Enough of Scaling Laws!

Let's Focus on ***Downscaling***

Tanmoy Chakraborty
Rajiv Khemani Chair Professor in AI
Associate Professor, IIT Delhi

The GPT-3 Story (Estimation)!!

Power Consumption

GPT-3 consumed around **1,287 MWh**

 Running the **London Eye** continuously for **over 5 years**

https://arxiv.org/abs/2104.10350
https://arxiv.org/abs/2005.14165

The GPT-3 Story (Estimation)!!

GPU Consumption

GPT-3 training used **~355 GPU-years** on V100s

 Running 10,000 high-end gaming PCs at full load for 13 days straight

https://lambdalabs.com/blog/demystifying-gpt-3/

The GPT-3 Story (Estimation)!!

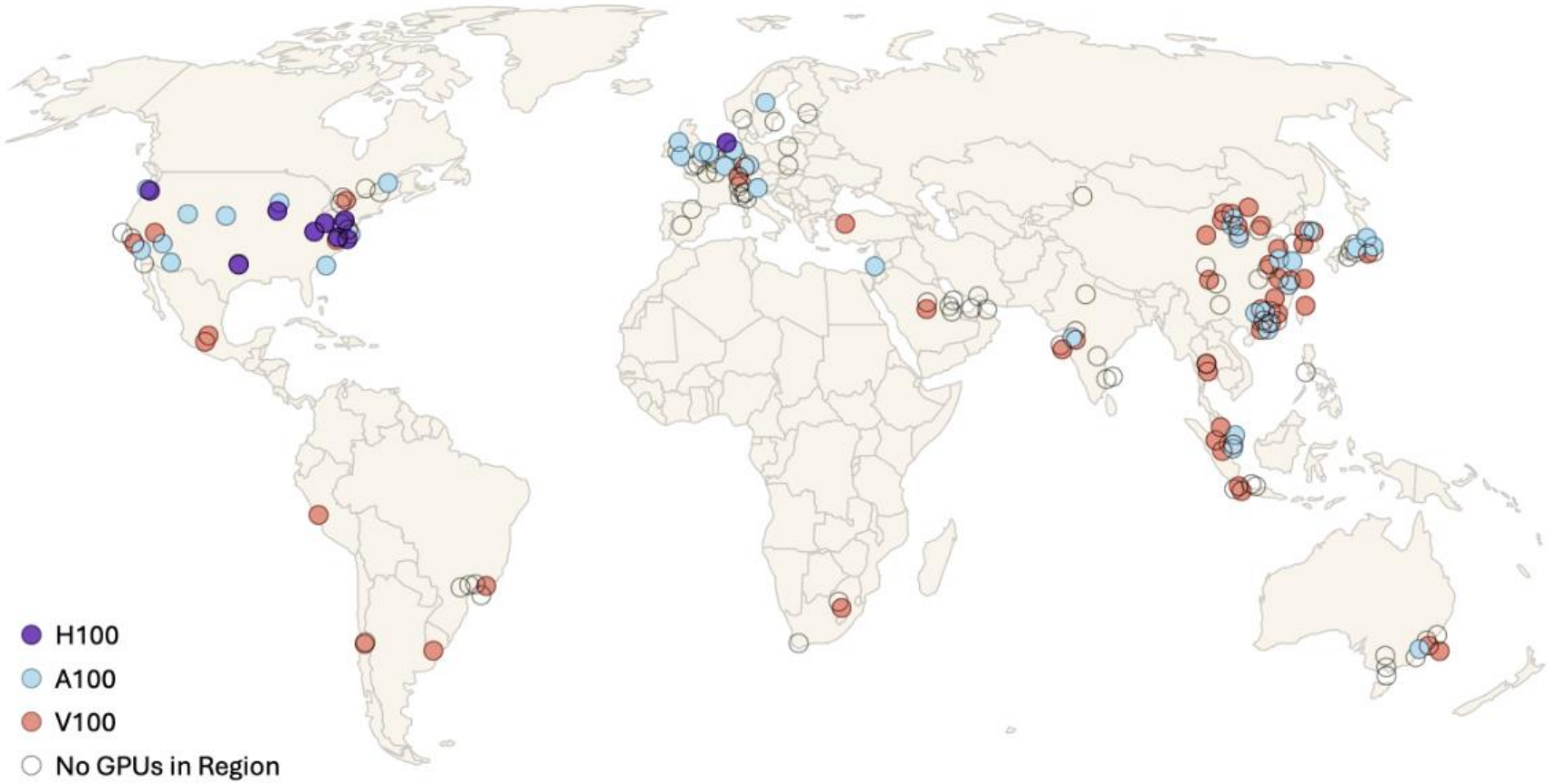
Water Consumption

GPT-3 required ~ **700K liters** of water

- Running a standard shower for 6 years non-stop
- Filling 1.5 million water bottles (500ml each)

https://arxiv.org/abs/2304.03271

AI is science for the rich, not the poor!

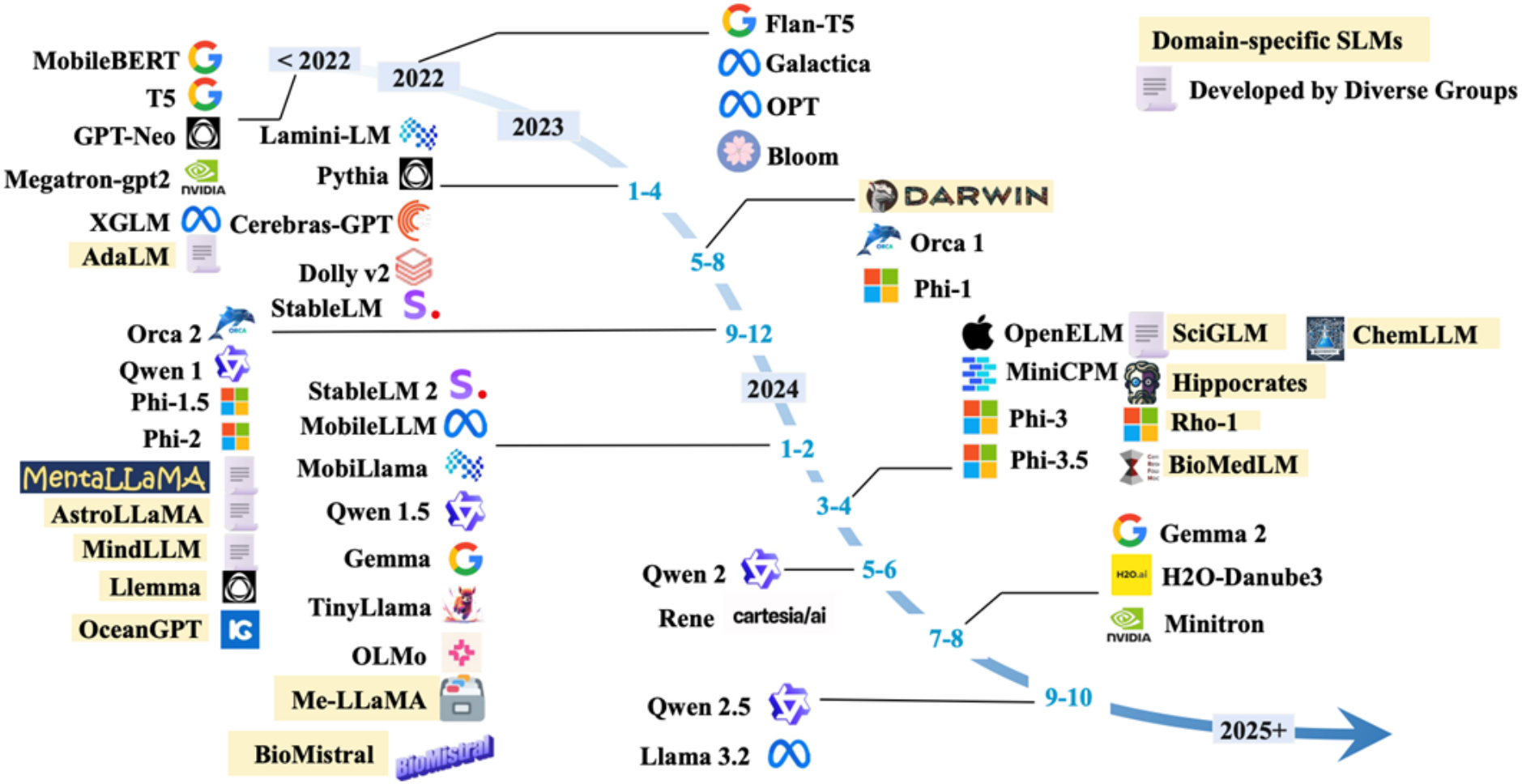


Approximate locations of public cloud regions and the most advanced GPU type available in each region

Image Source: Lehdonvirta, Wú, Hawkins. Compute North vs. Compute South: The Uneven Possibilities of Compute-based AI Governance Around the Globe. AIES 2024

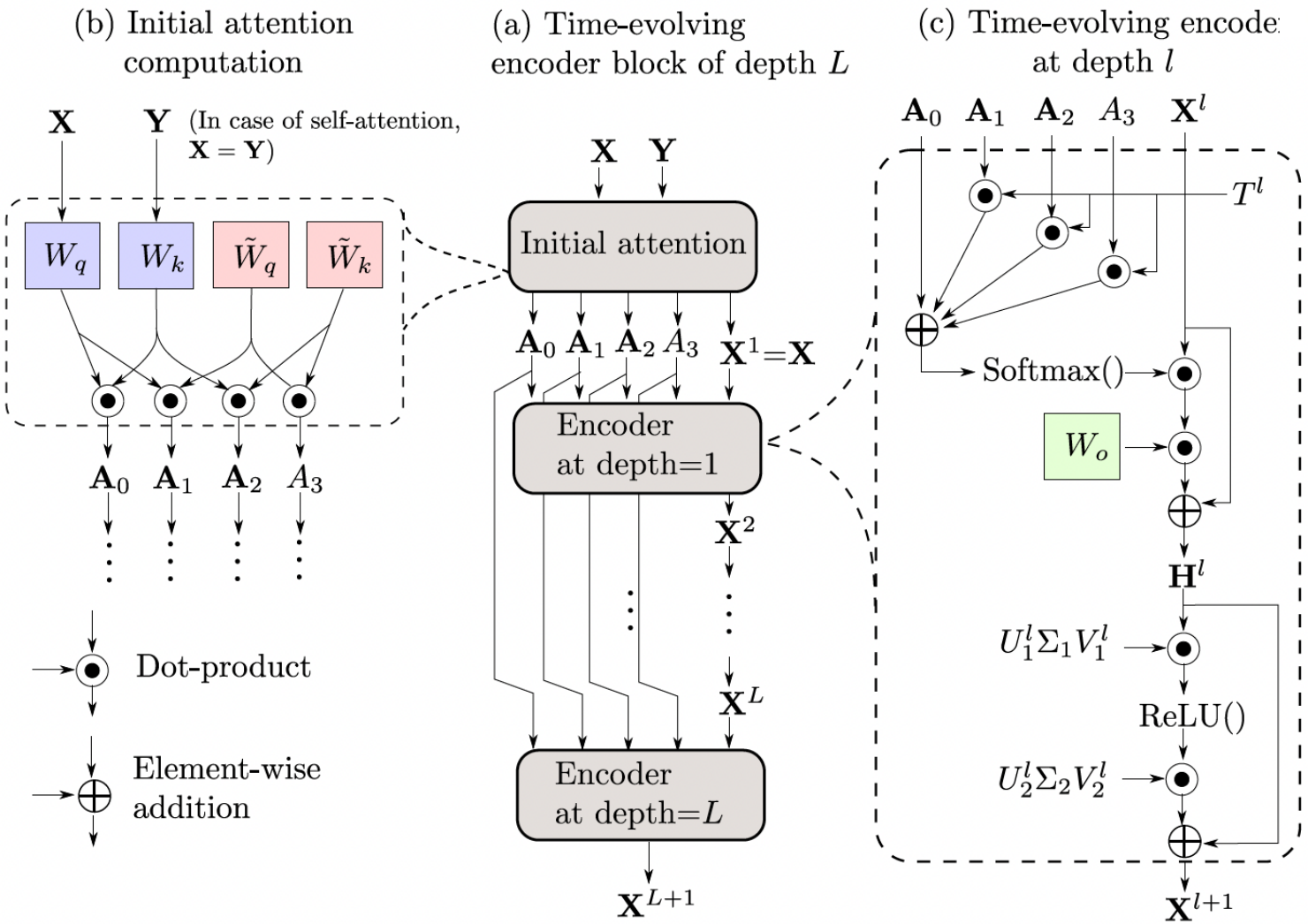
Lets make modern LLMs
less hungry for resource
and **less thirsty** for water!!

Downscaling has already started



https://www.marktechpost.com/2024/11/09/a-deep-dive-into-small-language-models-efficient-alternatives-to-large-language-models-for-real-time-processing-and-specialized-tasks/

TransEvolve



Redesigning the Transformer Architecture with Insights from Multi-particle Dynamical Systems

Subhabrata Dutta Tanya Gautam Soumen Chakrabarti Tanmoy Chakraborty

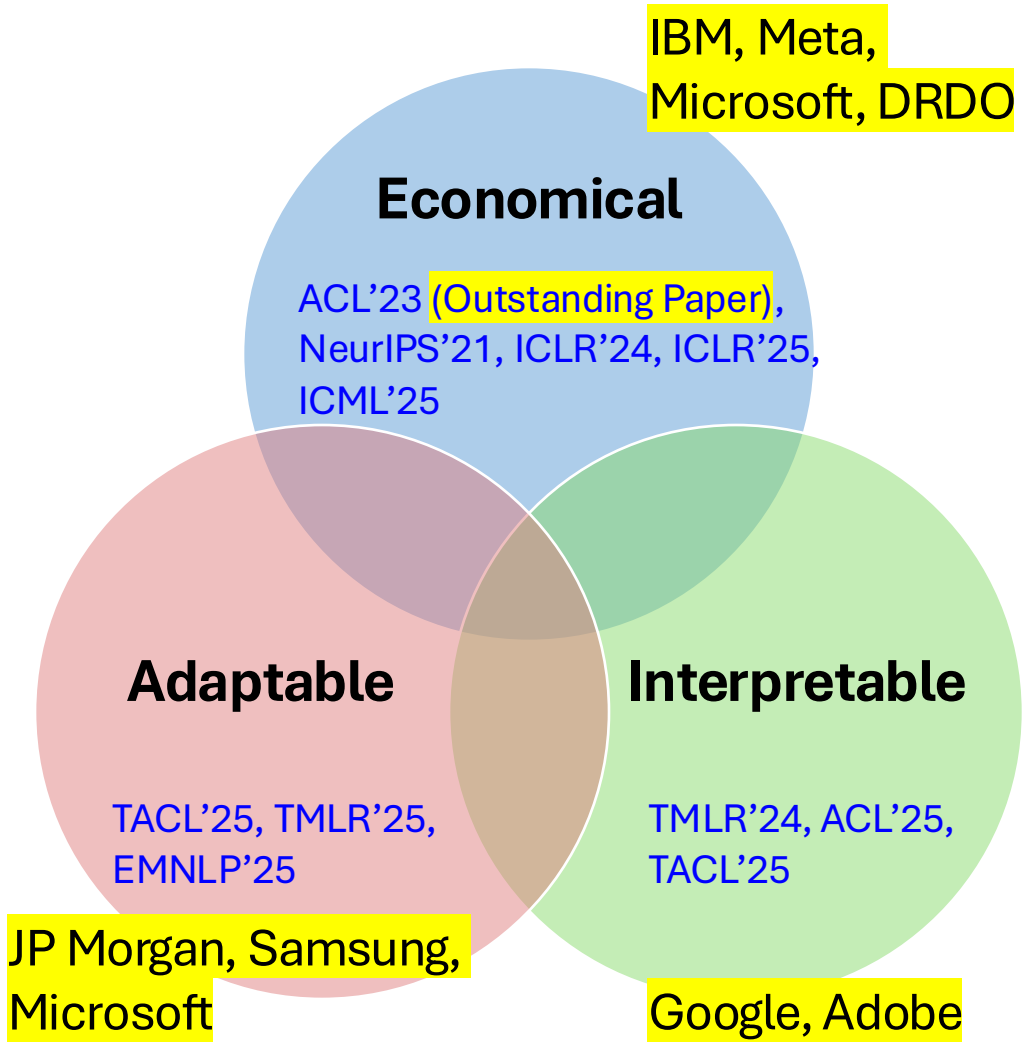
NeurIPS'21 - Spotlight

Deep neural networks as numerical solvers of ordinary differential equations

On all the encoder-only tasks, TransEvolve outperforms Transformer, as well as several strong baselines, with 50% fewer trainable parameters and more than 3× training speedup.

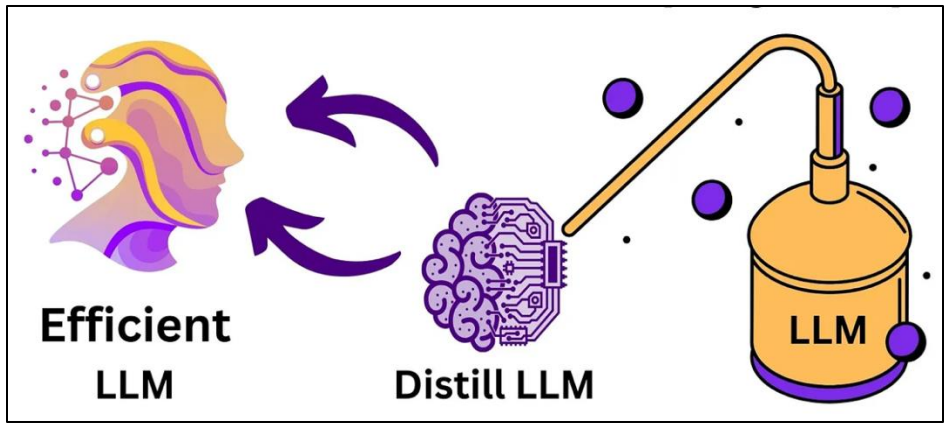
Economical, **Adaptable** and **Interpretable** Models that can reason ***faithfully***

1. Economical – How can we achieve powerful performance with fewer resources?
2. Adaptable – How do we make models generalize to new and low-resource domains?
3. Interpretable – Can we understand ‘why’ and ‘how’ they make predictions? Can we control them?



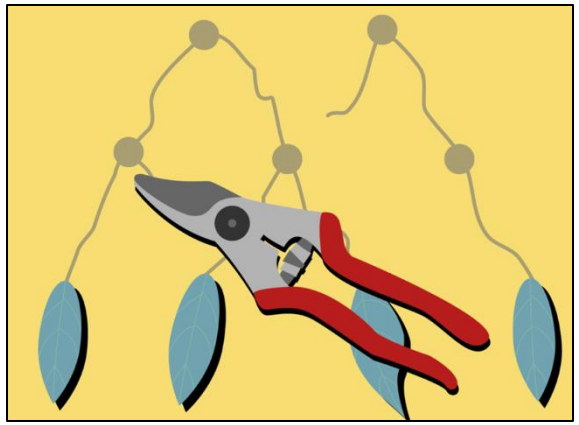
Economical Models

Knowledge Distillation



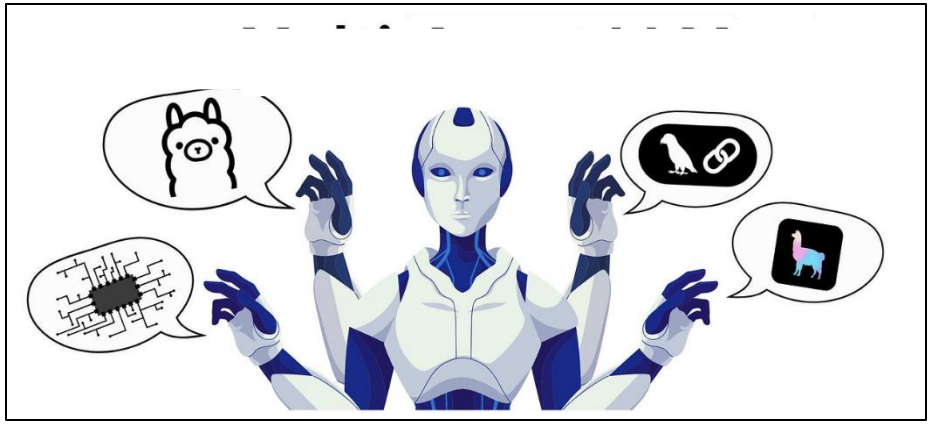
(ICLR'24, TMLR'24, ACL'25)

Model Pruning



(ICLR'25, NeurIPS'24)

Model Coordination



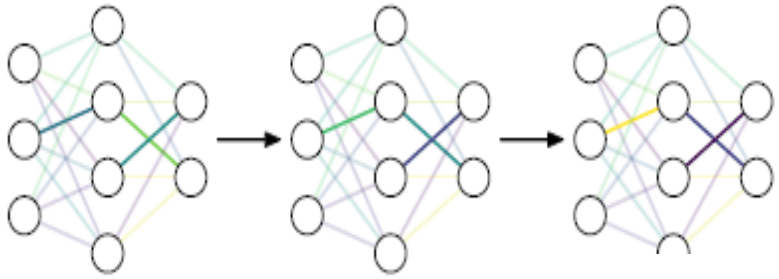
(EMNLP'23, EMNLP'24, AAAI'24, EMNLP'25)

Adaptable Models

Parameter Efficient Fine-Tuning

TACL'25

ID3: Adaptive Selective Fine-tuning of LLMs



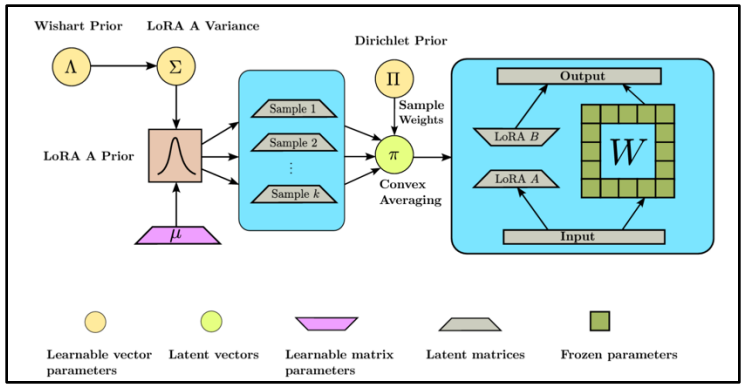
$$\mathcal{H}(\theta^i) = \frac{|\nabla_{\theta^i}|}{(|\theta^i| + \epsilon)^{exp}}$$

Heuristic function calculates the ratio of parameter gradient and magnitude. Epsilon and exp hyperparameters are used to balance exploration-exploitation

Robust Fine-tuning

(TMLR'25)

MontecLoRA: Robust Domain Adaptation



A Bayesian parameterization of low-rank adaptation reduces the variance of posterior estimate, stabilizing the fine-tuning model under different hyperparameters

In-context Adaptation

(ACL'24)

Cross-lingual In-Context Learning (ACL'23)

We proposed X-InSTA - a novel and effective prompt construction strategy for cross-lingual ICL.

ACL 2023 (Outstanding Paper Award)

Cross-task In-Context Learning

We showed how LLMs can leverage cross-task signals to solve novel tasks.

LLM Interpretability

Mechanistic Understanding of CoT

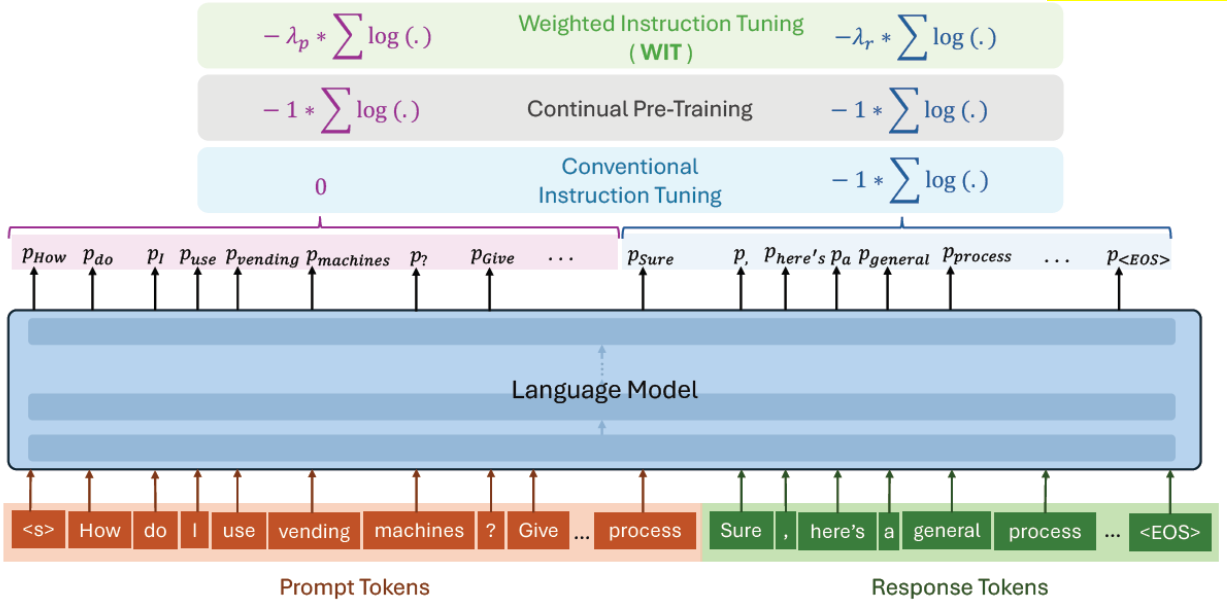
(TMLR'24)

Takeaways:

- Multiple different neural pathways are deployed to compute the answer, that too in parallel.
- parallel answer generation pathways collect answers from different segments of the input.
- Lower layers store pre-trained knowledge, whereas upper layers store in-context knowledge

How Instruction Fine-tuning works?

(TACL'25)

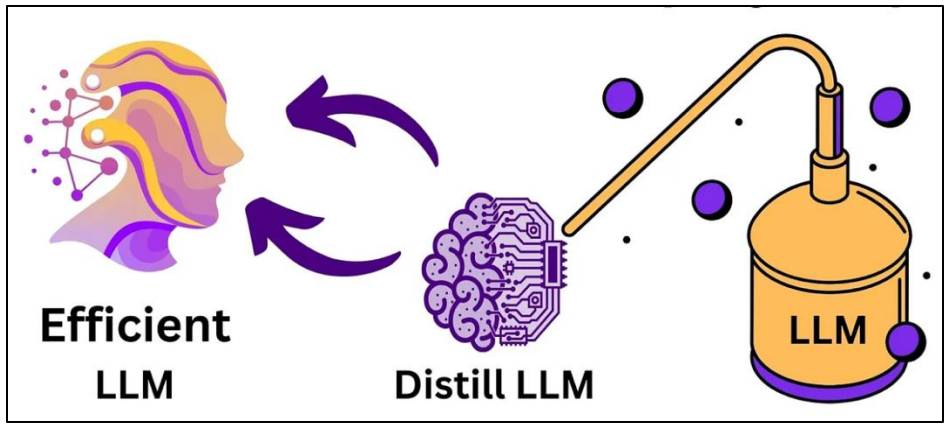


Takeaways:

- The conventional instruction tuning loss rarely yields the best-performing model.
- A moderately high response weight not only enhances performance but also improves model robustness to minor prompt.

Economical Models

Knowledge Distillation



(ICLR'24, TMLR'24, ACL'25)

Model Pruning

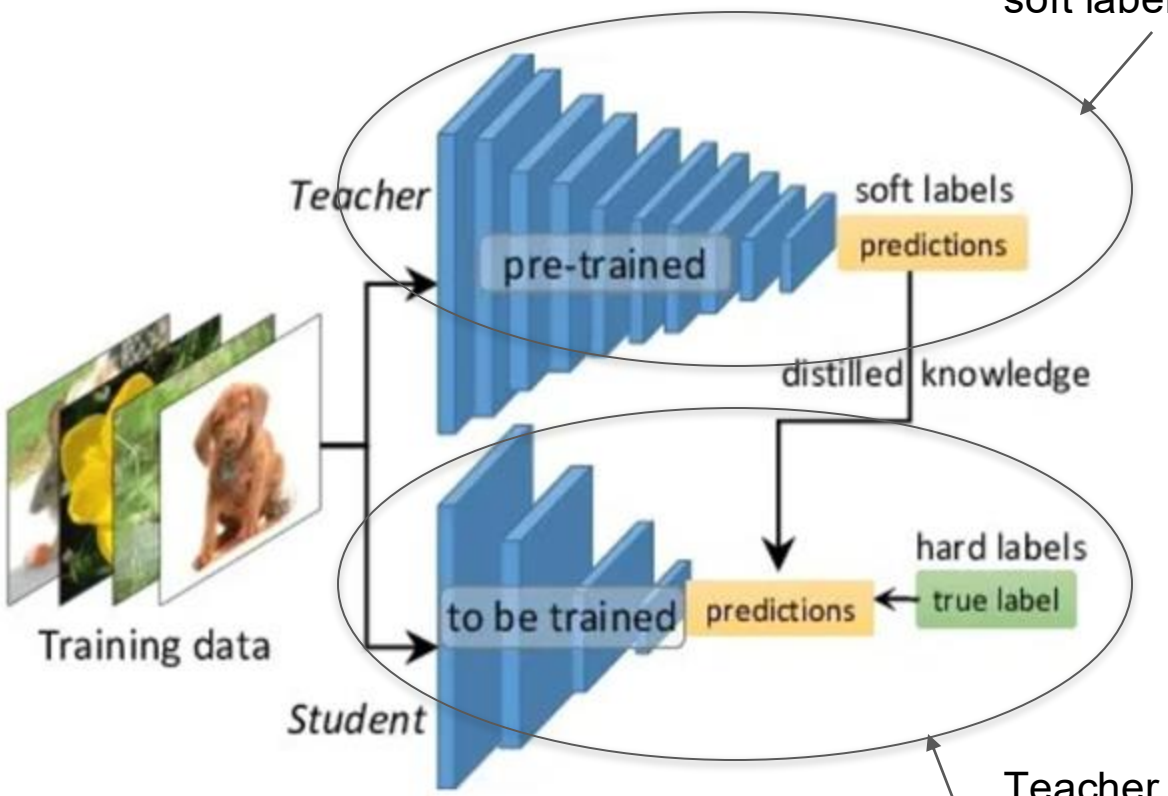
(ICLR'25, NeurIPS'24)

Model Coordination

(EMNLP'23, EMNLP'24, AAAI'24)

Knowledge Distillation (KD): Types

Hinton et al., 2015



Teacher model generate soft labels (logits)

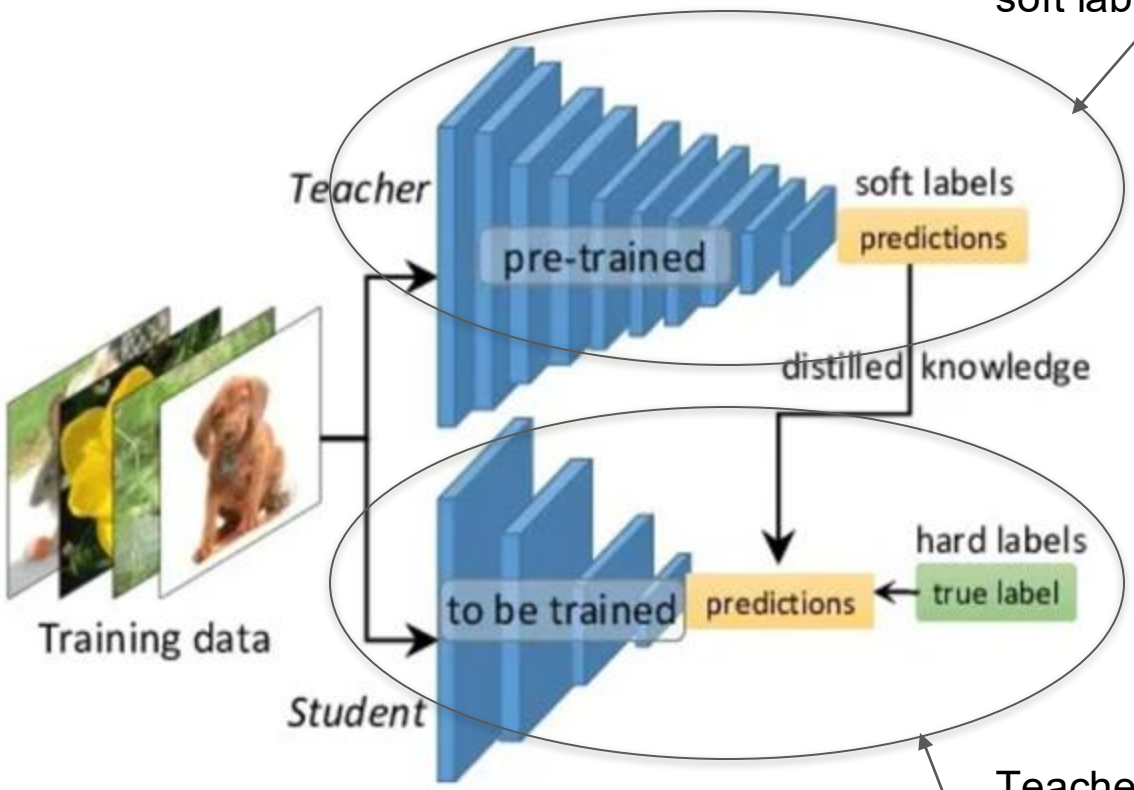
Teacher generated logits used to fine-tune the student

$$\left\| \Phi\left(\frac{\hat{y}_{(.,T)}}{\tau}\right) - \Phi\left(\frac{\hat{y}_{(.,S)}}{\tau}\right) \right\|_2$$

Knowledge Distillation (KD): Types

Hinton et al., 2015

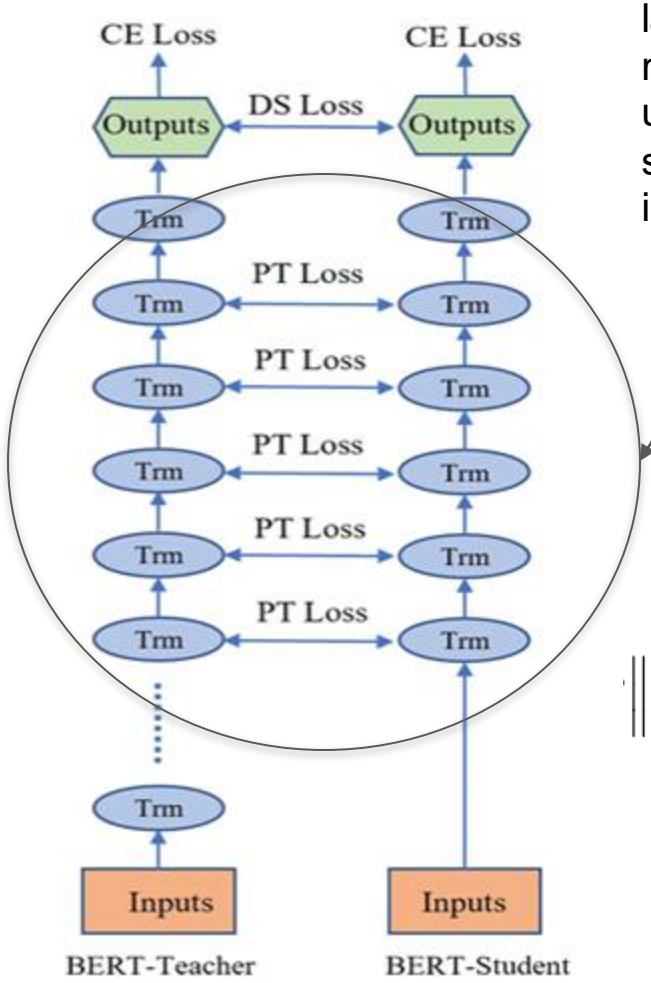
Teacher model generate soft labels (logits)



Teacher generated logits used to fine-tune the student

$$\left\| \Phi\left(\frac{\hat{y}_{(.,T)}}{\tau}\right) - \Phi\left(\frac{\hat{y}_{(.,S)}}{\tau}\right) \right\|_2$$

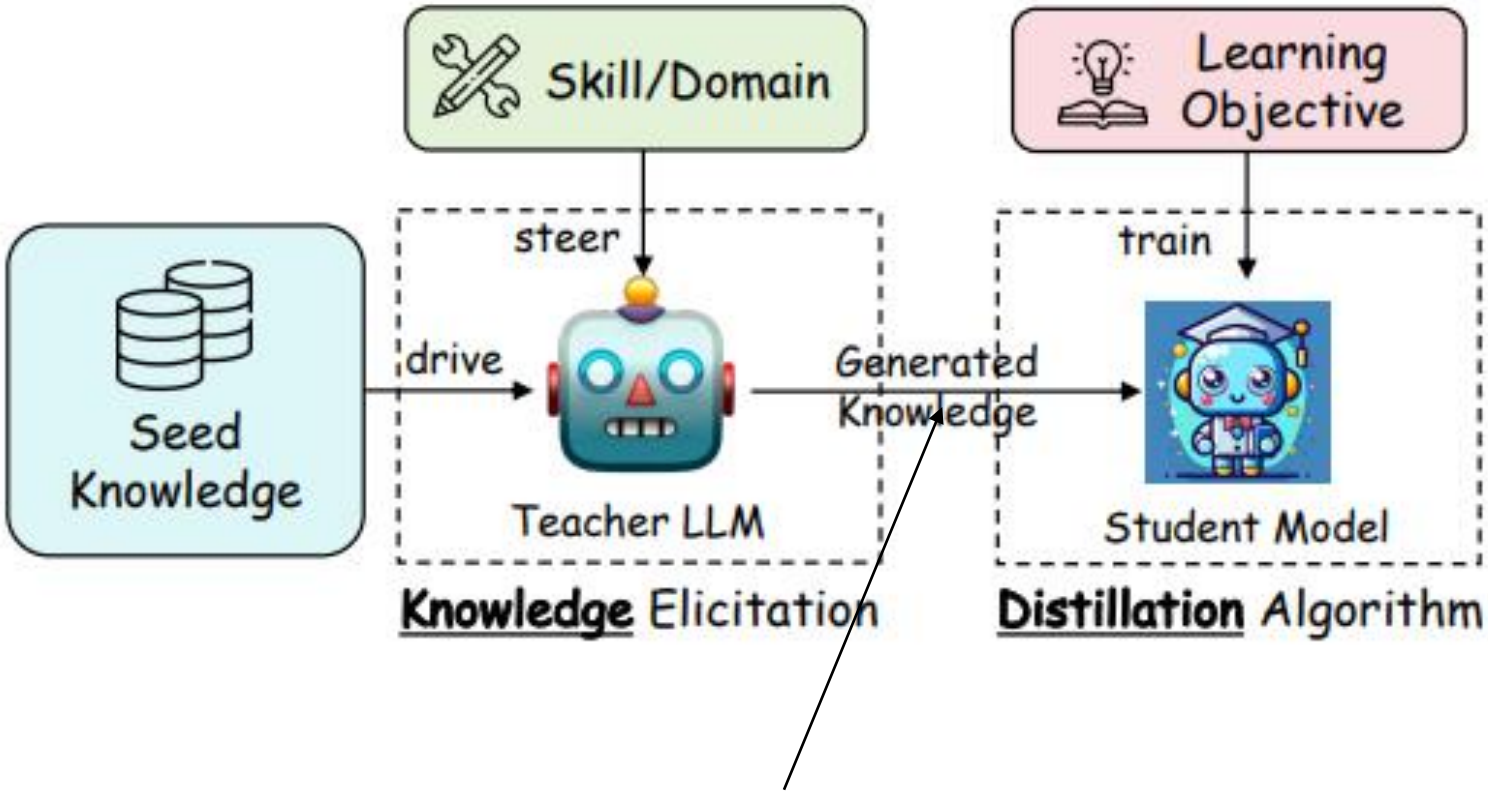
Distance between layer-wise representations used to enforce student model to imitate the teacher



$$\left\| h_{(.,T)} - h_{(.,S)} \right\|_2$$

Sun et al., 2019

Limitations of Vanilla KD

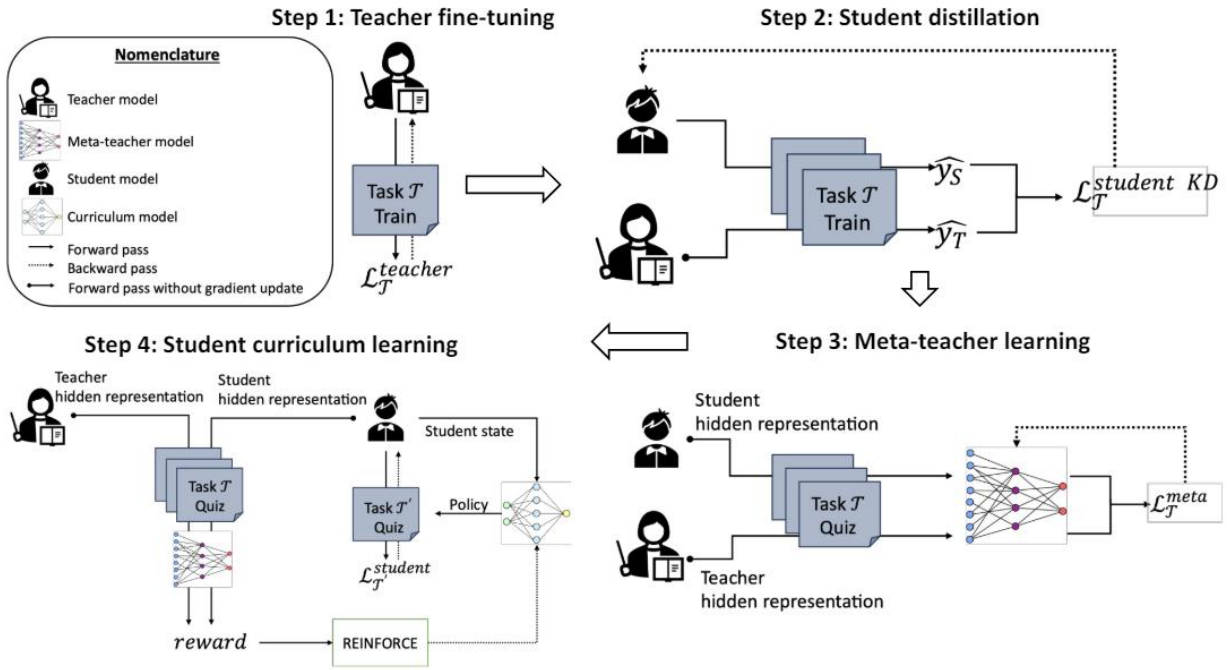


Knowledge sharing is **unidirectional**, *i.e.*, teacher is not aware of student's capacity

Image reference: Xu et al., 2024

MPDistil: Student-Aware Meta Distillation: *Learning to teach*

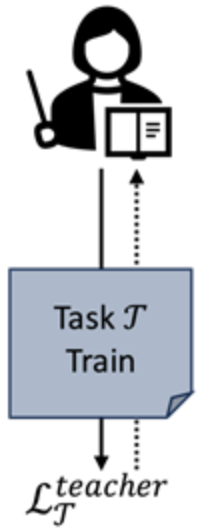
- A **healthy competition** between the teacher and student can encourage both the models to perform better.
- A **better teacher can set a higher benchmark for the student**, enhancing student's performance.
- The student can devise **better learning strategy** (curriculum) to perform better than the teacher.



Sengupta, Dixit, Akhtar, **Chakraborty**. A Good Learner Can Teach Better: Teacher-Student Collaborative Knowledge Distillation. **ICLR** 2024.

MPDistil: Step 1 -- Teacher Fine-tuning

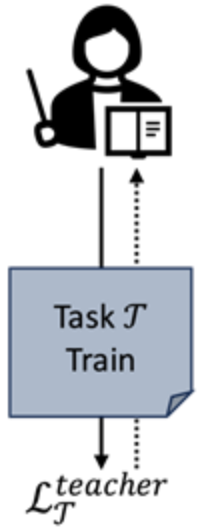
1. Teacher Fine-tuning



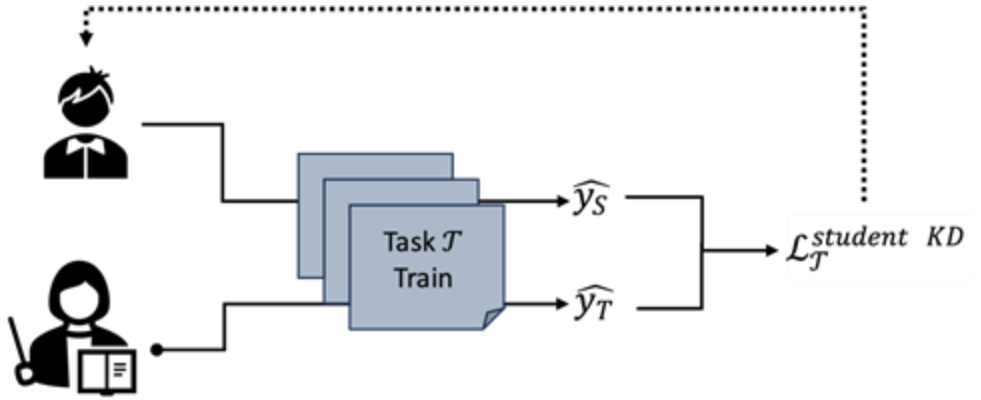
$$\mathcal{L}_{\mathcal{T}}{}^{teacher} = \frac{1}{N}\sum_{i=1}^{N}\mathcal{L}_{\mathcal{T}}(y_i, \hat{y}_{(i,T)}), \text{ with } \hat{y}_{(i,T)} = T(x_i; \theta_T)$$

MPDistil: Step 2 -- Student Distillation

1. Teacher Fine-tuning



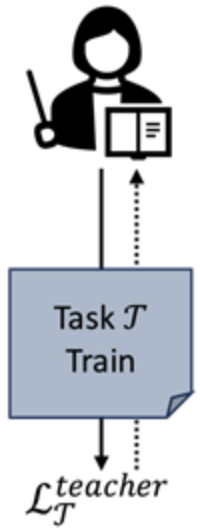
2. Student Distillation



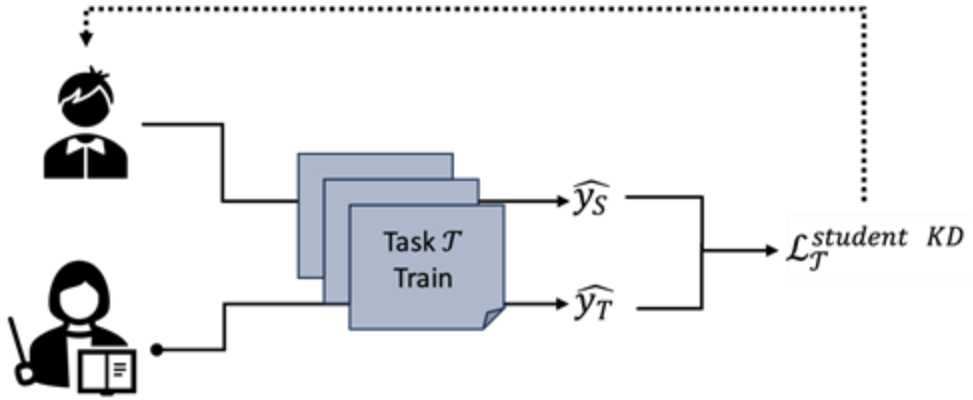
$$\mathcal{L}_{\mathcal{T}}^{\text{student KD}} = \frac{1}{N}\sum_{i=1}^{N} \alpha \mathcal{L}_{\mathcal{T}}(y_i, \hat{y}_{(i,S)}) + (1-\alpha)\left\|\Phi\left(\frac{\hat{y}_{(.,T)}}{\tau}\right) - \Phi\left(\frac{\hat{y}_{(.,S)}}{\tau}\right)\right\|_2 + \beta\left\|h_{(.,T)} - h_{(.,S)}\right\|_2$$

MPDistil: Step 3 -- Meta-teacher Learning

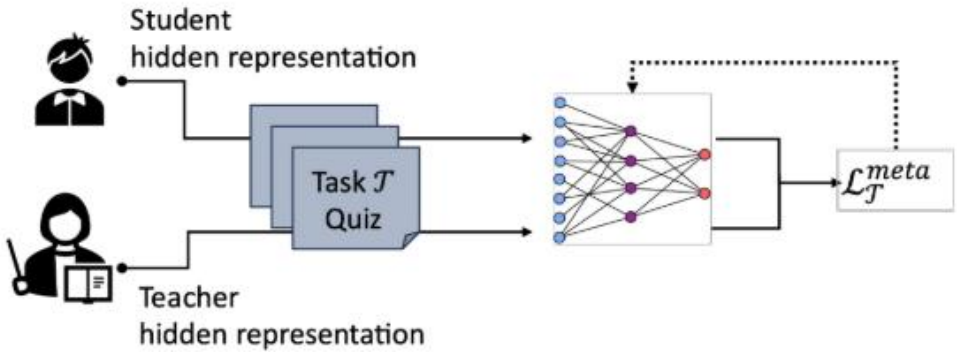
1. Teacher Fine-tuning



2. Student Distillation



3. Teacher Meta Learning
(on a quiz dataset)



Collaborative Loss

$$\mathcal{L}_{\mathcal{T}}^{\text{meta col}} = \begin{cases} -\frac{1}{2N}\sum_{i=1}^{N}\left[\log \bar{y}_{(i,T)} + \log \bar{y}_{(i,S)}\right], & \text{if } \mathcal{T} \text{ is a classification task} \\ \frac{1}{2}\left\|y - \hat{\hat{y}}_{(.,T)}\right\|_2 + \frac{1}{2}\left\|y - \hat{\hat{y}}_{(.,S)}\right\|_2, & \text{if } \mathcal{T} \text{ is a regression task} \end{cases}$$

Competitive Loss

$$\mathcal{L}_{\mathcal{T}}^{\text{meta com}} = \begin{cases} -\frac{1}{N}\sum_{i=1}^{N}\left[2\log \bar{y}_{(i,T)} - \log \bar{y}_{(i,S)}\right], & \text{if } \mathcal{T} \text{ is a classification task} \\ \left\|y - \hat{\hat{y}}_{(.,T)}\right\|_2 - \frac{1}{2}\left\|y - \hat{\hat{y}}_{(.,S)}\right\|_2, & \text{if } \mathcal{T} \text{ is a regression task} \end{cases}$$

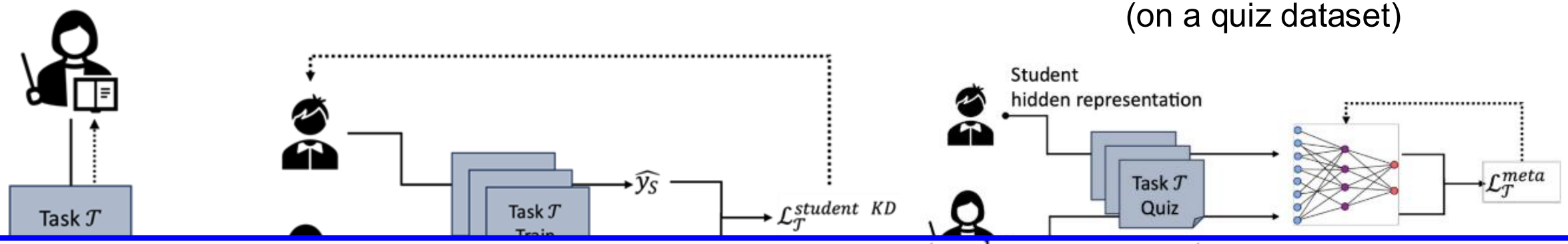
Intuition: The meta-teacher obtains the hidden states from both teacher and student and creates a healthy competition between the models.

MPDistil: Step 3 -- Meta-teacher Learning

1. Teacher Fine-tuning

2. Student Distillation

3. Teacher Meta Learning (on a quiz dataset)



Proposition 1. For any classification task $\mathcal{T}$, having $\mathbb{E}[\mathcal{L}_{\mathcal{T}}^{\text{meta col}}] < \mathbb{E}[\mathcal{L}_{\mathcal{T}}^{\text{meta com}}]$ ensures stronger student with $\mathbb{E}[\bar{y}_S] > \mathbb{E}[\bar{y}_T]$.

Collaborative Loss

$$\mathcal{L}_{\mathcal{T}}^{\text{meta col}} = \begin{cases} -\frac{1}{2N}\sum_{i=1}^{N}\left[\log \bar{y}_{(i,T)} + \log \bar{y}_{(i,S)}\right], & \text{if } \mathcal{T} \text{ is a classification task} \\ \frac{1}{2}\left\|y - \hat{\hat{y}}_{(.,T)}\right\|_2 + \frac{1}{2}\left\|y - \hat{\hat{y}}_{(.,S)}\right\|_2, & \text{if } \mathcal{T} \text{ is a regression task} \end{cases}$$

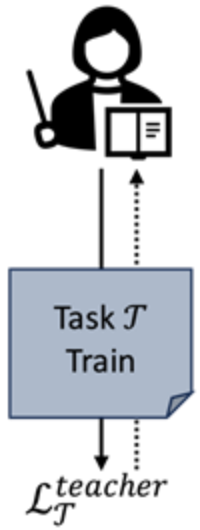
Competitive Loss

$$\mathcal{L}_{\mathcal{T}}^{\text{meta com}} = \begin{cases} -\frac{1}{N}\sum_{i=1}^{N}\left[2\log \bar{y}_{(i,T)} - \log \bar{y}_{(i,S)}\right], & \text{if } \mathcal{T} \text{ is a classification task} \\ \left\|y - \hat{\hat{y}}_{(.,T)}\right\|_2 - \frac{1}{2}\left\|y - \hat{\hat{y}}_{(.,S)}\right\|_2, & \text{if } \mathcal{T} \text{ is a regression task} \end{cases}$$

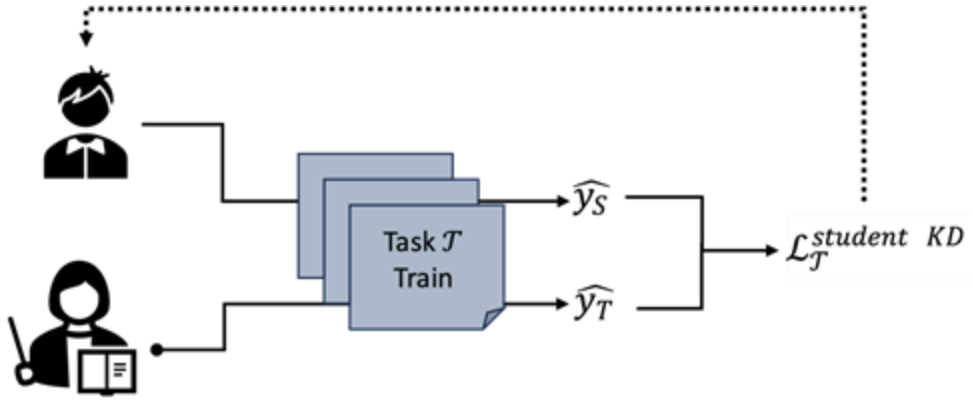
Intuition: The meta-teacher obtains the hidden states from both teacher and student and creates a healthy competition between the models.

MPDistil: Step 4 -- Student Curriculum Learning

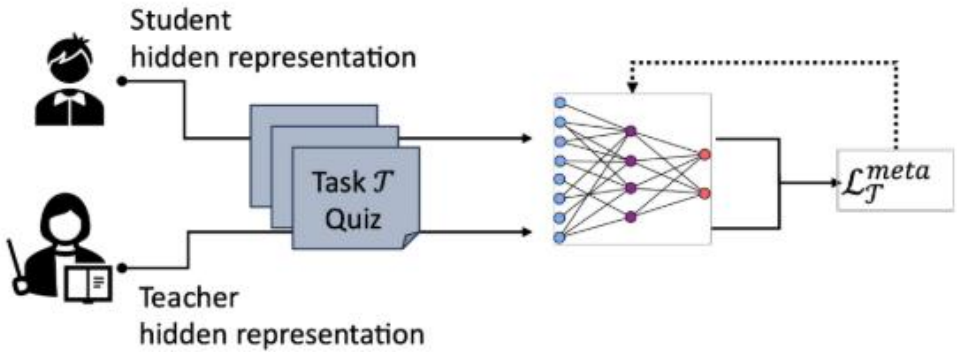
1. Teacher Fine-tuning



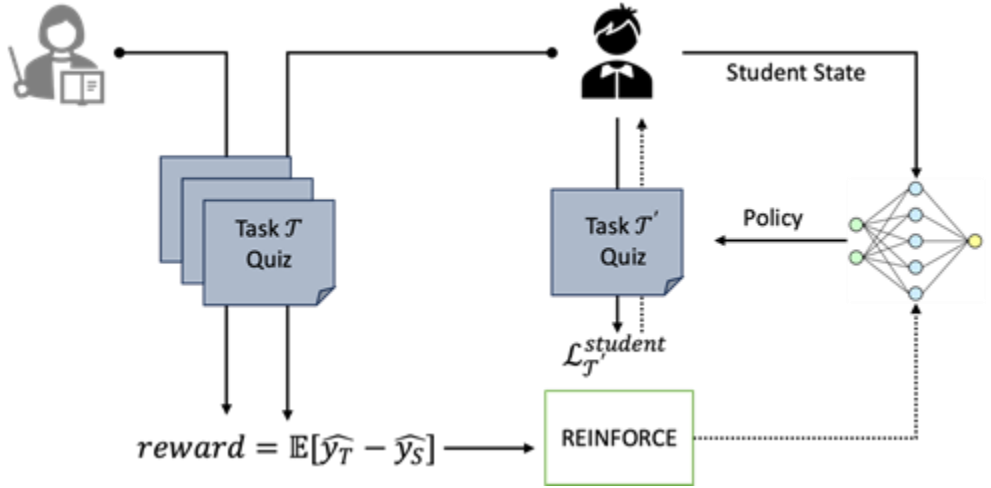
2. Student Distillation



3. Teacher Meta Learning
(on a quiz dataset)



4. Student Curriculum Learning

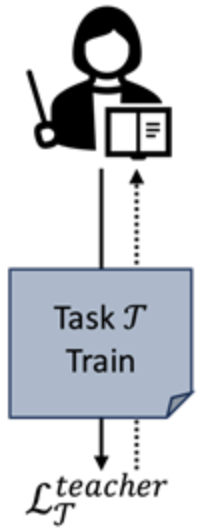


Why Curriculum Learning in KD?

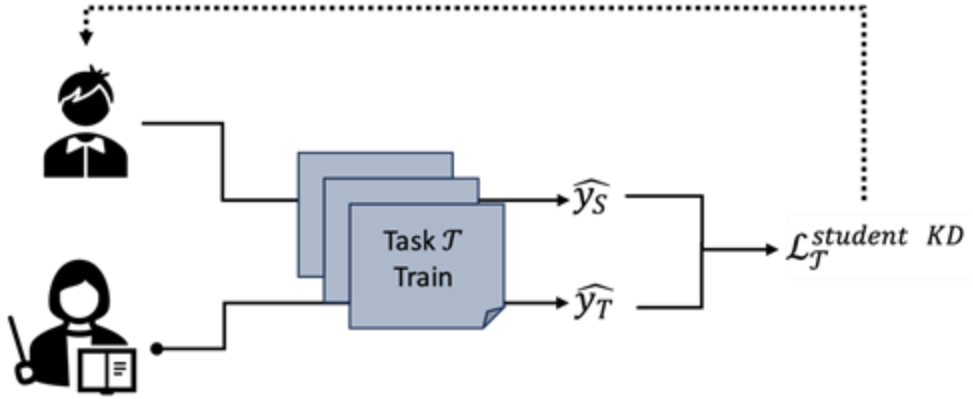
In real world, a student might aim to improve her understanding of Physics by studying selected concepts from Mathematics.

MPDistil: Step 4 -- Student Curriculum Learning

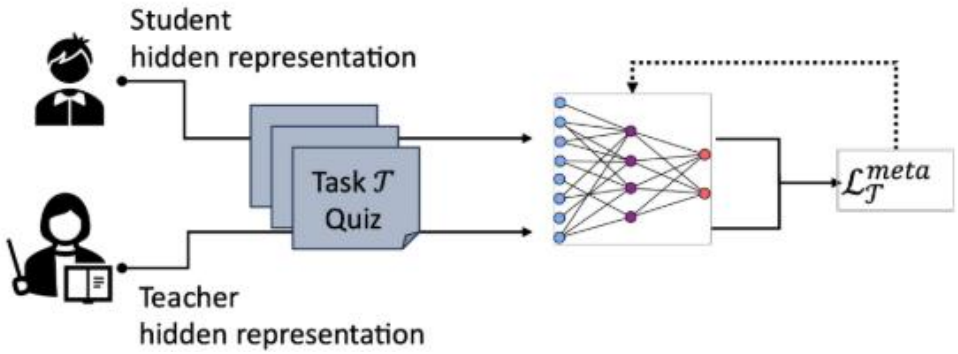
1. Teacher Fine-tuning



2. Student Distillation



3. Teacher Meta Learning
(on a quiz dataset)



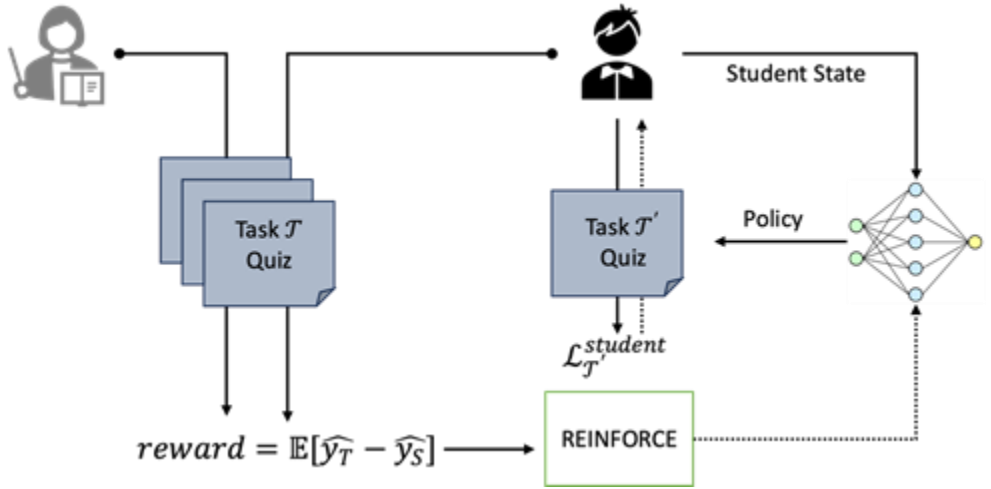
Competing student tries to beat the teacher

A policy network selects optimal curriculum to fine-tune the student by maximizing the reward

$$R^{\text{binary}} = \begin{cases} \mathbb{I}_{\hat{y}_{(i,S)} > \hat{y}_{(i,T')}}, & \text{if } \mathcal{T} \text{is a classification task} \\ \mathbb{I}_{\left\|y_i - \hat{y}_{(i,T')}\right\|_2 > \left\|y_i - \hat{y}_{(i,S)}\right\|_2}, & \text{if } \mathcal{T} \text{is a regression task} \end{cases}$$

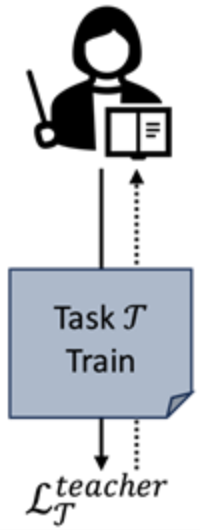
$$R^{\text{real}} = \begin{cases} \hat{y}_{(i,S)} - \hat{y}_{(i,T')}, & \text{if } \mathcal{T} \text{is a classification task} \\ \left\|y_i - \hat{y}_{(i,T')}\right\|_2 - \left\|y_i - \hat{y}_{(i,S)}\right\|_2, & \text{if } \mathcal{T} \text{is a regression task} \end{cases}$$

4. Student Curriculum Learning

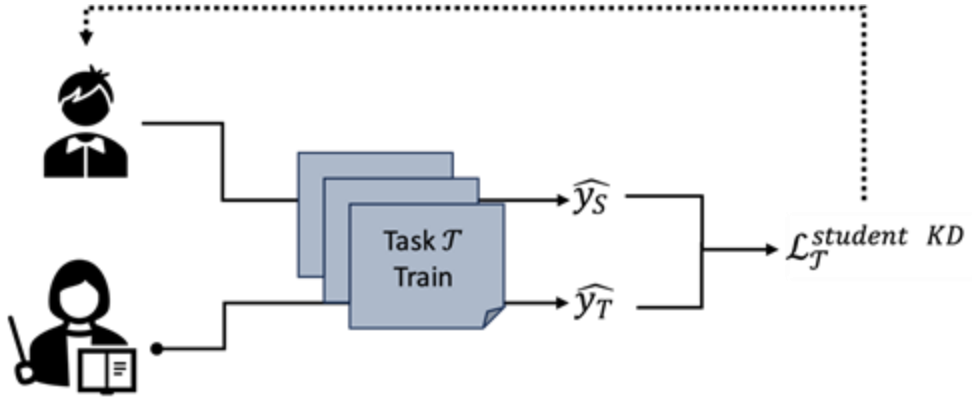


MPDistil: Step 4 -- Student Curriculum Learning

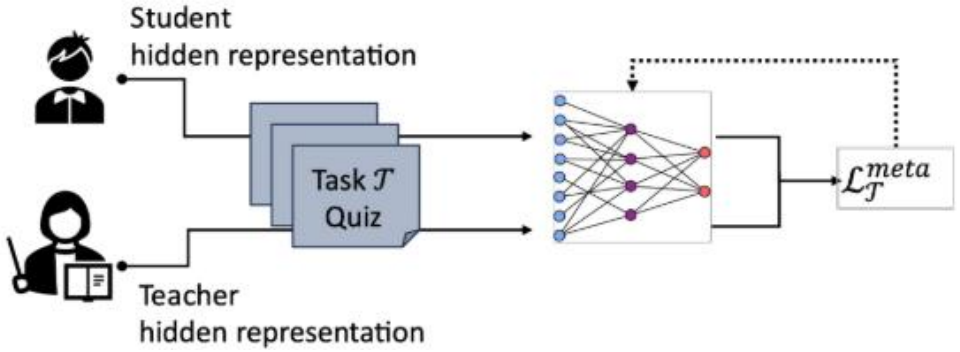
1. Teacher Fine-tuning



2. Student Distillation



3. Teacher Meta Learning
(on a quiz dataset)



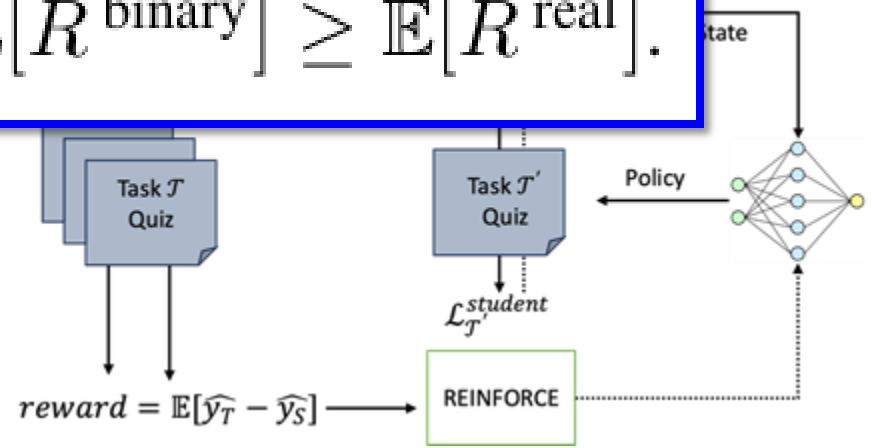
4. Student Curriculum Learning

Proposition 2. For any classification task $\mathcal{T}$, $\mathbb{E}[R^{\text{binary}}] \geq \mathbb{E}[R^{\text{real}}]$.

A policy network selects optimal curriculum to fine-tune the student by maximizing the reward

$$R^{\text{binary}} = \begin{cases} \mathbb{I}_{y_{(i,S)} > y_{(i,T')}}, & \text{if } \mathcal{T} \text{is a classification task} \\ \mathbb{I}_{\left\|y_i - \hat{y}_{(i,T')}\right\|_2 > \left\|y_i - \hat{y}_{(i,S)}\right\|_2}, & \text{if } \mathcal{T} \text{is a regression task} \end{cases}$$

$$R^{\text{real}} = \begin{cases} \hat{y}_{(i,S)} - \hat{y}_{(i,T')}, & \text{if } \mathcal{T} \text{is a classification task} \\ \left\|y_i - \hat{y}_{(i,T')}\right\|_2 - \left\|y_i - \hat{y}_{(i,S)}\right\|_2, & \text{if } \mathcal{T} \text{is a regression task} \end{cases}$$



ICLR'24

A "smart" student can beat a teach!!

Methods	BoolQ	CB	COPA	RTE	WiC	WSC
KD Hinton et al. (2015)	-13.3	-19.1	-4.3	-3.7	-9.1	-14.4
PD Turc et al. (2019) †	-9.6	-9.5	-0.3	-13.5	-6.9	-11.2
PKD Sun et al. (2019)	-1.7	-5.9	-6.0	-3.8	-0.4	-12.5
DistilBERT Sanh et al. (2019) †	-6.0	-7.7	-1.0	-12.0	-5.8	-9.3
Theseus Xu et al. (2020) †	-1.6	-3.6	-4.3	-4.8	-1.8	-11.5
TinyBERT Jiao et al. (2019)	**-1.4**	-1.2	4.3	-3.7	1.7	-2.9
MobileBERT Sun et al. (2020) †	-4.8	-2.4	-0.7	-14.0	-2.3	-9.3
SID Aguilar et al. (2020) †	-10.1	-17.3	-1.0	-14.8	-9.0	-12.8
MiniLM Wang et al. (2020b) †	-3.5	-11.9	-4.0	-5.3	-1.2	-14.4
MiniLMv2 Wang et al. (2020a) †	-2.7	-14.3	-4.0	-6.3	-2.5	-15.1
ALP-KD Passban et al. (2021) †	-2.2	-11.3	-5.3	-4.8	-1.3	-13.1
LRC-BERT Fu et al. (2021) †	-4.5	-9.5	-0.3	-16.4	-8.5	-11.2
Annealing-KD Jafari et al. (2021) †	-8.8	-5.9	3.3	-14.0	-6.3	-11.2
CKD Park et al. (2021) †	-7.8	-6.6	-1.0	-11.7	-7.3	-11.2
Universal-KD Wu et al. (2021a) †	-1.8	-5.4	-7.3	-2.8	-0.6	-11.2
DIITO Wu et al. (2021b) †	-3.9	-5.9	6.0	-7.5	-5.4	-8.6
Continuation-KD Jafari et al. (2022) †	-8.0	-7.1	2.7	-14.2	-7.9	-13.1
RAIL-KD Haidar et al. (2021) †	-10.4	-7.7	0.7	-12.4	-5.8	-7.7
MGSKD Liu et al. (2022a)) †	-6.1	-6.6	-1.0	-7.0	-3.0	-12.8
MetaDistil Zhou et al. (2021)	-2.7	-1.8	1.0	-2.0	-1.6	0.9
`MPDistil` (Ours)	-1.9	**0.0**	**7.0**	**0.4**	**2.5**	**1.0**
(-) Curriculum learning	-2.8	-5.3	-4.0	-1.8	1.2	0.0

Positive value indicates the **student model** is **better than the teacher model**

ACL 2025

Explaining Knowledge Distillation

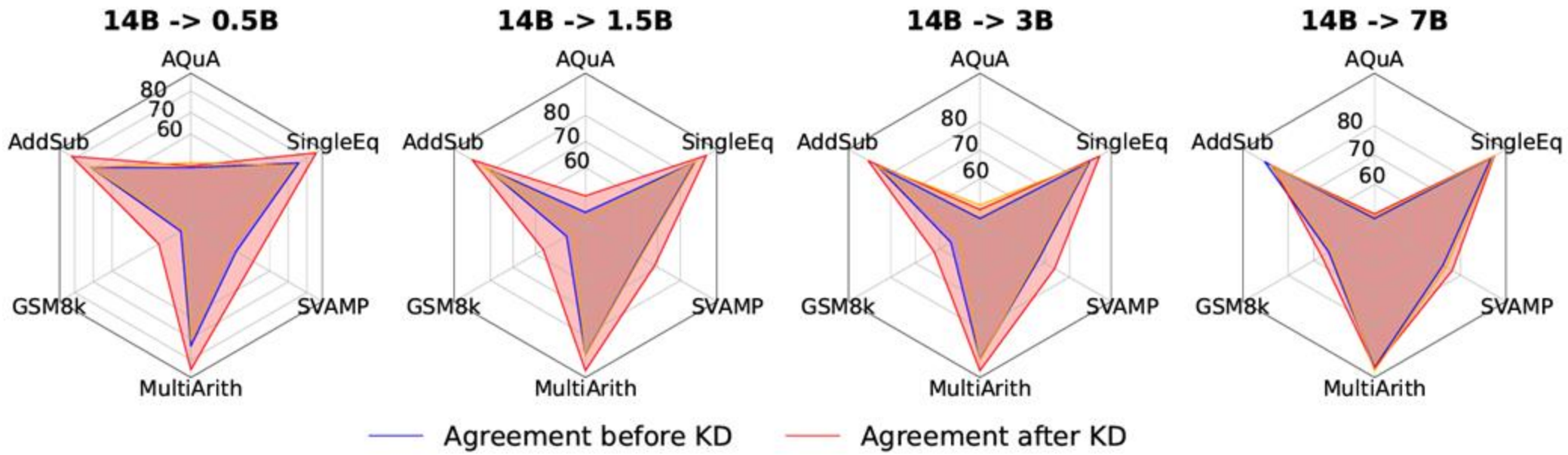
Known: KD improves generalization abilities of student models.

Questions

(i) Post-KD, does student perfectly *imitate* a teacher?
(ii) What are the *key drivers* influencing the effectiveness of KD methods?

ACL 2025

Agreement b/w Teacher-Student Post-KD

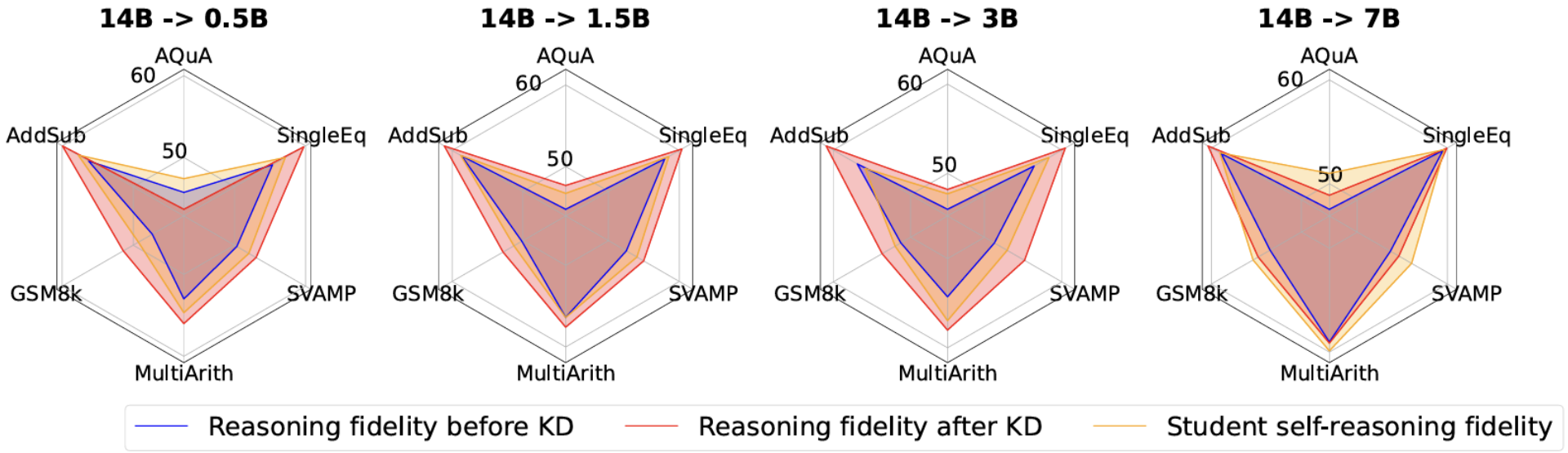


Agreement: Overlap between the final output generated by teacher and students.

Teacher-student agreement improves post KD, mostly for smaller LMs (<7B).

ACL 2025

Fidelity b/w Teacher-Student Post-KD



Fidelity: Ability of the student to imitate the teacher's reasoning behaviors.

- Smaller LMs tend to have better fidelity post-KD.
- However, statistical tests show that fidelity does not necessarily improve the generalization abilities of student models!!

ACL 2025

Fidelity vs Generalization Paradox of KD

Instruction	Student KD Output	Teacher SFT Output	Fidelity
Robin has 28 packages of gum and 13 packages of candy. There are 4 pieces in each package. How many pieces of gum does Robin have?	To solve this problem, we need to use multiplication. 1. Identify the given information: Robin has 28 packages of gum and 13 packages of candy. Each package has 4 pieces of gum and candy. 2. Determine the operation needed to find the total number of pieces of gum: We need to multiply the number of packages of gum by the number of pieces of gum in each package. 3. Multiply: 28 by 4 to find the total number of pieces of gum. 56. The answer is 56. Therefore, Robin has 56 pieces of gum.<|endoftext|>	To solve this problem, we need to use multiplication. 1. Multiply the number of packages of gum by the number of pieces in each package: 28 packages x 4 pieces = 112 pieces of gum 2. Therefore, Robin has 112 pieces of gum. The answer in Arabic numerals is: 112.<|endoftext|>	0.79

High teacher-student fidelity, but wrong answer predicted by student (poor generalization)

ACL 2025

Fidelity vs Generalization Paradox of KD

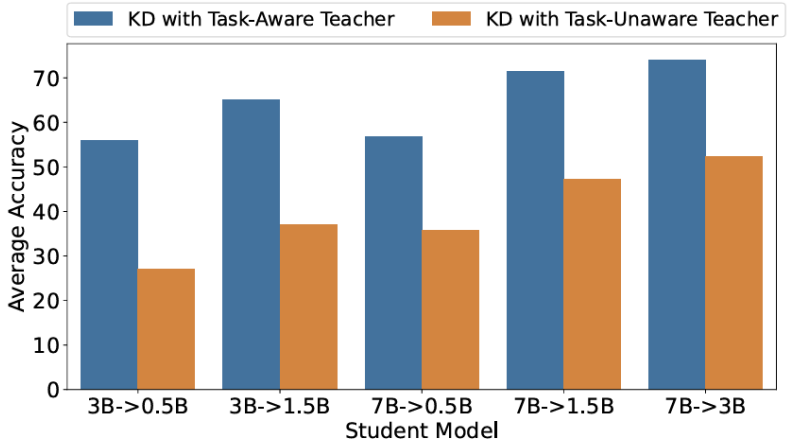
Instruction	Student KD Output	Teacher SFT Output	Fidelity
Because of the decision Sofia asked 310 students to suggest specific types of food. 185 students suggested adding mashed potatoes while others suggested adding bacon to the menu. How many students suggested bacon?	To solve the problem, we can use subtraction:number of students = 310of students who suggested mashed potatoes = 185find the number of students who suggested bacon, we subtract the number of students who suggested mashed potatoes from the total number of students: 310 - 185 = 125 Therefore, 125 students suggested adding bacon to the menu.<|endoftext|>	We can start by using the concept of complementary counting. Let x be the number of students who suggested bacon. We know that the total number of students is 310, so we can set up an equation: 185 + x = 310 Subtracting 185 from both sides: x = 310 - 185 Simplifying: x = 125 Therefore, 125 students suggested adding bacon to the menu. Answer: 125.<|endoftext|>	0.13

Low teacher-student fidelity, but good generalization

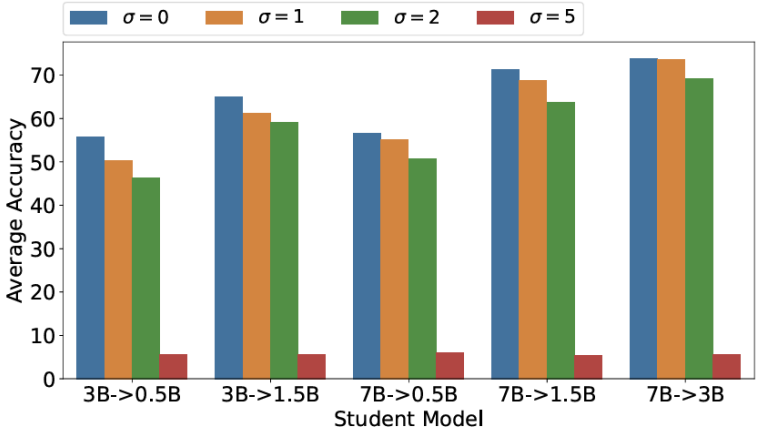
Therefore, the tradeoff between generalization vs fidelity-agreement remains prominent.

ACL 2025

Drivers behind Successful KD



1. Teacher model should be task-aware



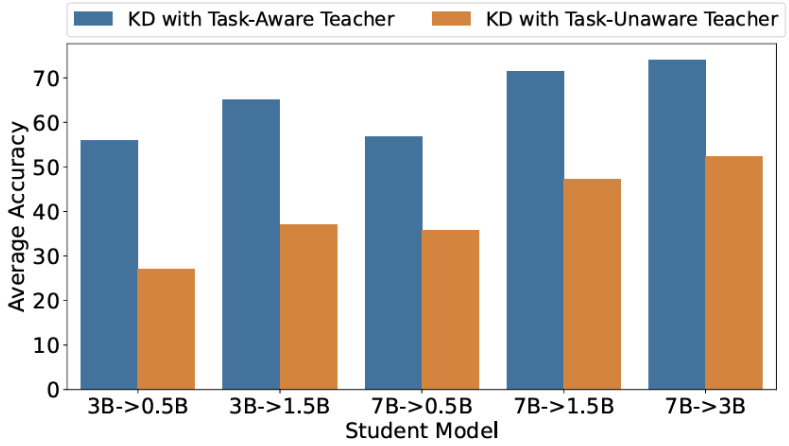
2. Teacher signals to student should be noise-free.

Here σ is the amount of Gaussian noise added to the teacher logits before distilling to student. For σ, student performance drops drastically.

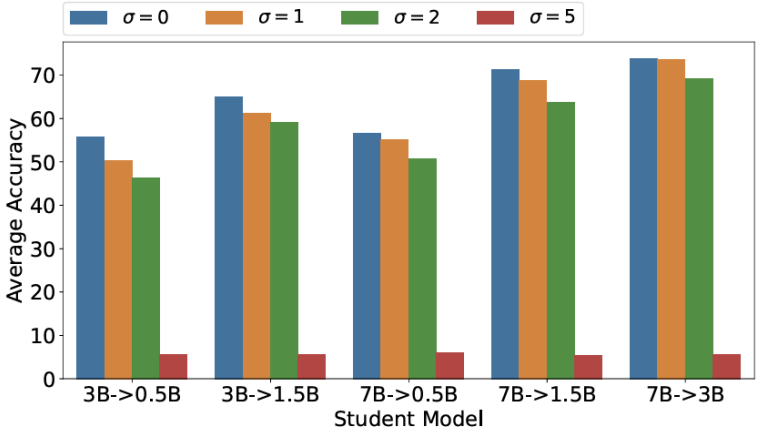
Teacher model performance ***minimally affects*** student outcomes; however, the teacher's ***task-specific expertise is crucial***

ACL 2025

Drivers behind Successful KD

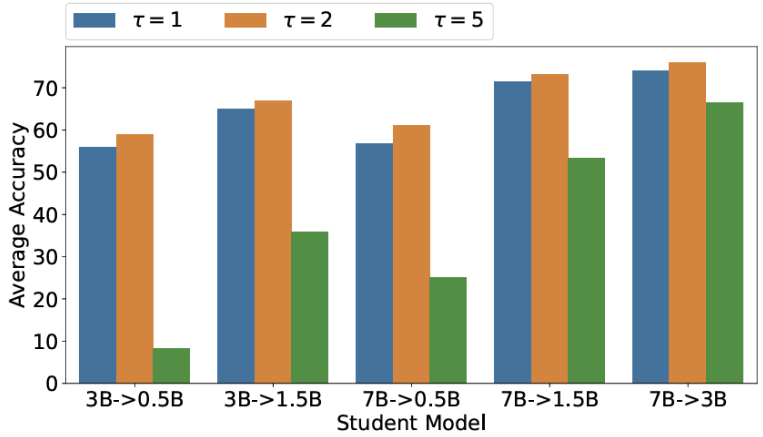


1. Teacher model should be task-aware



2. Teacher signals to student should be noise-free.

Here σ is the amount of Gaussian noise added to the teacher logits before distilling to student. For σ, student performance drops drastically.



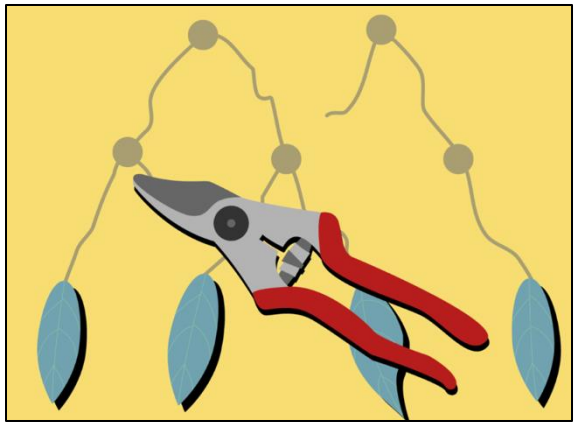
3. Logit smoothing is important

Here τ is the temperature used to smoothen the teacher logits. Too much smoothing hurts student performance, but moderate smoothing shows benefit.

Temperature (τ) in KD balances precision ($\tau \downarrow$) and recall ($\tau \uparrow$) of the student model.

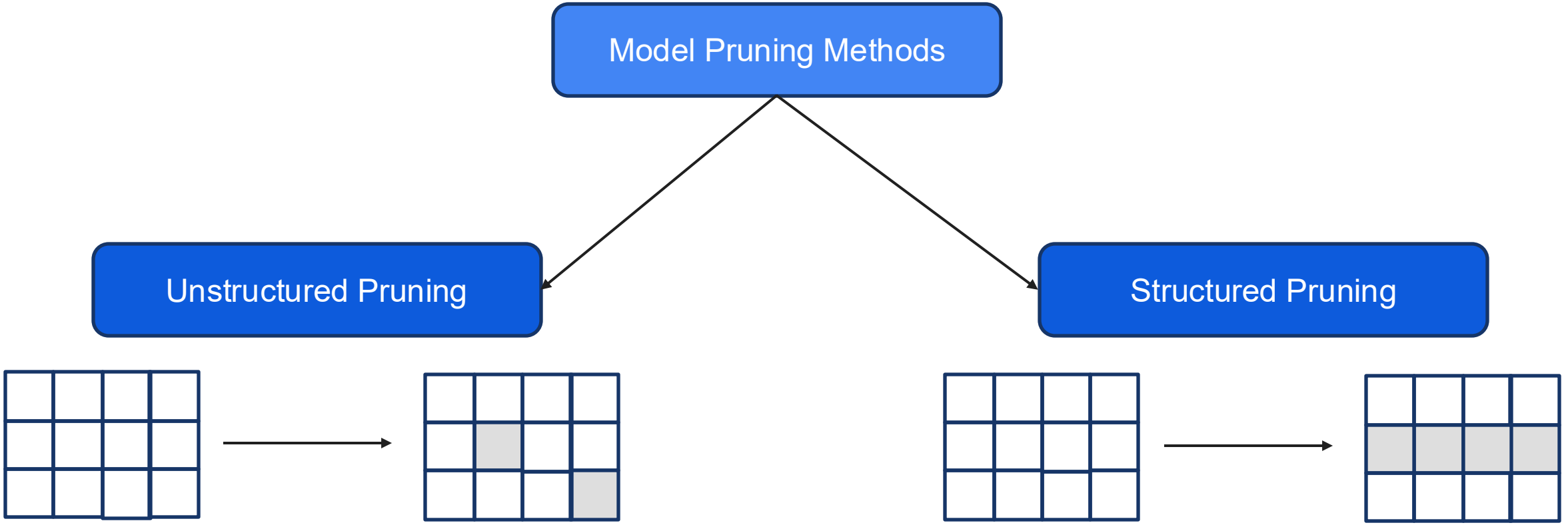
Economical Models

Model Pruning



(ICLR'25, NeurIPS'24)

Efficient Model Pruning

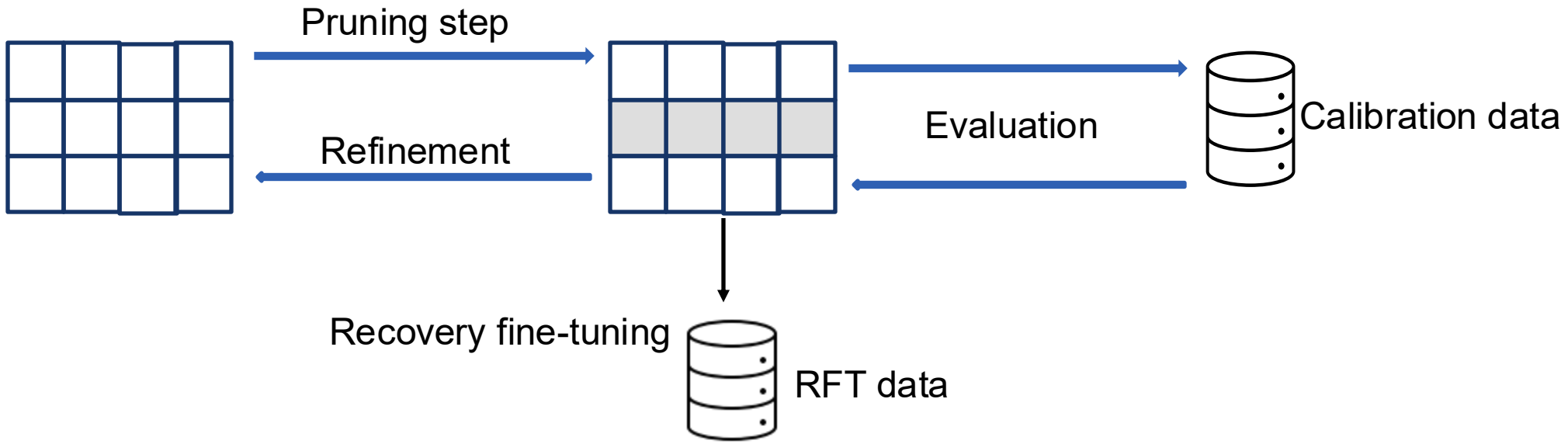


Benefit - Better retention of model performance
Limitation – Sparse matrix operations are hardware-dependent and requires complex implementation for inference benefits

Benefit - Easier to implement, flexible and hardware friendly
Limitation – Over pruning hurts performance, needs careful considerations for different model architectures

ICLR'25

Structure pruning requires calibration data



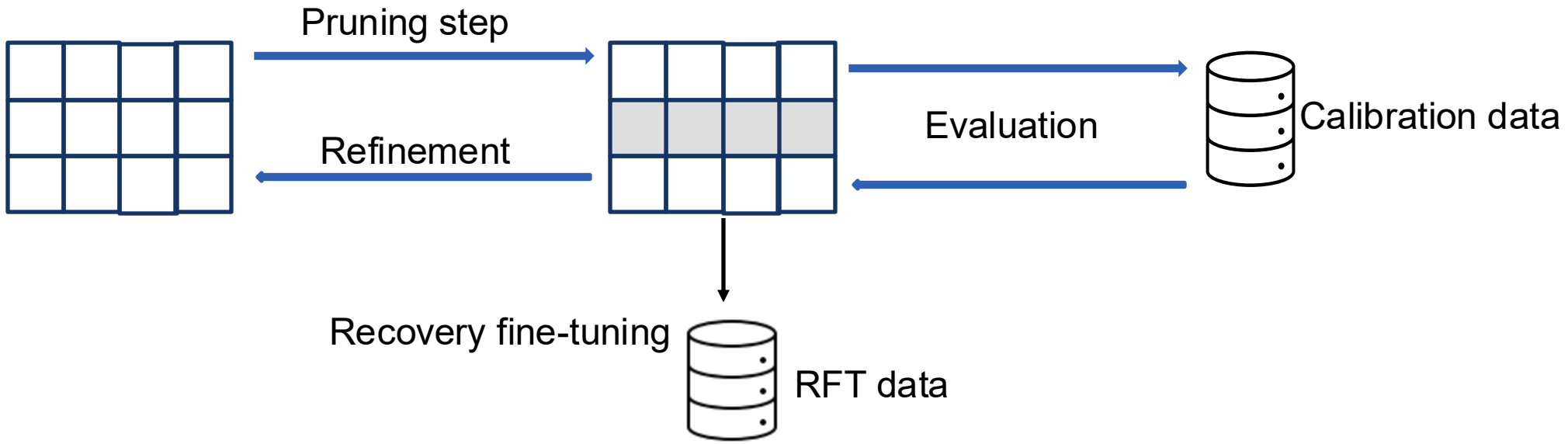
Existing structured pruning methods – SliceGPT (Ashkboos et al., 2024), LLM Pruner (Ma et al., 2023), Layer Collapse (Yang et al., 2024) use calibration data to determine the unimportant components of a pre-trained model for pruning.

Limitations

1. Over-reliance on calibration data makes the compressed model sensitive to the data selection, becomes less reliable on downstream tasks (Ji et al., 2025)
2. Recovery fine-tuning (RFT) is crucial for preserving performance of the models, post-compression

ICLR'25

Structure pruning requires calibration data

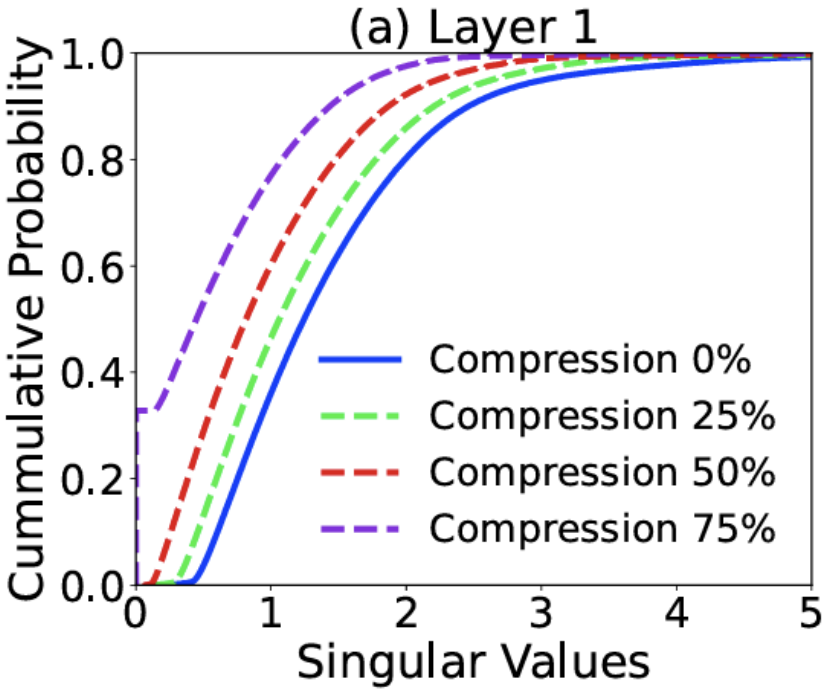


Lemma 3.1 (Limitations of Intrinsic Model Compression). *Given an LLM with hidden dimension* d_{hidden} *and intermediate FFN dimension* $d_{intermediate}$*, any intrinsic model compression method that introduces new parameters within the model will reduce model size only if the compression ratio* $r > \frac{d_{hidden}+d_{intermediate}}{5d_{hidden}+3d_{intermediate}}$.

ICLR'25

Can we use Intrinsic Metrics for Pruning

Corollary 3.3 (Slicing shrinks the range of the spectrum). *Let $\boldsymbol{W} \in \mathbb{R}^{n \times d}$ be a weight matrix, and let $\boldsymbol{W}' \in \mathbb{R}^{m \times d}$ be a matrix obtained by slicing off rows of $\boldsymbol{W}$ so that $m \leq n$. Then, the range of singular values of $\boldsymbol{W}'$ is a subset of the range of singular values of $\boldsymbol{W}$.*

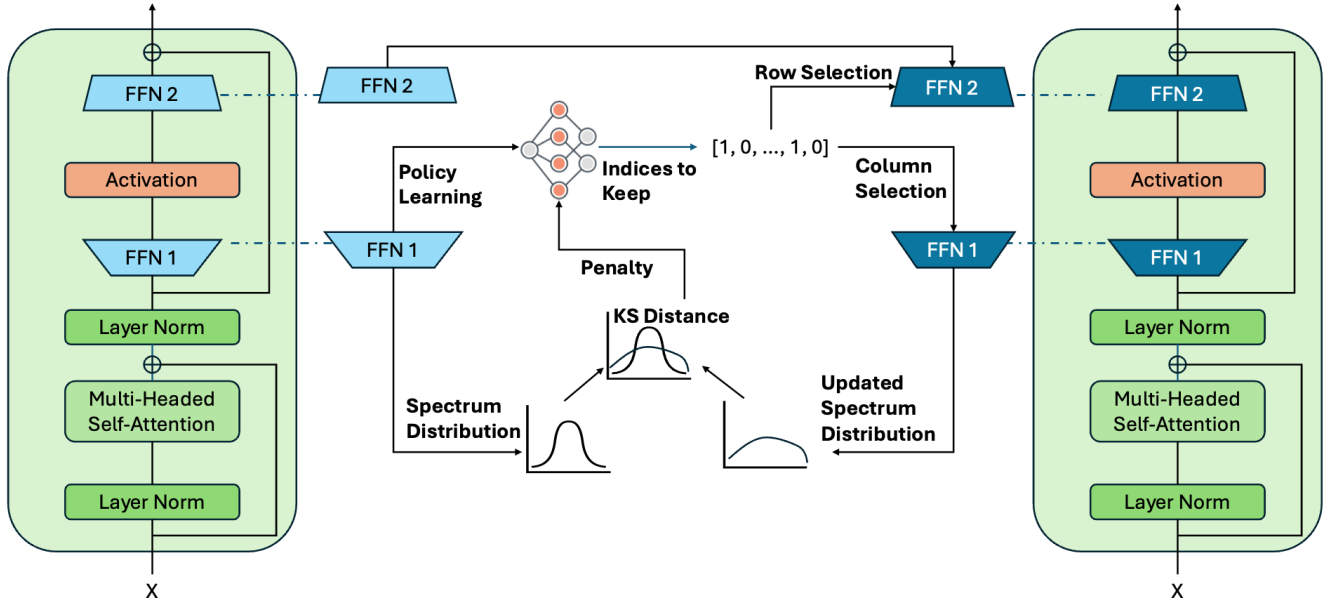


Singular values of a matrix determine the importance of each component.

Can we preserve the singular value structure (spectral structure) to preserve the performance of compressed model?

ICLR'25

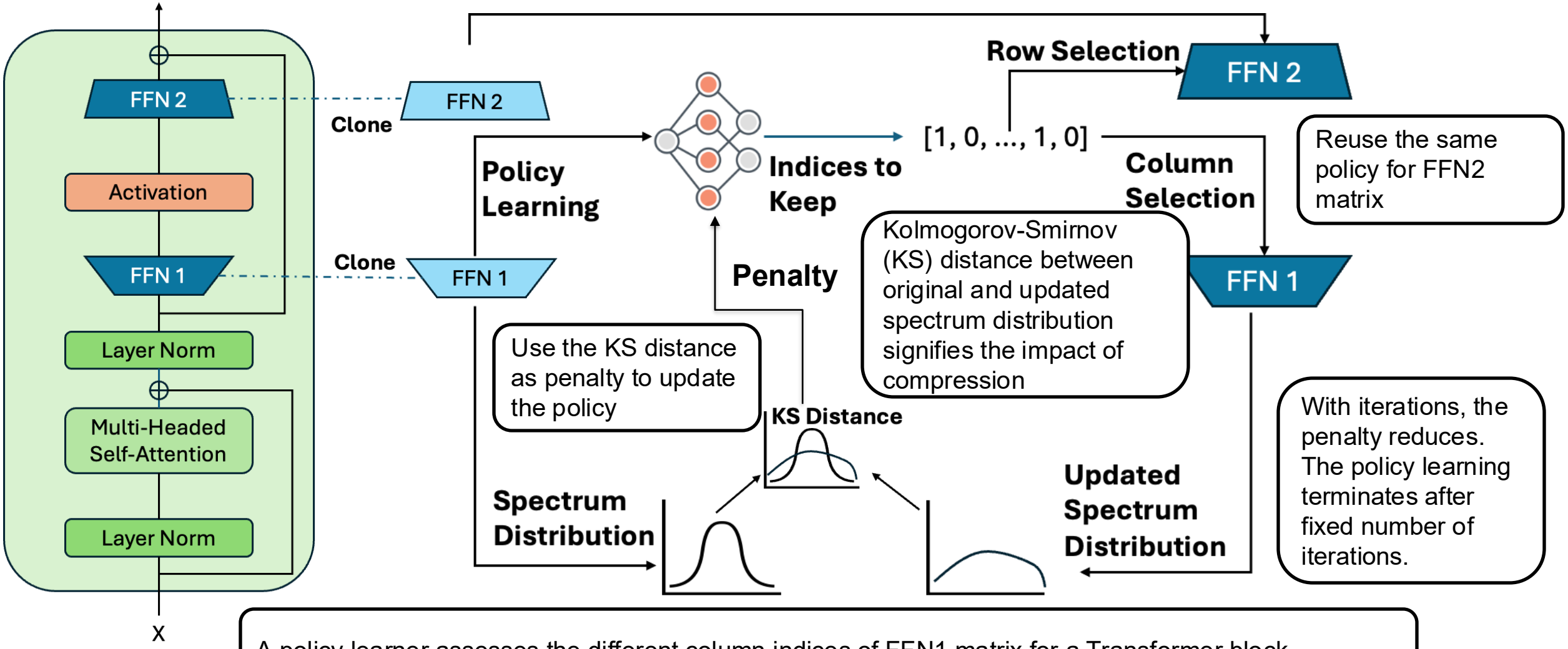
PruneNet: Calibration-free Structured Pruning



- ***PruneNet*** treats model compression as a **policy-learning process** that assesses the parameter importance once (using intrinsic methods) and can reuse the policy to compress the model at multiple compression ratios, at once.
- PruneNet is highly flexible, reusable and does not use sensitive and unreliable mechanisms like calibration.

ICLR'25

PruneNet: Calibration-free Structured Pruning



A policy learner assesses the different column indices of FFN1 matrix for a Transformer block

ICLR'25

Effectiveness of PruneNet: Empirical Evidence

Method	Sparsity	Effective Sparsity	FLOPs	Avg. Zero-shot Acc
Dense	0%	0.0%	1.35e+13 (1.00x)	69.0
SliceGPT	20%	9.4%	1.23e+13 (1.10x)	58.2
PruneNet		**12.0%**	**1.18e+13 (1.15x)**	**61.7**
SliceGPT	25%	15.3%	1.14e+13 (1.18x)	55.5
PruneNet		**16.0%**	**1.13e+13 (1.20x)**	**58.6**
SliceGPT	30%	**21.4%**	**1.07e+13 (1.27x)**	51.5
PruneNet		19.0 %	1.09e+13 (1.24x)	**55.5**

Model	Method	Throughput (Token/sec)
LLaMA-2-7B	Dense	11.96
	SliceGPT	12.82
	PruneNet	**20.74**
Phi-2	Dense	20.20
	SliceGPT	18.48
	PruneNet	**29.50**

PruneNet achieves higher effective sparsity and efficiency while maintaining better performance on downstream tasks.

Effective sparsity indicates the memory reduction in the compressed model.

LLaMA-2-7B compressed with PruneNet exhibits 73% better inference throughput than the original model.

ICLR'25

Takeaways

- LLaMA-2-7B compressed with PruneNet exhibits **73% better inference throughput** than the original model.

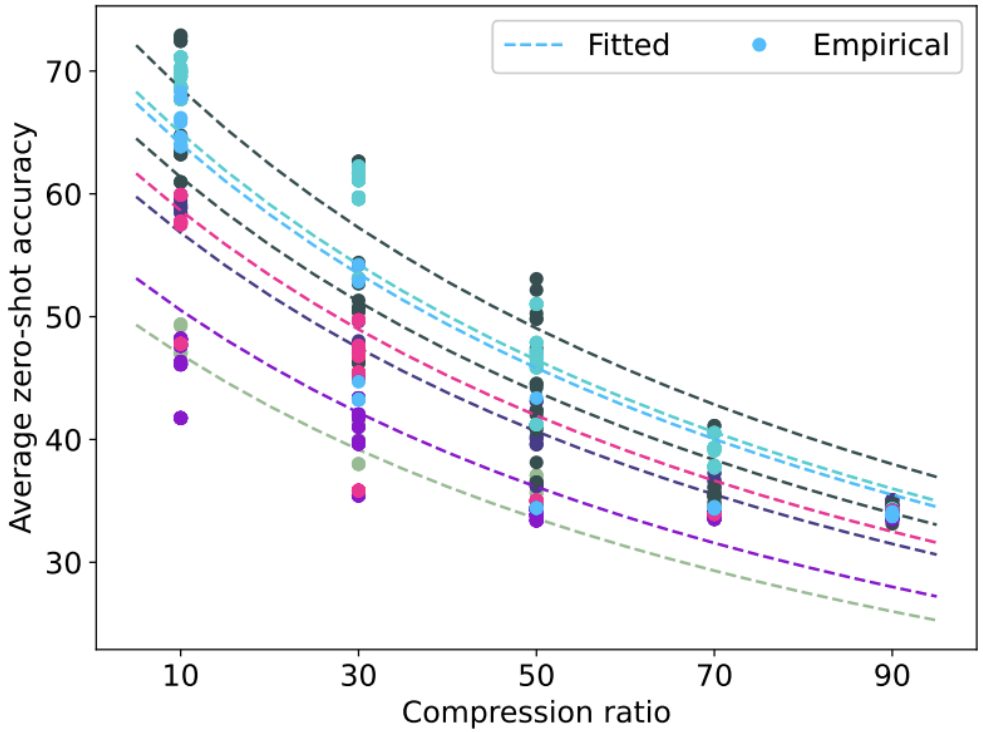
- PruneNet can compress LLaMA-2-7B in just 15 minutes by 30%, achieving over 80% retention of its zero-shot performance.

- PruneNet is **architecture-agnostic** and can be applied on any pre-trained network, without the need for any calibration.

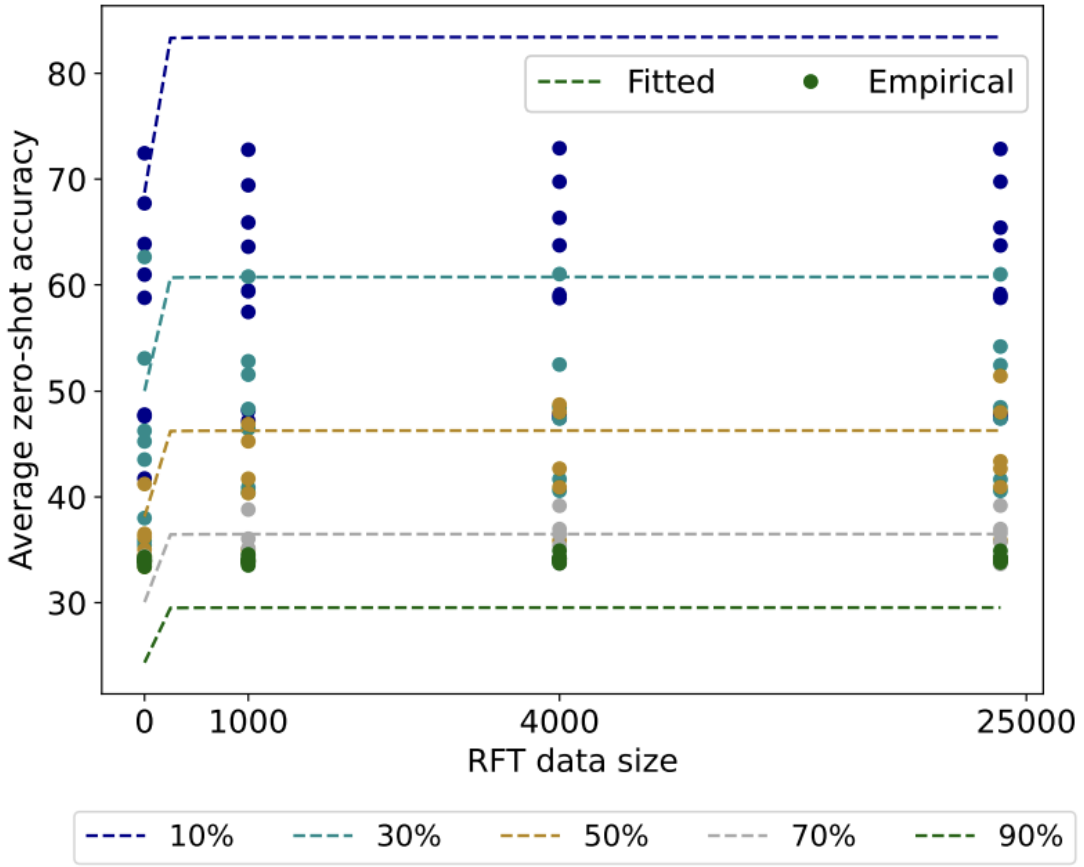
https://github.com/LCS2-IIITD/PruneNet

- SliceGPT, PruneNet still face some loss in the performance
- Can we analytically estimate how much performance can be dropped post compression?
- Can we estimate how much inference speedup is expected post-compression to overweigh the performance drop?

Accuray vs compression ratio is power law

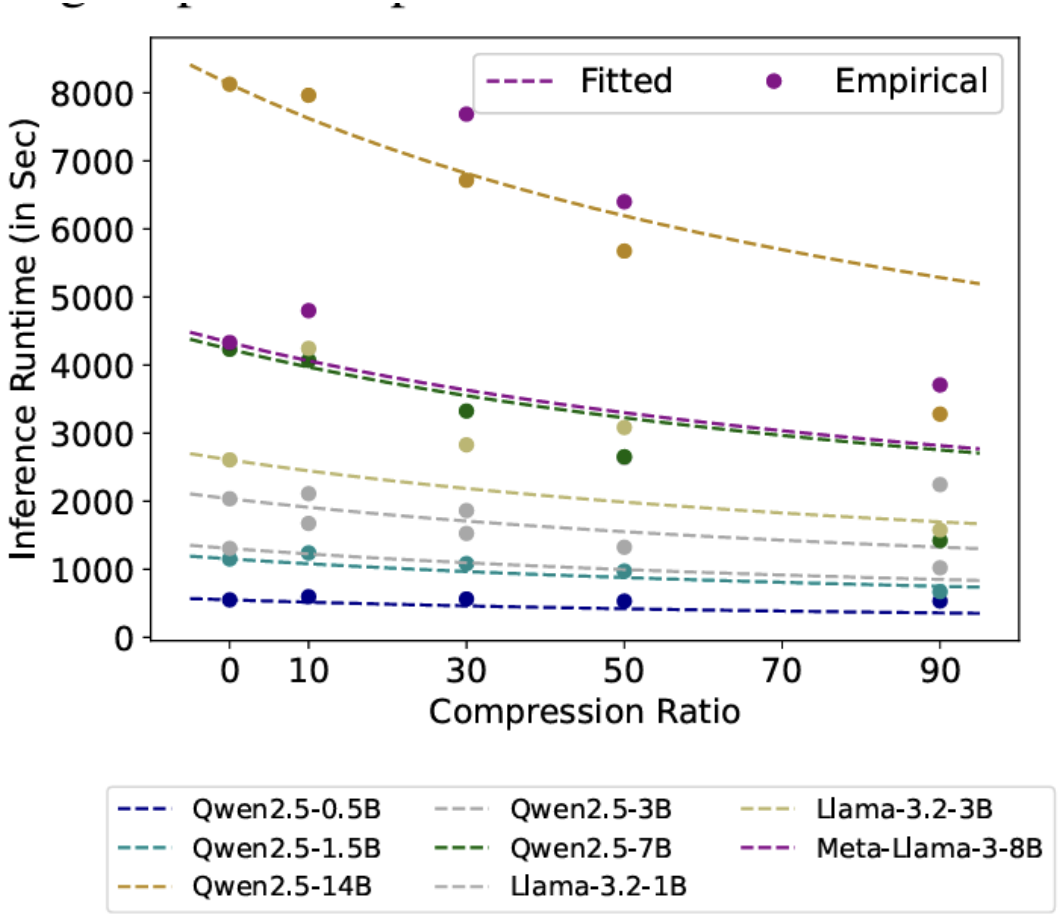


(b) Extrinsic performance follows $\mathcal{L} = \mathcal{L}_0^{1.01}(r+1)^{-1.05}$



(d) $\mathcal{L} = \mathcal{L}_0^{0.98}(r+1)^{-1.03}\left(1+\frac{1}{D+1}\right)^{-0.14}$

Inference time vs compression ratio is power law



Inference runtime of a compressed LLM follows power-law

$\mathcal{L} = \mathcal{L}_0{}^{1.0}(1+r)^{-0.67}$, with $\mathcal{L}_0$ being the uncompressed model runtime, r being the compression ratio.

Increasing the compression ratio for 0% to 30%, leads to 17% reduction in runtime (1.19X speedup). Increasing compression ratio to 60% leads to 1.37X speedup.

ICML'25

Compression Laws for Large Language Models

Ayan Sengupta [*1] **Siddhant Chaudhary** [*1] **Tanmoy Chakraborty** [1]

$$\mathcal{L}(\mathcal{L}_0, r, D) = \mathcal{L}_0 P_0 (1-r)^{\alpha} \left(1 + \frac{D}{D+\epsilon}\right)^{\beta}$$

- Performance of pruned LLM
- Performance of unpruned LLM
- Pruning constant
- Effect of pruning ratio
- Effect of recovery fine-tuning

https://arxiv.org/pdf/2504.04342

Critical Compression Ratio

$r_{critical}(\sigma) := \sigma^{\frac{1}{\beta}} - 1$, *which we call the* ***critical compression ratio for recovery threshold*** $\sigma \in (0, 1)$. *Then the following hold:*

1. *If* $\sigma \in (0, 2^{\beta})$, *then for any compression ratio* $r \in (0, 1)$, *there exists* D *such that RFT on the compressed model with a dataset size of* D *will result in* $\frac{\mathcal{L}}{\mathcal{L}_0^{\alpha}} \geq \sigma$.

2. *If* $\sigma \in [2^{\beta}, 1)$, *then for any* $r \geq r_{critical}(\sigma)$, *no amount of RFT can recover the compressed model accuracy* $\mathcal{L}$ *to satisfy* $\frac{\mathcal{L}}{\mathcal{L}_0^{\alpha}} \geq \sigma$. *On the other hand, for any* $r < r_{critical}$, *there is a large enough* D *such that RFT with a dataset of size* D *will result in* $\frac{\mathcal{L}}{\mathcal{L}_0^{\alpha}} \geq \sigma$. [2]

If we prune the LLM below the critical compression ratio, we can achieve the performance recovery target with some arbitrarily large recovery fine-tuning data.

if we compress the LLM more than the critical compression ratio, we can never reach the performance target, no matter how much we fine-tune the pruned model.

Laboratory for Computational Social Systems (LCS2), IIT Delhi

http://lcs2.in

@lcs2lab

Thanks to our sponsors

Flipkart

accenture

J.P.Morgan

Adobe

Logically

IBM

Microsoft

wipro

Department of Science & Technology, Government of India

Project under MHRD GOVERNMENT OF INDIA