Knowledge Distillation from A Stronger Teacher

Tao Huang[1,2] Shan You[1] Fei Wang[3] Chen Qian[1] Chang Xu[2]
[1]SenseTime Research [2]School of Computer Science, Faculty of Engineering, The University of Sydney
[3]University of Science and Technology of China
Correspondence to: youshan@sensetime.com

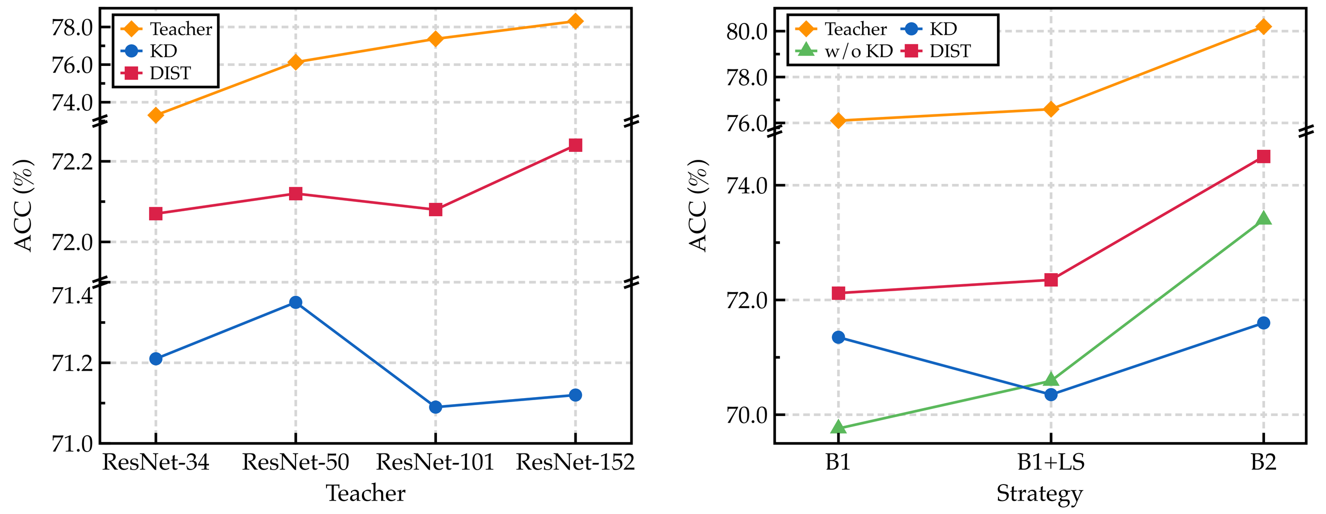
Motivation

Stronger Teacher: Current KD methods mainly focus on baseline training settings, while today's state-of-the-art approaches are using much stronger models and training strategies.

- **Stronger models:** larger capacity, advanced architectures *e.t.c.*
- **Stronger strategies:** auto augmentation, MixUp, AdamW optimizer, *e.t.c.*

Frustrating Performance of KD from a Stronger Teacher: We train the student with stronger teachers in vanilla KD (KL div.).

- **Larger teachers:** the ACCs of KD with R152 and R101 are lower than R34.
- **Stronger strategies:** the ACCs of KD with stronger strategies are even lower than standalone training.



What makes stronger teachers abnormal compared to baselines?

Catastrophic Discrepancy with A Stronger Teacher

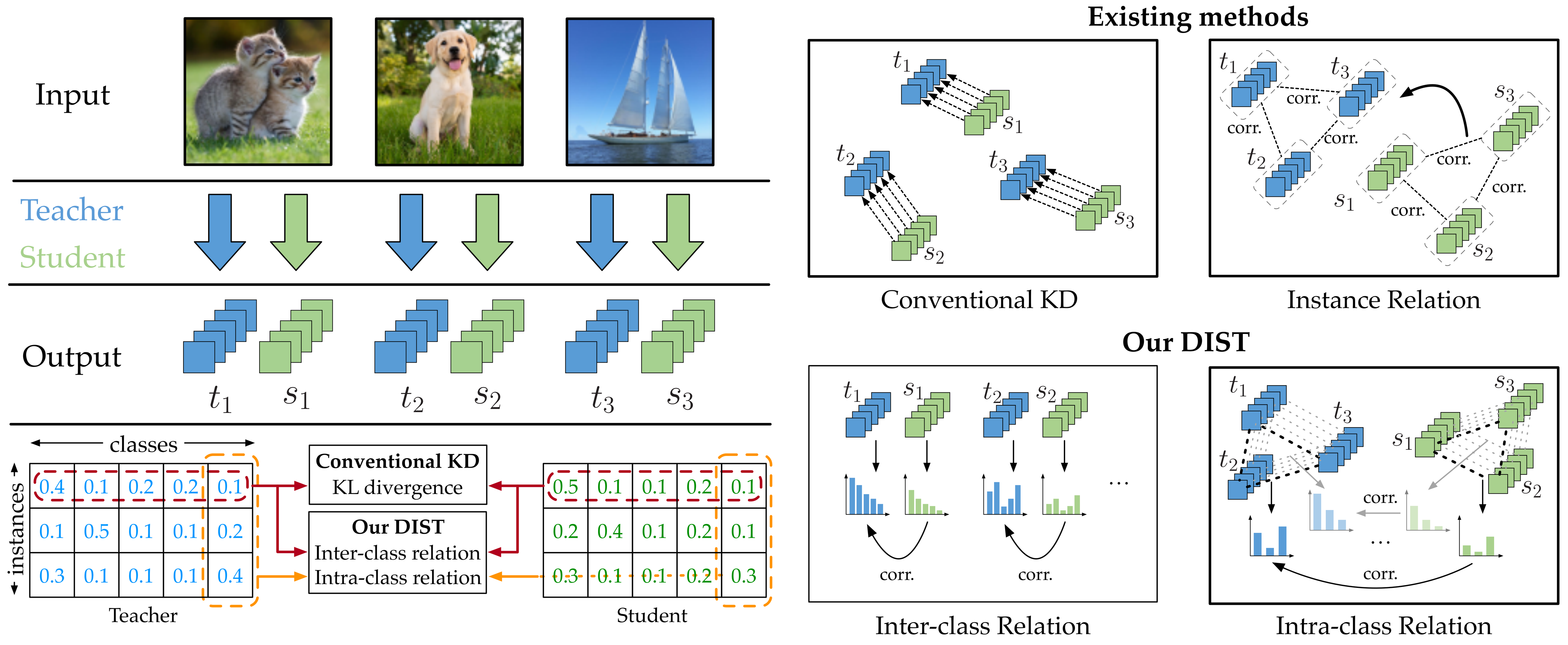
By measuring the outputs of trained baseline and stronger models, we find that

- It tends to be fairly challenging for the student to exactly match the teacher's outputs as their discrepancy becomes larger.
- When the teacher and student are trained with stronger strategies, their discrepancy would be larger.

The exact match in KL divergence seems way too overambitious and demanding when the discrepancy becomes large.

Intuition: Relax the match with relations.

Relaxed Match with DIST



DIST replaces KL divergence with Pearson distance

$$d_p = 1 - \rho_p(\boldsymbol{u}, \boldsymbol{v}) := \frac{\mathrm{Cov}(\boldsymbol{u}, \boldsymbol{v})}{\mathrm{Std}(\boldsymbol{u})\mathrm{Std}(\boldsymbol{v})}.$$

Experiments

Baseline settings on ImageNet:

Stu. (Tea.)	Tea.	Stu.	KD	CRD	Review	DIST
Res18 (Res34)	73.31	69.76	70.66	71.17	71.61	**72.07**
MBV1 (Res50)	76.16	70.13	70.68	71.37	72.56	**73.24**

Stronger teachers:

Tea.	Stu.	tea.	stu.	KD	RKD	SRRL	DIST
Res50SB	Res18	80.1	73.4	72.6	72.9	71.2	**74.5**
	Res34		76.8	77.2	76.6	76.7	**77.8**
	MBV2		73.6	71.7	73.1	69.2	**74.4**
	Eff.B0		78.0	77.4	77.5	77.3	**78.6**
Swin-L‡	ResNet-50	86.3	78.5	80.0	78.9	78.6	**80.2**
	Swin-T		81.3	81.5	81.2	81.5	**82.3**

Comparisons of training speed (batches / second):

KD	RKD	SRRL	CRD	DIST
14.28	11.11	12.98	8.33	14.19

Pytorch implementation of DIST:

```python
import torch.nn as nn

def cosine_similarity(a, b, eps=1e-8):
    return (a * b).sum(1) / (a.norm(dim=1) * b.norm(dim=1) + eps)

def pearson_correlation(a, b, eps=1e-8):
    return cosine_similarity(a - a.mean(1).unsqueeze(1), b - b.mean(1).unsqueeze(1), eps)

def inter_class_relation(y_s, y_t):
    return 1 - pearson_correlation(y_s, y_t).mean()

def intra_class_relation(y_s, y_t):
    return inter_class_relation(y_s.transpose(0, 1), y_t.transpose(0, 1))

class DIST(nn.Module):
    def __init__(self, beta, gamma):
        super(DIST, self).__init__()
        self.beta = beta
        self.gamma = gamma

    def forward(self, z_s, z_t):
        y_s = z_s.softmax(dim=1)
        y_t = z_t.softmax(dim=1)
        inter_loss = inter_class_relation(y_s, y_t)
        intra_loss = intra_class_relation(y_s, y_t)
        kd_loss = self.beta * inter_loss + self.gamma * intra_loss
        return kd_loss
```