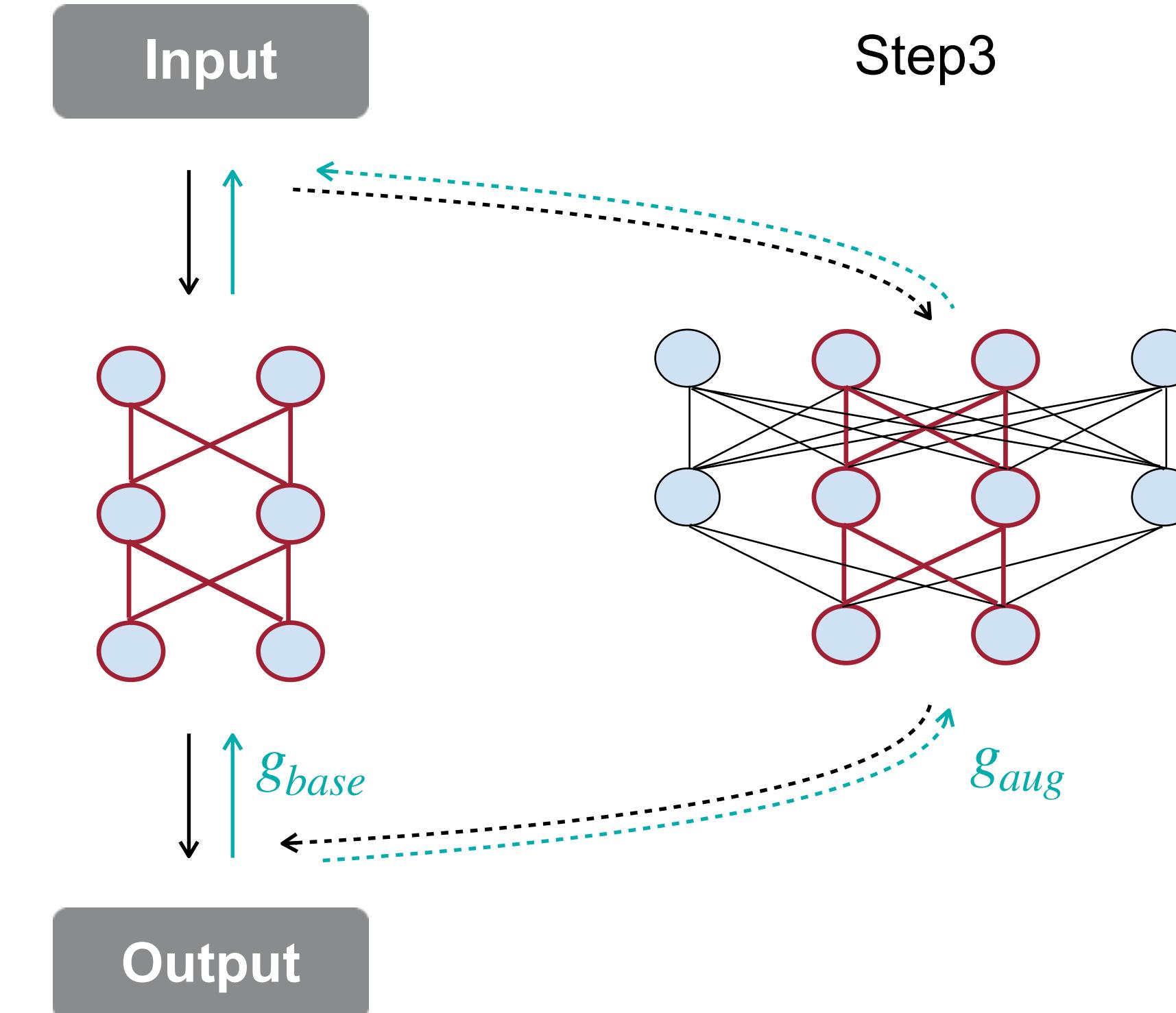


$$\mathcal{L}_{\text{aug}} = \underbrace{\mathcal{L}(W_{\text{base}})}_{\text{base supervision}} + \alpha \underbrace{\mathcal{L}([W_{\text{base}}, W_{\text{aug}}])}_{\text{auxiliary supervision}}$$

Network Augmentation for Tiny Deep Learning [Cai et al., ICLR 2022]

NetAug: Training Process

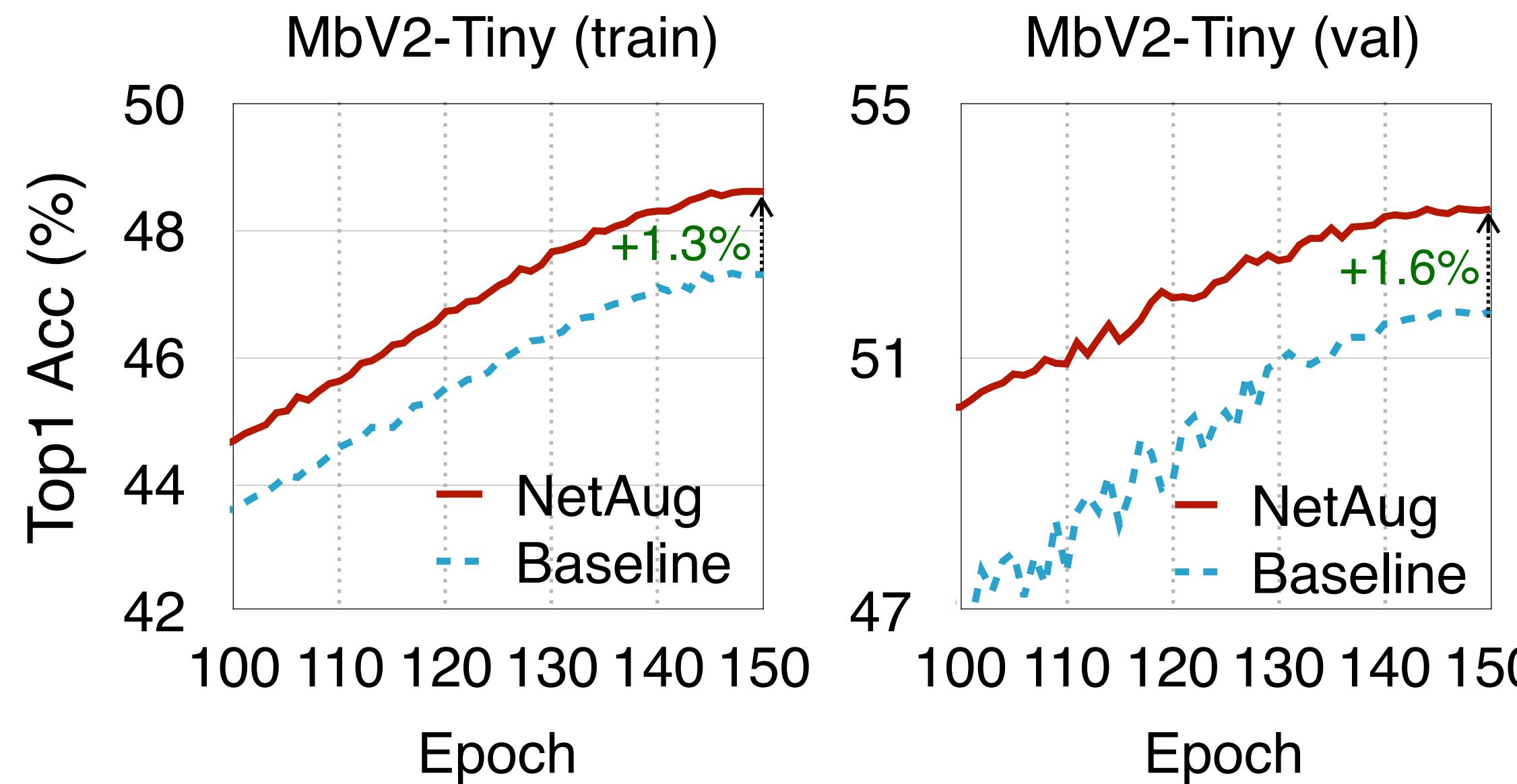
Augment the model to get extra supervision during training for tiny models



$$\mathcal{L}_{\text{aug}} = \underbrace{\mathcal{L}(W_{\text{base}})}_{\text{base supervision}} + \alpha \underbrace{\mathcal{L}([W_{\text{base}}, W_{\text{aug}}])}_{\text{auxiliary supervision}}$$

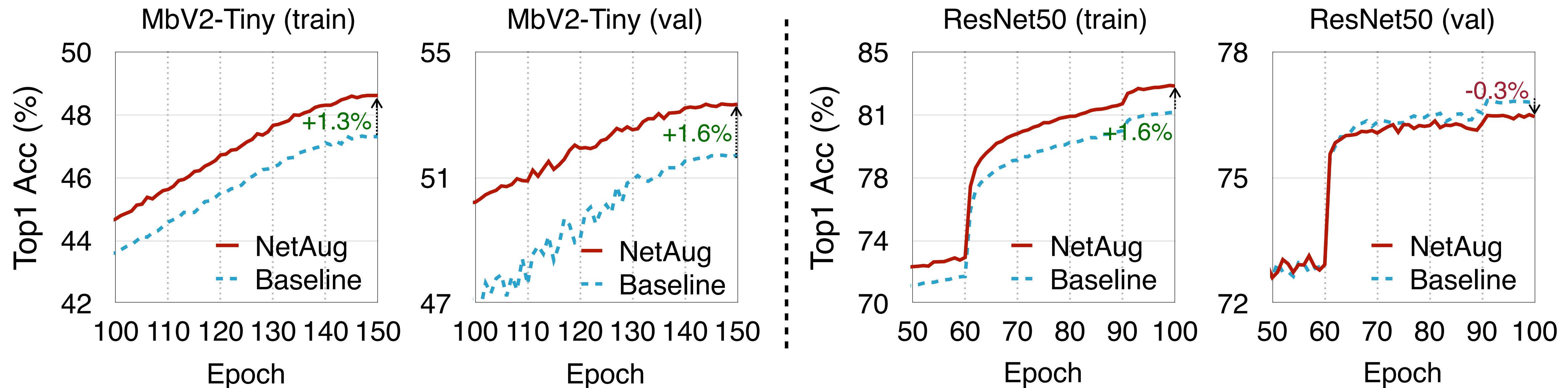
Network Augmentation for Tiny Deep Learning [Cai et al., ICLR 2022]

NetAug: Learning Curve



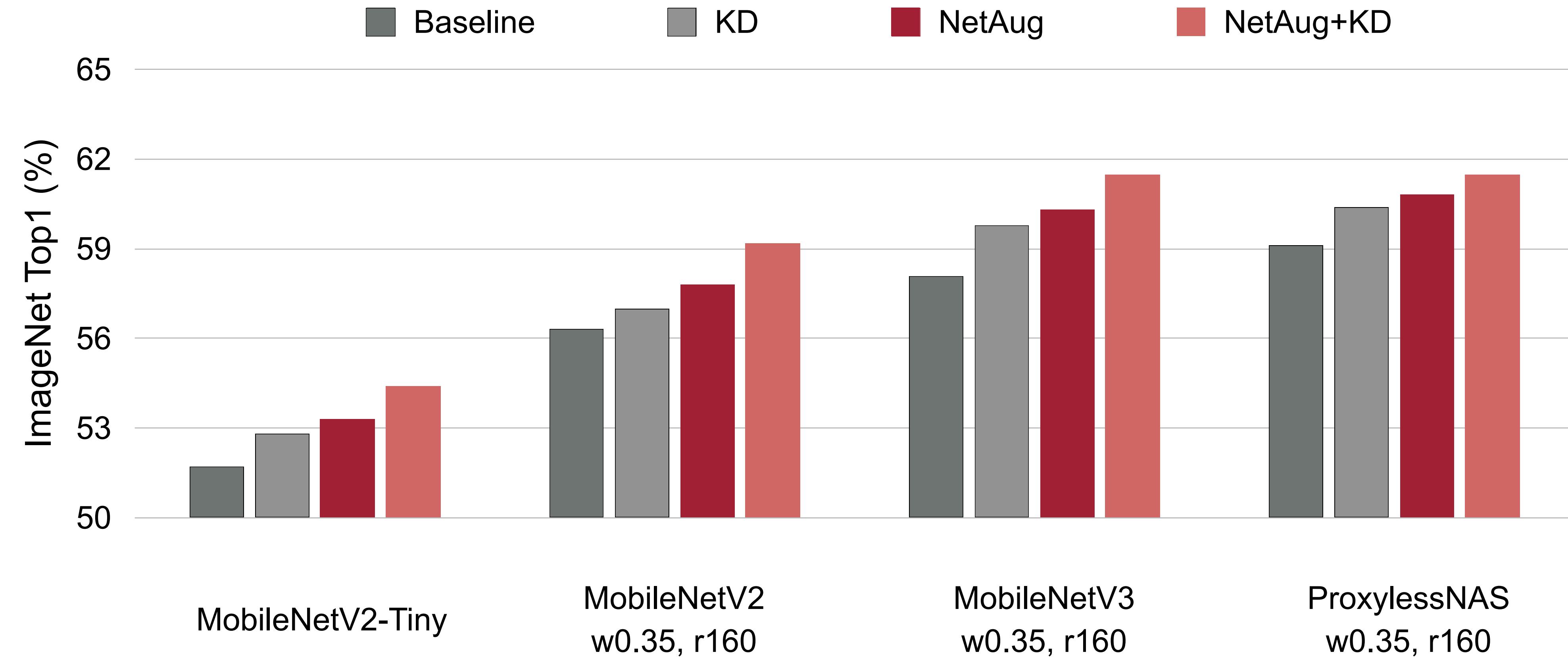
- For a tiny neural network, NetAug improves both the training accuracy and val accuracy.

NetAug: Learning Curve



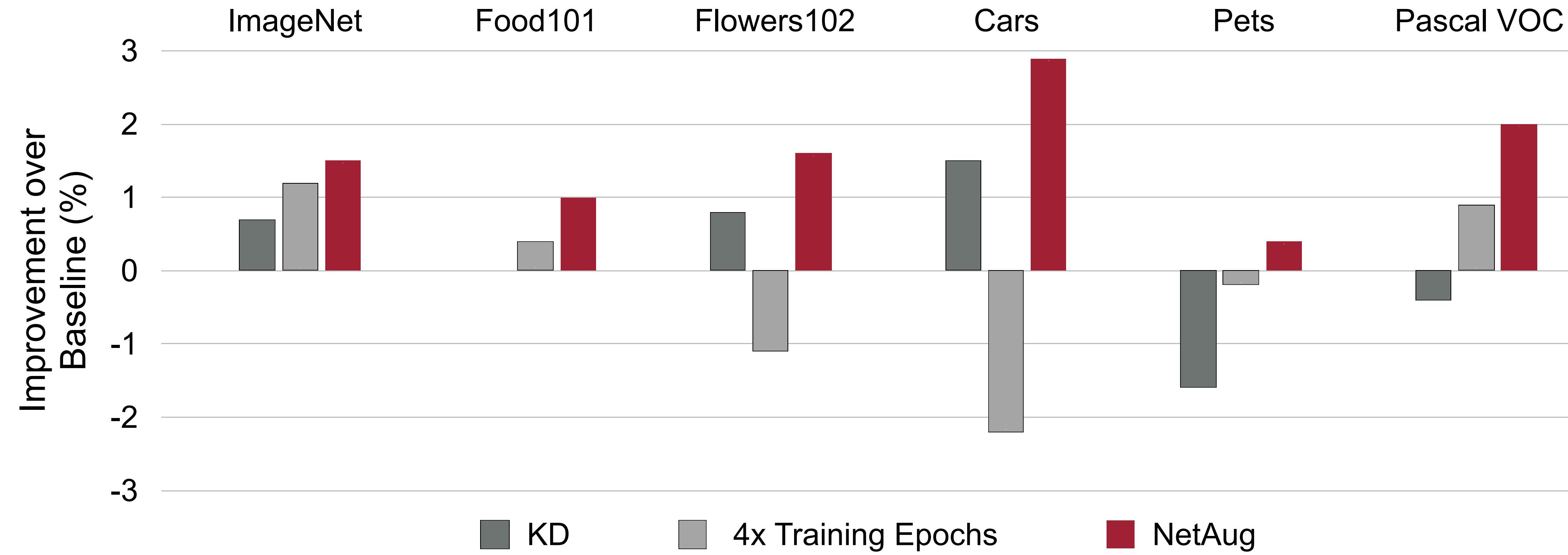
- For a tiny neural network, NetAug improves **both** the training accuracy and val accuracy.
- For a large neural network, NetAug **improves** the training accuracy but **hurts** the val accuracy.

NetAug: Results



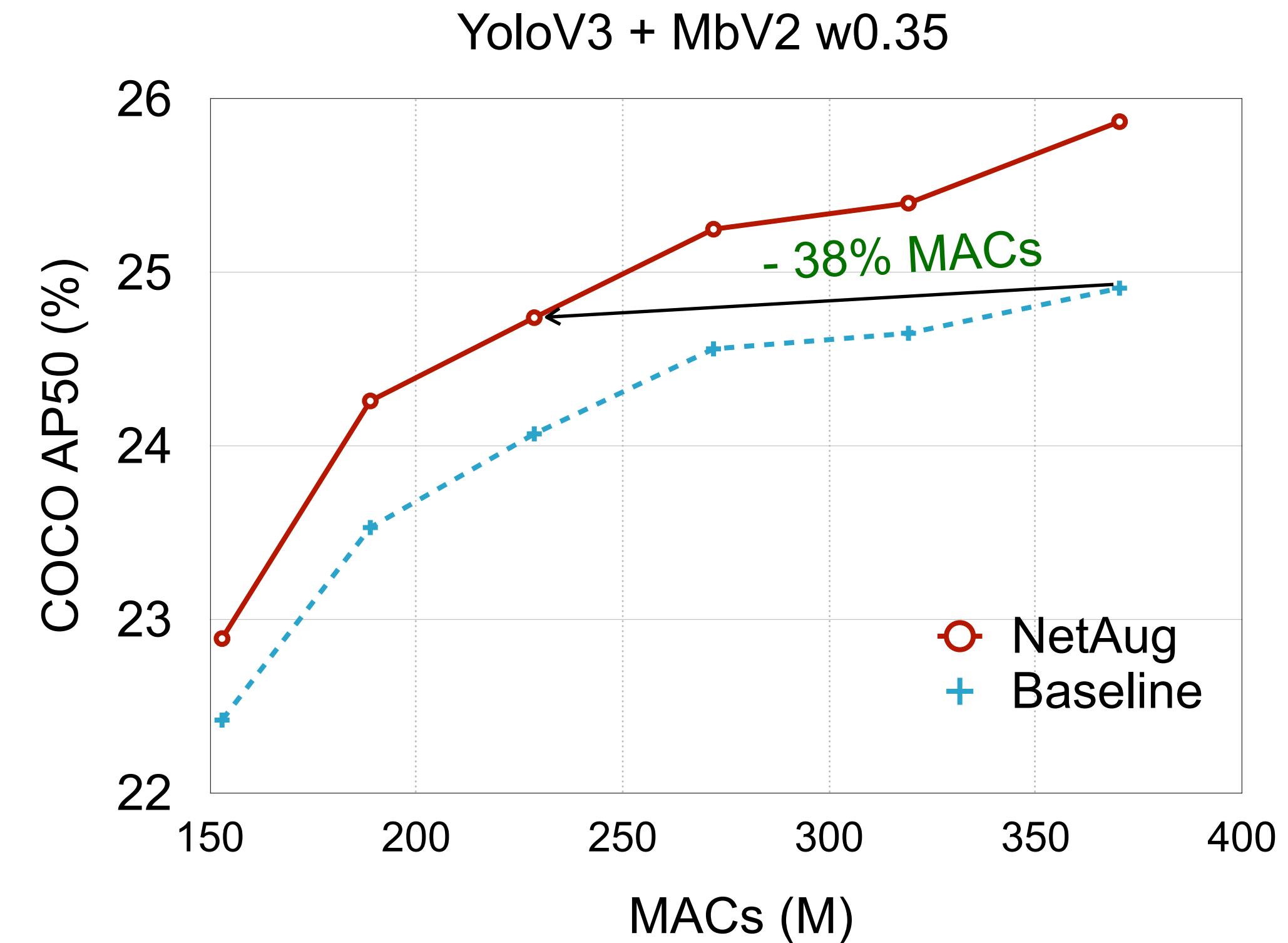
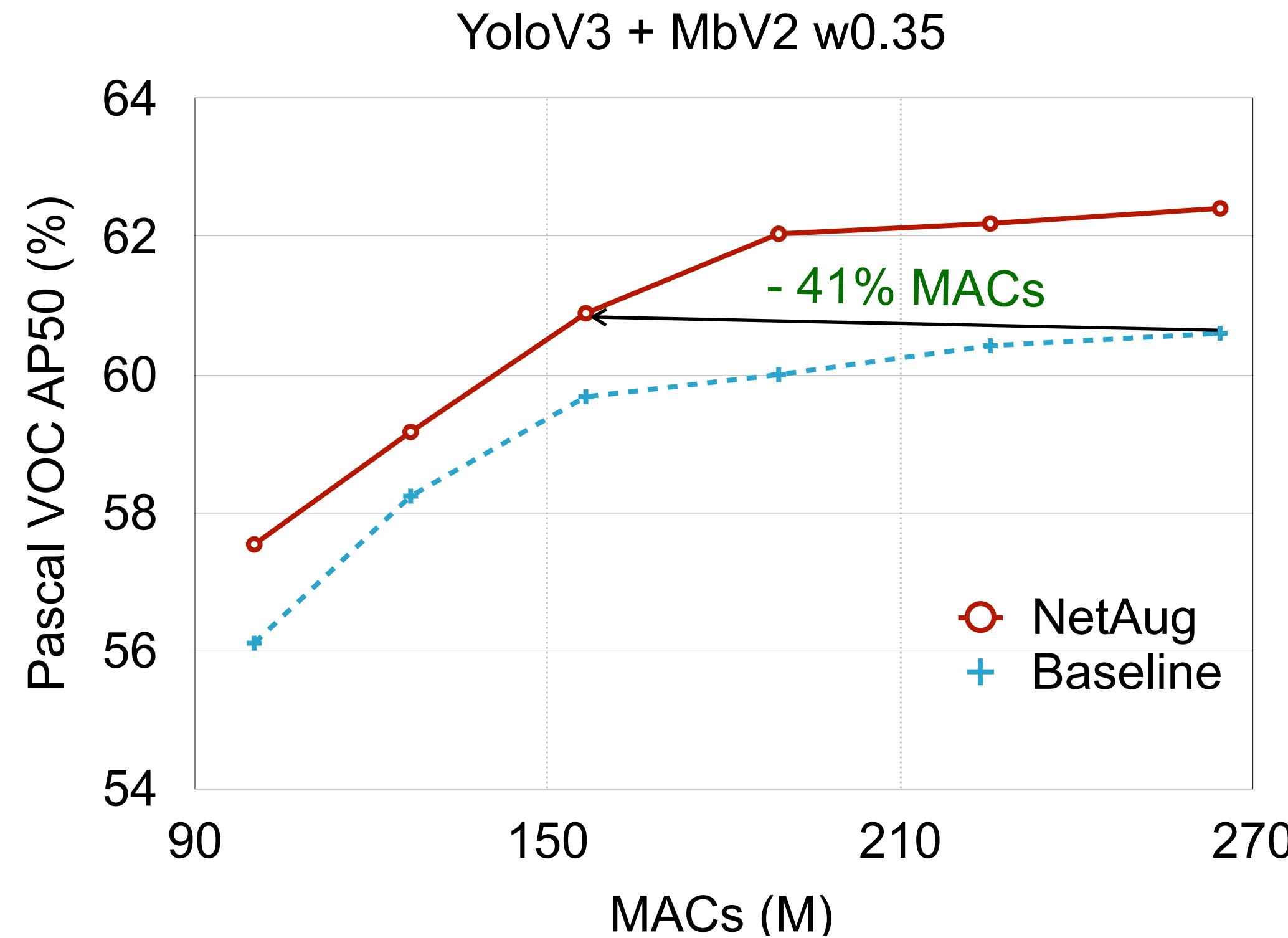
- NetAug is orthogonal to KD.

NetAug: Transfer Learning



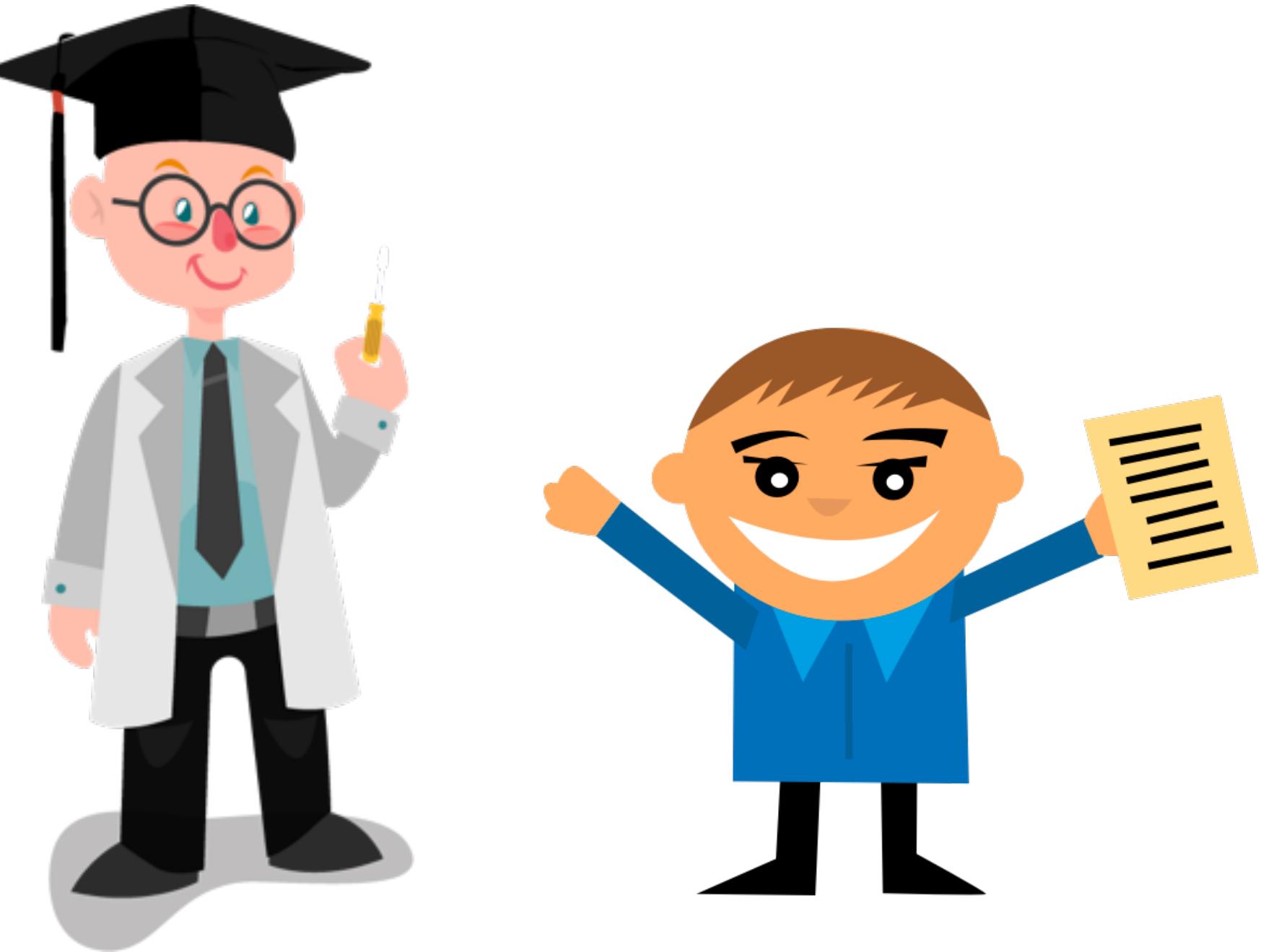
- NetAug provides better transfer learning performances than KD and 4x training schedule, thought their ImageNet performances are similar.

NetAug: Transfer to Object Detection



Summary of Today's Lecture

- In this lecture, we introduce:
 - What is knowledge distillation (KD);
 - What to match;
 - Self and online KD;
 - KD for different tasks;
 - Network augmentation.
- In the next lecture, we will introduce MCUNet, an algorithm-system co-design framework for TinyML on microcontrollers.



References

- 1. Network Augmentation for Tiny Deep Learning [Cai *et al.*, ICLR 2022]
- 2. Distilling the Knowledge in a Neural Network [Hinton *et al.*, NeurIPS Workshops 2014]
- 3. Knowledge Distillation: A Survey [Gou *et al.*, IJCV 2020]
- 4. Do Deep Nets Really Need to be Deep? [Ba and Caruana, NeurIPS 2014]
- 5. FitNets: Hints for Thin Deep Nets [Romero *et al.*, ICLR 2015]
- 6. Like What You Like: Knowledge Distill via Neuron Selectivity Transfer [Huang and Wang, arXiv 2017]
- 7. Paying More Attention to Attention: Improving the Performance of Convolutional Neural Networks via Attention Transfer [Zagoruyko and Komodakis, ICLR 2017]
- 8. Paraphrasing Complex Network: Network Compression via Factor Transfer [Kim *et al.*, NeurIPS 2018]
- 9. Knowledge Transfer via Distillation of Activation Boundaries Formed by Hidden Neurons [Heo *et al.*, AAAI 2019]
- 10. A Gift from Knowledge Distillation: Fast Optimization, Network Minimization and Transfer Learning [Yim *et al.*, CVPR 2017]
- 11. Relational Knowledge Distillation [Park *et al.*, CVPR 2019]

References

- 12. Distillation as a Defense to Adversarial Perturbations against Deep Neural Networks [Papernot *et al.*, SP 2016]
- 13. Born-Again Neural Networks [Furlanello *et al.*, ICML 2018]
- 14. Deep Mutual Learning [Zhang *et al.*, CVPR 2018]
- 15. Knowledge Distillation by On-the-Fly Native Ensemble [Lan *et al.*, NeurIPS 2018]
- 16. Be Your Own Teacher: Improve the Performance of Convolutional Neural Networks via Self Distillation [Zhang *et al.*, ICCV 2019]
- 17. Learning Efficient Object Detection Models with Knowledge Distillation [Chen *et al.*, NeurIPS 2017]
- 18. Localization Distillation for Dense Object Detection [Zheng *et al.*, CVPR 2022]
- 19. Improving Object Detection by Label Assignment Distillation [Nguyen *et al.*, WACV 2022]
- 20. Structured Knowledge Distillation for Semantic Segmentation [Liu *et al.*, CVPR 2019]
- 21. MobileBERT: a Compact Task-Agnostic BERT for Resource-Limited Devices [Sun *et al.*, ACL 2020]
- 22. Improved Regularization of Convolutional Neural Networks with Cutout [DeVries *et al.*, arXiv 2017]

References

- 23. mixup: Beyond Empirical Risk Minimization [Zhang *et al.*, ICLR 2018]
- 24. AutoAugment: Learning Augmentation Policies from Data [Cubuk *et al.*, CVPR 2019]
- 25. Efficient Object Localization Using Convolutional Networks [Tompson *et al.*, CVPR 2015]
- 26. DropBlock: A regularization method for convolutional networks [Ghiasi *et al.*, NeurIPS 2018]