MIT HAN LAB

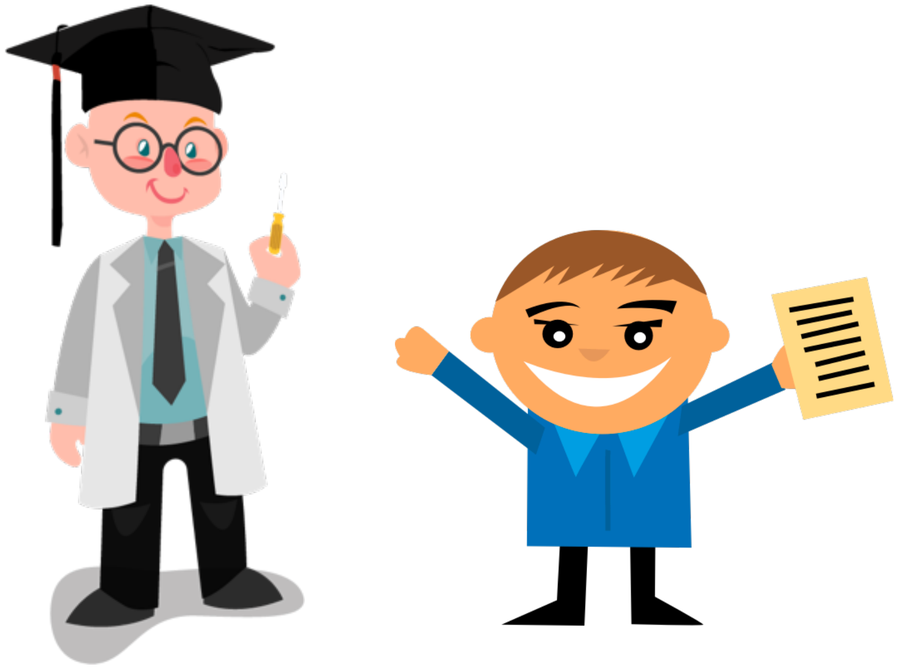
Lecture 10
Knowledge Distillation

Song Han
songhan@mit.edu

Lecture Plan

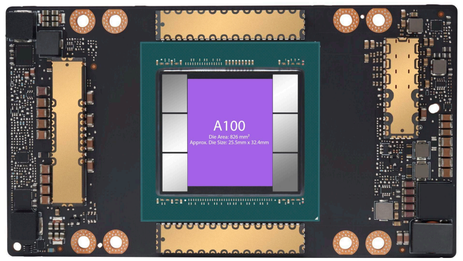
Today we will:

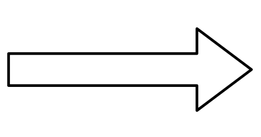
1. What is knowledge distillation;
2. What to match;
3. Self and online distillation;
4. Distillation for different tasks;
5. Network Augmentation, a training technique for tiny machine learning models.



What is knowledge distillation?

Challenge: limited hardware resources





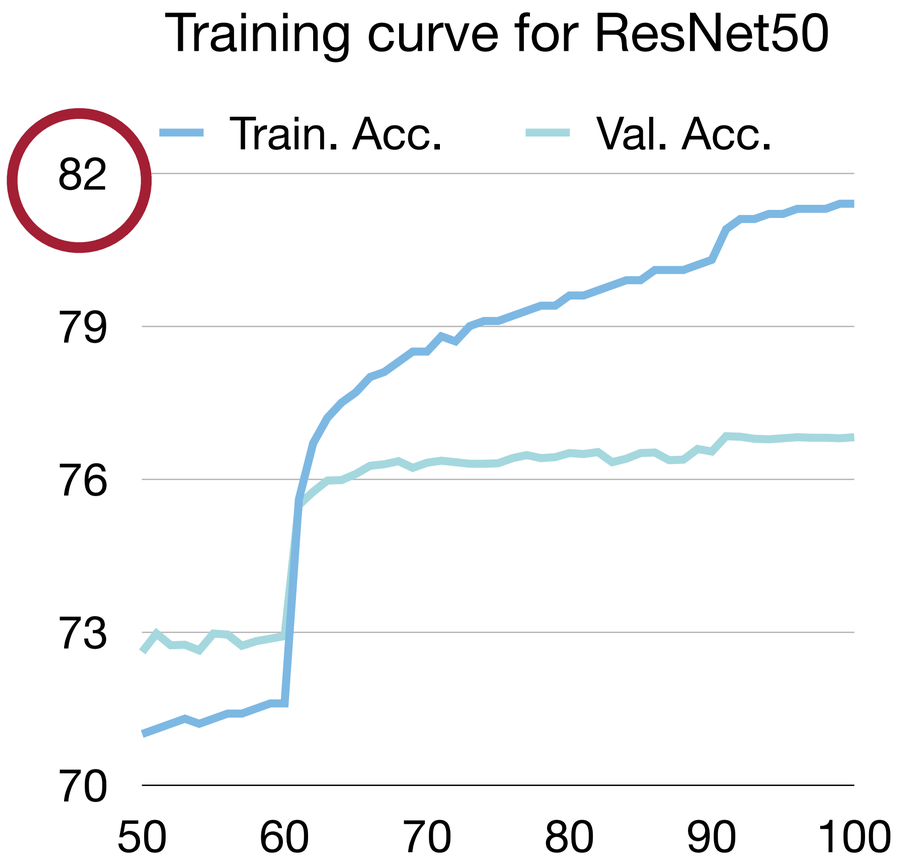
	Cloud AI	Tiny AI
Computation (fp32)	19.5 TFLOPS	MFLOPs
Memory	80GB	256kB
Neural Network	ResNet ViT-Large …	MCUNet MobileNetV2-Tiny …

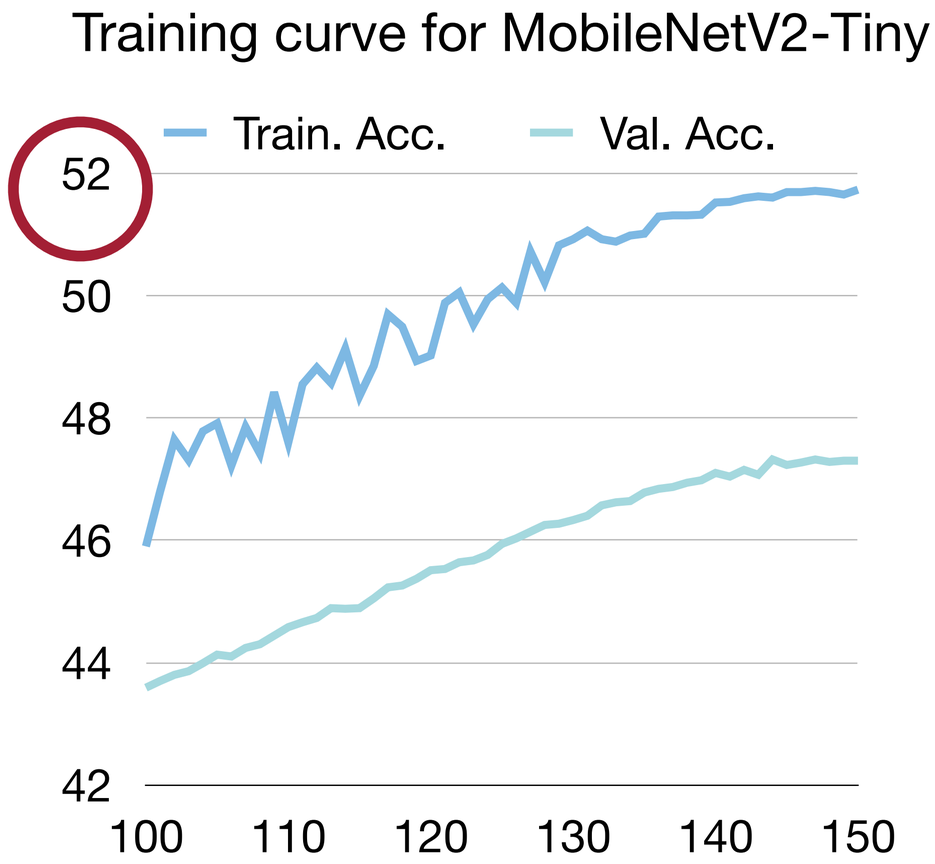
- Neural network must be **tiny** to run efficiently on tiny edge devices.

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]

Knowledge Distillation

Distilling the Knowledge in a Neural Network

Geoffrey Hinton[*†]
Google Inc.
Mountain View
geoffhinton@google.com

Oriol Vinyals[†]
Google Inc.
Mountain View
vinyals@google.com

Jeff Dean
Google Inc.
Mountain View
jeff@google.com

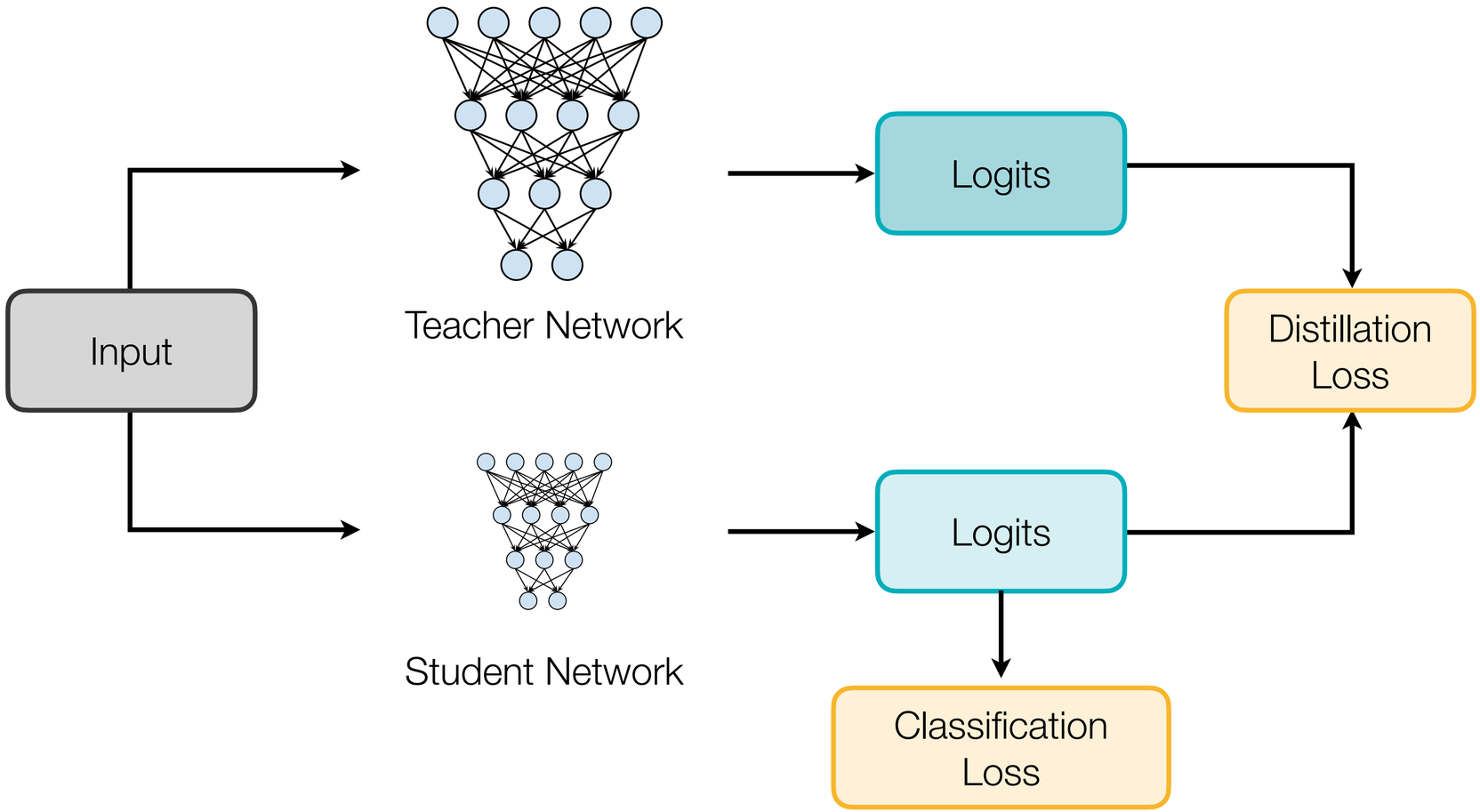
Abstract

A very simple way to improve the performance of almost any machine learning algorithm is to train many different models on the same data and then to average their predictions [3]. Unfortunately, making predictions using a whole ensemble of models is cumbersome and may be too computationally expensive to allow deployment to a large number of users, especially if the individual models are large neural nets. Caruana and his collaborators [1] have shown that it is possible to compress the knowledge in an ensemble into a single model which is much easier to deploy and we develop this approach further using a different compression technique. We achieve some surprising results on MNIST and we show that we can significantly improve the acoustic model of a heavily used commercial system by distilling the knowledge in an ensemble of models into a single model. We also introduce a new type of ensemble composed of one or more full models and many specialist models which learn to distinguish fine-grained classes that the full models confuse. Unlike a mixture of experts, these specialist models can be trained rapidly and in parallel.

Distilling the knowledge in a neural network
G Hinton, O Vinyals, J Dean
arXiv preprint arXiv:1503.02531 2 (7)
10910 2015

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

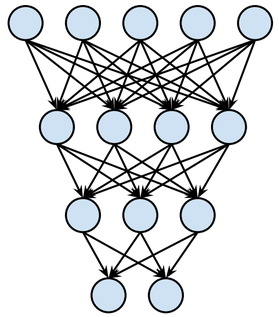
Illustration of knowledge distillation



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

Intuition of knowledge distillation

Matching prediction probabilities between teacher and student

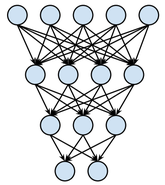


Teacher Network

	Logits	Probabilities
Cat	5	0.982
Dog	1	0.017

$$\frac{\exp(5)}{\exp(5)+\exp(1)}$$

$$\frac{\exp(1)}{\exp(5)+\exp(1)}$$



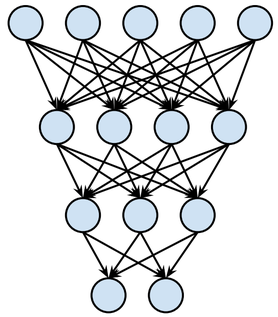
Student Network

	Logits	Probabilities
Cat	3	0.731
Dog	2	0.269

The student model is less confident

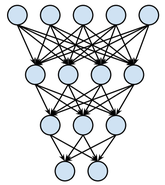
Intuition of knowledge distillation

Matching prediction probabilities between teacher and student



Teacher Network

	Logits	Probabilities
Cat	5	0.982
Dog	1	0.017

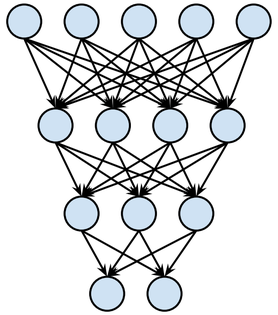


Student Network

	Logits	Probabilities
Cat	3	0.731 ↑
Dog	2	0.269 ↓

Intuition of knowledge distillation

Concept of temperature



Teacher Network

$$\frac{\exp(5/1)}{\exp(5/1)+\exp(1/1)}$$

	Logits	Probabilities (T=1)	Probabilities (T=10)
Cat	5	0.982	0.599
Dog	1	0.017	0.401

$$\frac{\exp(5/10)}{\exp(5/10)+\exp(1/10)}$$

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,$i, j = 0,1,2,...,C-1$, where C is the number of classes. T is the temperature, which is normally set to 1.

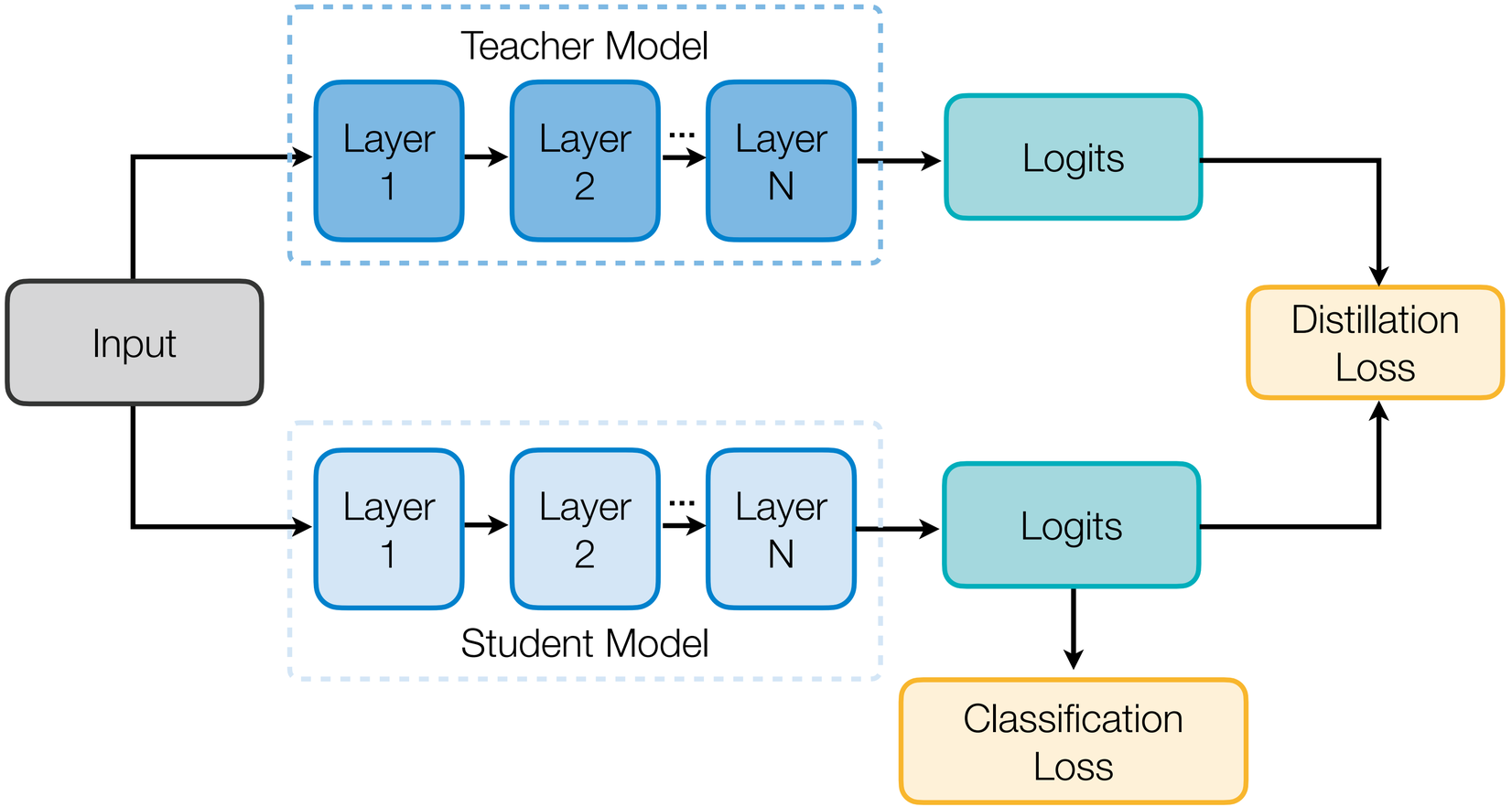
- The goal of knowledge distillation is to **align the class probability distributions from teacher and student networks**.

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

What to match?

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

Matching output logits



Cross entropy loss:

$$\mathbb{E}(-p_t \log p_s);$$

L2 loss:

$$E(\|p_t - p_s\|_2^2)$$

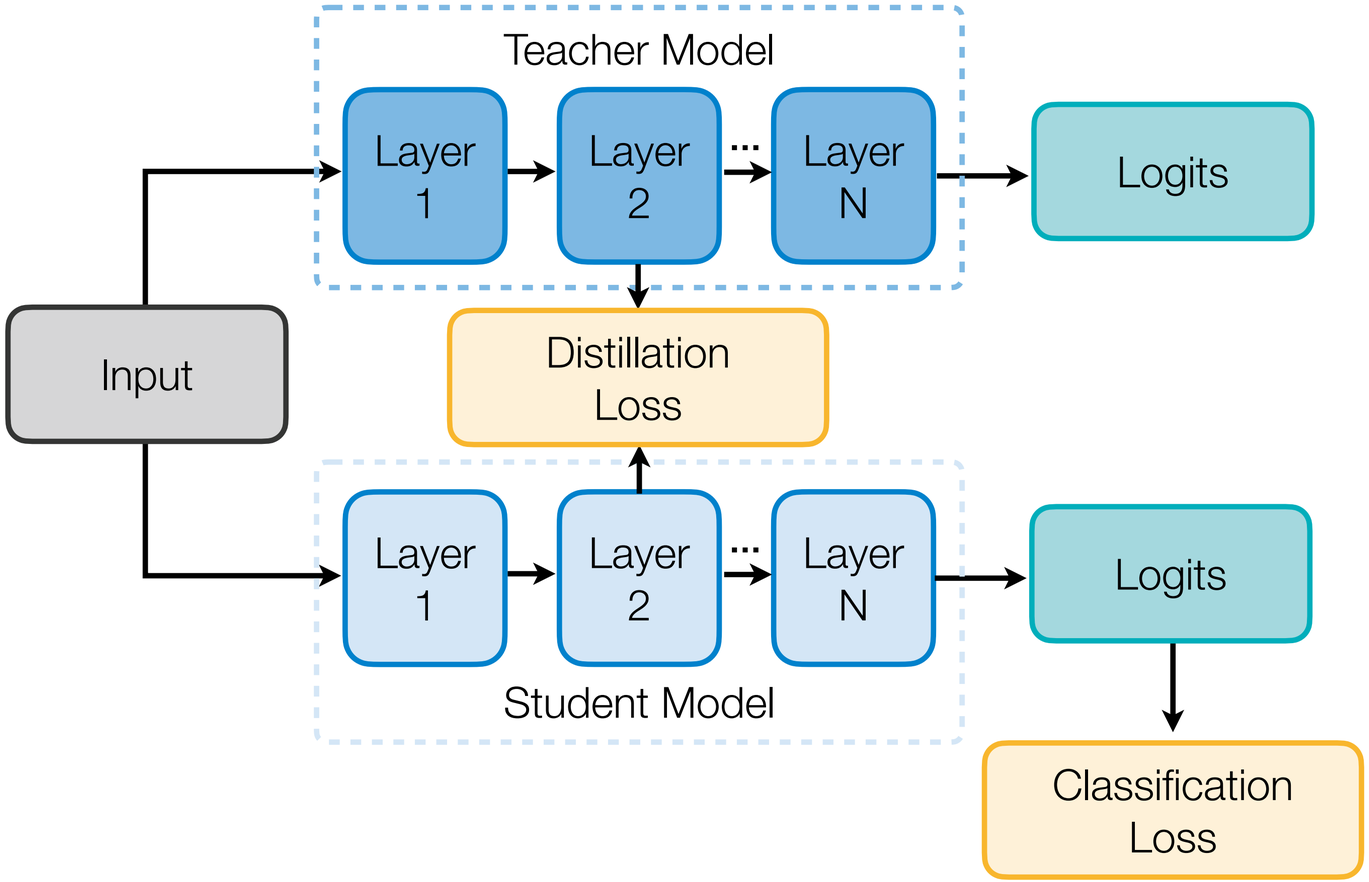
Distilling the Knowledge in a Neural Network [Hinton *et al.*, NeurIPS Workshops 2014]
Do Deep Nets Really Need to be Deep? [Ba and Caruana, NeurIPS 2014]

What to match?

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

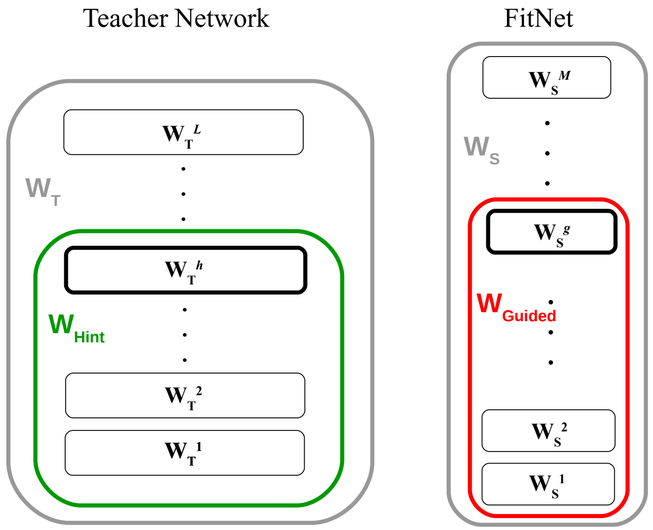
What else to match other than output logits?

Matching intermediate weights

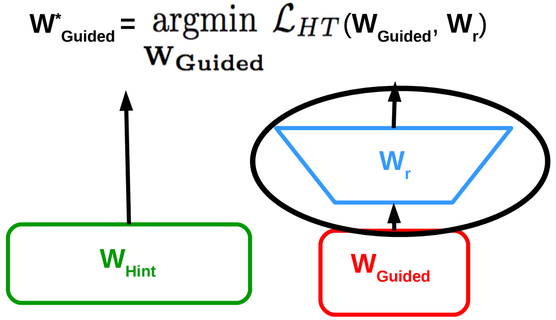


Knowledge Distillation: A Survey [Gou *et al.*, IJCV 2020]

Matching intermediate weights



(a) Teacher and Student Networks



(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).

FitNets: Hints for Thin Deep Nets [Romero *et al.*, ICLR 2015]

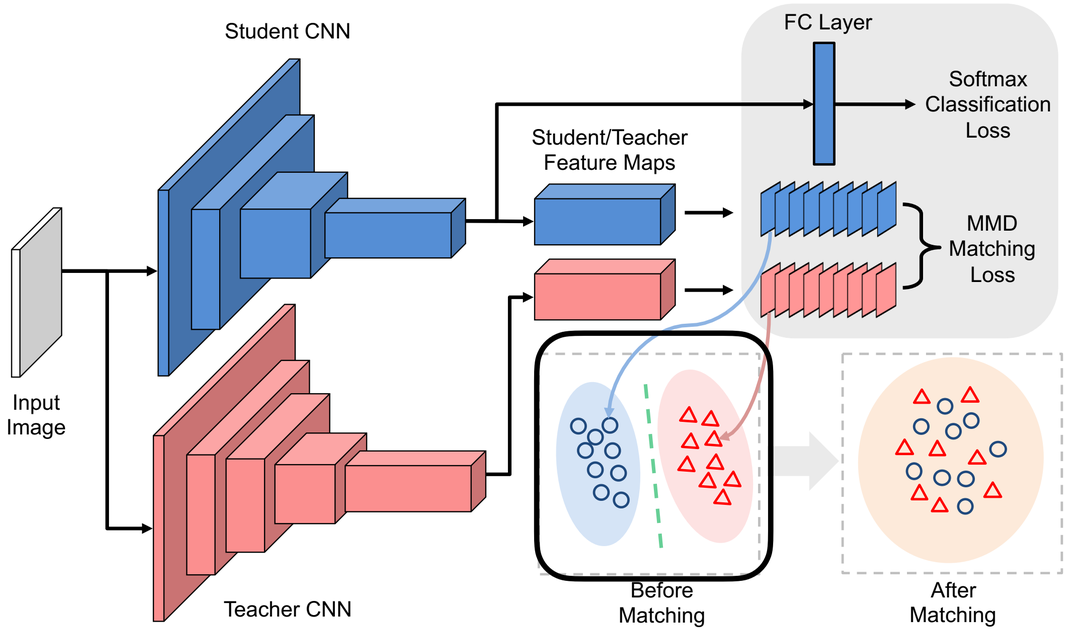
What to match?

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.



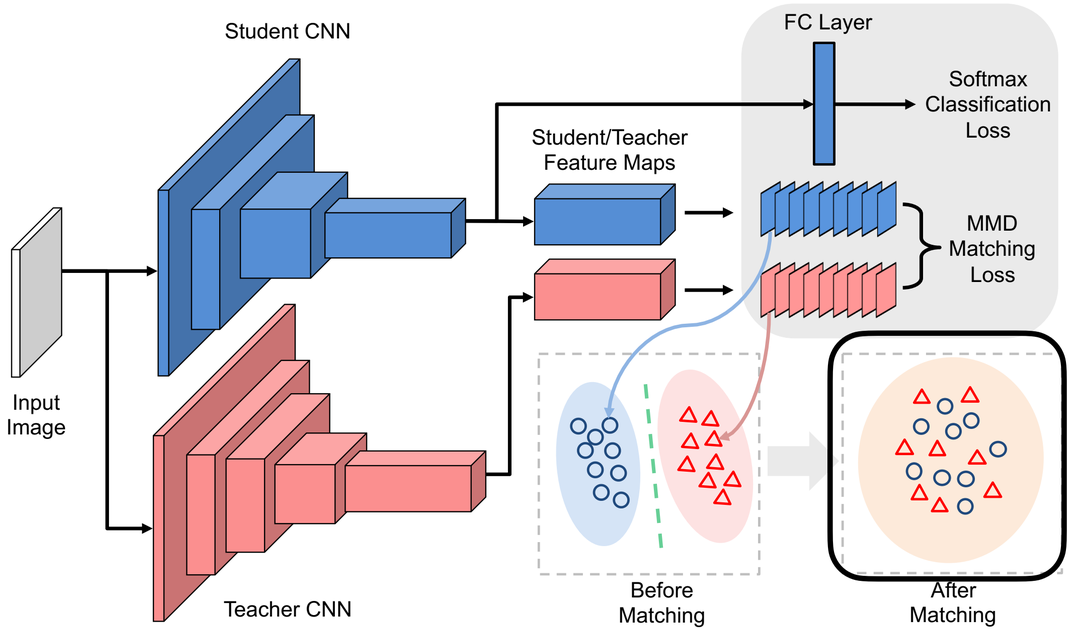
Teacher and student have very different feature distributions without distillation

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

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.



With the distillation objective, teacher and student feature distributions are similar.

$$\mathcal{L}_{\text{MMD}^2}(\mathbf{F}_T, \mathbf{F}_S) = \frac{1}{{C_T}^2}\sum_{i=1}^{C_T}\sum_{i'=1}^{C_T} k\left(\frac{\mathbf{f}_T^{i\cdot}}{\|\mathbf{f}_T^{i\cdot}\|_2}, \frac{\mathbf{f}_T^{i'\cdot}}{\|\mathbf{f}_T^{i'\cdot}\|_2}\right) + \frac{1}{{C_S}^2}\sum_{j=1}^{C_S}\sum_{j'=1}^{C_S} k\left(\frac{\mathbf{f}_S^{j\cdot}}{\|\mathbf{f}_S^{j\cdot}\|_2}, \frac{\mathbf{f}_S^{j'\cdot}}{\|\mathbf{f}_S^{j'\cdot}\|_2}\right) - \frac{2}{C_T C_S}\sum_{i=1}^{C_T}\sum_{j=1}^{C_S} k\left(\frac{\mathbf{f}_T^{i\cdot}}{\|\mathbf{f}_T^{i\cdot}\|_2}, \frac{\mathbf{f}_S^{j\cdot}}{\|\mathbf{f}_S^{j\cdot}\|_2}\right).$$

Cosine of angle between teacher/student feature vectors

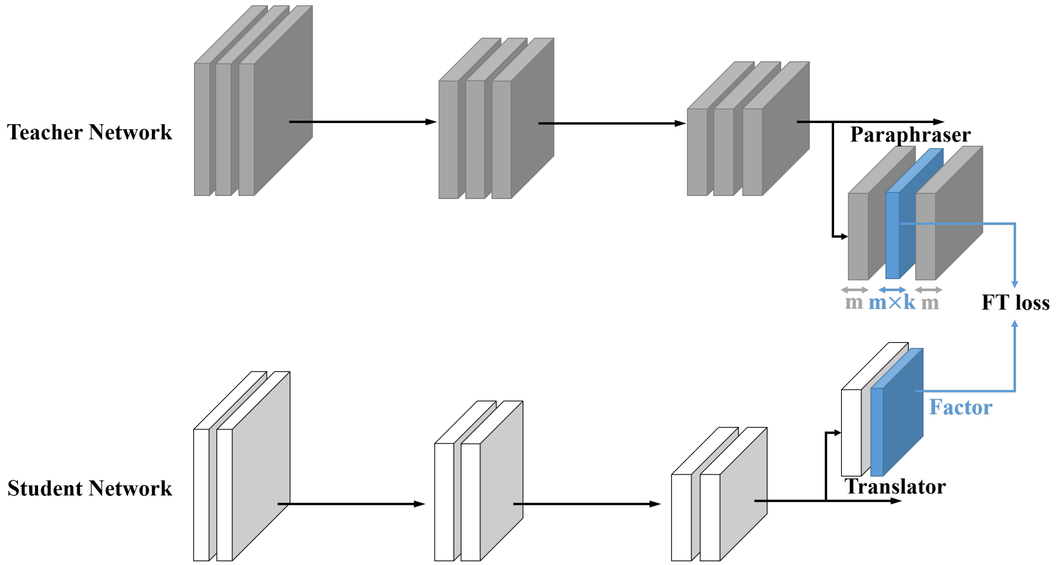
Use maximum mean discrepancy (MMD) as an objective. k is the kernel function, and its simplest form is the dot product.

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

Matching intermediate features

Minimizing the L2 distance between feature maps

- The paraphraser shrinks the output teacher feature map from m dimensions to $m \times k$ dimensions (called **factor**, typically k=0.5) and then expands the dimensionality back to m.
- The output of paraphraser is supervised with a **reconstruction loss** against the original m -dimensional output.
- The student uses one layer of MLP to obtain a **factor** with the same dimensionality of $m \times k$.
- FT minimizes the distance between teacher and student factors.



Paraphrasing Complex Network: Network Compression via Factor Transfer [Kim *et al.*, NeurIPS 2018]

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.

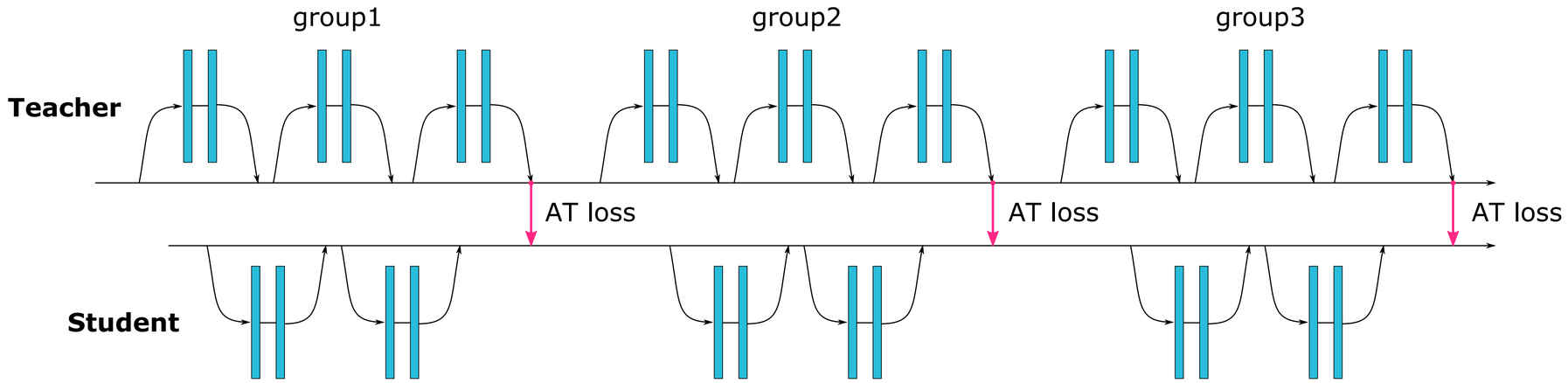
input image

attention map

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

Matching intermediate attention maps

- The attention transfer objective is defined as: $\frac{\beta}{2}||J_S - J_T||_2^2$, here J_S is the student attention map (gradient of student feature map) and J_T is the teacher attention map. β is a constant.

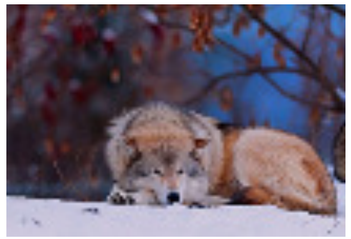


Paying More Attention to Attention: Improving the Performance of Convolutional Neural Networks via Attention Transfer [Zagoruyko and Komodakis, ICLR 2017]

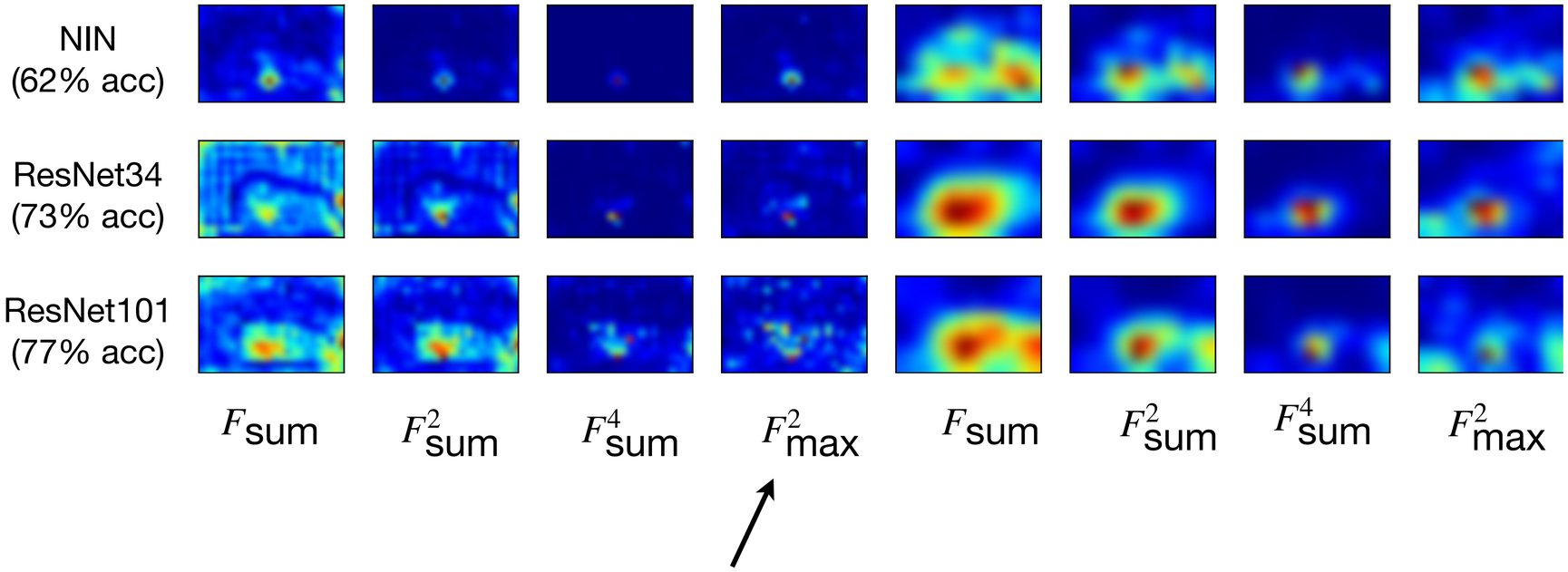
Intermediate attention maps

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.



Input



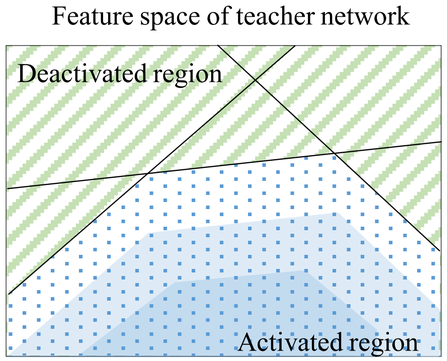
Paying More Attention to Attention: Improving the Performance of Convolutional Neural Networks via Attention Transfer [Zagoruyko and Komodakis, ICLR 2017]

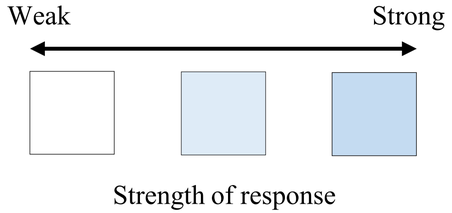
What to match?

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 $\rho(x) = 1[x > 0]$.
- We want to minimize $\mathscr{L}(I) = ||\rho(T(I)) - \rho(S(I))||_1$, where S and T corresponds to student and teacher networks, respectively.





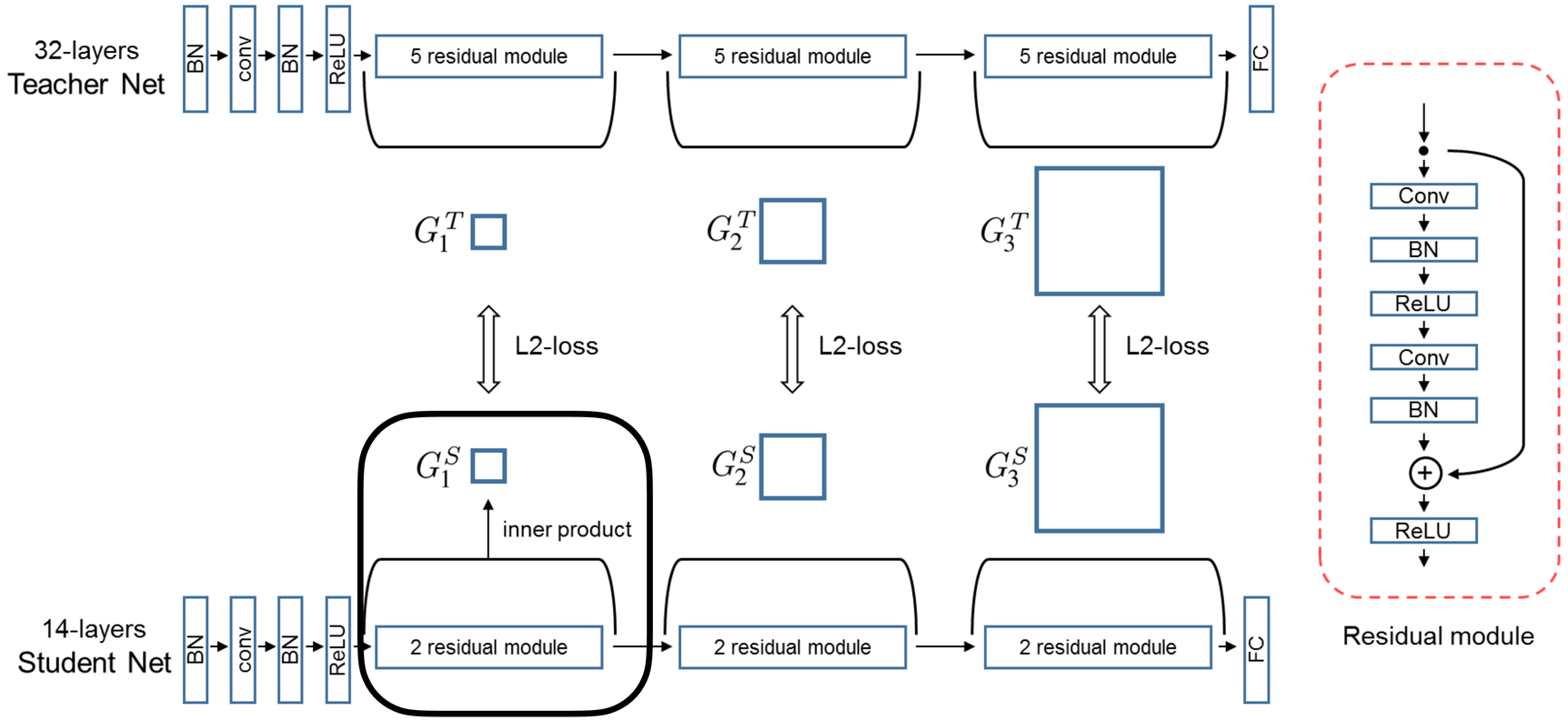
Knowledge Transfer via Distillation of Activation Boundaries Formed by Hidden Neurons [Heo *et al.*, AAAI 2019]

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



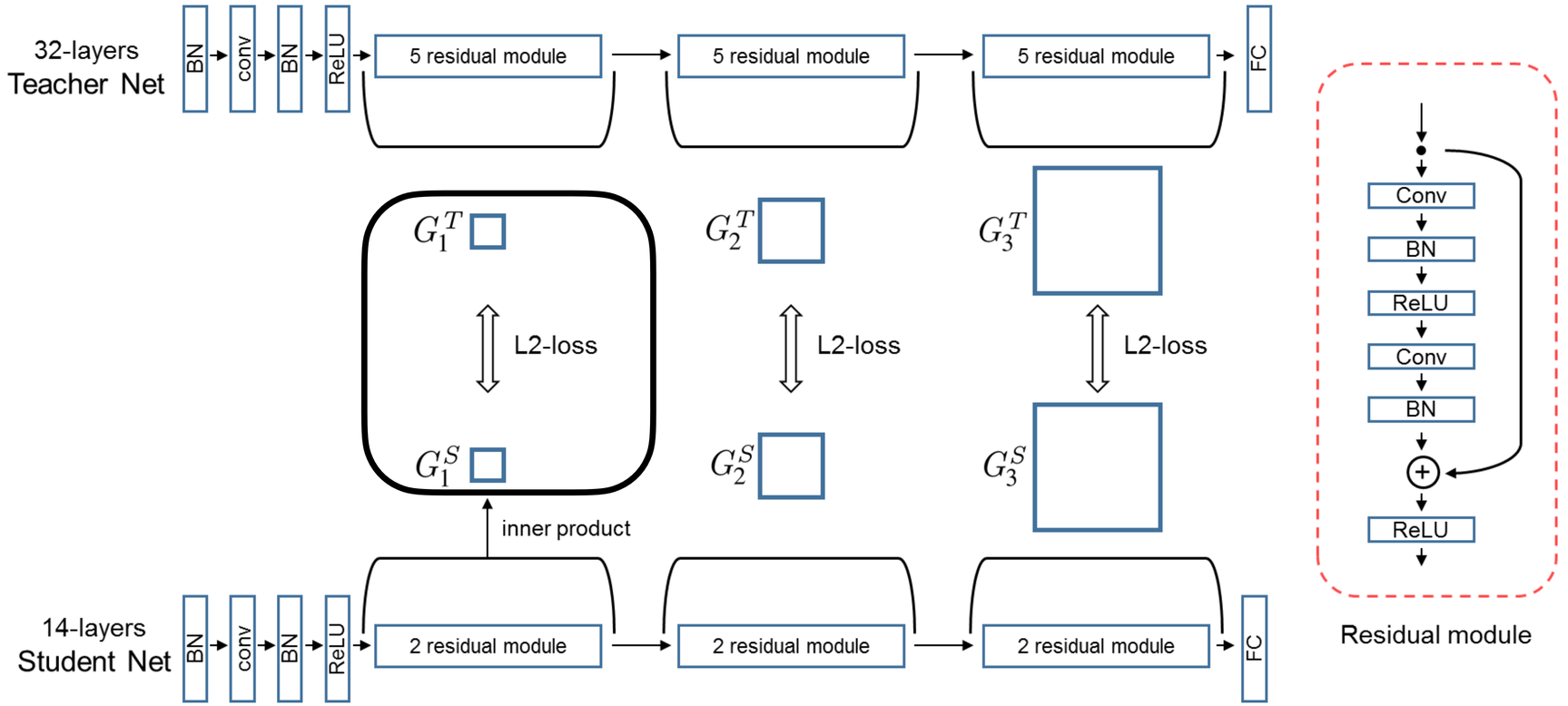
Use inner product to extract relational information (a matrix of shape $C_{\text{in}} \times C_{\text{out}}$, reduction on the spatial dimensions) for both student and teacher networks.

Note: the student and teacher networks only differ in number of layers, not number of channels.

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

Matching relational information

Relations between different layers

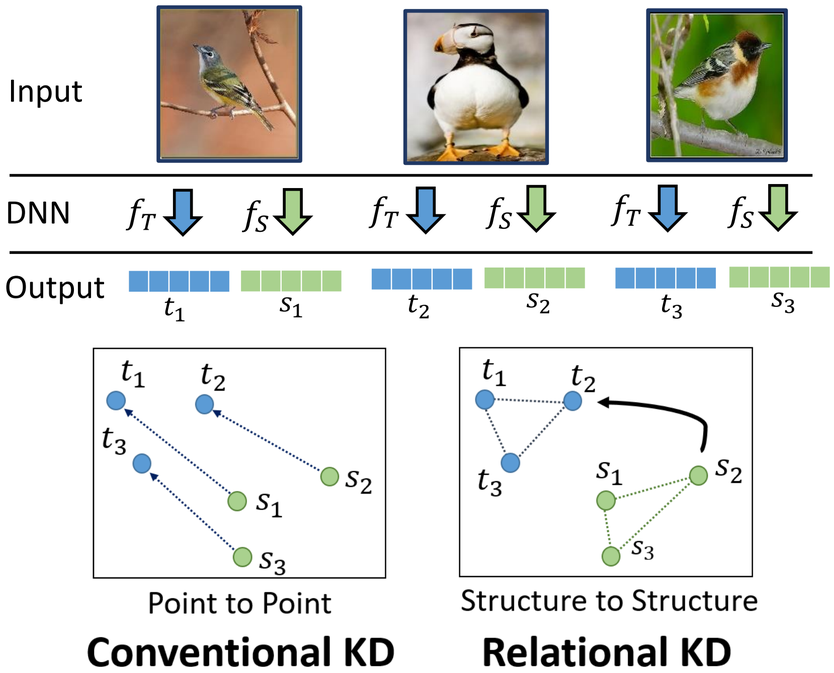


Then match the resulting dot products between teacher and student networks (G_1^T, G_1^S).

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

Matching relational information

Relations between different samples



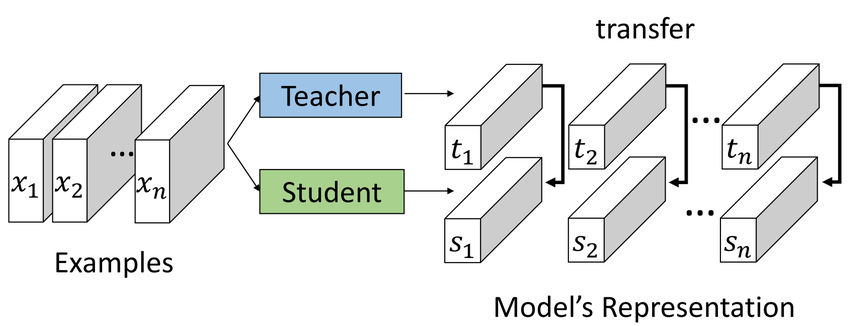
Conventional KD focuses on matching features / logits for **one input**.
Relational KD looks at the **relations** between intermediate features from **multiple inputs**.

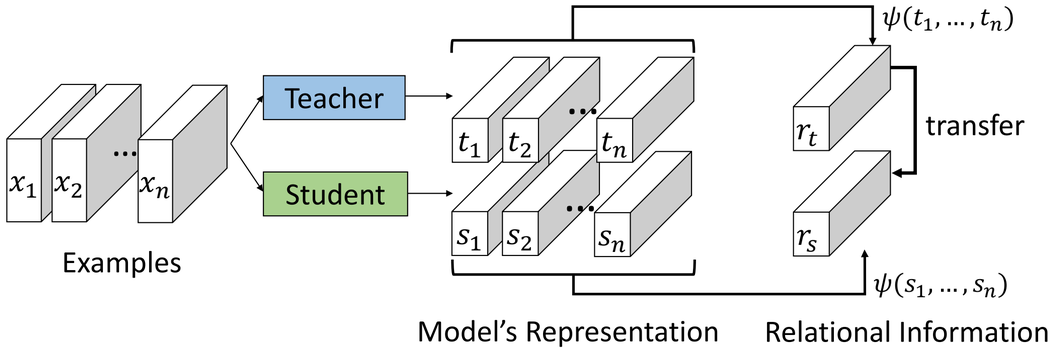
Relational Knowledge Distillation [Park *et al.*, CVPR 2019]

Matching relational information

Relations between different samples

$\psi(s_1, s_2, \ldots, s_n) = (\|s_1 - s_2\|_2^2, \|s_1 - s_3\|_2^2, \ldots, \|s_1 - s_n\|_2^2, \ldots \|s_{n-1} - s_n\|_2^2)$ 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**

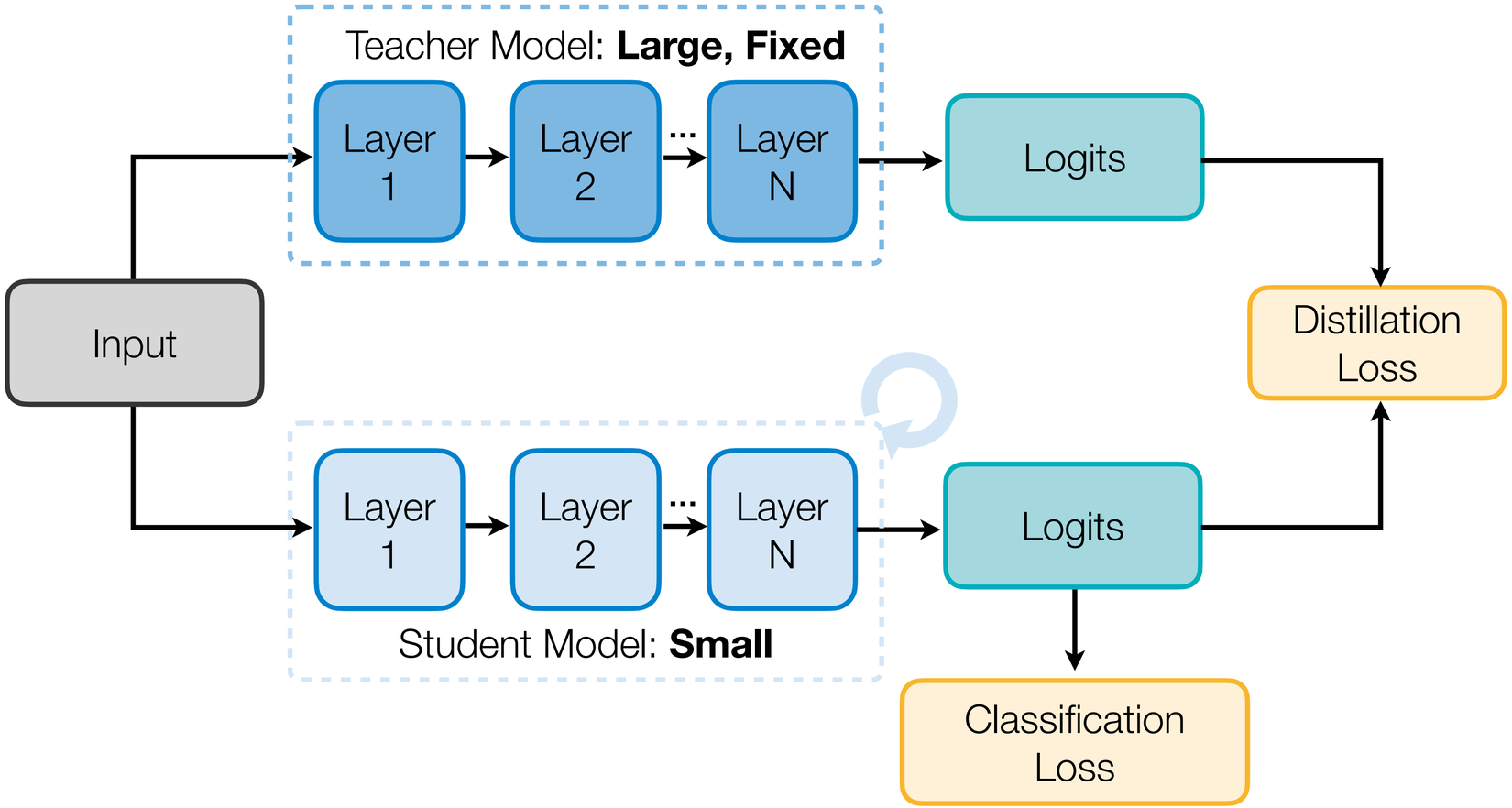
Relational Knowledge Distillation [Park *et al.*, CVPR 2019]

Self and Online Distillation

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

Overview of knowledge distillation

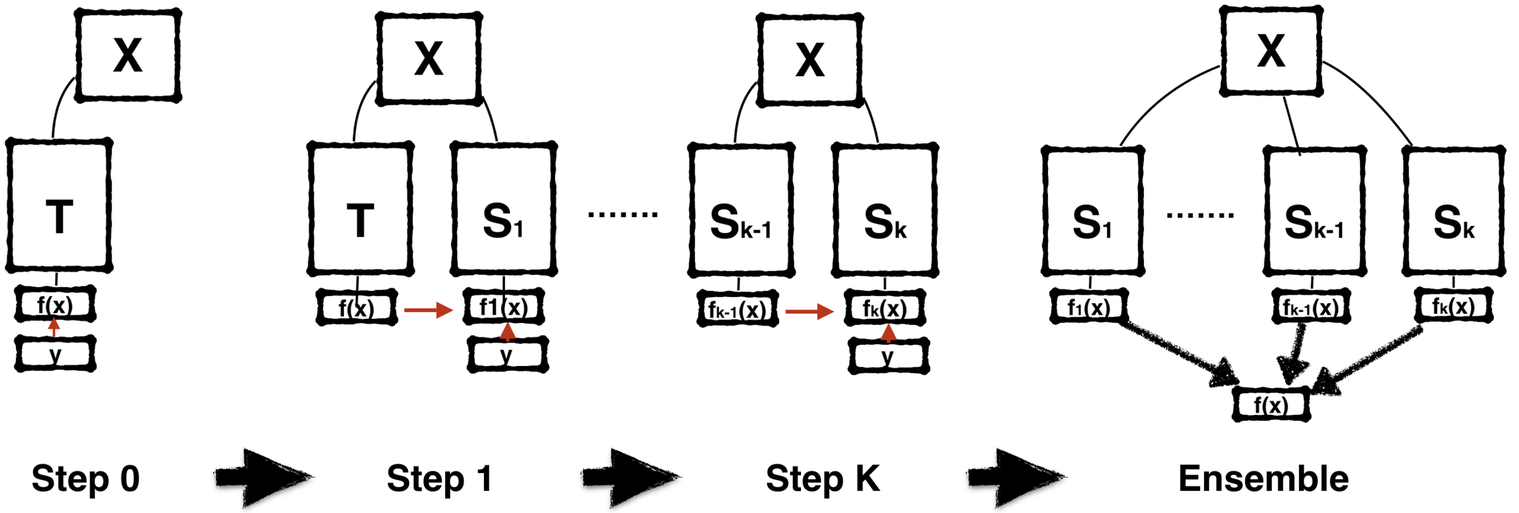
Teacher model is usually larger than the student model and is fixed



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 generalizes defensive distillation by **adding iterative training stages** and **using both classification objective and distillation objective** in subsequent stages.
- Network architecture $T = S_1 = S_2 = \ldots = S_k$.
- Network accuracy $T < S_1 < S_2 < \ldots < S_k$.
- Can alternatively ensemble $T, S_1, S_2, \ldots, S_k$ to get even better performance.

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

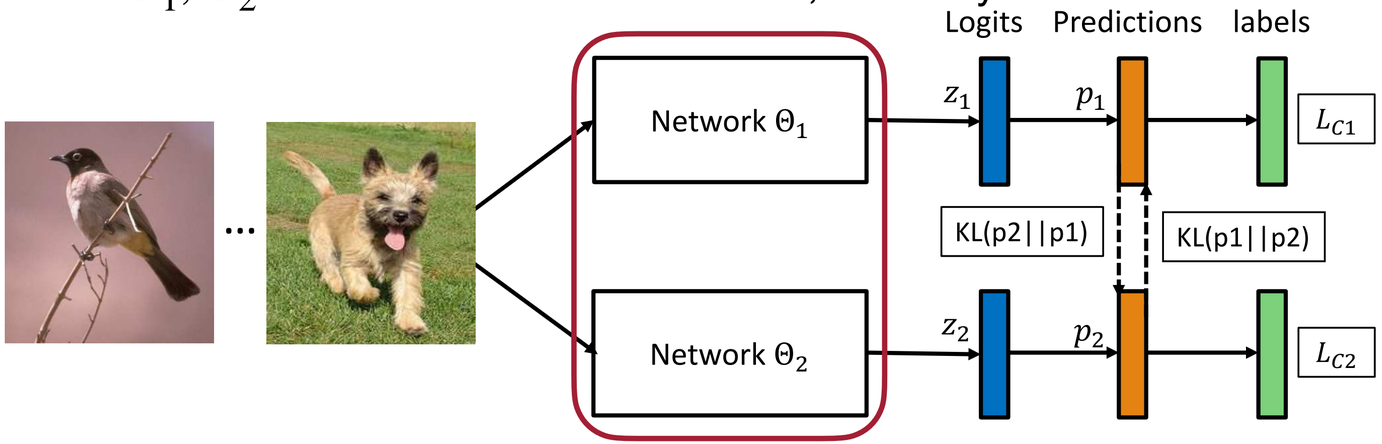
Self and Online Distillation

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.
- $\mathscr{L}(S) = \text{CrossEntropy}(S(I), y) + \text{KL}(S(I), T(I))$;
- $\mathscr{L}(T) = \text{CrossEntropy}(T(I), y) + \text{KL}(T(I), S(I))$.
- Note: it is not necessary to pretrain T and $S = T$ is allowed.

Deep Mutual Learning [Zhang *et al.*, CVPR 2018]

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.

Deep Mutual Learning [Zhang *et al.*, CVPR 2018]

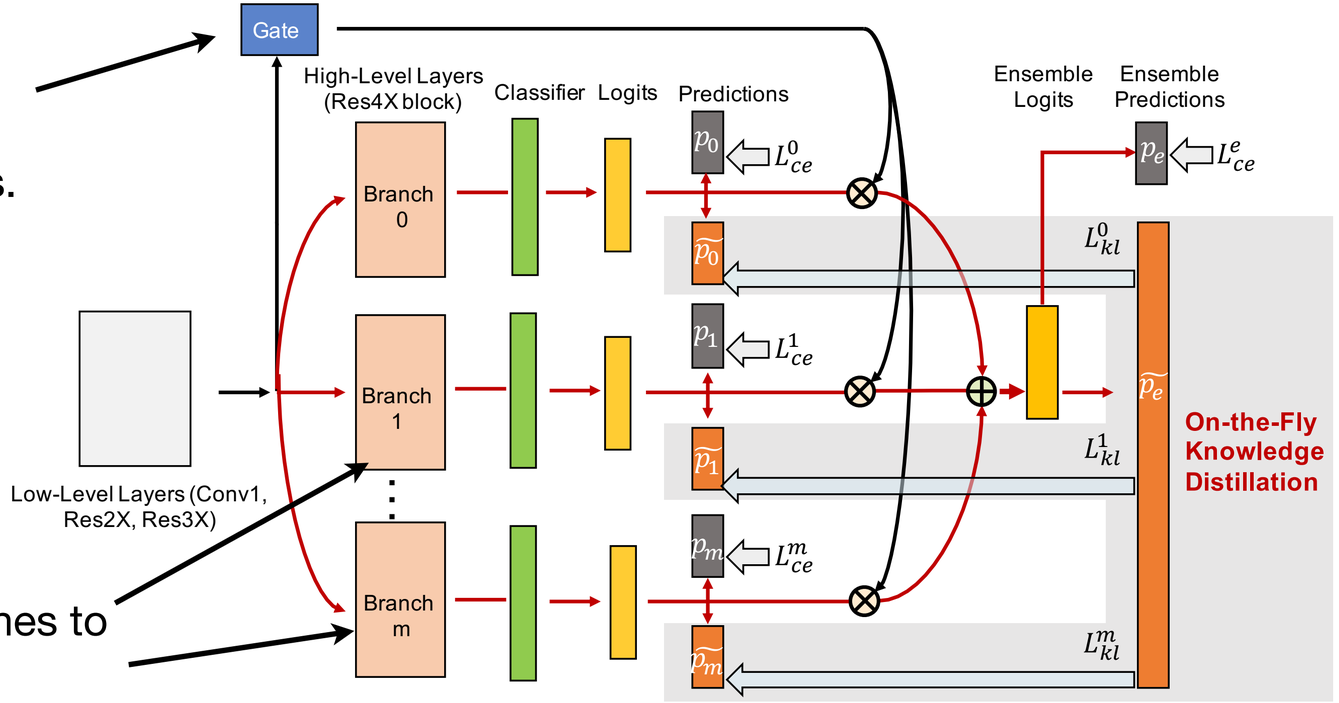
Self and Online Distillation

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

Combining Online and Self-Distillation

ONE: On-the-Fly Native Ensemble as the teacher network

Gate is used for ensembling logits from different branches.



Add more parallel branches to the backbone

Note: $\tilde{p}_i = \tilde{p}_e$

- Idea: generating **multiple** output probability distributions and **ensemble** them as the target distribution for knowledge distillation.
- Similar to DML, ONE allows the teacher model to be exactly the same as the student model, and it does not require pretraining the teacher network first. It is also not necessary to train two models as in DML.

Knowledge Distillation by On-the-Fly Native Ensemble [Lan *et al.*, NeurIPS 2018]

Combining Online and Self-Distillation

ONE: On-the-Fly Native Ensemble as the teacher network

Method	Top-1	Top-5
ResNet-18 [4]	30.48	10.98
ResNet-18 + **ONE**	**29.45±0.23**	**10.41±0.12**
ResNeXt-50 [23]	22.62	6.29
ResNeXt-50 + **ONE**	**21.85±0.07**	**5.90±0.05**
SeNet-ResNet-18 [31]	29.85	10.72
SeNet-ResNet-18 + **ONE**	**29.02±0.17**	**10.13±0.12**

Error rates on ImageNet

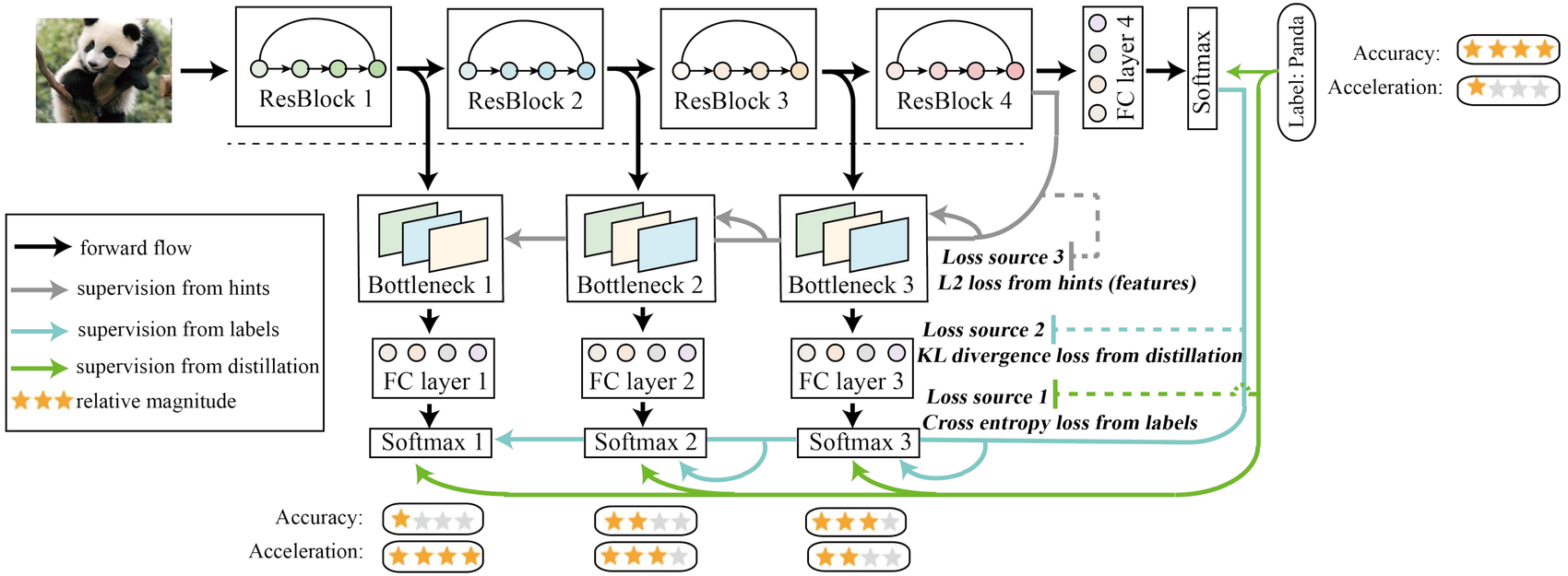
Target Network	ResNet-32			ResNet-110		
Metric	Error (%)	TrCost	TeCost	Error (%)	TrCost	TeCost
KD [10]	**28.83**	6.43	1.38	N/A	N/A	N/A
DML [17]	29.03±0.22*	**2.76**	1.38	**24.10±0.72**	**10.10**	5.05
ONE	**26.61±0.06**	**2.28**	1.38	**21.62±0.26**	**8.29**	5.05

Comparison with DML (TrCost: training cost, TeCost: testing cost)

Knowledge Distillation by On-the-Fly Native Ensemble [Lan *et al.*, NeurIPS 2018]

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

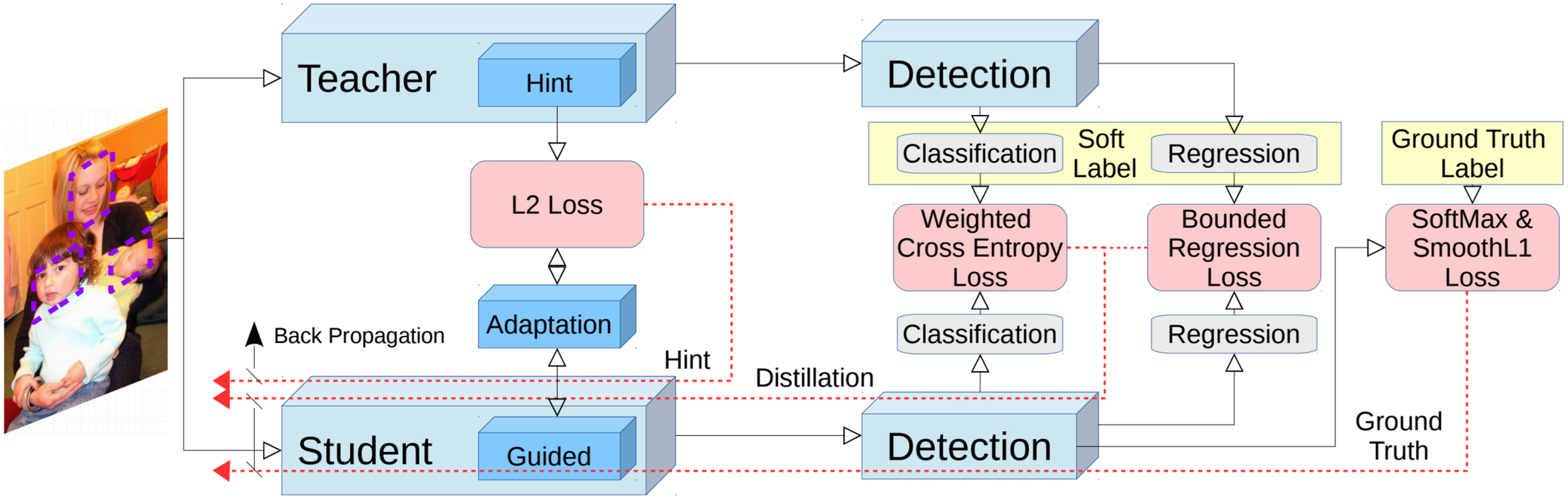
- Results on CIFAR100 shows consistent performance improvements over the baseline.
- In addition, predictions from intermediate classifiers (1/4, 2/4, 3/4) can sometimes outperform the baseline. As such, inference efficiency can be improved.

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

Distillation for Other Applications

KD for Object Detection

Feature Imitation

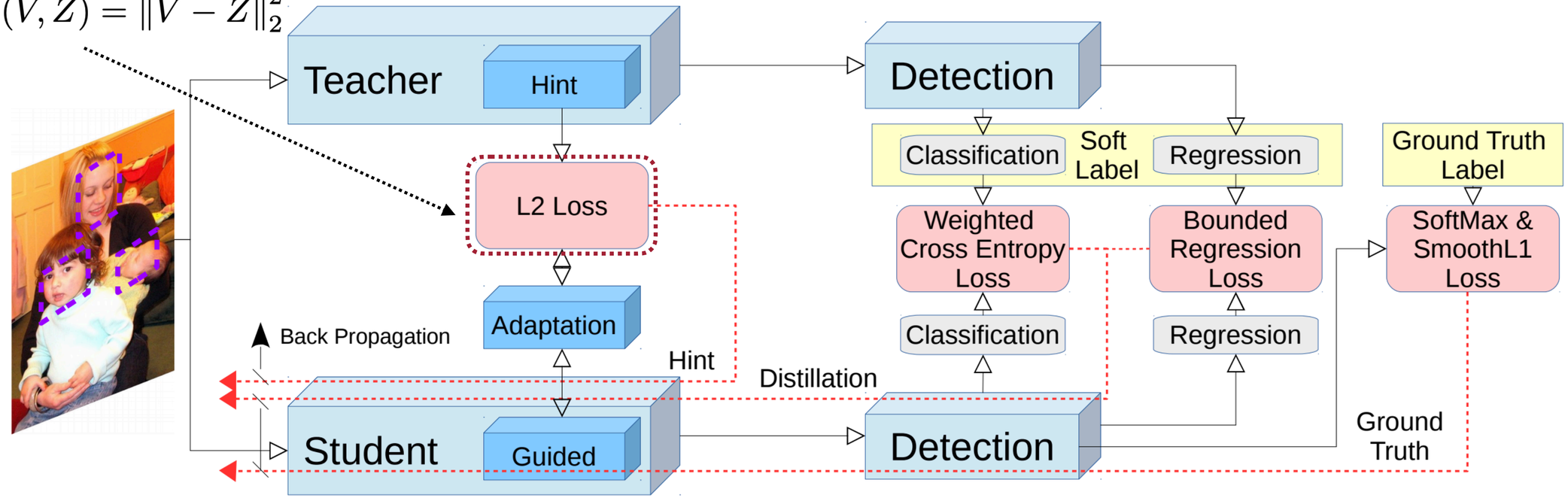


Learning Efficient Object Detection Models with Knowledge Distillation [Chen *et al.*, NeurIPS 2017]

KD for Object Detection

Feature Imitation

$$L_{Hint}(V, Z) = \|V - Z\|_2^2$$



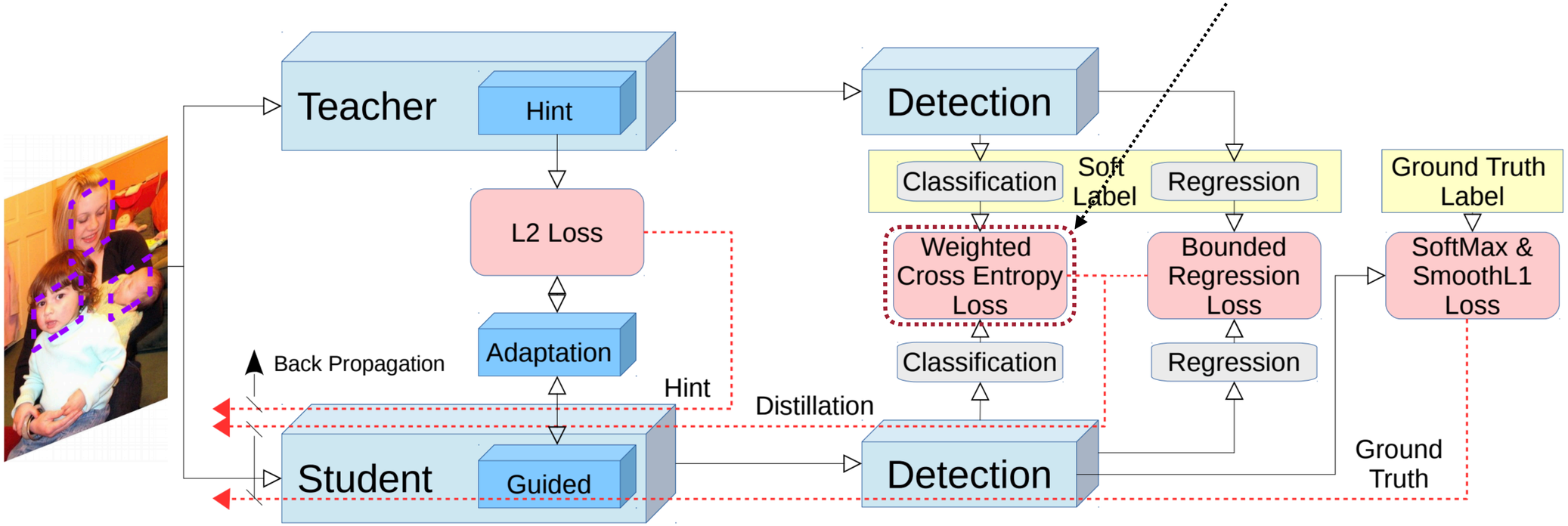
- Add a 1x1 convolution layer to match the shape.

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

KD for Object Detection

Feature Imitation

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



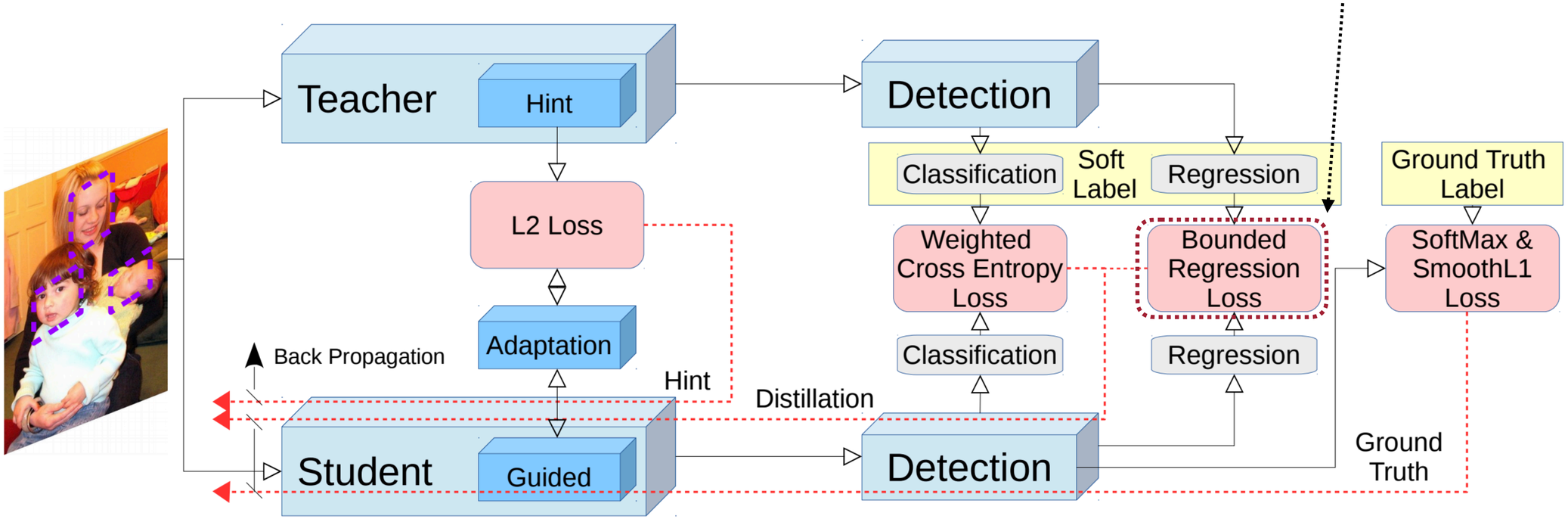
- Use different weights for foreground and background classes to handle the class imbalance problem.

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

KD for Object Detection

Feature Imitation

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

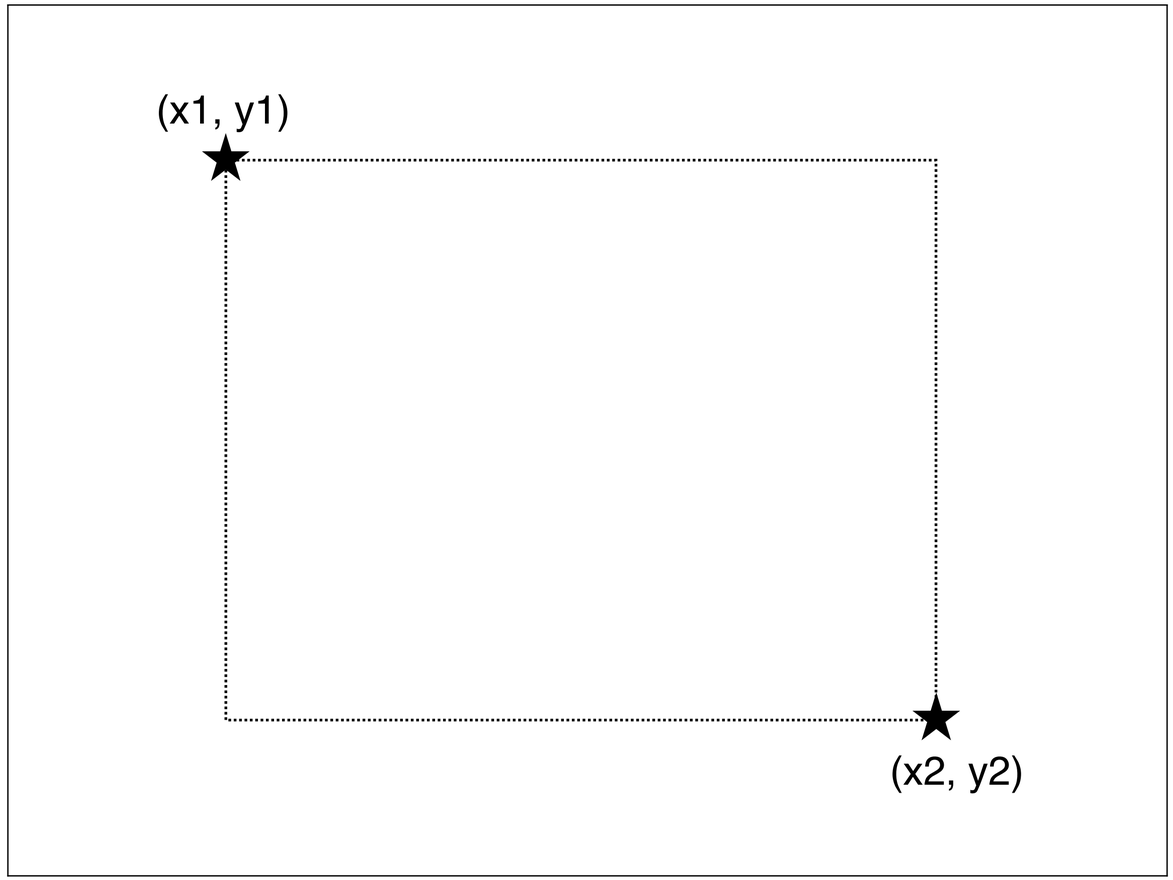


- Exploit teacher's prediction as an upper bound for the student to achieve. Once the quality of the student surpasses that of the teacher with a certain margin, the loss becomes zero.

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

KD for Object Detection

Convert bounding box regression to classification problem

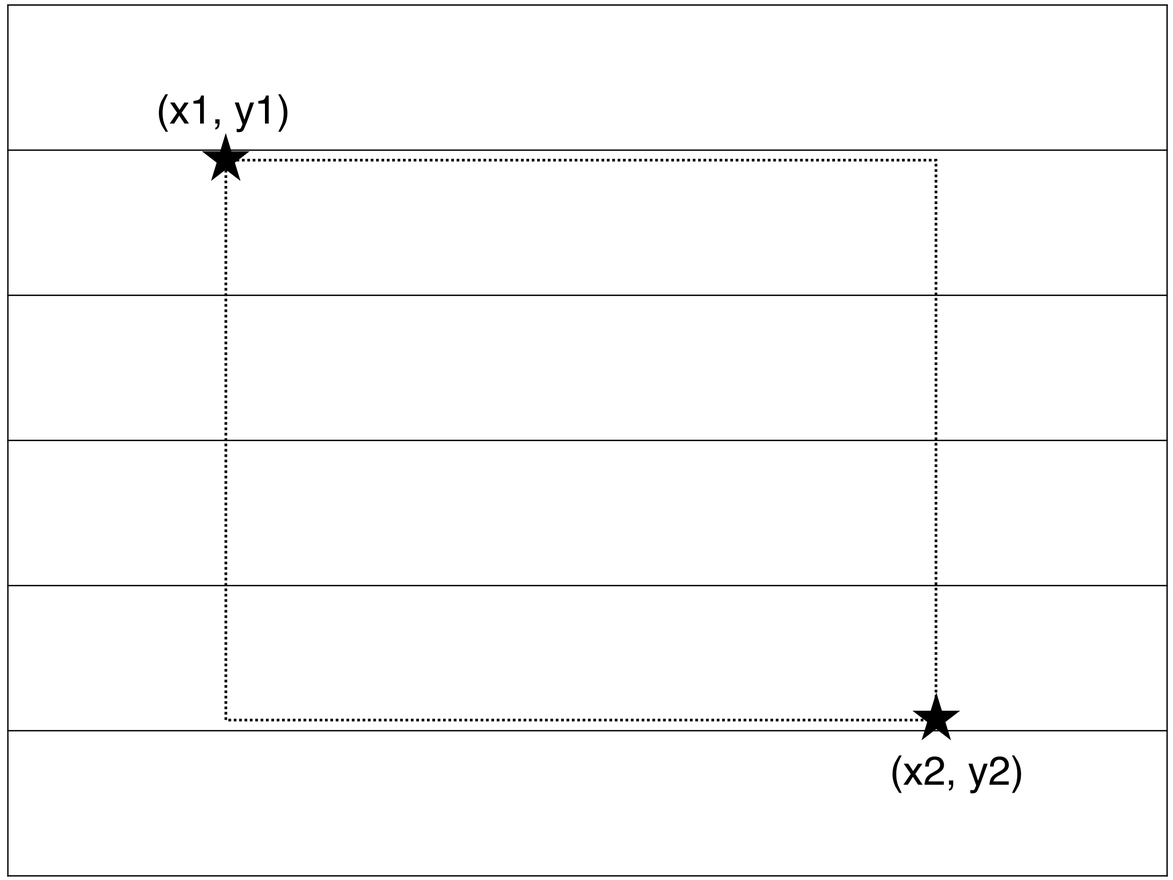


Localization Distillation for Dense Object Detection [Zheng *et al.*, CVPR 2022]

KD for Object Detection

Convert bounding box regression to classification problem

Divide the y-axis into 6 bins

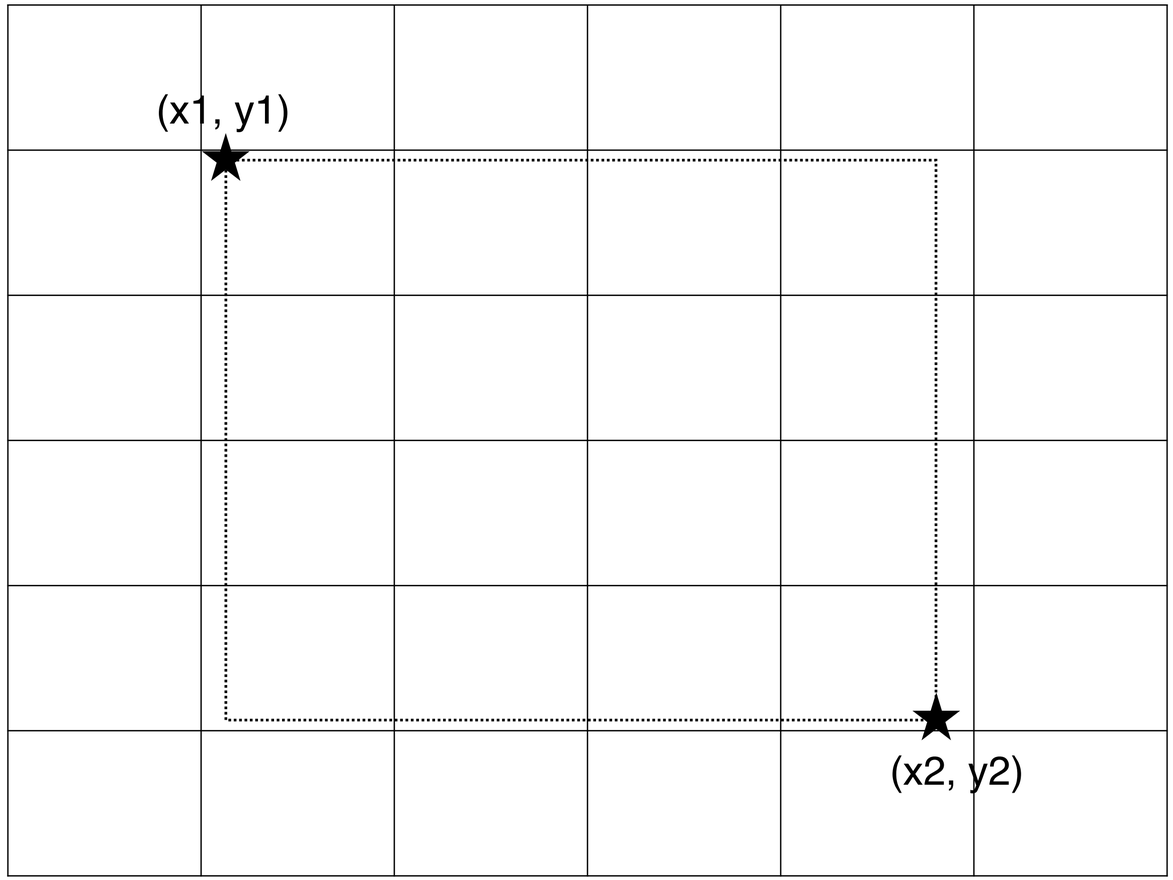


Localization Distillation for Dense Object Detection [Zheng *et al.*, CVPR 2022]

KD for Object Detection

Convert bounding box regression to classification problem

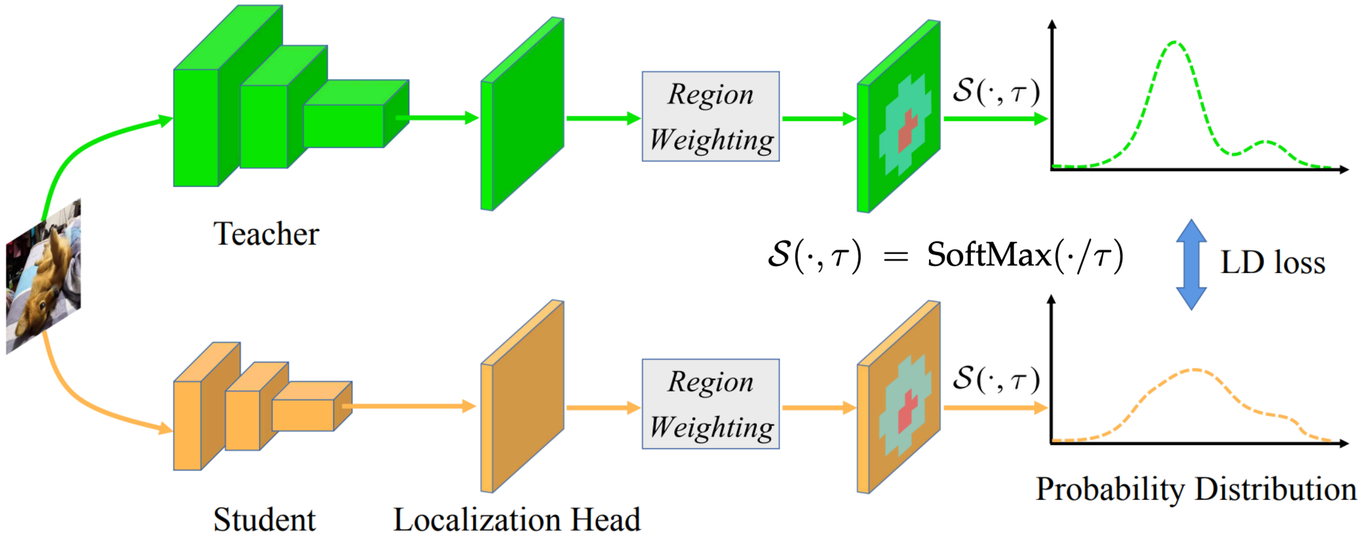
Divide the y-axis into 6 bins & divide the x-axis into 6 bins



Localization Distillation for Dense Object Detection [Zheng *et al.*, CVPR 2022]

KD for Object Detection

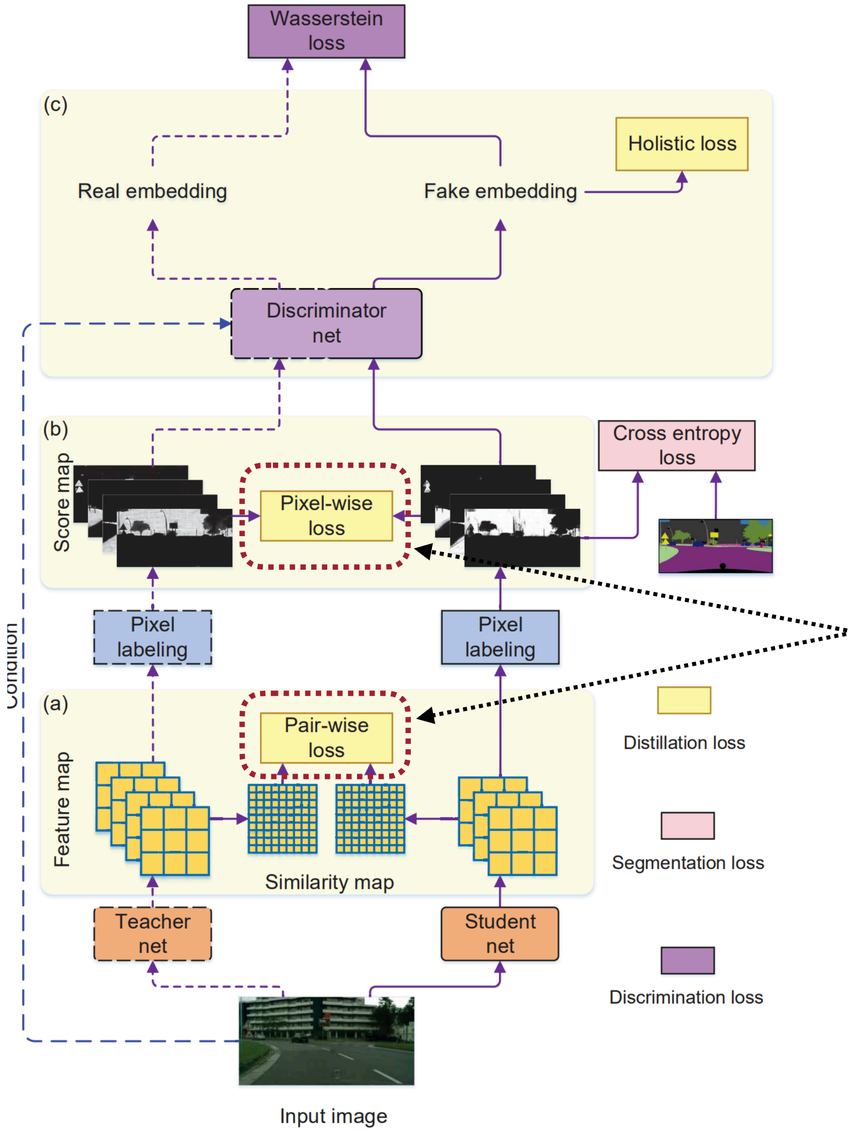
Localization Distillation



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

Localization Distillation for Dense Object Detection [Zheng *et al.*, CVPR 2022]

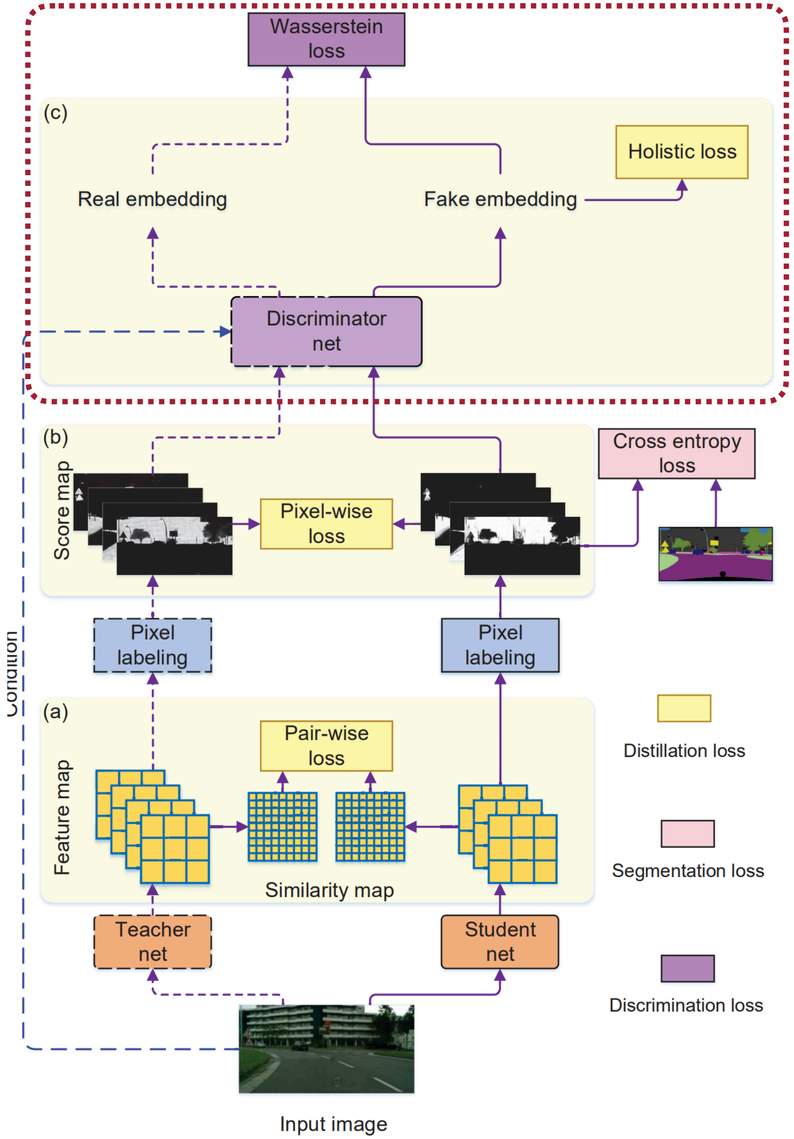
KD for Semantic Segmentation



Feature imitation similar to classification and detection

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

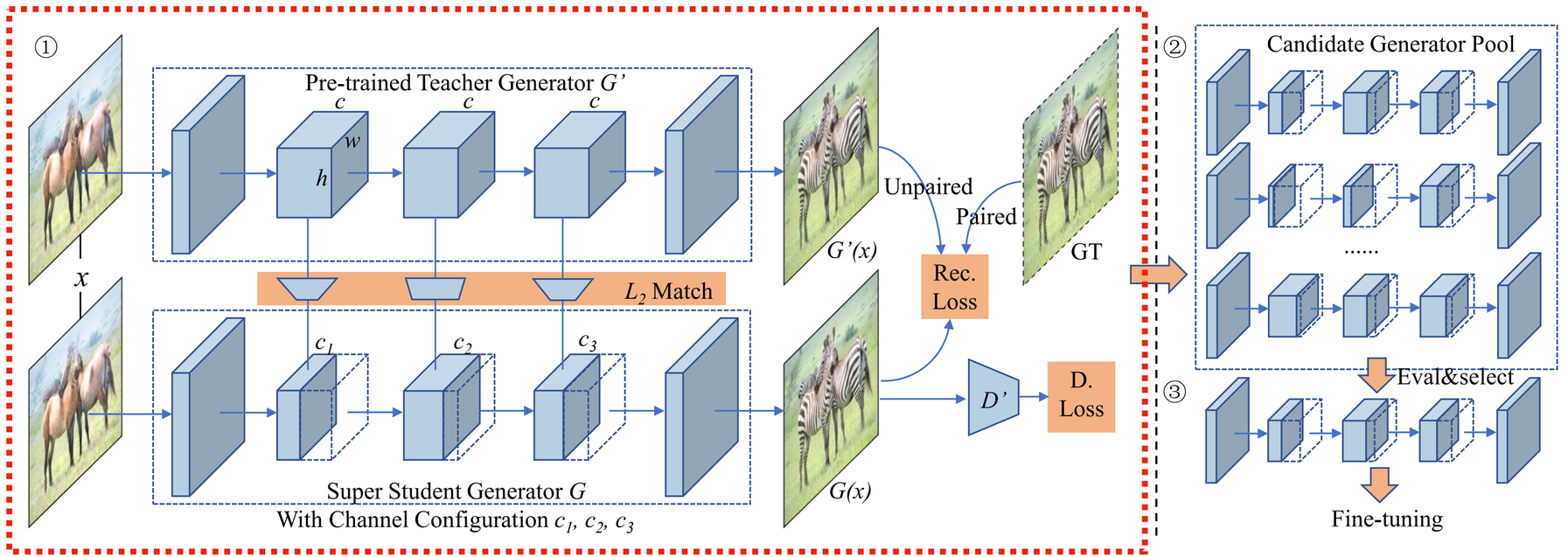
KD for Semantic Segmentation



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



Reconstruction Loss

$$\mathcal{L}_{\text{recon}} = \begin{cases} \|G(x) - y\| & \text{paired cGANs} \\ \|G(x) - G'(x)\| & \text{unpaired cGANs} \end{cases}$$

Distillation Loss

$$\mathcal{L}_{\text{distill}} = \sum_{k=1}^{n} \|G_k(x) - f_k(G'_k(x))\|$$

cGAN Loss

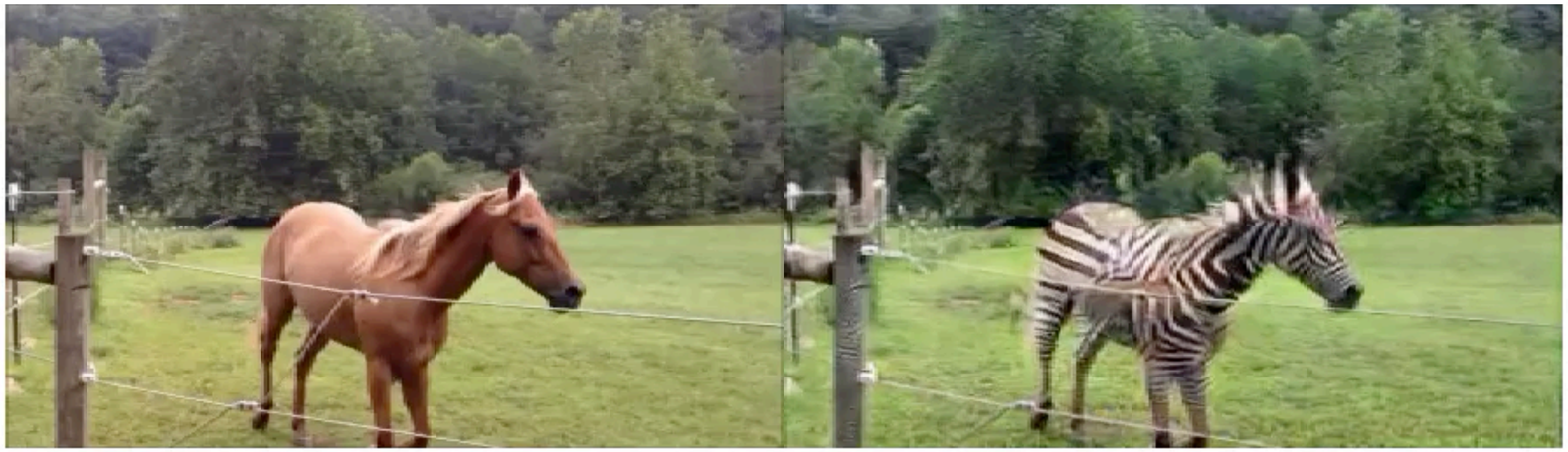
$$\mathcal{L}_{\text{cGAN}} = \mathbb{E}_{x,y}[\log D(x,y)] + \mathbb{E}_x[\log(1 - D(x, G(x)))]$$

Training Objective

$$\mathcal{L}(x) = \mathcal{L}_{\text{cGAN}}(x) + \lambda_{\text{recon}}\mathcal{L}_{\text{recon}}(x) + \lambda_{\text{distill}}\mathcal{L}_{\text{distill}}(x)$$

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

Demos on Horse2zebra Dataset



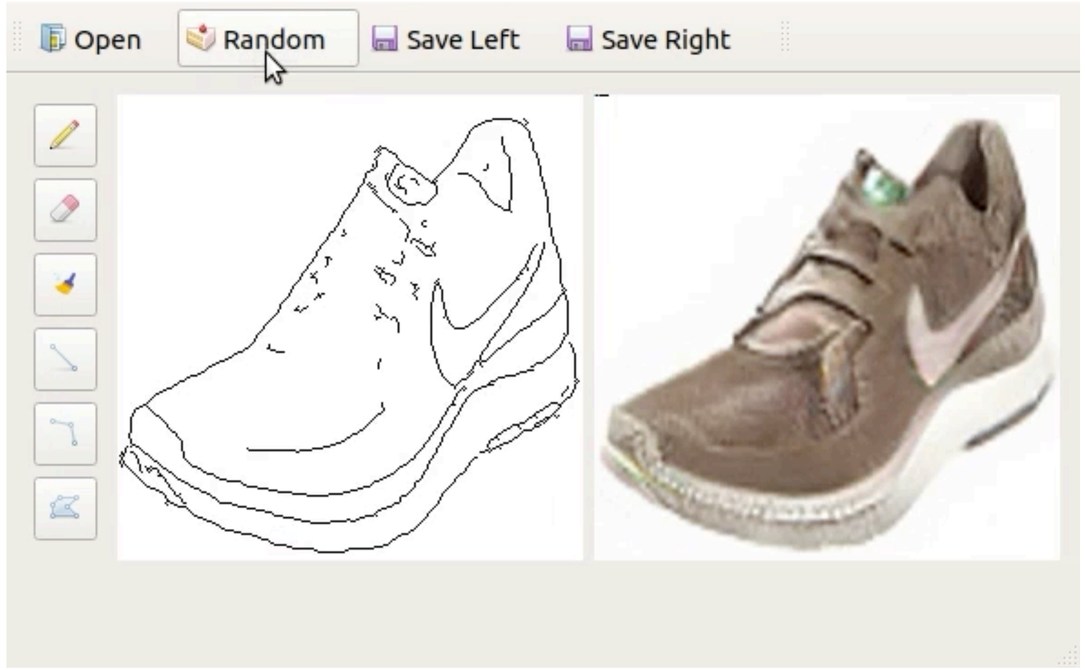
Original CycleGAN; FLOPs: 56.8G; FPS: 12.1; FID: 61.5

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



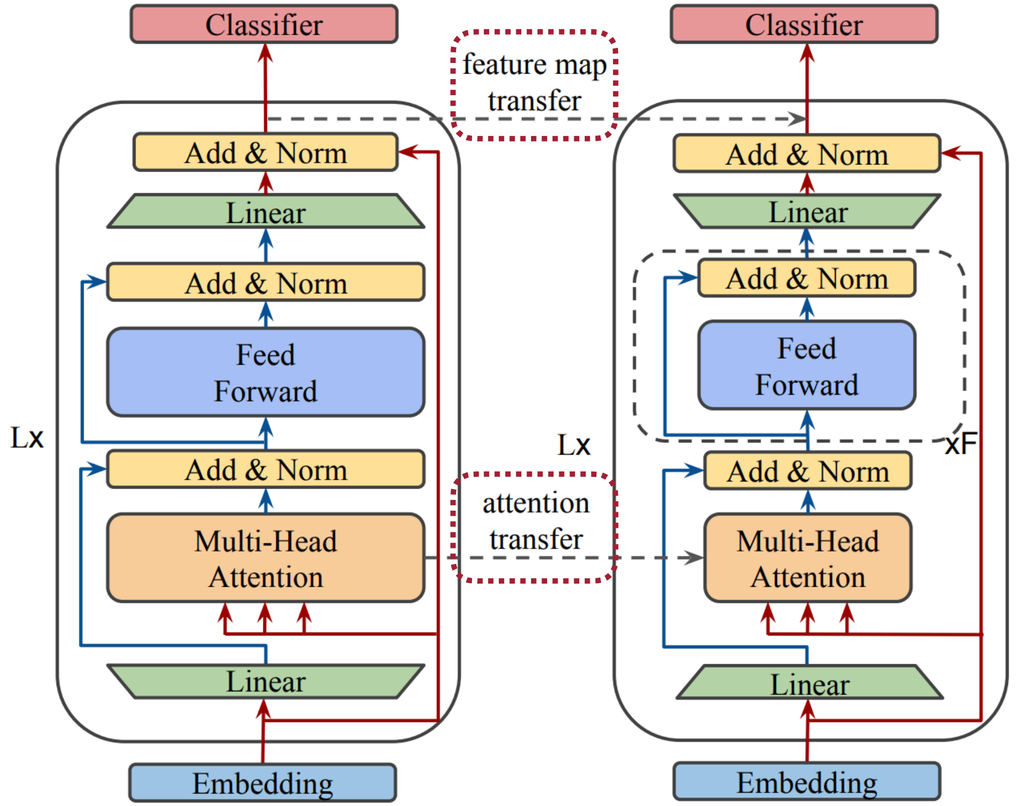
GAN Compression; MACs: 4.81G (11.8x); FPS: 3.9 (2.5x); FID: 26.6

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 train to mimic teacher model's attention maps.

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

Network Augmentation

Conventional Approach

Data augmentation during training to avoid overfitting

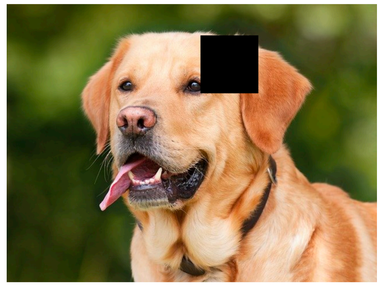
Data Augmentation

Conventional Approach

Data augmentation during training to avoid overfitting

Data Augmentation

Cutout



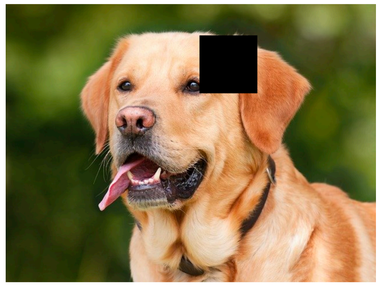
Improved Regularization of Convolutional Neural Networks with Cutout [DeVries *et al.*, arXiv 2017]

Conventional Approach

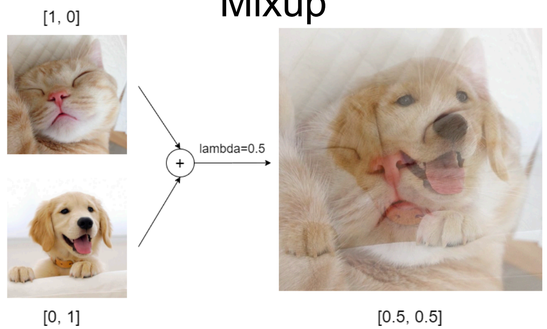
Data augmentation during training to avoid overfitting

Data Augmentation

Cutout



Mixup



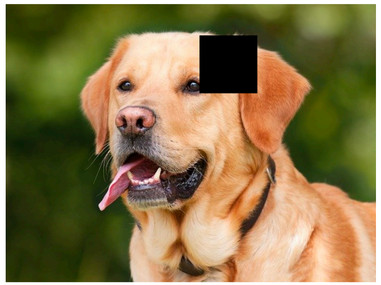
mixup: Beyond Empirical Risk Minimization [Zhang *et al.*, ICLR 2018]

Conventional Approach

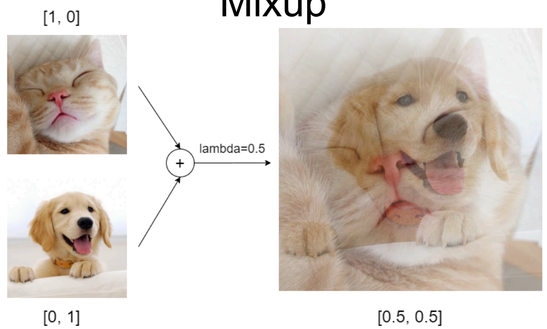
Data augmentation during training to avoid overfitting

Data Augmentation

Cutout



Mixup



AutoAugment

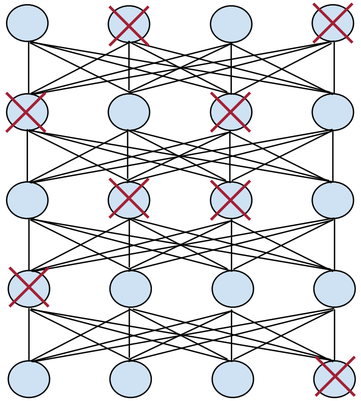
AutoAugment: Learning Augmentation Policies from Data [Cubuk *et al.*, CVPR 2019]

Conventional Approach

Dropout during training to avoid overfitting

Data Augmentation

Dropout

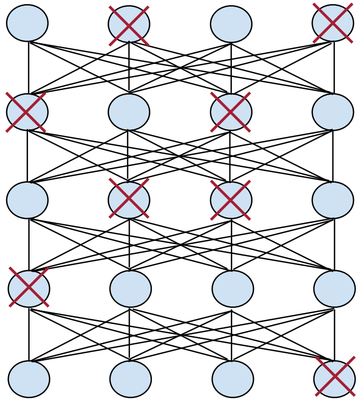


Conventional Approach

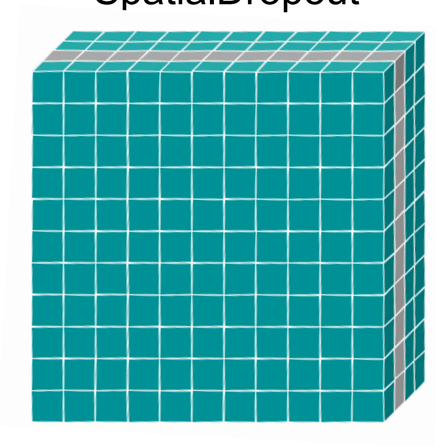
Dropout during training to avoid overfitting

Data Augmentation

Dropout



SpatialDropout



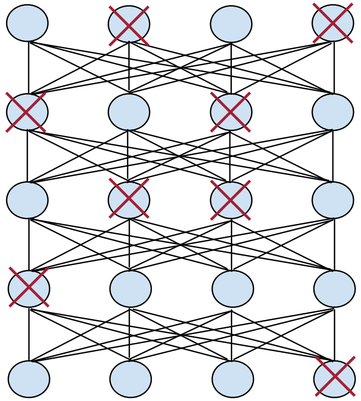
Efficient Object Localization Using Convolutional Networks [Tompson *et al.*, CVPR 2015]

Conventional Approach

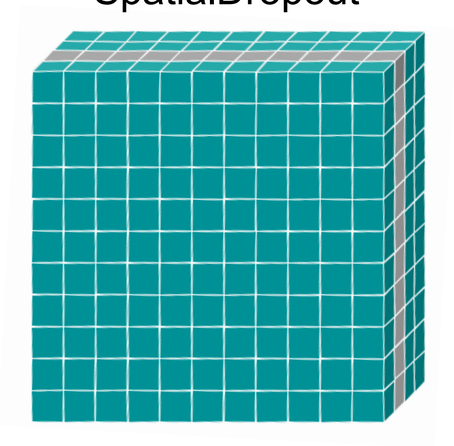
Dropout during training to avoid overfitting

Data Augmentation

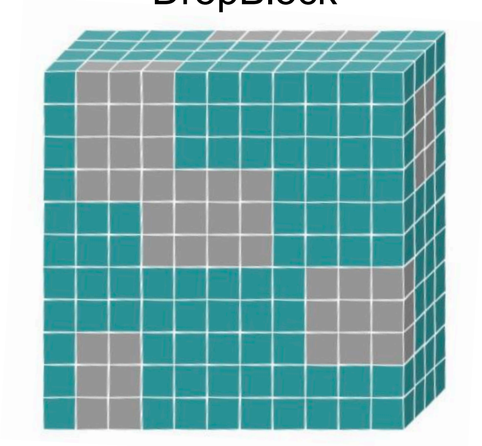
Dropout



SpatialDropout



DropBlock

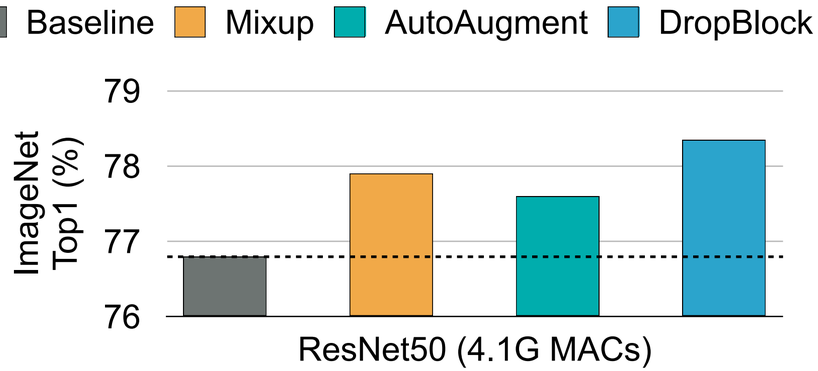


DropBlock: A regularization method for convolutional networks [Ghiasi *et al.*, NeurIPS 2018]

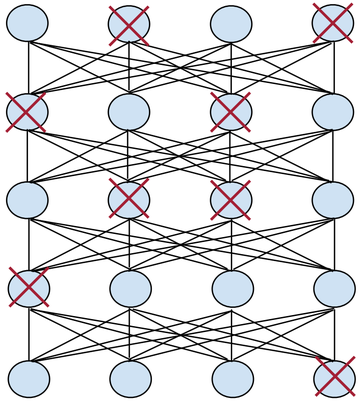
Conventional Approach

Dropout/Data augmentation improves large neural networks' performance

Data Augmentation



Dropout

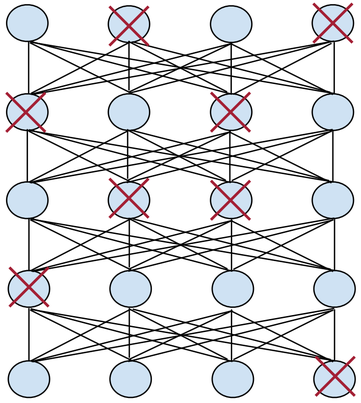


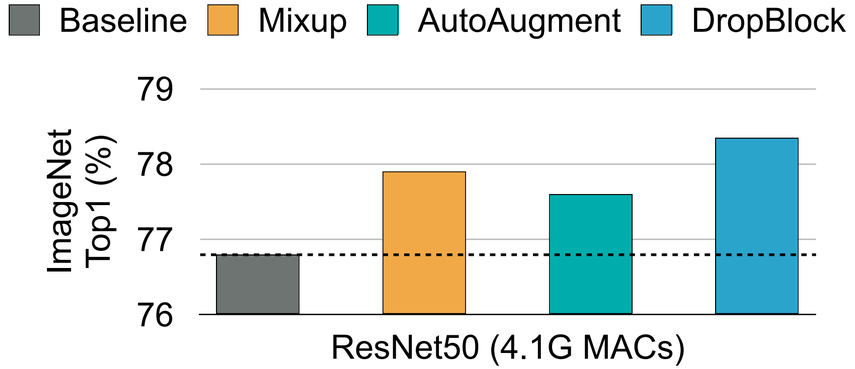
Conventional Approach

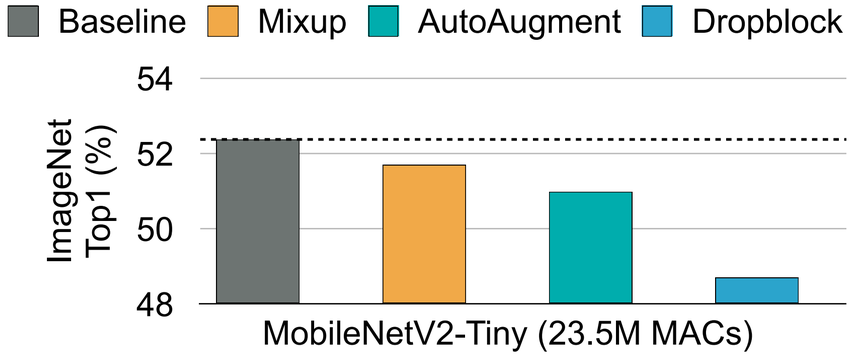
Dropout/Data augmentation hurts tiny neural networks' performance

Data Augmentation

Dropout



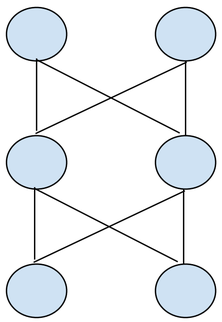


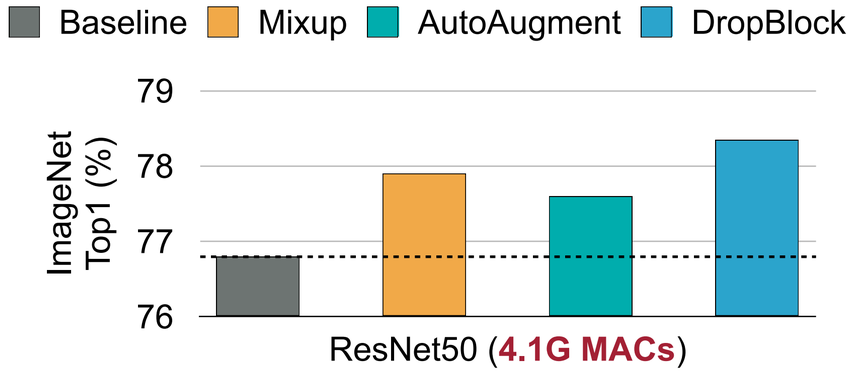


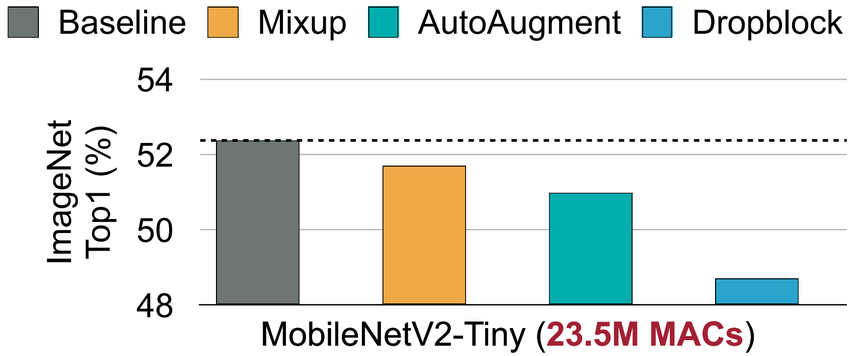
Conventional Approach

Tiny neural network lacks capacity

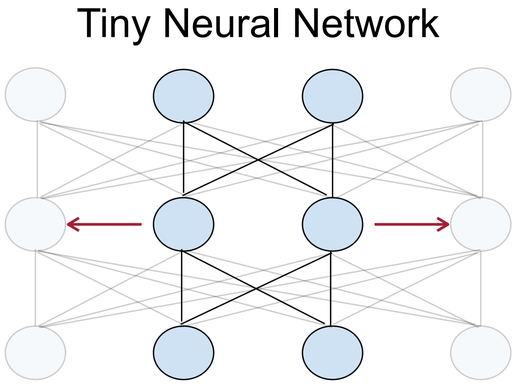
Tiny Neural Network

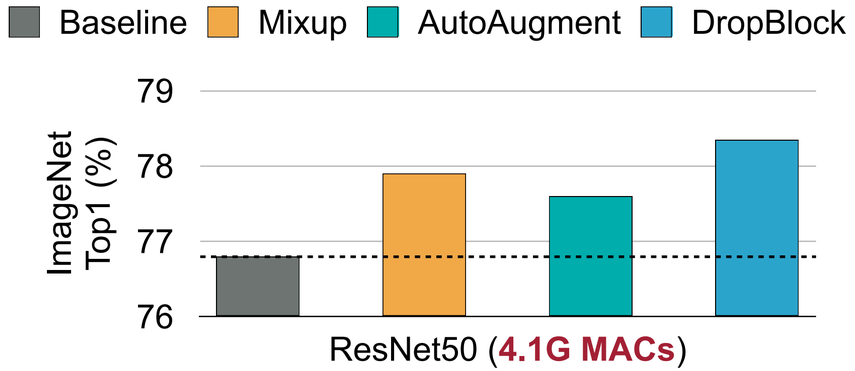


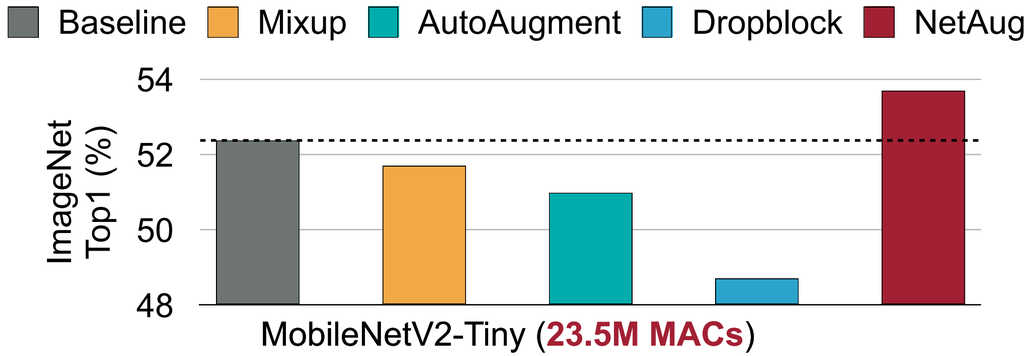




Network Augmentation



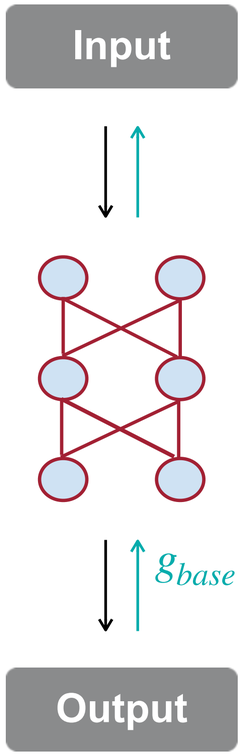




- Augment the model to get extra supervision during training for tiny models.

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

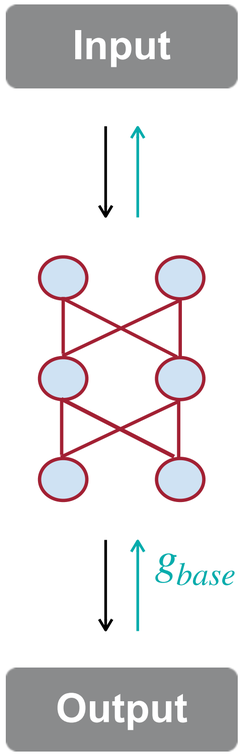
NetAug: Training Process



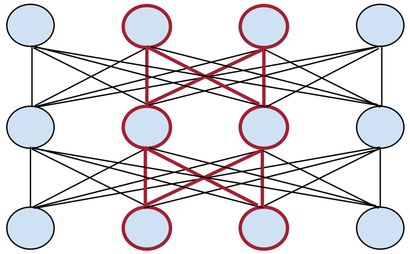
$$\mathcal{L}_{\text{aug}} = \underbrace{\mathcal{L}(W_{\text{base}})}_{\text{base supervision}}$$

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

NetAug: Training Process



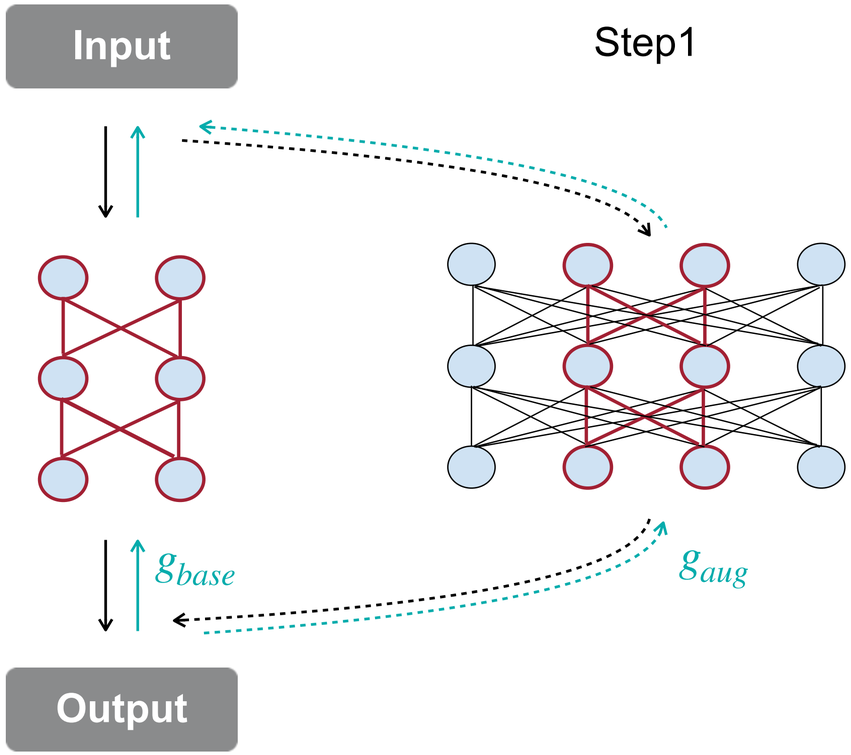
Build an Augmented Model



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

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

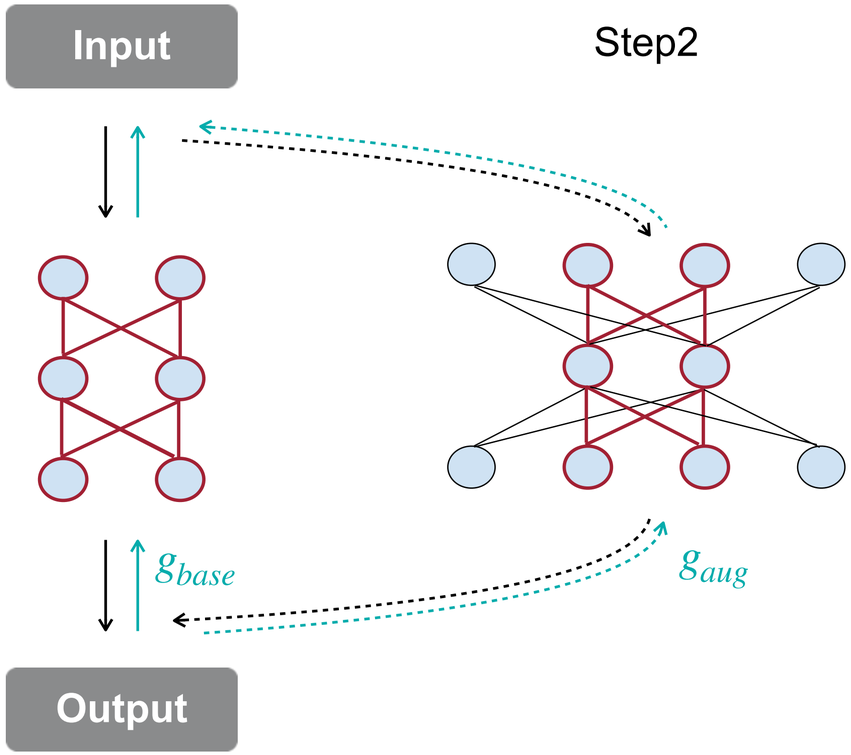
NetAug: Training Process



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

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

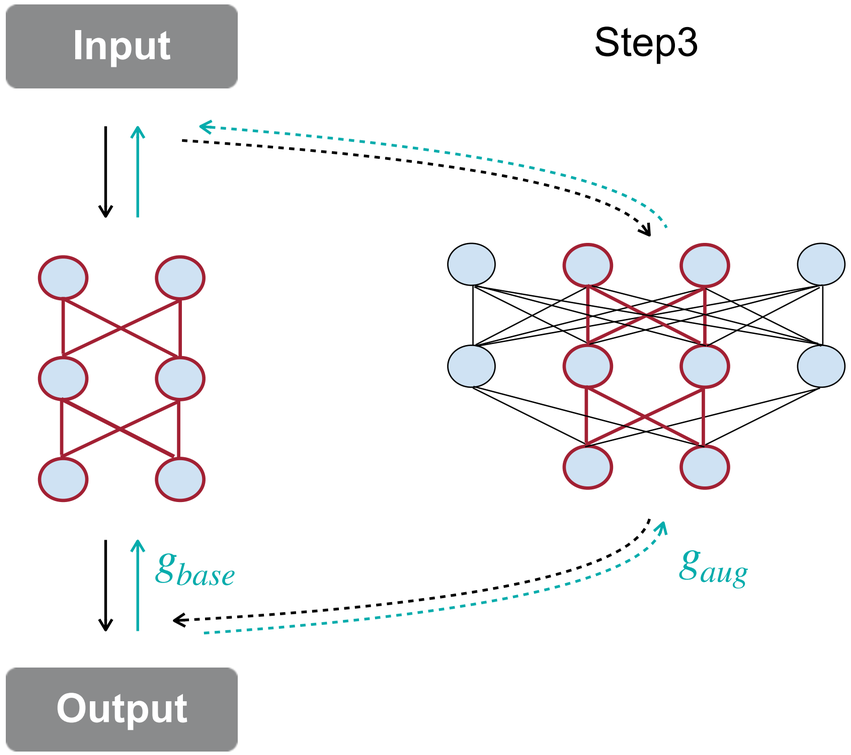
NetAug: Training Process



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

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

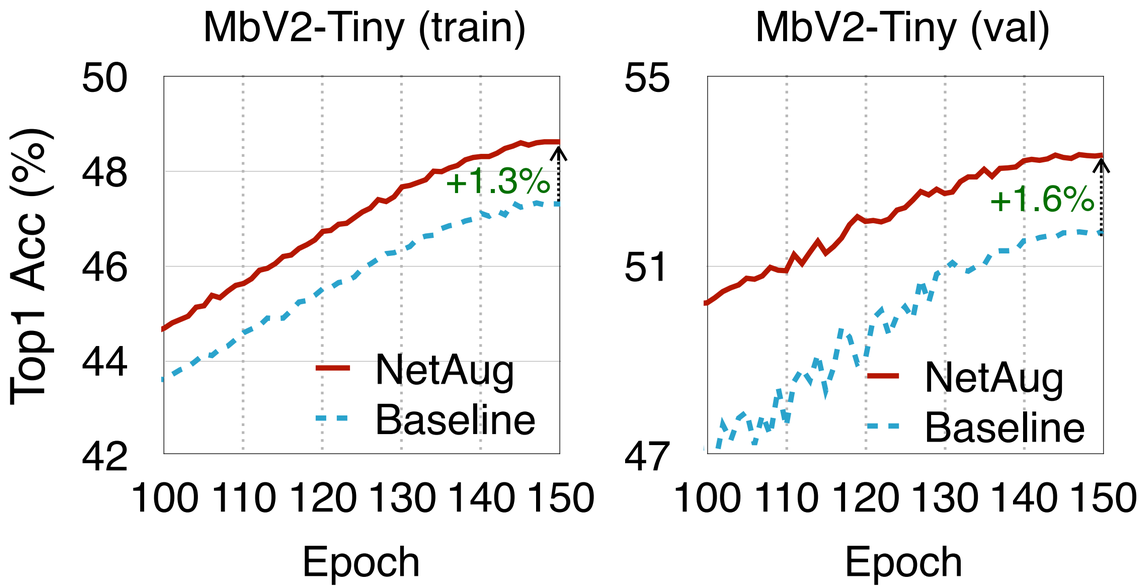
NetAug: Training Process



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

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

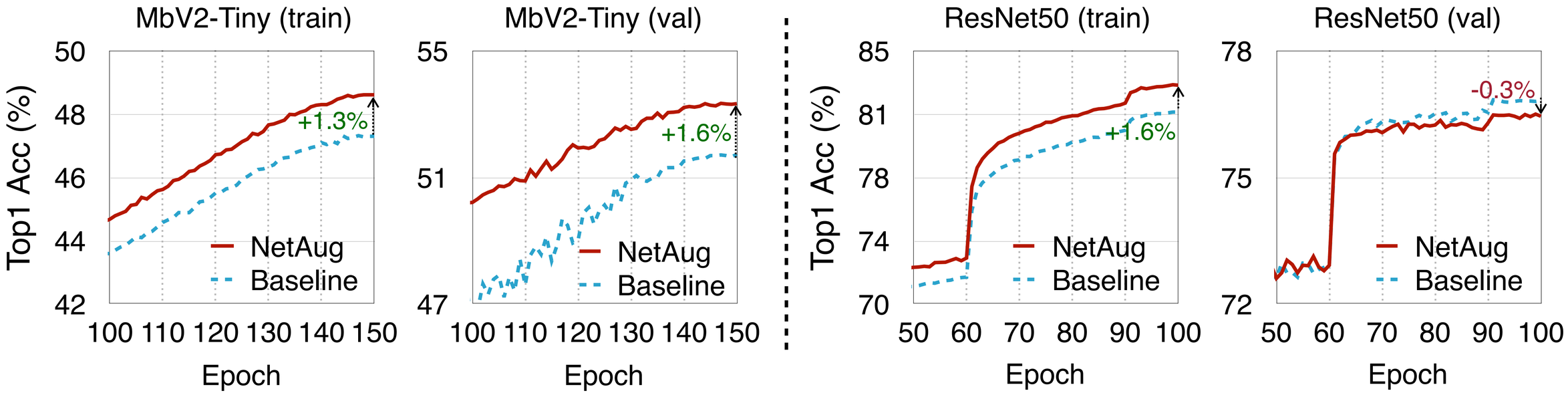
NetAug: Learning Curve



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

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

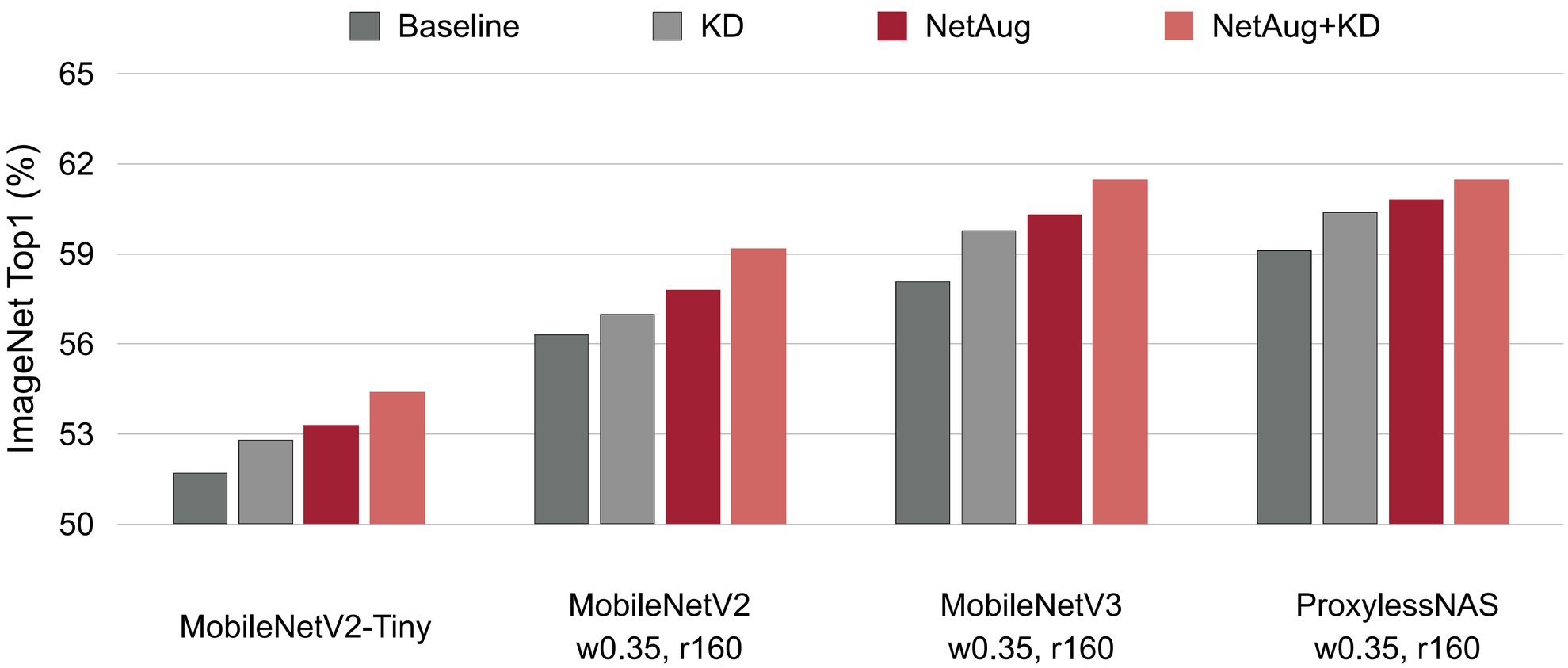
NetAug: Learning Curve



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

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

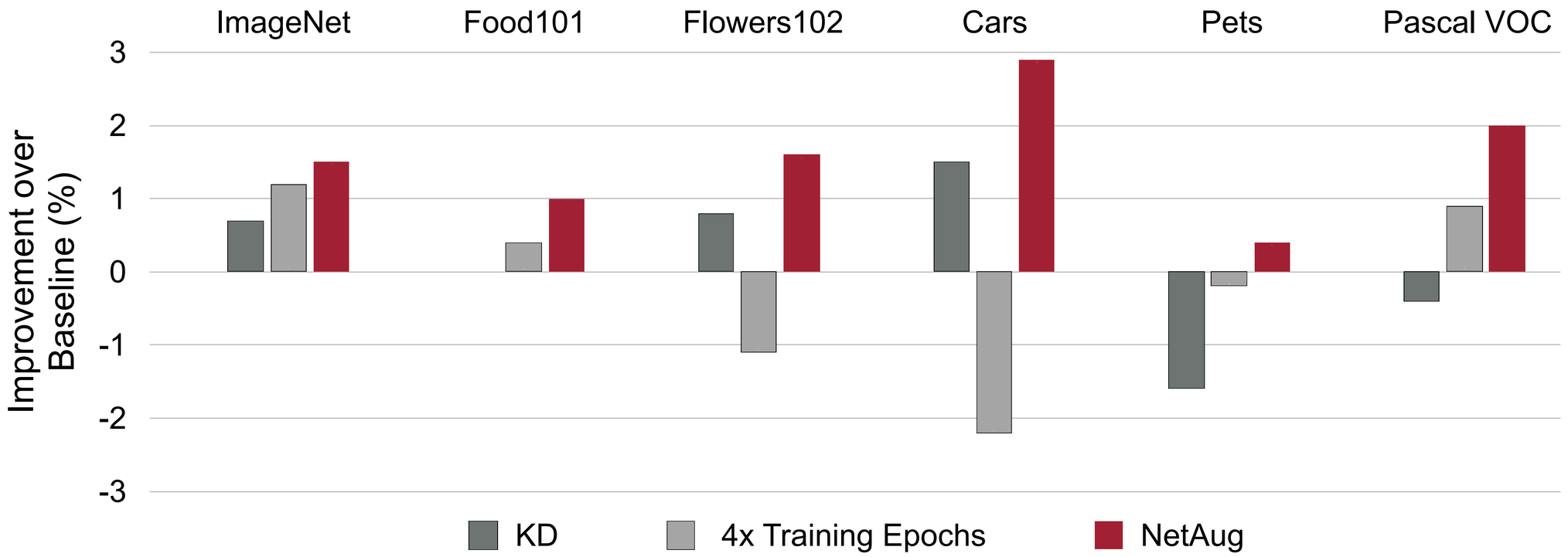
NetAug: Results



- NetAug is orthogonal to KD.

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

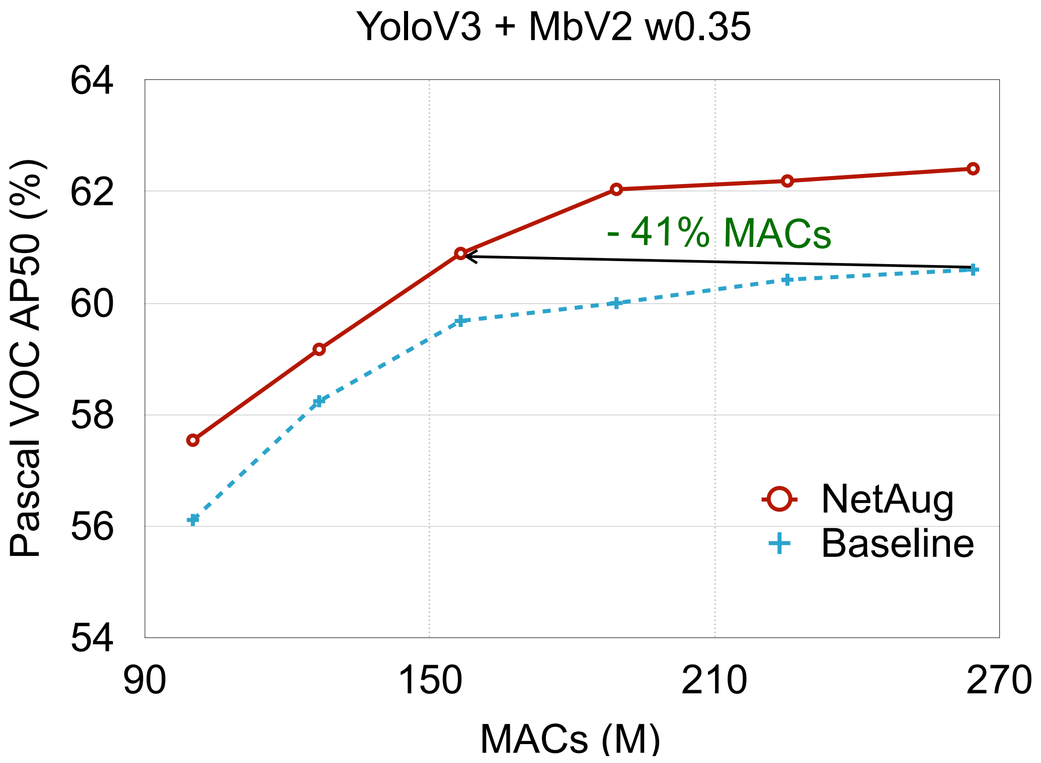
NetAug: Transfer Learning

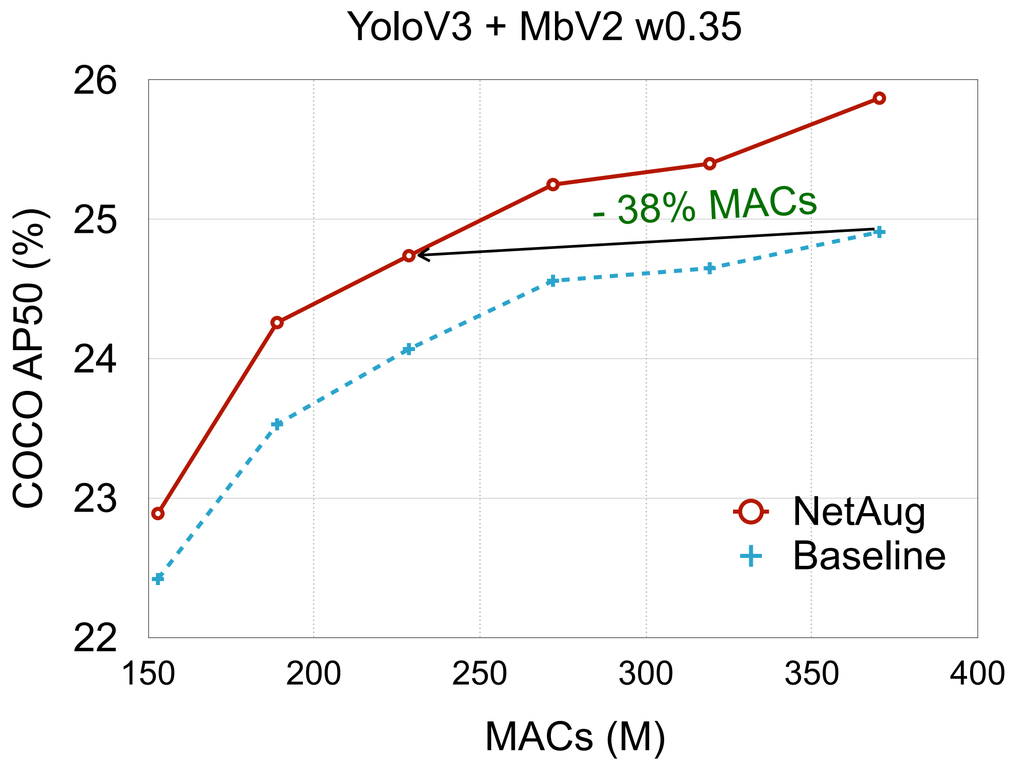


- NetAug provides better transfer learning performances than KD and 4x training schedule, thought their ImageNet performances are similar.

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

NetAug: Transfer to Object Detection

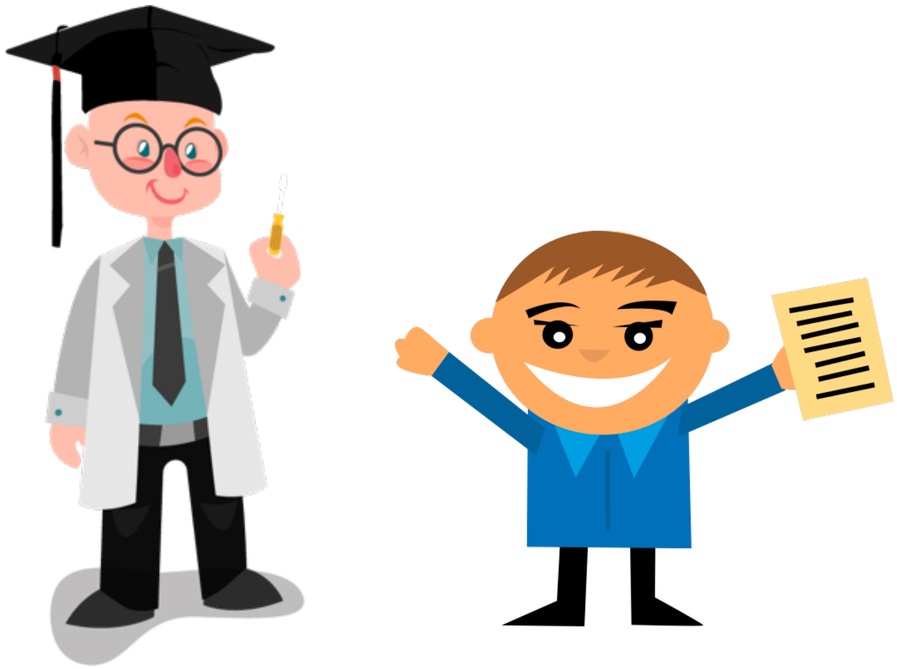




Network Augmentation for Tiny Deep Learning [Cai *et al.*, ICLR 2022]

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]