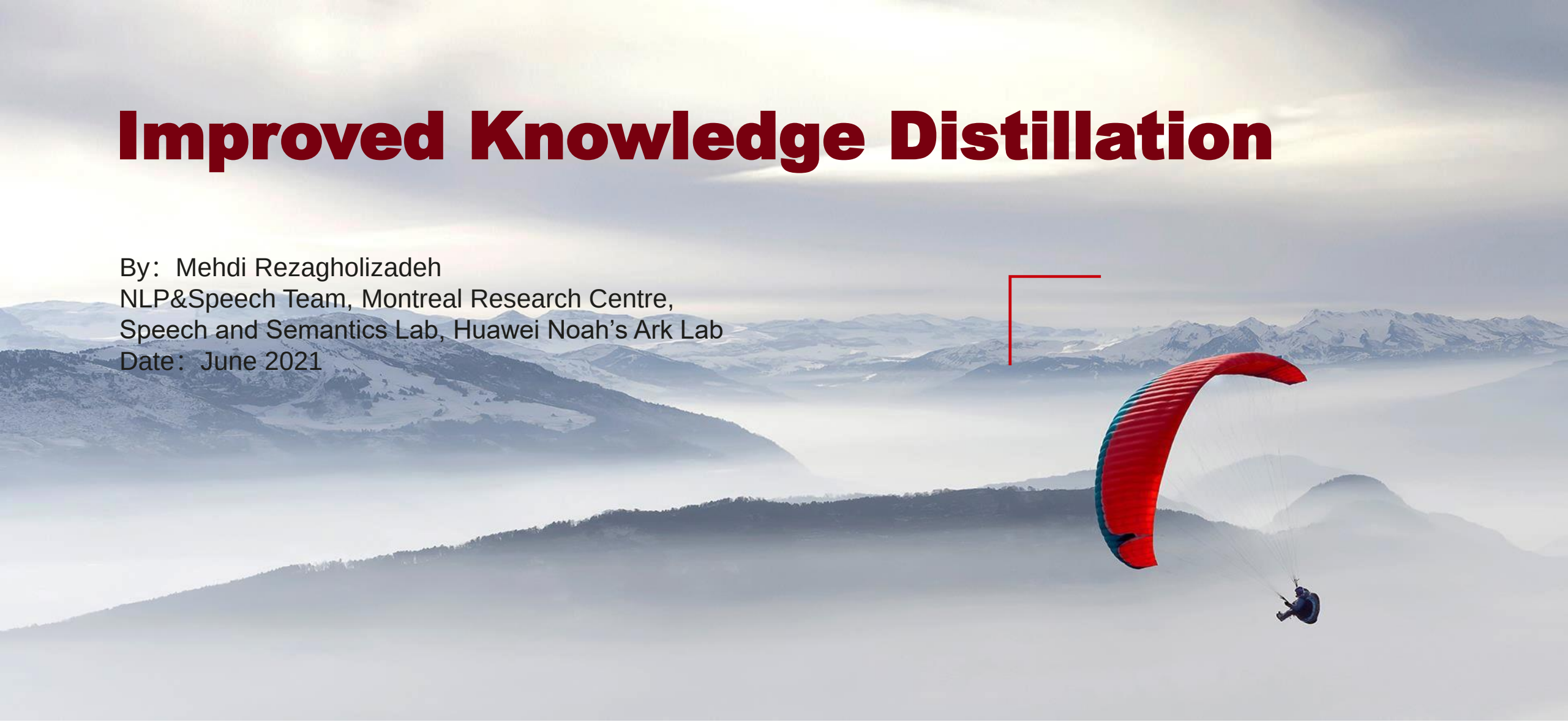
Improved Knowledge Distillation

By: Mehdi Rezagholizadeh
NLP&Speech Team, Montreal Research Centre,
Speech and Semantics Lab, Huawei Noah's Ark Lab
Date: June 2021



NOAH'S ARK LAB

HUAWEI

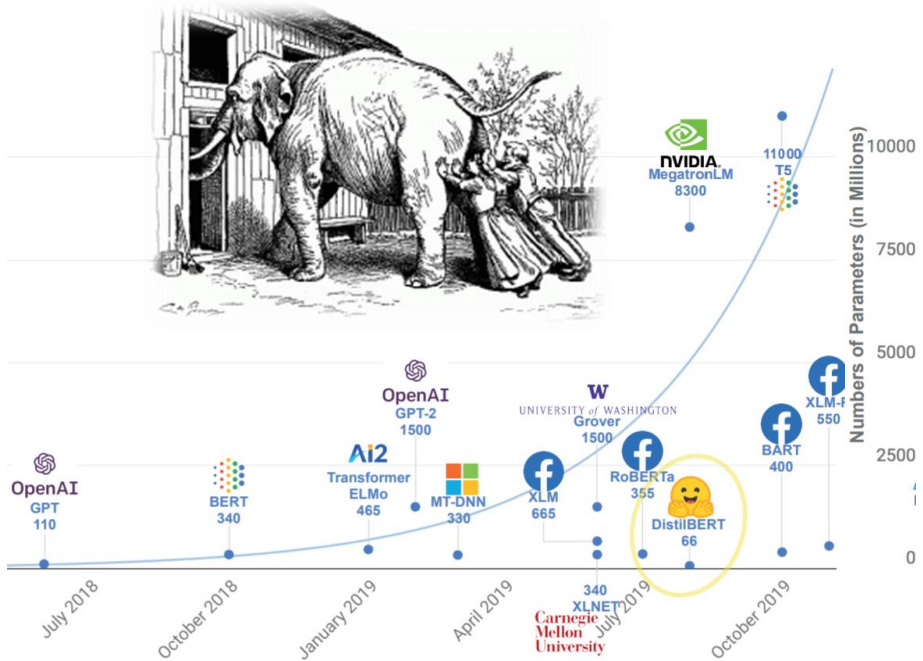
Content

- Introducing Knowledge Distillation
- How Does KD Help?
- Improving KD
  - Training: Annealing KD
  - Data: MATE-KD
  - Structure: ALP-KD
- Open Problems in KD

Motivation: Why Knowledge Distillation?

Model Compression

- KD is one of the most prominent neural model compression techniques.
- Goal is to distill the knowledge of a large model to a smaller model.
- Over parameterization is a common problem in deep neural models.



Model Improvement

- The number of parameters of the teacher is the same as the student.
- Goal is to improve the model rather than compressing it.
- It is called Born-Again Setting (2018).

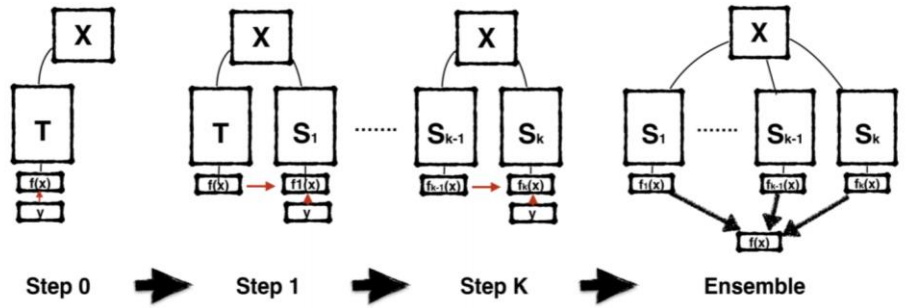


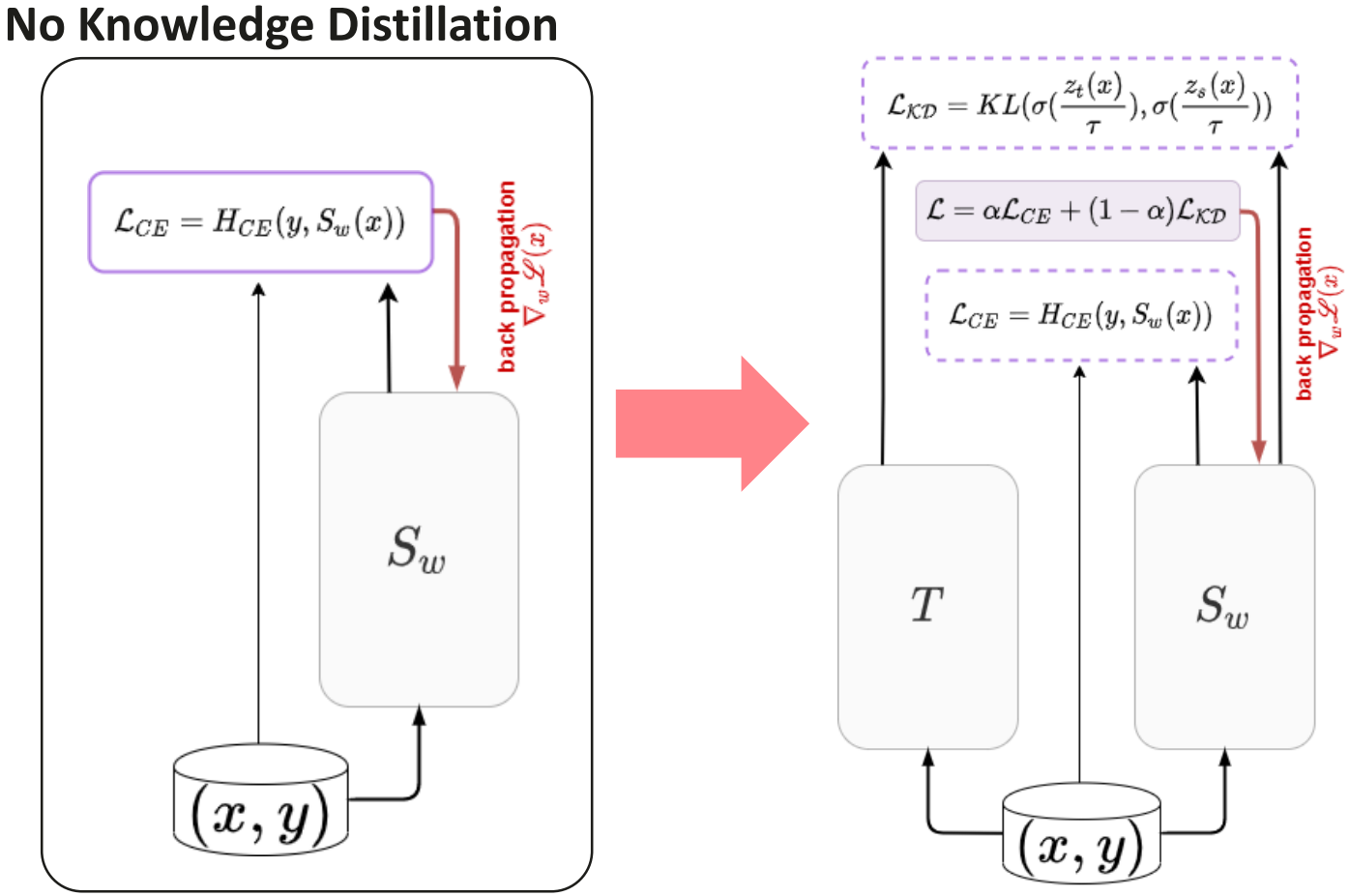
Figure 1. **Graphical representation of the BAN training procedure**: during the first step the teacher model T is trained from the labels Y. Then, at each consecutive step, a new identical model is initialized from a different random seed and trained from the supervision of the earlier generation. At the end of the procedure, additional gains can be achieved with an ensemble of multiple students generations.

[4] Furlanello, Tommaso, et al. "Born again neural networks." *International Conference on Machine Learning*. PMLR, 2018.

HUAWEI

Knowledge Distillation

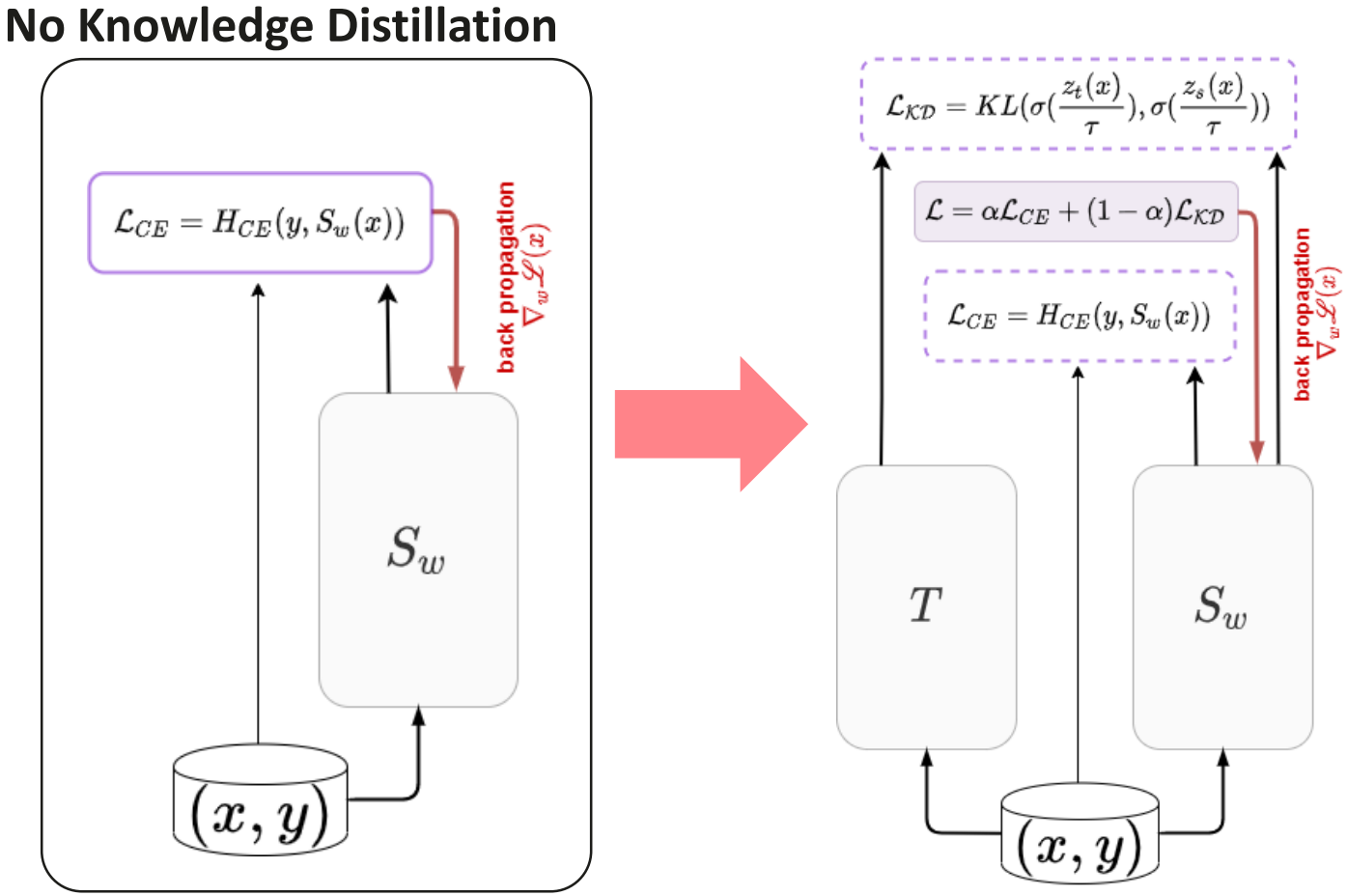
- Model Compression was originally proposed by Bucila et al. in 2006.
  - They deal with ensemble of models and try to match the logits of the compressed model and the logits of the ensemble model.
  - They solve a regression task.



[1] Bucilă, Cristian, Rich Caruana, and Alexandru Niculescu-Mizil. "Model compression." *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*. 2006.
[2] Hinton, Geoffrey, Oriol Vinyals, and Jeff Dean. "Distilling the knowledge in a neural network." *arXiv preprint arXiv:1503.02531* (2015).

Knowledge Distillation

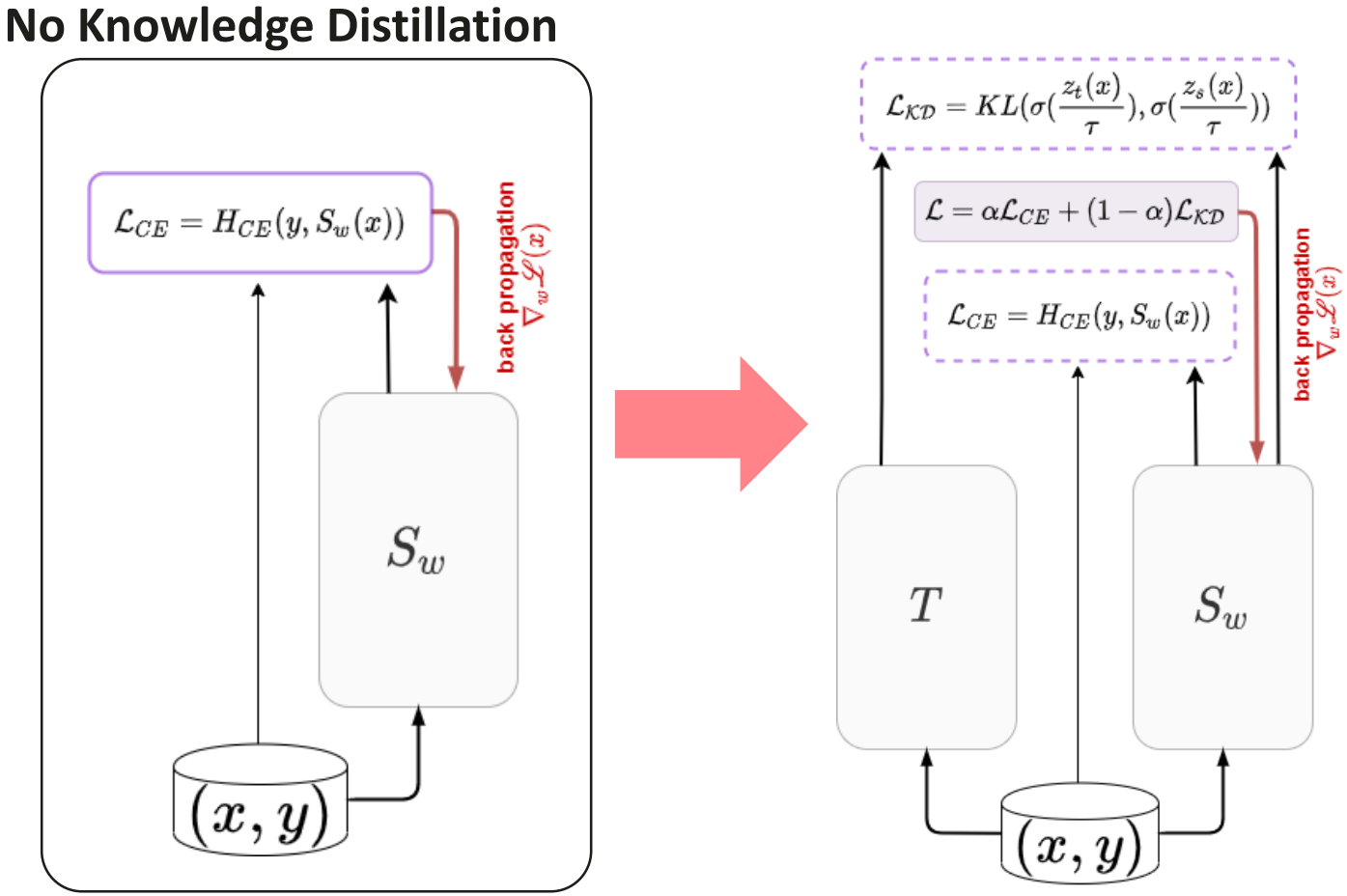
- Model Compression was originally proposed by Bucila et al. in 2006.
- The idea of Knowledge Distillation (KD) became prominent after the paper of “Distilling the Knowledge in a Neural Network” by Hinton et al. in 2015.
- Goal of KD:
  - Transferring the generalizability of a complex neural net to a smaller model
  - We can use a transfer set (i.e. the training set used for distilling knowledge from the large model to the small model) to do distillation. This transfer set can be the same as the original training set of the teacher.



[1] Bucilă, Cristian, Rich Caruana, and Alexandru Niculescu-Mizil. "Model compression." *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*. 2006.
[2] Hinton, Geoffrey, Oriol Vinyals, and Jeff Dean. "Distilling the knowledge in a neural network." *arXiv preprint arXiv:1503.02531* (2015).

Knowledge Distillation

- In KD, we have an extra term in the training loss $\mathcal{L}_{KD}$ which tries to match the output predictions of the two networks.
- In the KD loss there is a temperature factor τ which controls the softness of predictions.
- The output probabilities of the teacher is called "***soft target***" for the student in contrast to one-hot groundtruth labels (or so called hard labeles) in the training data.
- The term ***Dark Knowledge*** is coined by Hinton, which refers to class similarity information in the soft-targets.

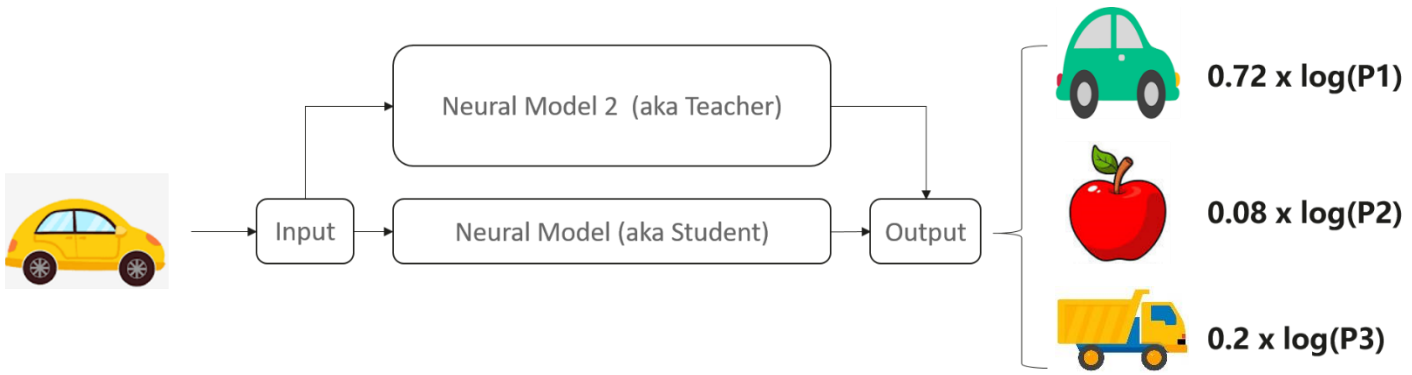


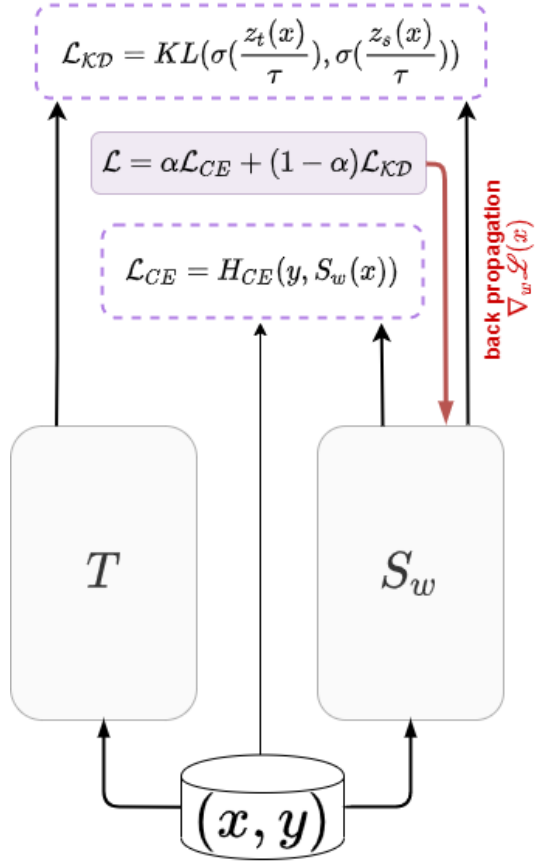
[1] Bucilă, Cristian, Rich Caruana, and Alexandru Niculescu-Mizil. "Model compression." *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*. 2006.
[2] Hinton, Geoffrey, Oriol Vinyals, and Jeff Dean. "Distilling the knowledge in a neural network." *arXiv preprint arXiv:1503.02531* (2015).

Knowledge Distillation

- The term ***Dark Knowledge*** is coined by Hinton, which refers to class similarity information in the soft-targets.

- What is the “Dark Knowledge”?





[3] Hinton, Geoffrey, Oriol Vinyals, and Jeff Dean. “Dark Knowledge," *https://www.ttic.edu/dl/dark14.pdf*.

How Does KD Help?

- For around 5 years the dominant answer was **Dark Knowledge**.
- Is really the Dark Knowledge play the most important role in making KD successful?
- According to the results of the following paper, the answer might change a bit:

Revisiting Knowledge Distillation via Label Smoothing Regularization

Li Yuan[1] Francis EH Tay[1] Guilin Li[2] Tao Wang[1] Jiashi Feng[1]

[1]National University of Singapore [2]Huawei Noah's Ark Lab
{ylustcnus, twangnh}@gmail.com, {mpetayeh,elefjia}@nus.edu.sg, guilinli2@huawei.com

- KD is a regularizer like Label Smoothing (LS) or better to say it is a type of learned LS.
  - **Observation1**: they argue that the success of KD is not fully due to the similarity information between categories from teachers, but also to the regularization of soft targets, which is equally or even more important.
  - **Observation 2**: In KD, beyond the acknowledgment that the teacher can improve the student, the student can also enhance the teacher significantly by reversing the KD procedure.
  - **Observation 3:** A poorly-trained teacher with much lower accuracy than the student can still improve the student significantly.

[ ] Yuan, Li, et al. "Revisiting knowledge distillation via label smoothing regularization." *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020.

How Does KD Help?

– Let's show why KD can be interpreted as a trained LS regularizer.

Regular Training with CE

Model prediction:

$$p(k|x) = p(k) \quad \text{(k is the kth label)}$$

Target:

$$q(k|x) = q(k)$$

(for label y , $q(y|x) = 1$ and $q(k|x) = 0, \forall k \neq y$)

Cross-Entropy:

$$H(q,p) = -\sum_{k=1}^{K} q(k) log p(k)$$

Label Smoothing Regularization

Model prediction:

$$p(k|x) = p(k) \quad \text{(k is the kth label)}$$

Target:

$$q'(k) = \alpha q(k) + (1-\alpha)u(k)$$

(Where $u(k)$ is a fix distribution over classes. Usually it is a uniform distribution $u(k) = 1/K$)

Cross-Entropy:

$$H(q',p) = -\sum_{k=1}^{K} q'^{(k)} log p(k)$$

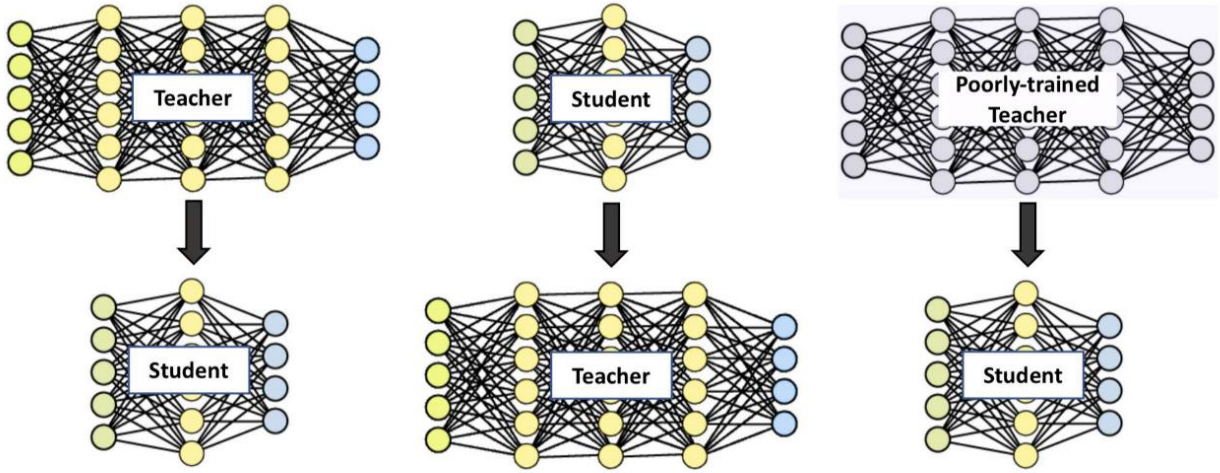
$$\approx \alpha H(q,p) + (1-\alpha) KL(u(k)||p(k))$$ **[Why?]**

– If we replace $u(k)$ with the teacher output distribution and $p(k)$ as student output distribution then KD would be a special case of label smoothing.

[ ] Yuan, Li, et al. "Revisiting knowledge distillation via label smoothing regularization." *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020.

How Does KD Help?

- If we replace $u(k)$ with the teacher output distribution and $p(k)$ as student output distribution then KD would be a special case of label smoothing. **What does this mean?**
  - In general any $u(k)$ distribution can have regularization effect.
  - It can be student distribution for training teacher (reverse KD)
  - It can be poorly trained teacher (defective KD)
  - It can be a uniform distribution (LSR)
  - Or even the student distribution for its own training (***Self-Distillation*** or Teacher-Free Distillation)

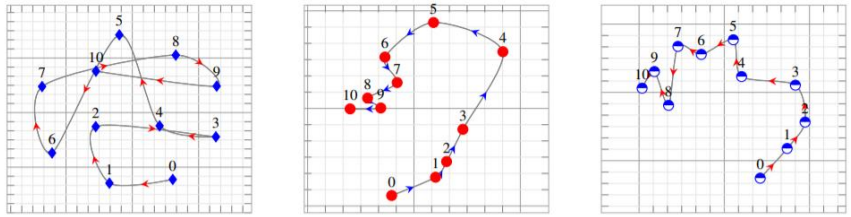


(a) Normal KD (b) Reversed KD (c) Defective KD

[ ] Yuan, Li, et al. "Revisiting knowledge distillation via label smoothing regularization." *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020.

How Does KD Help?

- Aside from *Dark Knowledge* and *Regularization effects*, there is another observation: **Inductive Bias**
- Inductive Bias: Inductive biases are the characteristics of learning algorithms that influence their generalization behaviour, independent of data (Abnar et al. 2020): architectural choices, the objective function, the curriculum, or the optimization regime, …
- **Related work and their observations:**
  - ❖ "Scalable Syntax-Aware Language Models Using Knowledge Distillation" [ACL' 2019] from DeepMind:
    - ✓ KD from recursive NN (with latent tree bias) to LSTM improved the performance of LSTM on syntax task
  - ❖ "Transferring Inductive Biases through Knowledge Distillation" [2020] from Google Brain
    - ✓ KD from CNN to MLP improved its performance on OOD data (trained on MNIST, tested on corrupted MNIST)
  - ❖ "Training data-efficient image transformers & distillation through attention" [2021] from FAIR
    - ✓ KD from CNN to Transformers improved its performance for vision

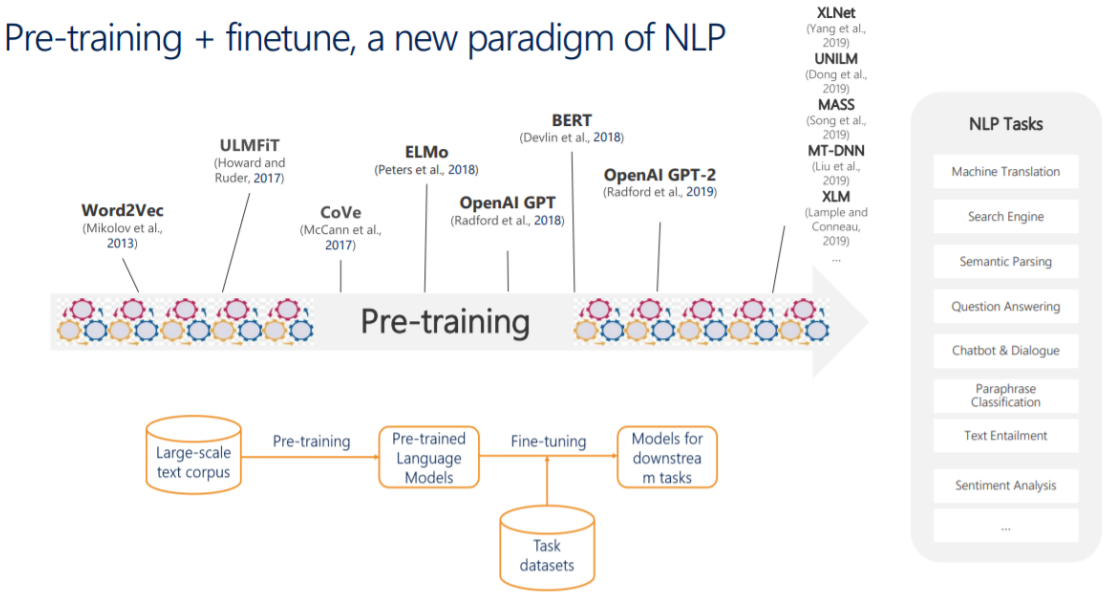


(a) MLP (b) CNN (c) CNN → MLP

Figure 1: Training paths of different models on the Translated MNIST task. Different points represent the state of the model at different epochs, from the initial state to the convergence. The visualization is based on a 2D projection of the representational similarity of the activations from the penultimate layer for the examples from the validation set, i.e. Translated MNIST (more details in Appendix B).

[ ] Abnar, Samira, Mostafa Dehghani, and Willem Zuidema. "Transferring inductive biases through knowledge distillation." *arXiv preprint arXiv:2006.00555* (2020).

KD for NLP Applications



- Pretrained models such as BERT is a new paradigm in NLP domain which gives SOTA results in many tasks.
- However, these pre-trained models are extremely large for deployment on mobile devices.

Naive solution:

- Vanilla KD on a narrower/shallower model

So we need to think of other potential solutions for retaining the performance of the compressed pre-trained models.

Alternatives:

- [ALBERT] Parameter sharing & matrix factorization
- Patient KD (PKD) : not for attentions and embedding
- [MobileBERT] Progressive Knowledge Transfer
- Tiny-BERT: everything in the pre-training and fine-tuning stages

Table 1: A summary of KD methods for BERT. Abbreviations: INIT(initializing student BERT with some layers of pre-trained teacher BERT), DA(conducting data augmentation for task-specific training data). Embd, Attn, Hidn, and Pred represent the knowledge from embedding layers, attention matrices, hidden states, and final prediction layers, respectively.

KD Methods	KD at Pre-training Stage					KD at Fine-tuning Stage				
	INIT	Embd	Attn	Hidn	Pred	Embd	Attn	Hidn	Pred	DA
Distilled $BiLSTM_{SOFT}$									✓	✓
BERT-PKD	✓							✓[3]	✓	
DistilBERT	✓				✓[4]				✓	
TinyBERT (our method)		✓	✓	✓		✓	✓	✓	✓	✓

Images are taken from: http://www.emc2-ai.org/assets/docs/neurips-19/emc2-neurips19-sanh-poster.pdf and Xin Jiang's presentation
TinyBERT Paper: https://arxiv.org/abs/1909.10351

HUAWEI

Improving KD

- We can improve KD from 3 perspectives:

Data

- Minimax KD
- Data-free KD

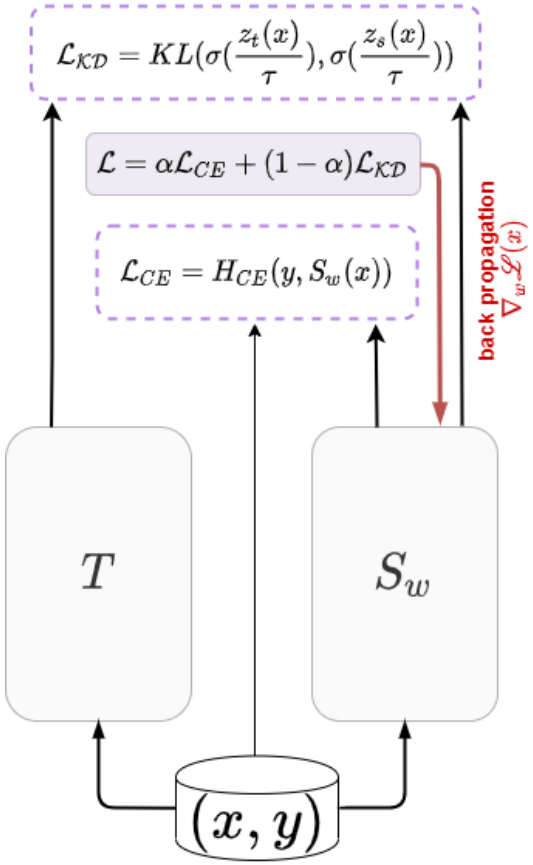
Structure

- Dynamic KD
- Intermediate Layers
- Cross Architecture

Training

- Online/Offline
- Self-KD
- Born-again KD
- Efficient KD
- Multi-Teacher
- TAKD

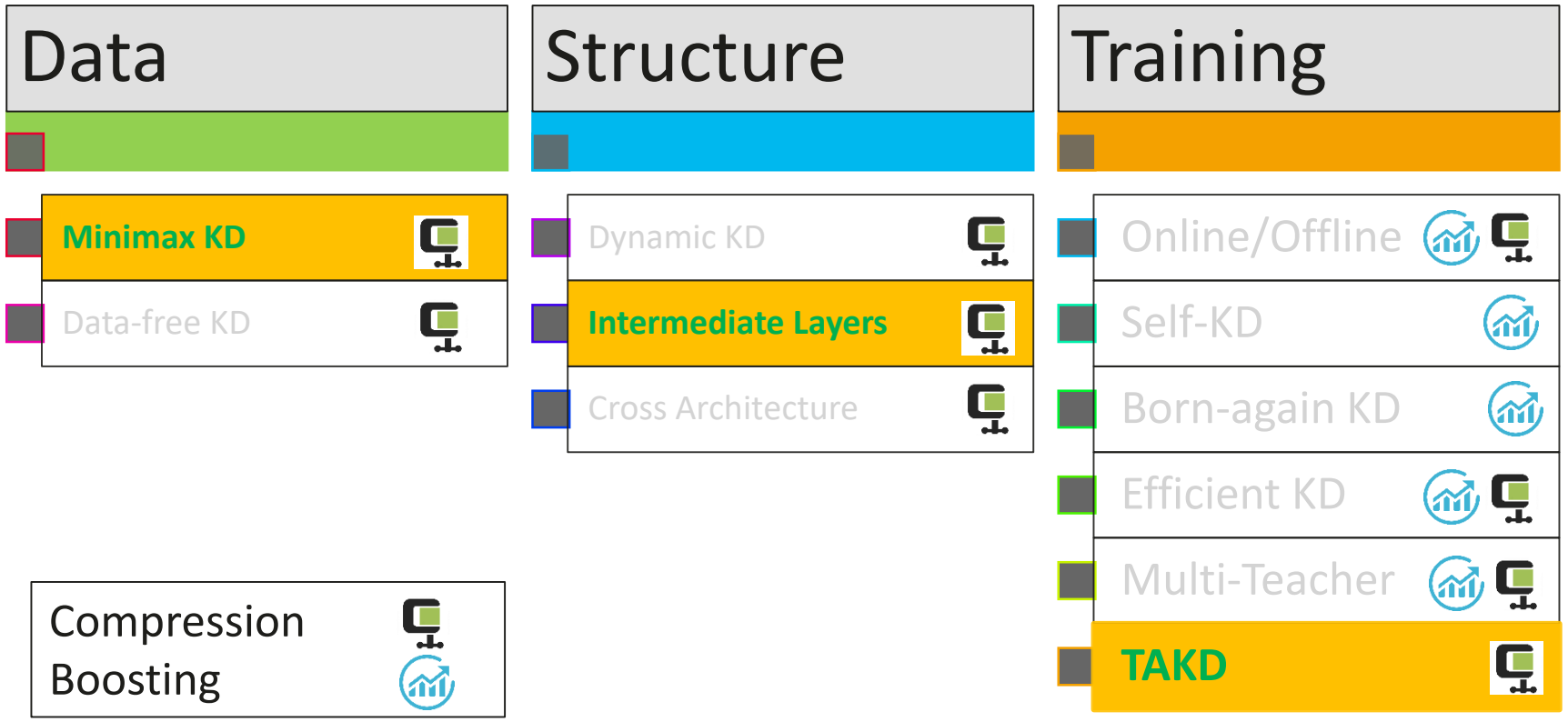
Compression
Boosting



HUAWEI

Improving KD

- We can improve KD from 3 perspectives:



Improving Training of Knowledge Distillation

Problem Statement:

- **Capacity GAP Problem:** KD starts performing poorly when the capacity gap between the teacher and student model becomes larger.
- **Why is it IMPORTANT?** These days since the size of neural models is ever growing especially in NLP with the emergence of transformer based models like BERT and GPT.

- **Existing Solutions:**
  - Mirzadeh, Seyed-Iman, et al. "Improved Knowledge Distillation via Teacher Assistant." (2019).
  - TA-KD proposes a multi-step KD in which some intermediate student networks is defined to fill the gap between the teacher and the student.

- **Problem of Existing Solutions:**
  - It needs training multiple intermediate networks (not feasible for NLP)—[Very Expensive and not scalable]
  - Error propagation
  - Just evaluated in the CV domain.

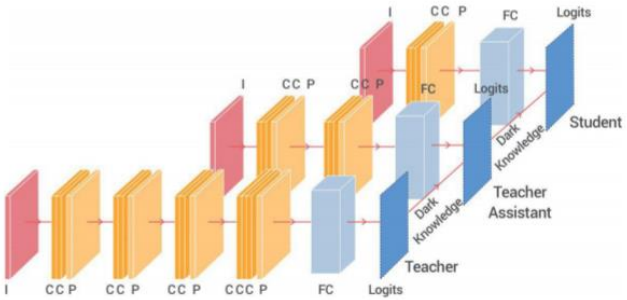


Figure 1: TA fills the gap between student & teacher

Image is taken from: https://arxiv.org/pdf/1902.03393.pdf

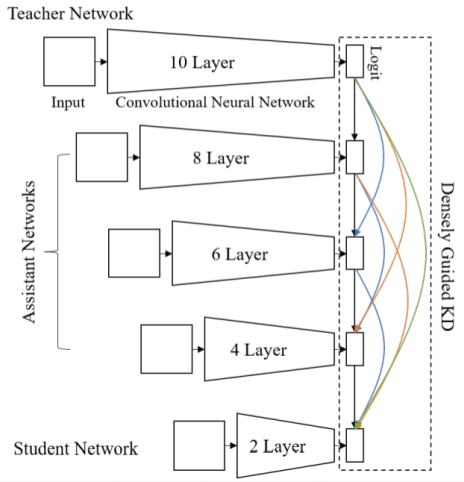


Image is taken from: https://arxiv.org/pdf/2009.08825.pdf

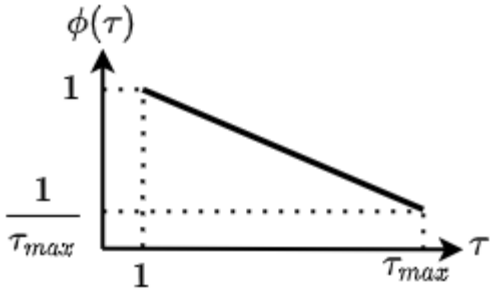
HUAWEI

Our Solution: Annealing KD

- Instead of pushing the student network to learn a sharp function, we can reduce the sharpness of the target function at the beginning of training process and then **gradually** increase it during the training process.
- **Benefit:** we can have a smooth transition from a soft function into a coarse function and training the student during this transition can transfer the behavior of the teacher to the student better. (Teacher can guide the student network gradually).
- **How?** We propose a two-stage solution:
  - **Stage I:** Train the student to gradually follow the teacher

$$\mathcal{L}_{\text{KD}}^{\text{Annealing}}(i) = \|z_s(x) - z_t(x) \times \Phi(\mathcal{T}_i)\|_2^2$$

$$\Phi(\mathcal{T}) = 1 - \frac{\mathcal{T}-1}{\tau_{\max}}, 1 \leq \mathcal{T} \leq \tau_{\max}, \mathcal{T} \in \mathbb{N}$$



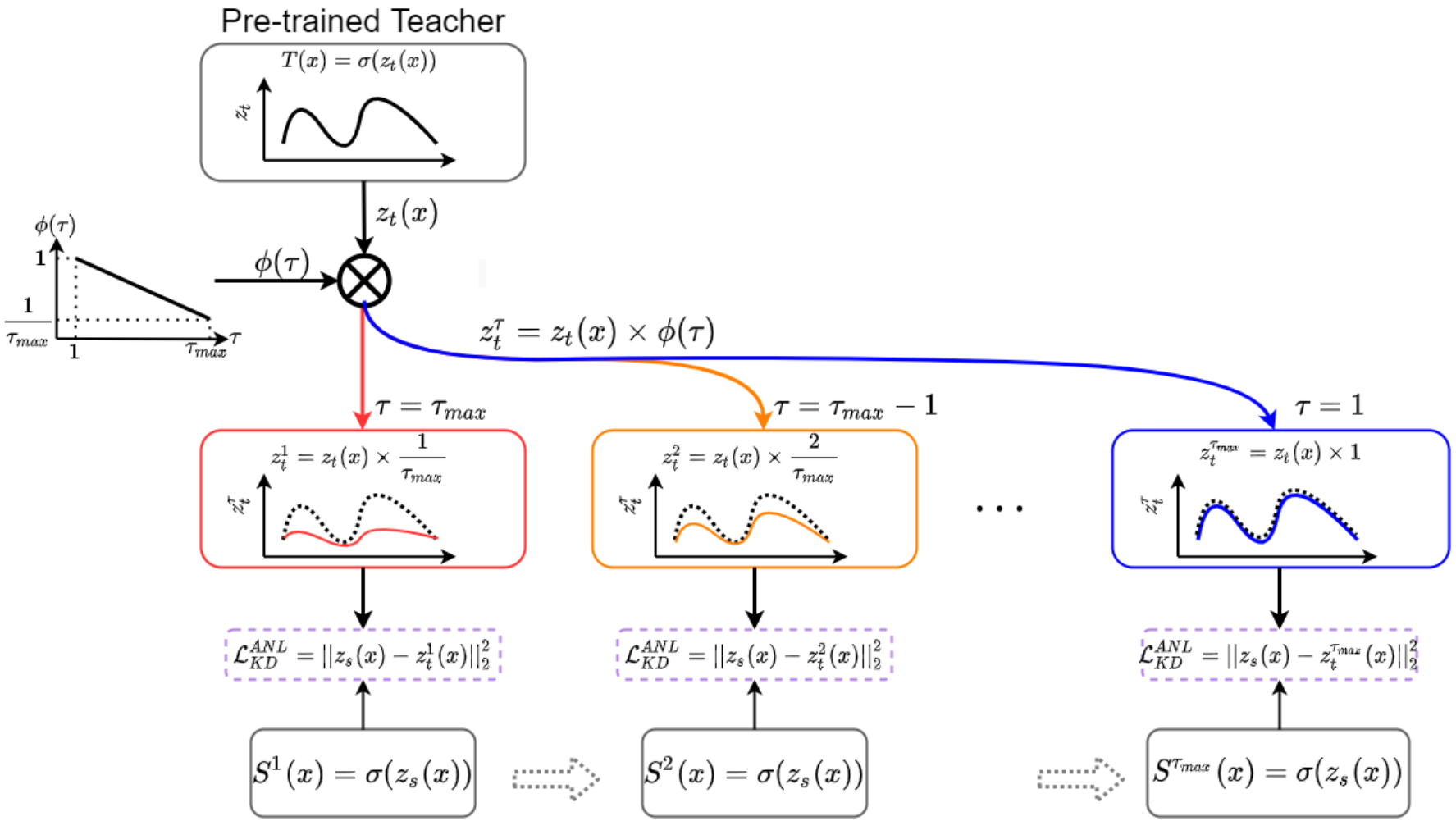
  - **Stage II:** Fine tune with the true labels

$$\mathcal{L} = \begin{cases} \mathcal{L}_{\text{KD}}^{\text{Annealing}}(i), & \textbf{Stage I: } 1 \leq i \leq n, 1 \leq \mathcal{T}_i \leq \tau_{\max} \\ \mathcal{L}_{CE}, & \textbf{Stage II: } i = n, \mathcal{T}_n = 1 \end{cases}$$

[] Aref Jafari, Mehdi Rezagholizadeh, Pranav Sharma, Ali Ghodsi "Annealing Knowledge Distillation." *Accepted in EACL* (2021).

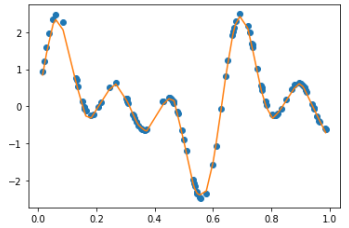
Our Solution: Annealing KD

[Stage I]

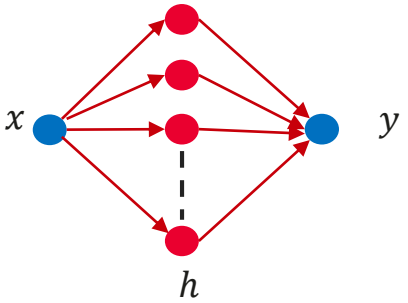


[] Aref Jafari, Mehdi Rezagholizadeh, Pranav Sharma, Ali Ghodsi "Annealing Knowledge Distillation." *Accepted in EACL* (2021).

Our Solution: Annealing KD [Example]



Dataset



Teacher network (|h|=100)

Stage I:

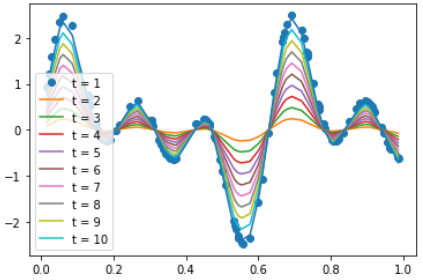
- **Step1)** Consider a dataset $\{(\boldsymbol{x_i}, \boldsymbol{y_i})\}_{\boldsymbol{i=1}}^{\boldsymbol{n}}$ and a trained teacher network approximating the underlying function of the dataset.
- **Step2)** For training the student network, consider the following loss function:

$$L_{KD}^{Annealing} = \|z_s(x) - z_t(x) \times \phi(\tau)\|^2$$

Where, $\boldsymbol{1 \leq \tau \leq \tau_{max}}$ is the temperature function changing from 1 to the maximum temperature $\boldsymbol{\tau_{max}}(\tau, \tau_{max} \in \mathbb{N})$

- Now if we consider the total number of training epochs to be equal to $\boldsymbol{n = k \times \tau_{max}}$ and by starting from $\boldsymbol{\tau = \tau_{max}}$, we decrease the temperature gradually to $\boldsymbol{\tau = 1}$, and in each step for each $\boldsymbol{\tau}$ we train the network for $\boldsymbol{k}$ epochs.

$T(x) \times \phi(\tau)$



Behavior of teacher function in different temperatures

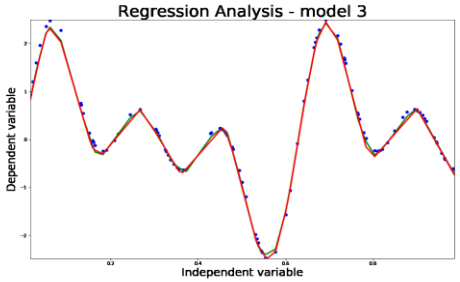
Teacher

$|h| = 100$

Regular KD

$|h| = 10$

Annealing KD



$|h| = 10$

HUAWEI

Experiments and Results

[Image Classification Tasks]

- We followed TAKD experimental setup.
- Dataset:
  - CIFAR-10 and CIFAR-100
  - 32 × 32 color images with 10 and 100 classes respectively
- Our Networks:
  - Teacher: ResNet-110 → Student: ResNet-8
  - Teacher: CNN-10 → Student: CNN-2

❖ Annealing-KD outperforms all other baselines and TAKD is the second-best performing student without significant distinction compared to KD.

No KD < KD < TAKD < Annealing KD (Ours) <Teacher

CIFAR 10

Model	Type	Training method	Accuracy
ResNet	Teacher(110)	from scratch	93.8
	TA(20)	KD	92.39
	Student(8)	from scratch	88.44
	Student(8)	KD	88.45
	Student(8)	TAKD	88.47
	Student(8)	**Annealing KD (ours)**	**89.44**
CNN	Teacher(10)	from scratch	90.1
	TA(4)	KD	82.39
	Student(2)	from scratch	72.75
	Student(2)	KD	72.43
	Student(2)	TAKD	72.62
	Student(2)	**Annealing KD (ours)**	**73.17**

CIFAR 100

Model	Type	Training method	Accuracy
ResNet	teacher(110)	from scratch	71.92
	TA(20)	KD	67.6
	student(8)	from scratch	61.37
	student(8)	KD	61.41
	student(8)	TAKD	61.82
	student(8)	**Annealing KD (ours)**	**63.1**
CNN	Teacher(10)	from scratch	64.89
	TA(4)	KD	60.73
	student(2)	from scratch	51.35
	student(2)	KD	51.62
	student(2)	TAKD	51.85
	student(2)	**Annealing KD (ours)**	**53.35**

[] Aref Jafari, Mehdi Rezagholizadeh, Pranav Sharma, Ali Ghodsi "Annealing Knowledge Distillation." *Accepted in EACL* (2021).

Experiments and Results

[GLUE Benchmark]

MNLI (Multi-Genre Natural Language Inference)	entailment classification	a pair of sentences, predict whether the second sentence is an entailment, contradiction, or neutral with respect to the first one.
QQP (Quora Question Pairs)	binary classification	determine if two questions asked on Quora are semantically equivalent
QNLI (Question Natural Language Inference)	binary classification	a version of the Stanford Question Answering Dataset.The positive examples are (question, sentence) pairs which do contain the correctanswer, and the negative examples are (question,sentence) from the same paragraph which do not contain the answer.
SST-2 (The Stanford Sentiment Treebank)	binary single-sentence classification task	sentences extracted from movie reviews with human annotations of their sentiment
CoLA (The Corpus of Linguistic Acceptability)	binary single-sentence classification task	predict whether an English sentence is linguistically “acceptable” or not
STS-B (The Semantic Textual Similarity Benchmark)	Class 1-5	Sentence pairs drawn from news headlines and other sources. They were annotated with a score from 1 to 5 denoting how similar the two sentences are in terms of semantic meaning.
MRPC (Microsoft Research Paraphrase Corpus)	binary classification task	sentence pairs automatically extracted from online news sources, with human annotations for whether the sentences in the pair are semantically equivalent
RTE (Recognizing Textual Entailment)		binary entailment task similar to MNLI
WNLI (Winograd NLI)	binary classification task	small natural language inference dataset deriving from (Levesque et al., 2011). when two examples contain the same sentence, that usually means they'll have opposite labels.

Experiments and Results

[GLUE Benchmark: DistilRoBERTa]

- Dataset: GLUE
- Our Networks:
  - ➢ Teacher: RoBERTa-Large (24 Layers)
  - ➢ TA: RoBERTa-Base (12 Layers)
  - ➢ Student: DistilRoBERTa (6 Layers)

- We train Annealing-KD for 14 epochs in phase 1 and 6 epochs in phase 2

❖ Annealing-KD outperforms all other baselines and TAKD is the second-best performing student without significant distinction compared to KD.

DistilRoBERTa results for Annealing KD on **dev set**. F1 scores are reported for MRPC, pearson correlations for STB-B,and accuracy scores for all other tasks.

KD Method	CoLA	RTE	MRPC	STS-B	SST-2	QNLI	QQP	MNLI	WNLI	Score
Teacher	68.1	86.3	91.9	92.3	96.4	94.6	91.5	90.22/89.87	56.33	85.29
From scratch	59.3	67.9	88.6	88.5	92.5	90.8	90.9	84/84	52.1	79.3
Vanilla KD	60.97	71.11	90.2	88.86	92.54	91.37	91.64	84.18/84.11	56.33	80.8
TAKD	61.15	71.84	89.91	88.94	92.54	91.32	**91.7**	83.89/84.18	56.33	80.85
Annealing KD	**61.67**	**73.64**	**90.6**	**89.01**	**93.11**	**91.64**	91.5	**85.34/84.6**	56.33	**81.42**

No KD < KD < TAKD < Annealing KD (Ours) <Teacher

[] Aref Jafari, Mehdi Rezagholizadeh, Pranav Sharma, Ali Ghodsi "Annealing Knowledge Distillation." *Accepted in EACL* (2021).

Experiments and Results

[GLUE Benchmark: DistilRoBERTa]

- Dataset: GLUE
- Our Networks:
  - ➢ Teacher: RoBERTa-Large (24 Layers)
  - ➢ TA: RoBERTa-Base (12 Layers)
  - ➢ Student: DistilRoBERTa (6 Layers)

- We train Annealing-KD for 14 epochs in phase 1 and 6 epochs in phase 2

❖ Annealing-KD outperforms all other baselines and TAKD is the second-best performing student without significant distinction compared to KD.

Performance of DistilRoBERTa trained by annealing KD on the **GLUE leaderboard** compared with Vanilla KD and TAKD. We applied the standard tricks to all 3 methods and fine-tune RTE, MRPC and STS-B from trained MNLI student model.

KD Method	CoLA	MRPC	STS-B	SST-2	MNLI-m	MNLI-mm	QNLI	QQP	RTE	WNLI	Score
Vanilla KD	**54.3**	86/80.8	85.7/84.9	93.1	83.6	82.9	90.8	71.9/89.5	74.1	65.1	78.9
TAKD	53.2	86.7/82.7	85.6/84.4	93.2	83.8	83.2	**91**	72/89.4	**74.2**	65.1	79
Annealing KD	54	**88.0/83.9**	**87.0/86.6**	**93.6**	**83.8**	**83.9**	90.8	**72.6/89.7**	73.7	65.1	**79.5**

KD < TAKD < Annealing KD (Ours) <Teacher

[] Aref Jafari, Mehdi Rezagholizadeh, Pranav Sharma, Ali Ghodsi "Annealing Knowledge Distillation." *Accepted in EACL* (2021).

Experiments and Results

[GLUE Benchmark: DistilRoBERTa]

- Dataset: GLUE
- Our Networks:
  - Teacher: BERT-Large (24 Layers)
  - TA: BERT-Base (12 Layers)
  - Student: BERT-small (4 Layers)
- We use a maximum temperature of 7 for MRPC, SST-2, QNLI, and WNLI, and 14 for all other tasks.

- ❖ Annealing-KD outperforms all other baselines and TAKD is the second-best performing student without significant distinction compared to KD.
- ❖ when we reduce the size of the student to a 4 layer model (BERT-Small), we notice almost twice as big of a gap in the average score over Vanilla KD when compared with DistilRoBERTa

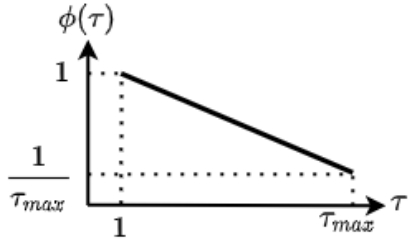
BERT-Small results for Annealing KD on dev set. F1 scores are reported for MRPC, pearson correlations for STS-B, and accuracy scores for all other tasks.

KD Method	CoLA	RTE	MRPC	STS-B	SST-2	QNLI	QQP	MNLI	WNLI	Score
Teacher	65.8	71.48	89.38	89.2	92.77	92.82	91.45	86.3/86.4	60.56	82.19
Vanilla KD	33.5	57	86	72.3	88.76	**83.15**	**87**	72.62/73.19	54.92	70.58
TAKD	34.24	59.56	85.23	71.1	89.1	82.62	87	72.32/72.45	54.92	70.76
Annealing KD	**35.98**	**61**	**86.2**	**74.54**	**89.44**	83.14	86.5	**73.85/74.84**	54.92	**71.68**

KD < TAKD < Annealing KD (Ours) <Teacher

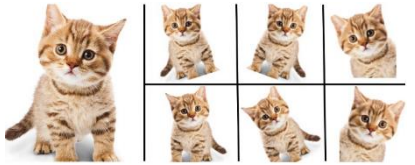
[] Aref Jafari, Mehdi Rezagholizadeh, Pranav Sharma, Ali Ghodsi "Annealing Knowledge Distillation." *Accepted in EACL* (2021).

Summary of Annealing KD

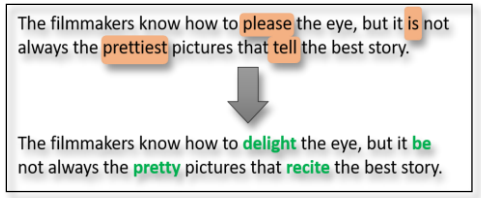
	Annealing KD	Vanilla KD
1	Unlike vanilla KD our algorithm is performed in two stages. Stage 1) $L_1 = L_{AKD}$ Stage2) $L_2 = L_{CE} = \sum_{i=1}^{n} y_i \log(S(x_i))$	$L = L_{CE} + L_{KD}$ $L = \alpha \sum_{i=1}^{n} y_i \log(S(x_i)) + (1-\alpha) L_{KD}(\frac{S(x)}{t}, \frac{T(x)}{t})$
2	Our KD loss is based on mean square error of the student output and annealed teacher output $L_{AKD} = \|z_s(x) - z_t(x) \times \phi(\tau)\|^2$	$L_{KD}(\frac{S(x)}{t}, \frac{T(x)}{t})$
3	Annealing KD dynamically changes the temperature value during he stage 1 of the training process, using the function $\varphi(t)$. 	Vanilla KD uses a fixed temperature value during entire training process

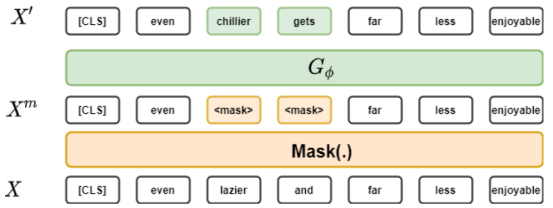
[] Aref Jafari, Mehdi Rezagholizadeh, Pranav Sharma, Ali Ghodsi "Annealing Knowledge Distillation." *Accepted in EACL* (2021).

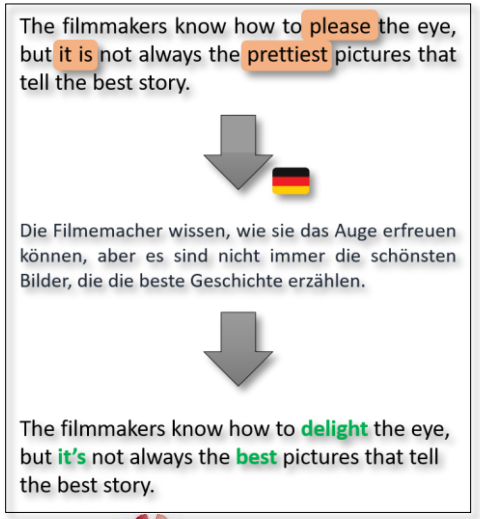
Improving Data Augmentation for KD



- Data Augmentation (DA): A popular technique to improve generalization of NNs
- Existing Solutions:
  - **Heuristic-based:**
    - [Examples in CV] image translation, horizontal reflection, altering RGB intensity, Mixup(Zhang et al. 2018)
    - [Examples in NLP] Replace words/phrases with their synonyms
  - **Model-based:**
    - **Without Training (Task Agnostic):** KNN-Retrieval-based Augmentation, Back Translation, Contextual DA
    - **With Training (Task-Aware):** Adversarial DA
- DA for CV is different from NLP







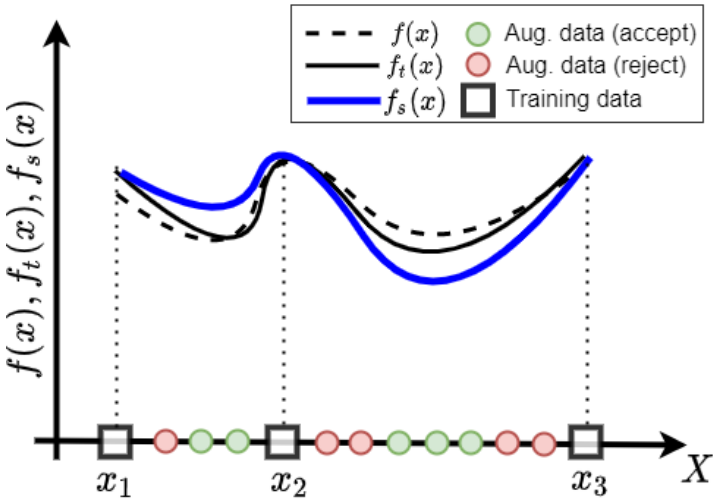
Problems:

- DA techniques are **not designed for KD** and even task-aware DA techniques are tailored for one network and not two
- **Data GAP Problem:** the teacher and student might **diverge** in areas in the input space that we do not have training samples
- Task agnostic DA techniques are **not sample efficient**
- Usually in NLP augmented samples are **not semantically meaningful**

HUAWEI

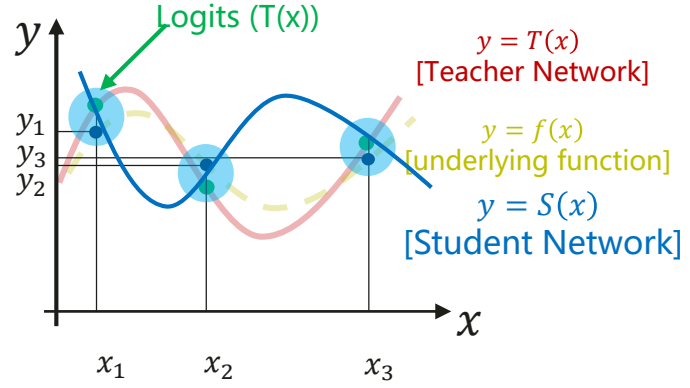
Improving Data Augmentation for KD

- Mathematically we can consider any neural network as a function. Therefore the problem of KD is matching student function into the teacher function.
- Although the conventional KD is effective for matching the two networks over the given data points, there is no guarantee that these models would match in other areas for which we do not have enough training samples.



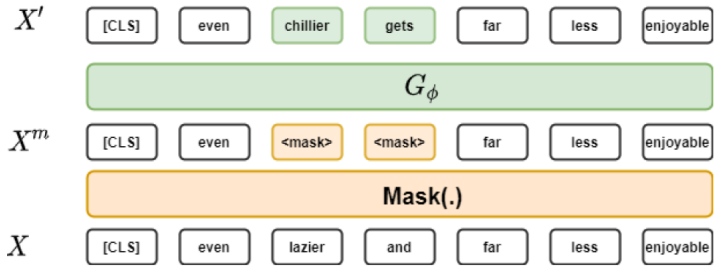
Problems:

- DA techniques are **not designed for KD** and even task-aware DA techniques are tailored for one network and not two
- **Data GAP Problem:** the teacher and student might **diverge** in areas in the input space that we do not have training samples
- Task agnostic DA techniques are **not sample efficient**
- Usually in NLP augmented samples are **not semantically meaningful**



HUAWEI

Improving Data Augmentation for KD

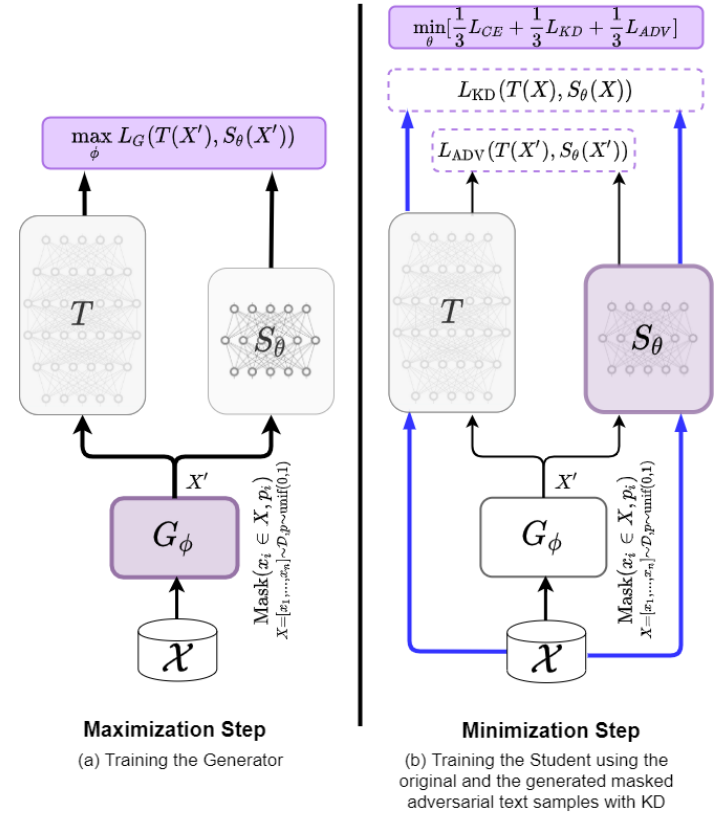


- **Solution: MiniMax + Generator [MATE-KD]**
- **(Phase I):Maximization Step**
  - We need to find the inputs which lead to maximizing the loss between the teacher and student networks
  - Taking the gradient of the loss function w.r.t the input samples, we perturb the training points in the direction of their gradients iteratively to increase the loss between two networks.

$$x' = \underset{x \in X}{\operatorname{argmax}}\, KL\left(T\left(G_\phi(x)\right), S_\theta(G_\phi(x))\right)$$

- **(Phase II): Minimization Step**
  - We add the augmented samples generated in the first Phase to the training data.
  - We use the same loss as original KD to minimize the CE and KL loss between the teacher and student.

$$\mathcal{L}_{KD} = (1-\lambda)\, H\left(y, \sigma\left(S(x)\right)\right) + \tau^2\, \lambda\, KL\left(\sigma\left(\frac{T(x)}{\tau}\right), \sigma\left(\frac{S(x)}{\tau}\right)\right)$$



[] Ahmad Rashid, Vasileios Lioutas, Mehdi Rezagholizade "MATE-KD: Masked Adversarial TExt, a Companion to Knowledge Distillation." *Accepted in ACL* (2021).

HUAWEI

Experiments and Results

- Dataset: GLUE
- Our Networks:
  - [GLUE]Teacher: RoBERTa-Large (24 Layers)
  - [GLUE]Student: DistilRoBERTa (6 Layers)

Remarks

- MATE-KD **outperforms** SOTA techniques

BUT

- **Needs** a generator
- Generated samples are **NOT** semantically meaningful

Method	CoLA	SST-2	MRPC	STS-B	QQP	MNLI	QNLI	RTE	Score
$\text{RoBERTa}_{\text{Large}}$ (teacher)	68.1	96.4	91.9	92.3	91.5	90.2	94.6	86.3	85.28
DistilRoBERTa (student)	56.6	92.7	89.5	87.2	90.8	84.1	91.3	65.7	78.78
Student + FreeLB	58.1	93.1	90.1	88.8	90.9	84.0	91.0	67.8	80.01
Student + FreeLB + KD	58.1	93.2	90.5	88.6	91.2	83.7	90.8	68.2	80.06
Student + Adversarial Learning (Adv)	62.0	93.1	86.1	88.9	91.9	84.5	91.3	70.7	80.53
Student + KD	60.9	92.5	90.2	89.0	91.6	84.1	91.3	71.1	80.77
Student + TinyBERT Augmentation + KD	61.3	93.3	90.4	88.6	91.7	84.4	91.6	72.5	81.12
Student + MATE-KD (Ours)	65.9	94.1	91.9	90.4	91.9	85.8	92.5	75.0	**82.64**

Table 1: Dev Set results for the GLUE benchmark. The score for the WNLI task is 56.3 for all models.

Model (Param.)	CoLA	SST-2	MRPC	STS-B	QQP	MNLI-m/mm	QNLI	RTE	Score
TinyBERT (66M)	51.1	93.1	87.3/82.6	85.0/83.7	71.6/89.1	84.6/83.2	90.4	70.0	78.1
$\text{BERT}_{\text{BASE}}$ (110M)	52.1	93.5	88.9/84.8	87.1/85.8	71.2/89.2	84.6/83.4	90.5	66.4	78.3
MobileBERT (66M)	51.1	92.6	88.8/84.5	86.2/84.8	70.5/88.3	84.3/83.4	91.6	70.4	78.5
DistilRoB. + KD (82M)	54.3	93.1	86.0/80.8	85.7/84.9	71.9/89.5	83.6/82.9	90.8	74.1	78.9
$\text{BERT}_{\text{LARGE}}$ (340M)	60.5	94.9	89.3/85.4	87.6/86.5	72.1/89.3	86.7/85.9	92.7	70.1	80.5
MATE-KD (82M)	56.0	94.9	91.7/88.7	88.3/87.7	72.6/89.7	85.5/84.8	92.1	75.0	**80.9**

Table 2: Leaderboard test results of experiments on GLUE tasks. The score for the WNLI task is 65.1 for all models.

Original	Generated
the new insomnia is a surprisingly faithful remake of its chilly predecessor, and	sinister new insomnia shows a surprisingly terrible remake of its hilarious predecessor, and
beautifully shot, delicately scored and powered by a set of heartfelt performances	beautifully sublime, delicately scored, powered by great dozens of heartfelt performances
a perfectly pleasant if slightly pokey comedy	a 10 pleasant if slightly pokey comedy
that appeals to me	Federal appeals punished me
good news to anyone who's fallen under the sweet, melancholy spell of this unique director's previous films	good news for anyone who's fallen under the sweet, melancholy spell of this unique director's previous mistakes

Table 6: Examples of original and adversarially generated samples during training for the SST-2 dataset

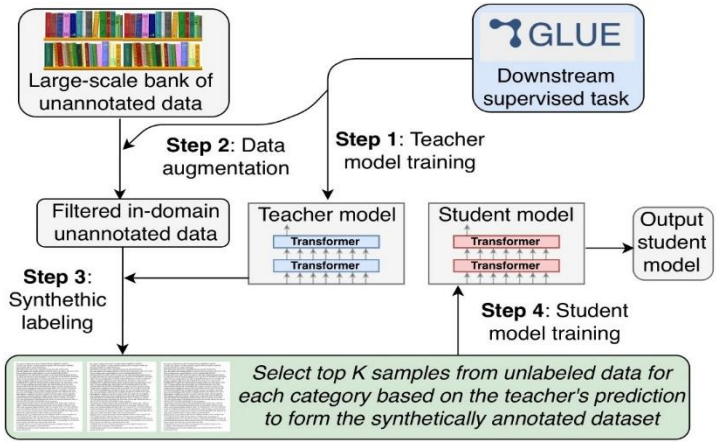
[] Ahmad Rashid, Vasileios Lioutas, Mehdi Rezagholizade "MATE-KD: Masked Adversarial TExt, a Companion to Knowledge Distillation." *Accepted in ACL* (2021).

Improving Data Augmentation for KD:

Semantic Data Augmentation

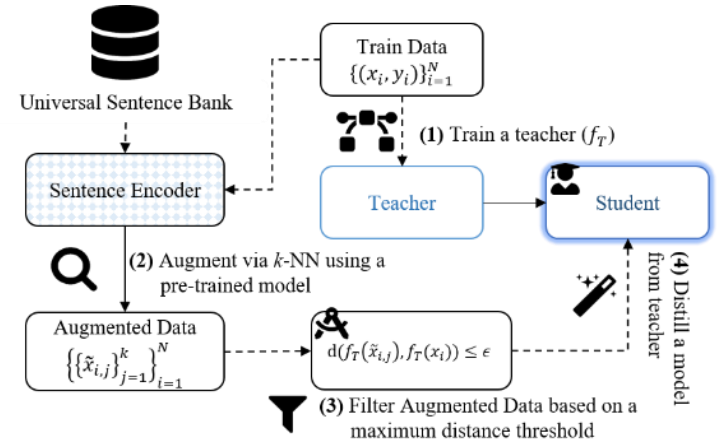
- In NLP we have access to a huge body of unlabeled text on the internet. Can we use them a source for our augmentation?
- **Existing Solution:**
  - **KNN or Retrieval-based Augmentation**: **Interpretable** augmentation by incorporating unannotated text from Web via kNN search → The retrieval part can be done model-based or model-free (e.g. TF-IDF)
- **Problems:**
  - ➢ Not sample efficient (use ~10-20 samples per training data)
  - ➢ Completely blind to the student model
- **Our Solution:**
  - **Minimax-KNN-KD:** In our approach we select **top** $k_1 \ll K$ samples from the retrieved K samples from KNN which leads to maximum divergence between the student and teacher networks.
  - In this solution we use less number of augmented samples.
  - Both the teacher and student networks are involved in selecting best augmented samples.

Existing (SentAug)



The image is taken from (Du et al., 2020)

Ours (Minimax-KNN-KD)



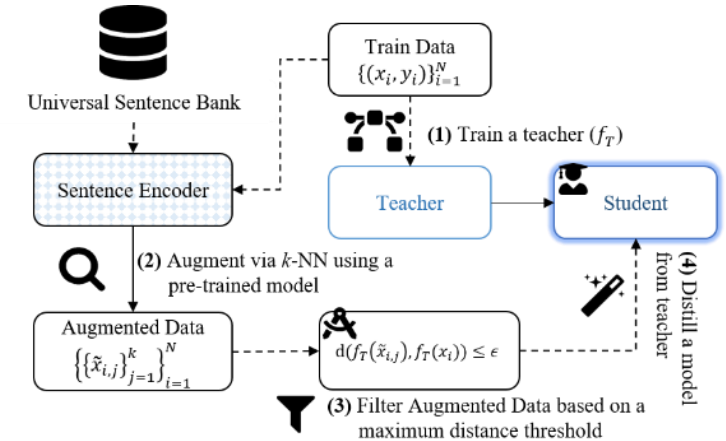
[] Ehsan Kamalloo*, Mehdi Rezagholizadeh*, Peyman Passban, Ali Ghodsi "Not Far Away, Not So Close: Sample Efficient Nearest Neighbour Data Augmentation via MiniMax." *Accepted in ACL* (2021).

Improving Data Augmentation for KD:

Semantic Data Augmentation

- **Results:**
  - **Improved the training time for > 25% and DA sample efficiency for 2x**

Model	SST-5	SST-2	CR	IMP	TREC
RoBERTa-large (Teacher)	57.6	96.2	94.1	89.1	98.0
DistilRoBERTa	52.9	93.5	**92.1**	86.8	96.0
DistilRoBERTa + KD	53.2	93.6	**92.1**	87.3	96.6
DistilRoBERTa + vanilla-8NN	55.2	94.7	91.3	**87.8**	97.0
DistilRoBERTa + Minimax-8NN (k1=4)	**55.4** (+0.2↑)	**95.2** (+0.5↑)	91.6 (+0.3↑)	**87.8**	**97.4** (+0.4↑)



Ours (Minimax-KNN-KD)

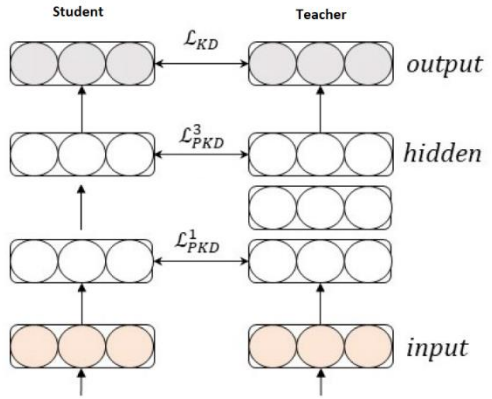
- **Remarks:**
  - **Minimax-KNN-KD** is sample efficient and does not need any further training models to provide augmented samples
  - Both the teacher and student networks are involved in selecting best augmented samples.
  - If you are using augmented retrieval, employ minimax for sample efficiency
  - Weakness: long sentence augmentation and multiple/paired sentence tasks

Name	Task	Domain	Labels
SST-5	Sentiment classification	Movie reviews	very pos., pos., neutral, neg., very neg.
SST-2	Sentiment classification	Movie reviews	positive, negative
CR	Sentiment classification	Product reviews	positive, negative
IMP	Hate-speech classification	Forum conversations	insulting, neutral
TREC	Question-type classification	Questions	entity, numeric, human, location, desc., abbr.

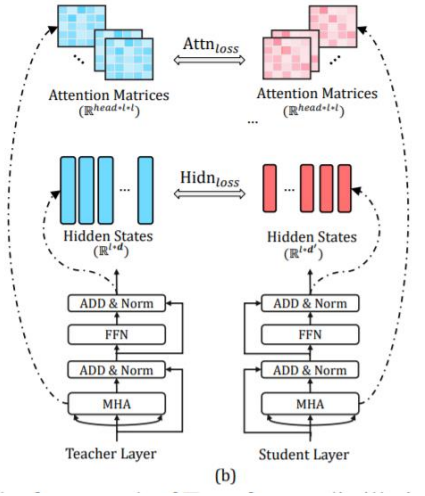
[] Ehsan Kamalloo*, Mehdi Rezagholizadeh*, Peyman Passban, Ali Ghodsi "Not Far Away, Not So Close: Sample Efficient Nearest Neighbour Data Augmentation via MiniMax." *Accepted in ACL* (2021).

Structure Efficient KD

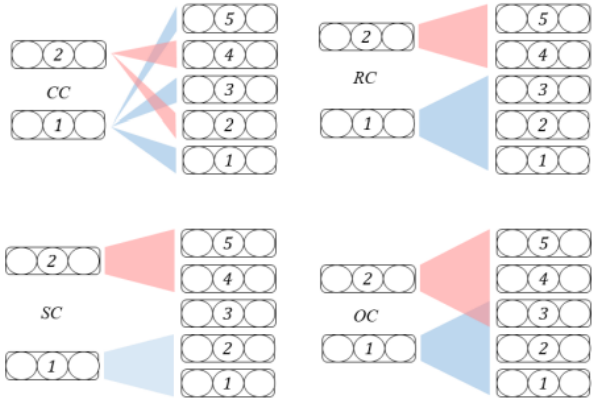
- Transferring Knowledge from intermediate layers is shown to improve KD results especially for BERT-based models.
- **Problem:**
  - Skip: Multiple intermediate layers from teacher are ignored when distillation performed by selecting the same *m* layers from *n* intermediate teacher layers (m>>n), where *m* is the number of intermediate student layers.
  - Search: Since the teacher intermediate layers are selected arbitrarily, other important teacher layers might be missed which could have significant information for knowledge distillation.
- **Existing Solutions:**
  - [1] Patient Knowledge Distillation (PKD) (Sun et al. 2019): selecting some layers of the teacher to distill to the student {Search&Skip}
  - [2] TinyBERT: mapping some arbitrary layers and self-attention matrices {Search&Skip}
  - [3] Combinatorial KD (CKD) (Wu et al. 2020): combining layers to solves the skip problem.



PKD



TinyBERT



CKD

[ ] Passban, Peyman, et al. "ALP-KD: Attention-Based Layer Projection for Knowledge Distillation." AAAI 2020, *arXiv preprint arXiv:2012.14022* (2020).

Our Solution: Attention-based Layer Projection (ALP-KD)

- In order to solve the search and skip problem at the same time, we propose that **each layer of the student to attend to all layers of the teacher.**
- **Benefit:** attention score will determine the weight or contribution of each layer of the teacher in the distillation process.

$$\mathcal{C}^j = \sum_{h_{\mathcal{T}}^k \in \mathcal{A}(j)} \alpha_{jk}\, h_{\mathcal{T}}^k$$

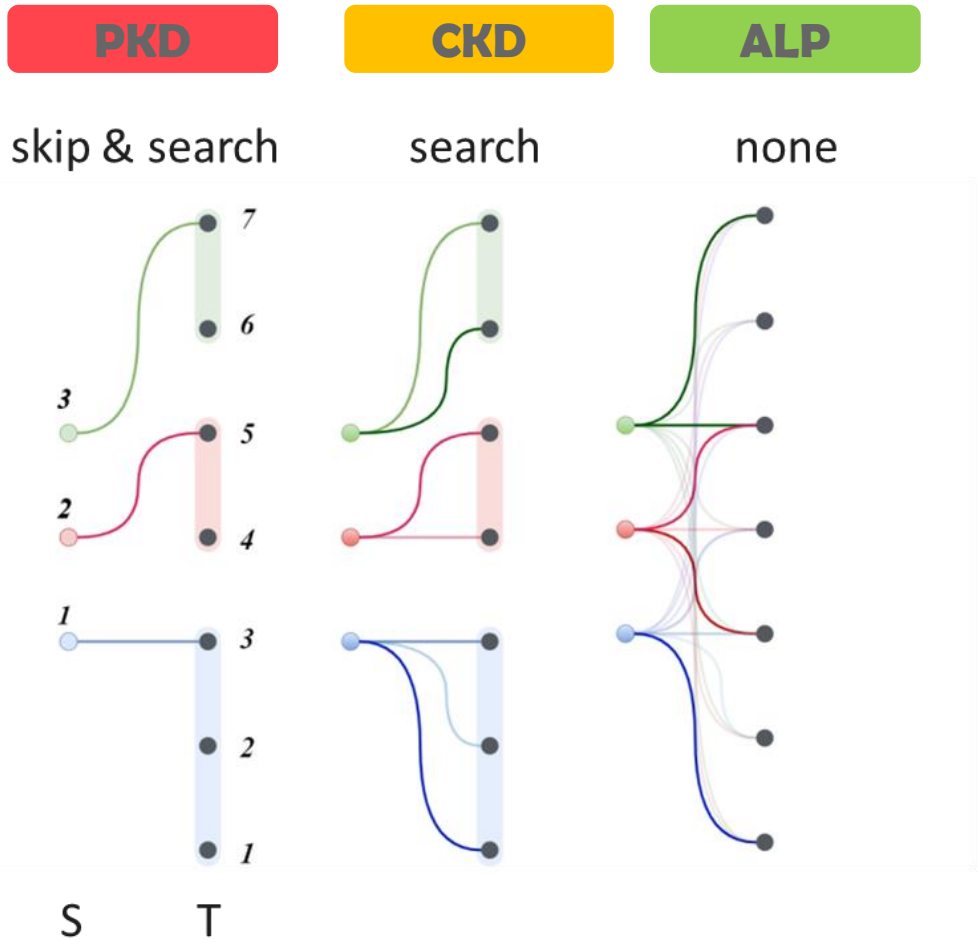
$$\alpha_{jk} = \frac{\exp(h_{\mathcal{S}}^j . h_{\mathcal{T}}^k)}{\sum_{h_{\mathcal{T}}^k \in \mathcal{A}(j)} \exp(h_{\mathcal{S}}^j . h_{\mathcal{T}}^{k'})}$$

$$\cup_{j=1}^{m} \mathcal{A}(j) = H_{\mathcal{T}} = \{h_{\mathcal{T}}^1, ..., h_{\mathcal{T}}^n\}$$

- The final training loss will be as following:

$$\mathcal{L} = \beta \mathcal{L}_{ce} + \eta \mathcal{L}_{KD} + \lambda \mathcal{L}_{ALP}$$

$$\mathcal{L}_{ALP} = \sum_{i=1}^{N} \sum_{j=1}^{m} MSE(h_{\mathcal{S}}^{i,j}, \mathcal{C}^{i,j})$$



[ ] Passban, Peyman, et al. "ALP-KD: Attention-Based Layer Projection for Knowledge Distillation." AAAI 2020, *arXiv preprint arXiv:2012.14022* (2020).

HUAWEI

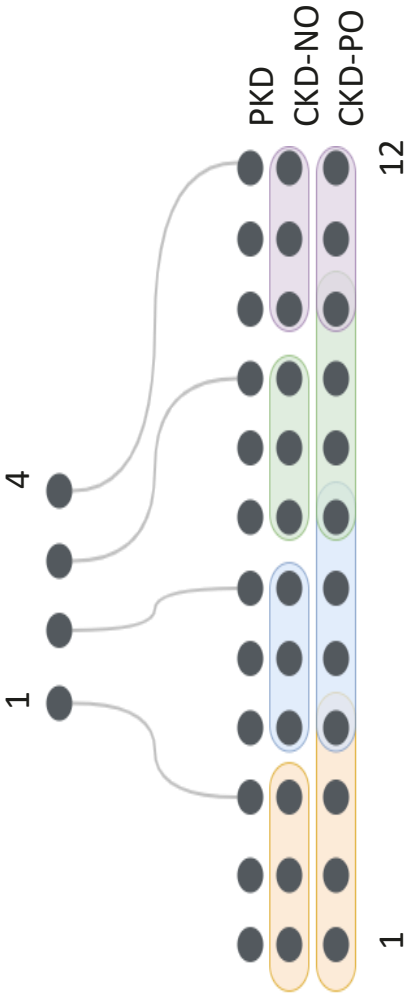
Experiments:

(12 → 4)

- ALP has the best average score among all intermediate-layer KD methods.

Problem	Model	CoLA	MNLI	MRPC	QNLI	QQP	RTE	SST-2	STS-B	Average
N/A	$\mathcal{T}_{\text{BERT}}$	57.31	83.39	86.76	91.25	90.96	68.23	92.67	88.82	82.42
N/A	$\mathcal{S}_{\text{NKD}}$	31.05	76.83	77.70	85.13	88.97	61.73	88.19	87.29	74.61
skip, search	$\mathcal{S}_{\text{RKD}}$	29.22	79.31	79.41	86.77	90.25	65.34	90.37	87.45	76.02
skip, search	$\mathcal{S}_{\text{PKD}}$	32.13	79.26	80.15	86.64	90.23	65.70	90.14	87.26	76.44
search	$\mathcal{S}_{\text{CKD-NO}}$	31.23	79.42	80.64	86.93	88.70	66.06	90.37	87.62	76.37
search	$\mathcal{S}_{\text{CKD-PO}}$	31.95	79.53	80.39	86.75	89.89	**67.51**	90.25	87.55	76.73
search	$\mathcal{S}_{\text{ALP-NO}}$	**34.21**	79.26	79.66	**87.11**	**90.72**	65.70	90.37	87.52	76.82
search	$\mathcal{S}_{\text{ALP-PO}}$	33.86	**79.74**	79.90	86.95	90.25	66.43	**90.48**	87.52	76.89
none	$\mathcal{S}_{\text{ALP}}$	33.07	**79.62**	**80.72**	87.02	90.54	67.15	90.37	**87.62**	**77.01**

Table 1: Except the teacher ($\mathcal{T}_{\text{BERT}}$) which is a 12-layer model, all other models have 4 layers. The firs column shows what sort of problems each model suffers from. NKD stands for *No KD* which means there is no KD technique involved during training the student model. *NO* and *PO* are different configurations for mapping internal layers. Boldfaced numbers show the best student score for each column over the validation sets. The scores in the first column are Matthew's Correlations. SST-B scores are Pearson correlations and the rest are accuracy scores.



[ ] Passban, Peyman, et al. "ALP-KD: Attention-Based Layer Projection for Knowledge Distillation." AAAI 2020, *arXiv preprint arXiv:2012.14022* (2020).

Experiments:

(12 → 2)

- the gap between PKD and ALP-KD is even more visible. This result points out to the fact that when teacher and student models differ much, intermediate layer combination becomes vital.

Problem	Model	CoLA	MNLI	MRPC	QNLI	QQP	RTE	SST-2	STS-B	Average
N/A	$\mathcal{T}_{\text{BERT}}$	57.31	83.39	86.76	91.25	90.96	68.23	92.67	88.82	82.42
N/A	$\mathcal{S}_{\text{NKD}}$	14.50	72.73	72.06	79.61	86.89	57.04	85.89	40.80	63.69
skip, search	$\mathcal{S}_{\text{RKD}}$	24.50	74.90	73.53	81.04	87.40	59.21	87.39	41.87	66.23
skip, search	$\mathcal{S}_{\text{PKD-1}}$	23.09	74.65	72.55	81.27	87.68	57.40	88.76	43.37	66.1
skip, search	$\mathcal{S}_{\text{PKD-6}}$	22.48	74.57	73.04	80.74	87.70	57.40	88.65	42.92	65.94
skip, search	$\mathcal{S}_{\text{PKD-12}}$	22.46	74.33	72.79	81.22	87.88	57.40	88.76	45.39	66.28
search	$\mathcal{S}_{\text{CKD}}$	**24.69**	74.67	73.04	**81.60**	87.10	58.84	88.65	43.71	66.54
none	$\mathcal{S}_{\text{ALP}}$	24.61	**74.78**	**73.53**	81.24	**88.01**	**59.57**	**88.88**	**46.04**	**67.08**

Table 3: The teacher model $\mathcal{T}_{\text{BERT}}$ and all other student models have 12 and 2 layers, respectively. $\mathcal{S}_{\text{PKD-X}}$ indicates that h_T^X from the teacher is used for distillation.

All-in-1 Comparison: ALP, Annealing, MATE

- **Goal:** Comparing our different KD technique in a unified setting

No KD < KD < PKD < ALP < Annealing < MATE

  - Minimax-DA (MATE-KD) is the most effective solution
  - Training mechanism can be more efficient than intermediate layer distillation

- We combined the best two models (Annealing + MATE)
  - ➢ We got rank 1 on GLUE leaderboard with Combined KD v. 1.0

dataset	BERT-base	DistilBERT	VanillaKD	PKD	ALP-KD	ALP-KD (DistilBERT)	AKD	MKD
CoLA	59.5	51.3	47.3	45.7	47.0	51.8	55.2	**60.4**
MNLI	84.6	82.1	82.8	82.1	81.9	82.9	83.8	84.5
MRPC	90.6	90.1	89.5	89.3	89.2	89.9	90.2	90.5
QNLI	91.5	89.2	89.9	89.3	89.7	89.9	89.8	**91.2**
QQP	91.0	88.5	90.5	90.7	90.7	91.2	91.2	**91.4**
RTE	68.2	59.9	66.0	68.2	68.6	67.5	67.9	70.0
SST-2	93.1	91.3	90.4	91.5	91.9	91.4	92.1	**92.2**
STS-B	88.4	86.9	86.7	88.6	88.6	87.3	87.5	88.5
Avg. score	83.4	79.9	80.2	80.7	81.0	81.5	82.2	83.6

Table 4: GLUE dev result for differnt KD models (BERT-base). Bold number are the best performance reached l
6-layer models in this table.

−	24	Huawei Noah's Ark Lab MTL	CombinedKD-TinyRoBERTa (6 layer 82M parameters, Annealing+MateKD)	81.5
			MATEKD-TinyRoBERTa (6 layer 82M parameters Adversarial KD)	80.9
	26	廖亿	u-PMLM-R (Huawei Noah's Ark Lab)	81.3
	27	Xinsong Zhang	AMBERT-BASE	81.0
+	30	Jacob Devlin	BERT: 24-layers, 16-heads, 1024-hidden	80.5
+	33	xiaok Liu	BERT-EMD(6-layer; Single model; No DA)	78.7
+	35	MobileBERT Team	MobileBERT	78.5
+	37	TinyBERT Team	TinyBERT (6-layer; Single model)	78.1

HUAWEI

Open Problems

We target efficiency of KD from the following point of views:

Data
- Sample Efficiency
- Data Augmentation
- Data Refining

Training
- Early Stopping
- Capacity-Gap problem
- Teacher-Free Distillation
- Improving the loss function: Meta-KD and Contrastive Loss

Structure
- Intermediate layer distillation
- Irregular Objective functions: CRF, Generative loss (GPT or Translation)

Open Problems) Training Efficient Knowledge Distillation

We target efficiency of KD from the following point of views:

- **Training**
  - **Early Stopping :**
    - More accurate teachers are NOT necessarily better teachers for Knowledge distillation: (1) How to know to choose which checkpoint from the teacher for distillation? (2) Is this selection architecture dependent or task dependent? {Grow-KD}
  - **Capacity-Gap problem :**
    - Distilling super-large models such as GPT-2 or GPT-3 can be very challenging. What is the best strategy to fill this gap smoothly? Will our Annealing or Grow-KD work on GPT distillation?
  - **Teacher-Free Distillation :**
    - Can we Remove the Teacher network from the training process?
  - **Improving the loss function:**
    - The training loss of KD can be improved from different perspective. KD loss at least has two or more components but the question is:
      - How to define the weights of these components during training {Meta-Reweighted-KD}
      - Should we include all losses the entire time? What is the best schedule for these losses?
      - What type of loss function is better for KD? (MSE, KL, Contrastive, …) CL has showed a great potential for improving KD training.

Open Problems) Data Efficient Knowledge Distillation

We target efficiency of KD from the following point of views:

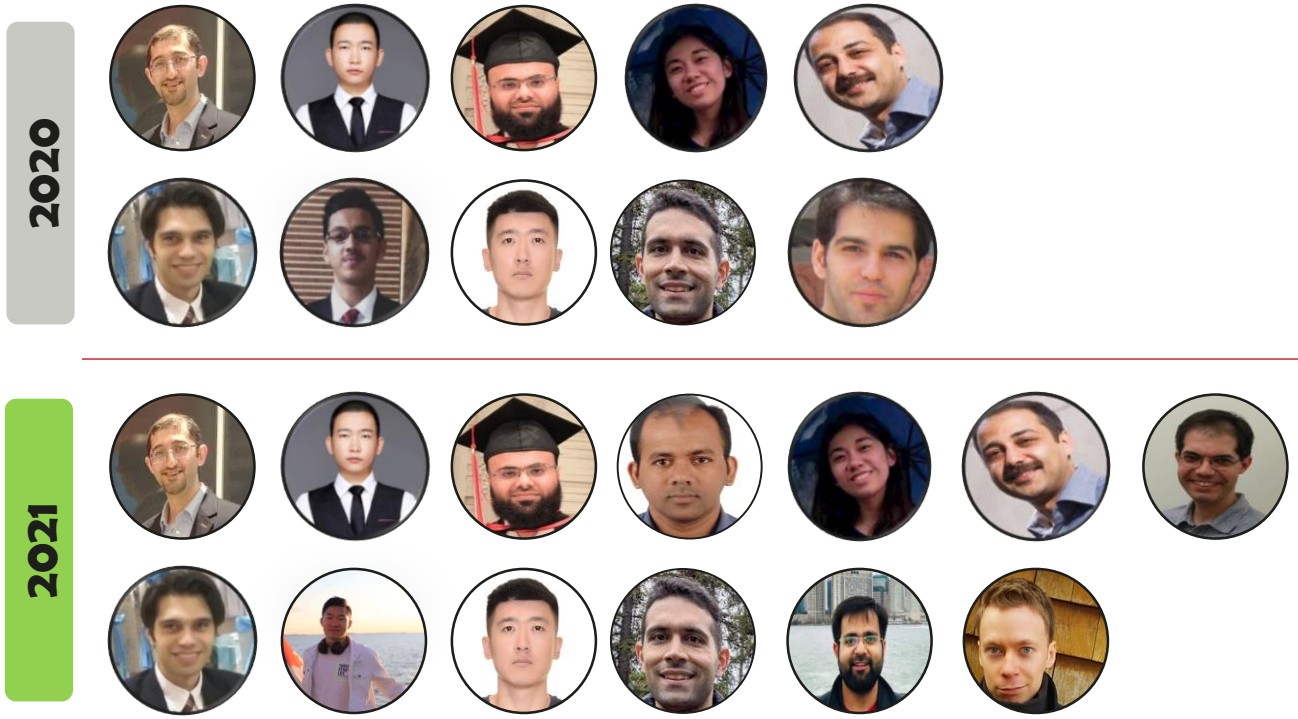
- **Data**
  - **Sample Efficiency :**
    - Improving sample efficiency especially during pre-training: Smallest architecture change requires us to redo the pre-training.
      - Curriculum learning, defining higher dimension labels (rather than one-hot)
      - Generalized Minimax for DA→ Extending to other DA techniques
  - **Data Augmentation :**
    - Dynamic DA: (1) How many augmented samples needs to be incorporated during the training process? (2) Do we need to augment all training samples? (3) How frequent do we need to feed augmented samples during training?
    - What is the most efficient DA technique in terms of in-domain, OOD performance and training time?
    - Further improving MATE-KD: MATE-KD + Contrastive Intermediate Layer Distillation
  - **Data Refining:**
    - In practice we deal with noise or bias in training datasets. How does KD respond to noisy labels? What is the best strategy to deal with label-noise in KD? Can KD help in refining noisy datasets? This step will be useful for task specific trainings.

Structure Efficient Knowledge Distillation

We target efficiency of KD from the following point of views:

- **Structure**
  - **Intermediate layer distillation {K1, K2, K3}**
    - It is shown to be effective in improving KD in the literature. However it is not clear how intermediate layer distillation helps because it is not interpretable. {Early-Exit KD}
    - Intermediate layer distillation is effective but at the same time it is computationally expensive. {Drop-out KD}
  - **Irregular Objective functions such as CRF, Generative loss (GPT or Translation) {K1, K3}:**
    - KD is originally proposed for classification tasks; however, when it comes to GPT distillation, the task is a generative task and we need to distill from a sequence to sequence and also the size of the softmax is much larger. For NER, we deal with CRF which is another irregular loss. So the question is whether our current solutions for BERT based models will work on GPT as well or not?

Thank you.



把数字世界带入每个人、每个家庭、
每个组织，构建万物互联的智能世界。

Bring digital to every person, home, and organization for a fully connected, intelligent world.

The information in this document may contain predictive statements including, without limitation, statements regarding the future financial and operating results, future product portfolio, new technology, etc. There are a number of factors that could cause actual results and developments to differ materially from those expressed or implied in the predictive statements. Therefore, such information is provided for reference purpose only and constitutes neither an offer nor an acceptance. Huawei may change the information at any time without notice.

HUAWEI